
趣味程序设计
集锦
煤炭工业出版社

封面设计：赵 鸿 亮

科技新书目：62—125
书号：15035·2609
定价：3.30元

趣味程序设计集锦

纪有奎 冀 翾 编著

煤炭工业出版社

趣味程序设计集锦

纪有奎　冀　翾　编著

煤炭工业出版社

内 容 提 要

本书题目选自国内外趣味数学、经典性程序设计等题。许多题出自著名数学家之手，用FORTRAN语言进行程序设计，有算法分析、设计思路、框图、源程序和上机结果。因此也适用于BASIC、PASCAL、ALGOL等算法语言。

题目趣味横生，引人入胜，使乏味程序设计趣味化。取材有简有繁，实用性强，适用面广，使初学者及有一定造诣的读者可各取所需，是一本普及和提高相结合的书。

本书可供大、中专师生、科技人员、计算机工作者阅读，也可作为教学补充资料，视为绝妙的习题集。

责任编辑：刘庆韶

趣味程序设计集锦

纪有奎 冀翾 编著

*

煤炭工业出版社 出版

（北京安定门外和平北路16号）

煤炭工业出版社印刷厂 印刷

新华书店北京发行所 发行

*

开本787×1092 1/16 印张31 1/2 插页2

字数753千字 印数1—20,000

1984年5月第1版 1984年5月第1次印刷

书号15035·2609 定价3.30元

前　言

去年我们用BASIC语言编写的《趣味程序设计100例》出版后，收到许多读者来信，热情鼓励我们再更换些题目，用另一种常用的语言继续编写这类书。于是，我们更换了绝大部分题目，加强了广度和深度，用FORTRAN语言编写了本书。

本书题目选自国内外趣味数学、数学竞赛、智力测验以及经典性的程序设计题。内容广泛，妙趣横生，引人入胜。不少题是出自著名数学家之手，也有我们自己改编的。

这些具有不同难度的题做为程序设计的“靶子”，体裁新颖，使乏味程序设计趣味化，可培养程序设计的兴趣和独立思考分析问题、解决问题的能力。通过多种类型的选题，不仅扩大了程序设计的视野和领域，复习并巩固所学的基础知识，还可锻练读者灵活运用算法语言进行程序设计的本领，提高设计技巧和水平。

该书每道题包括三部分：（一） 算法分析或笔算步骤和结果（笔算可启发算法）；（二） 程序设计部分——设计思路、框图、源程序和上机打印结果；（三） 思考题。因此也适用于BASIC、PASCAL、ALGOL等算法语言，达到“举一反三”的目的，具有通用性。每题都上机调通（机型是：TRS-80，BCM-2，HP/3000，NOVA/4，IBM360等）。因机型不同，输出入设备号等也各异，个别题采用机器配备的扩展FORTRAN语句，按照原样未做修改。

本书在编写过程中：得到中国科学院计算技术研究所夏培肃研究员的鼓励和支持，FORTRAN程序设计专家、北京工业大学计算机系系主任丘玉圃教授对本书提出许多宝贵意见，北京计算机技术研究所柳维长所长特为本书撰写序言并提出建设性意见，中国科学院沈阳计算技术研究所徐国红同志提供部分题，刘福生等同志协助调试部分题的程序，北京工业大学、北京计算机技术研究所等单位，均创方便提供上机条件，得到学生的协助，我们学校有关领导也给予大力支持，谨此表示感谢。由于水平所限，错误之处敬请广大读者指正。

北方交通大学电子工程系　纪有奎
中国科学技术大学计算机科学技术系　冀　翾

1983年10月

序　言

学习算法语言、编写程序，常使人们感到枯燥乏味。作者打破常规，编写了趣味程序设计专著，填补了空白。这是普及计算机里程中一个迷人境界。作者是这个领域的探索者，任何科学的探索者需要花费加倍的心血。去年出版了《趣味程序设计100例》，今又编写本书，两年内抽空著作近百万字，表现了作者的才华和勤奋。

本书不是长篇大论，而是摆在读者面前一道道新颖的趣味题，需要灵活运用所学的基础知识，分析算法、进行程序设计，是智力锻练的脑力操，可复习、巩固基础知识，锻练逻辑思维能力，开发智力，激发学习兴趣，开阔视野，从而提高程序设计技巧和水平。书中许多题本身就是实用题或是实用题的化身，被有趣的故事情节装饰着。文学作品有小品文，本书每题均是程序设计小品。

作者原收集的题是为课外辅导学生考研究生的算法语言、程序设计课而准备的，若全按这意图成书，将失去初学的广大读者；若全以初学者程度出书，又将失去许多有造诣的读者。本书选材难易结合，用不同的笔调去触及这些难易题。小题仅用十几个语句编写程序，大题有近千条语句。有些题可作课程设计的大习题，甚至可选为毕业论文题，初学者可暂不读这类题，各取所需。

本书将数学和程序设计水乳交融、巧妙结合在一起，成了孪生姊妹。读者在获得程序设计知识的同时，还能得到数学的享受。除大量的中学数学知识外，还涉及到数论、运筹学、图论、数理逻辑、博弈论、对策论以及人工智能等，这不仅显示了作者程序设计的本领，同时还显示了作者具有广泛而雄厚的数学基础和渊博知识。将内容广泛的趣味数学题、经典程序题，得心应手地运用算法语言，巧妙地编写程序上机通过，打印输出结果。

许多趣味数学题出自著名数学家之手，为了锻练智力、考验掌握基础知识和灵活运用的能力，常出现在考试、竞赛场合。趣味程序设计题也应如此。有的数学家以趣味数学而闻名，被称为趣味数学家。同理，如果说有程序设计专家的话，自然也应有趣味程序设计家。

作者说书中许多题是经过苦思瞑想上机反复调试出来的，回头再看，又觉得并不难，走了许多弯路，这是探索者的心声。因此，当你每读一题时，不妨先思索片刻，再往下看解法，因为每题的程序不是唯一的，“抛砖”能“引玉”。若过分追求技巧，可能使程序费解、难读。为了初学者，书中某些题描述得很详尽；为了有基础的读者，有些题则加强了技巧性。思考题留作习题，这是“山外青山楼外楼”——题外有题。

本书第107题名为“游览名胜十九处”，书中正好是十九个大题，从第一处入口分酒比赛、喝啤酒开始，到漫步迷宫，魔术师玩扑克牌，新婚夫妻蜜月旅行，里程碑……，在第十九处(出口)挂有世界名画摩娜丽莎。这些古今中外的十九处名胜古迹，构成全书奇异的游览区。科学著作具有艺术构思，耐人深思回味，别具匠心！请你漫步饱览这十九处名胜。

我也是一位普通读者，有幸在出版前粗览了手稿这十九处，写了以上体会和看法，供广大读者参考，不当处，请指正。

北京市计算机技术研究所所长　柳维长

1983年9月

目　录

一　分酒比赛 ……1
1　分蜂蜜 ……1
2　喝啤酒 ……3
3　分牛的传说 ……5
4　出售金鱼 ……7
5　巧分桔子 ……10
6　整元兑换硬币 ……13
7　老人分马 ……16
8　左右衣袋钱数相等 ……21
9　传递小球 ……23
10　分酒比赛 ……26
二　辨别四对夫妻关系 ……44
11　分辨阅览室的桌子和灯 ……44
12　挑选乒乓球 ……46
13　这样区分全钢和半钢表壳 ……49
14　求婚者的智慧 ……51
15　白兔智斗狡猾的狐狸 ……54
16　找出次品钮扣 ……57
17　生者与死者 ……61
18　如此辨认伪币 ……64
19　谁在说谎 ……69
20　辨别鸡蛋和鸭蛋 ……72
21　公安人员断案 ……75
22　确定值班大夫何日值班 ……78
23　辨别四对夫妻关系 ……80
三　哥德巴赫和欧拉书信往来 ……87
24　哥德巴赫和欧拉书信往来 ……87
25　互质二数加减得 1 ……92
26　找相邻二数之和为素数 ……94
四　魔术师玩扑克牌 ……97
27　魔术师玩扑克牌 ……97
28　魔术师又玩扑克牌 ……99
29　钟形牌游戏 ……102
五　数的趣谈 ……106
30　回文数猜想 ……106
31　数的趣谈 ……109
32　立方和恰好等于和的平方 ……113

33 胸中有数 ……115
34 北京市中学生智力竞赛题 ……118
35 刘徽《九章算术》中的勾股数 ……120
36 水仙花数 ……123
37 巧妙的算法 ……124
38 完全数 ……126
39 自然数可用不超过四个平方数之和来表示 ……129
40 惊人的记忆力 ……133
六 漫步迷宫 ……137
41 漫步迷宫 ……137
42 怪兽的踪迹 ……141
43 四通八达的迷宫 ……147
七 计算机会告诉你怎样度过年华 ……155
44 日历编排的规律 ……155
45 是否有每月此日都不是星期天 ……158
46 元旦遇到星期六还是星期天的机会多 ……162
47 老人今年几何 ……164
48 计算机会告诉你怎样度过年华 ……166
八 二人博弈 ……176
49 巧夺奇数 ……176
50 取尽者胜 ……180
51 有奖储蓄 ……183
52 请试试你几次猜中 ……187
53 先取者输 ……190
54 由你指定总数量和取法 ……193
55 由双方共同指定扑克牌总数 ……197
56 二人博弈 ……202
九 计算机算式 ……209
57 BC乘BC等于几 ……209
58 计算机算术字谜 ……211
59 除法游戏 ……213
60 古诗词算式 ……215
61 计算机算式 ……218
62 古纸残篇 ……221
63 祝圣诞节快乐 ……224
64 巧妙的算式 ……227
十 里程碑 ……231
65 发展家庭养殖事业 ……231
66 运动员和学习成绩 ……232
67 人口非控制不可 ……234
68 如何找这个数 ……238
69 苹果装箱 ……241

70 找出这样的四位数 ……245
71 求出特定的四位数 ……247
72 五个人的年龄 ……249
73 数学谜的年龄 ……251
74 里程碑 ……253
75 第十八届国际数学竞赛题 ……256
76 找出三个不同的数字 ……262
77 找出四个数之和的最大及最小 ……264
78 排队买票 ……267
79 乐队人数 ……270
十一 高斯八皇后92种解 ……272
80 行列各种四棵树 ……272
81 高斯八皇后92种解 ……275
82 小虫奇妙地移动 ……286
83 生命游戏 ……297
十二 巧妙填数 ……305
84 使每两个棋子跳在一起 ……305
85 按要求填表 ……307
86 钟表盘上数字分组 ……310
87 变换各数的位置 ……312
88 巧妙填数 ……316
十三 三对新婚夫妻蜜月旅行 ……320
89 皇后王子公主逃命 ……320
90 猎人监送狗兔白菜 ……324
91 跳棋 ……327
92 三对新婚夫妻蜜月旅行 ……332
93 古老的神话 ……336
十四 第五届国际数学竞赛试题 ……344
94 教练员智拆两队 ……344
95 乒乓球队员分组训练 ……349
96 砝码问题 ……352
97 五本书被三个人借 ……355
98 巧排分组 ……359
99 第五届国际数学竞赛试题 ……362
100 用金环付房租 ……366
十五 人工智能“重排九宫” ……371
101 巧排魔方阵 ……371
102 在方阵中寻找规律 ……376
103 对调四阶方阵中数字 ……380
104 人工智能“重排九宫” ……382
105 魔方游戏 ……394
十六 游览名胜十九处 ……416

106 如何画一笔画 ……416
107 游览名胜十九处 ……423
十七 条条大路都能走 ……438
108 任意六人必有三人两两相识 ……438
109 九个不同国际数学家如此对话 ……441
110 条条大路都能走 ……444
十八 茫然大海寻友人住址 ……450
111 每行的数是某数的平方 ……450
112 杨辉三角形 ……452
113 用0 1 2三个数字排成序列 ……456
114 加拿大第七届数学竞赛题 ……463
115 找座标上的点 ……465
116 茫然大海寻友人住址 ……469
117 数字串排序游戏 ……476
十九 世界名画摩娜丽莎 ……482
118 世界名画摩娜丽莎 ……482

一 分酒比赛

1 分蜂蜜

三家合伙养蜜蜂，年终清点时，有七个满桶蜂蜜，七个半桶和七个空桶。三家平分这些蜂蜜和装蜂蜜的桶，规定是不许打开桶后将蜂蜜倒来倒去，应该怎样分？

（一）笔算步骤和结果

分析可能有几种分法，如果想到这21个桶里装蜂蜜共有$7+3\frac{1}{2}$，即$10\frac{1}{2}$桶，便易找出办法。就是说每家应得到 $3\frac{1}{2}$桶蜂蜜和 7 个包装用的桶。有两种办法：

办法 1

第一家 { 3 个满桶
1 个半桶
3 个空桶 } 共 $3\frac{1}{2}$桶蜂蜜和 7 个桶

第二家 { 2 个满桶
3 个半桶
2 个空桶 } 共 $3\frac{1}{2}$桶蜂蜜和 7 个桶

第三家 { 2 个满桶
3 个半桶
2 个空桶 } 共 $3\frac{1}{2}$桶蜂蜜和 7 个桶

办法 2

第一家 { 3 个满桶
1 个半桶
3 个空桶 } 共 $3\frac{1}{2}$桶蜂蜜和 7 个桶

第二家 { 1 个满桶
5 个半桶
1 个空桶 } 共 $3\frac{1}{2}$桶蜂蜜和 7 个桶

第三家 { 3 个满桶
1 个半桶
3 个空桶 } 共 $3\frac{1}{2}$桶蜂蜜和 7 个桶

（二）程序设计

1）设计思路

设 I 、 J 、 K 三个变量，分别代表满桶、半桶、空桶。据抽屉原则第一家至少应分 3 个满桶，故程序开始时，令 I 1⇐3。然后利用二重循环试分第一家半桶J1，空桶K 1 各 应得多少。初步确定I1、J1、K1后，再用循环语句确定第二家试分的I2、J2、K2，第三家由上试分法是总数减去上述两家分得量，若不符合条件，依次递推返工，再重新试分，直到满足规定条件为止，分别打印出正确分配方法。

2）框图　见图 1

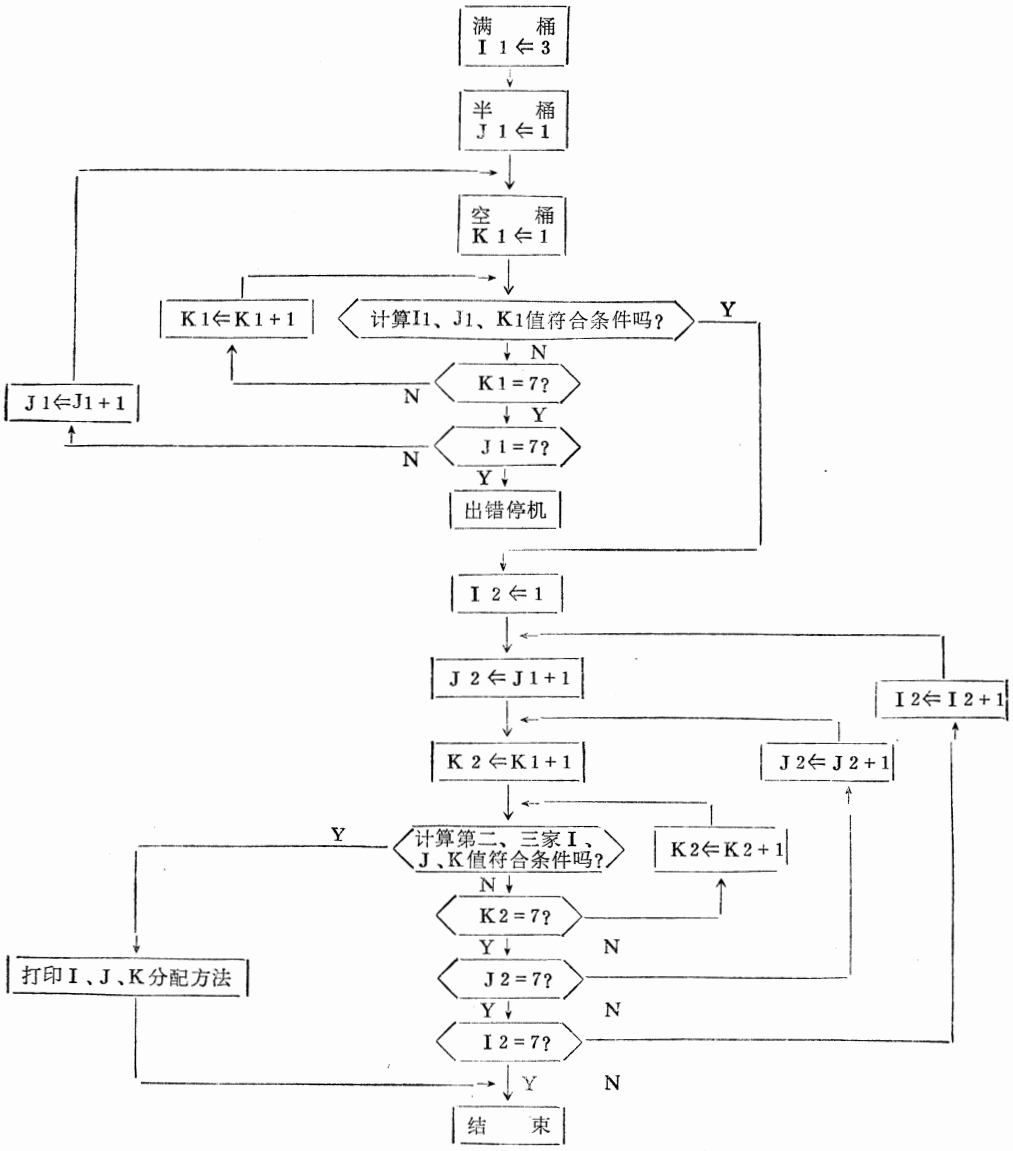


图 1

3）源程序及运行结果

```
      I1=3
      DO  10  J1=1, 7
      DO  15  K1=1, 7
      IF (FLOAT (I1) +FLOAT (J1) /2.0. NE. 3.5) GO TO 10
      IF (I1+J1+K1. NE. 7) GO TO 15
      GO TO 18
15    CONTINUE
10    CONTINUE
      STOP'ERR'
18    DO  20  I2=1, 3
```

```
      DO  22  J2=J1+1, 7
      DO  23  K2=K1+1, 7
      J21=J2-J1
      IF (FLOAT (I2) +FLOAT (J21) /2.0. NE. 3.5) GO TO 22
      K21=K2-K1
      IF (I2+J21+K21. NE. 7) GO TO 23
      I3=7-I2-I1
      J3=7-J2
      IF (FLOAT (I3) +FLOAT (J3) /2.0. NE. 3.5) GO TO 22
      K3=7-K2
      IF (I3+J3+K3. NE. 7) GO TO 23
      WRITE (6, 50) I1, J1, K1, I2, J21, K21, I3, J3, K3
50    FORMAT (1X, 3 (3I3, 5X) )
23    CONTINUE
22    CONTINUE
20    CONTINUE
      STOP
      END
```

程序运行后打印结果

```
3  1  3     1  5  1     3  1  3   第二种分配办法
3  1  3     2  3  2     2  3  2   第一种分配办法
```

（三）思 考 题

根据美国作家本·艾姆斯·威廉斯写的小故事改编此题，名为分椰子。说是五个男人和一只猴子因翻船而来到一个小岛上。第一天他们花了整天的功夫采集椰子。夜里有一个人醒了，他决定取走自己的一份椰子。于是他把椰子分成五堆，结果还剩下一个椰子，他便把它分给了猴子。然后他把自己的一份藏好，就回去睡觉了。过了一会儿，第二个人醒了，也干了这样的事，他把椰子分成五堆以后，还剩下一只，也给了猴子，然后他藏起自己的一份，也回去睡了。此后，第三个人、第四和第五个人一点不差地全都这样做了。第二天早晨他们起身以后，把剩下的椰子分成五份，恰巧又剩下一个椰子给了猴子。

试编写程序计算出他们原来采集了多少只椰子？每次分后每人所得椰子数？

2 喝 啤 酒

宴会上数学家出了道难题：假定桌子上有三瓶啤酒，每瓶平均分给几个人喝，但喝各瓶啤酒的人数不相等，不过其中一个人喝到每瓶啤酒的量加起来恰巧是一整瓶，请问喝这三瓶啤酒各有多少人？

（一）笔算步骤和结果

喝这三瓶啤酒的人数为 2 人，3 人，6 人。即第一瓶两人喝，每人平均喝半瓶；第二瓶 3 人喝，每人平均喝 1/3 瓶；第三瓶 6 人喝，每人平均喝 1/6 瓶。其中一个人喝过三瓶，加起来的量（1/2 + 1/3 + 1/6）正好是一瓶。

（二）程 序 设 计

1）设计思路

设JX、JY、JZ是喝第一瓶、第二瓶、第三瓶啤酒的人数，利用三重小循环进行推算。由于题意指出喝各瓶酒的人数不相等，所以各循环的控制变量也应不同。寻找出符合条件的结果时，形象地打印出JX、JY、JZ各等于多少人。

2）框图　见图 2

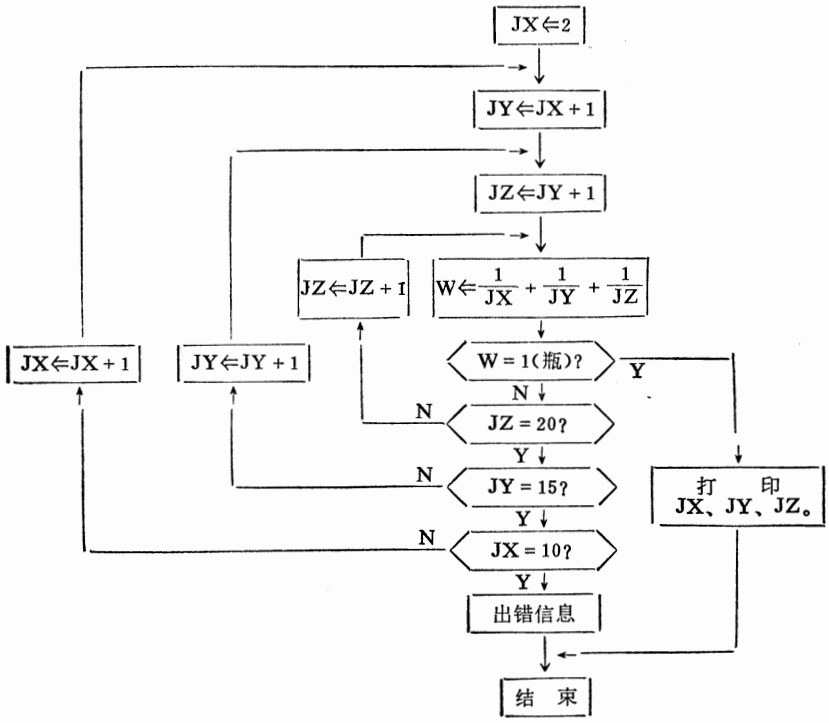


图 2

3）源程序及运行结果

```
      DO  10  JX=2, 10
      JX1=JX+1
      DO  10  JY=JX1, 15
      JZ1=JY+1
      DO  10  JZ=JZ1, 20
      W=1.0/FLOAT (JX) +1.0/FLOAT (JY) +1.0/FLOAT (JZ)
      IF ( (W-1.0).LE. 0.00001)  GO TO 20
10    CONTINUE
      STOP 'ERR'
20    WRITE (6, 30)  JX, JY, JZ
30    FORMAT (5X, 3HJX=, I1, 5X, 3HJY=, I1, 5X, 3HJZ=. I1)
      STOP
      END
JX=2     JY=3     JZ=6
```

（三）思 考 题

如果题中啤酒是四、五瓶，情况又是怎样？请编写程序推算出结果。

3 分牛的传说

老师为了鼓励学生练习上机操作，让学生自选算题上机练习，徐彼悦学生选道趣味题如下：

古印度有一个老人，他临死前把三个儿子叫到跟前说："我的遗产只有十九头牛，你们分了吧！老大分总数的1/2，老二分总数的1/4，老三分总数的1/5"。说完不久便死了。

（一）笔算步骤和结果

既要遵守不准宰牛的教规，又要执行老头的遗嘱。应该怎样分法，三个儿子都没想出恰当的办法，请教了当时有学问的人，也没有解决。有一天，一位农民牵了一头牛从门前路过，看见三兄弟唉声叹气，问明原因，思索片刻就说："这个问题很容易解决，我把自己这头牛借给你们，凑成二十头，老大分1/2应得十头，老二分 1/4 应得 5 头，老三分1/5应得 4 头，余下一头刚好还给我。"聪明的办法，绝妙的主意，事情就这样完满解决了。

过了若干年后，人们对这样分牛的方案提出了疑议：譬如老大似乎应分 9.5 头，最后怎么分了10头呢？老二、老三也不应该得那么多头。又过了若干年，随着数学的发展，有人证明这种分法是完全合理的。用等比数列求和的知识，就能予以证明。

我们知道，19头牛中，老大分19/2 头，老二分19/4 头，老三分 19/5 头，加起来是：$19\left(\frac{1}{2}+\frac{1}{4}+\frac{1}{5}\right)=19\times\frac{19}{20}<19$。他们并没有把牛都分完，还剩下$19\times\left(1-\frac{19}{20}\right)=\frac{19}{20}$（头），这剩下的 19/20 头应按照规定的比例再分，老大分$\frac{1}{2}\times\frac{19}{20}$头，老二分$\frac{1}{4}\times\frac{19}{20}$头，老三分$\frac{1}{5}\times\frac{19}{20}$头，然而仍没分完，仍剩下$\frac{19}{20}\left(1-\frac{19}{20}\right)=\frac{19}{20^2}$头，再按比例分下去，又剩下了$\frac{19}{20^2}\left(1-\frac{19}{20}\right)=\frac{19}{20^3}$（头）。从理论上讲，这样分下去永远也分不完，但每次所剩越来越少。

设老大分到的是S_1头，则$S_1=\frac{19}{2}+\frac{19}{2\times20}+\frac{19}{2\times20^2}+\frac{19}{2\times20^3}+\cdots\cdots=\lim\limits_{n\to\infty}\frac{\frac{1}{2}\left(1-\frac{1}{20^n}\right)}{1-\frac{1}{20}}=\lim\limits_{n\to\infty}10\left(1-\frac{1}{20^n}\right)=10$ 说明老大分 10 头是十分合理。同样，老二应分$S_2=\frac{19}{4}+\frac{19}{4\times20}+\frac{19}{4\times20^2}+\cdots=\lim\limits_{n\to\infty}5\left(1-\frac{1}{20^n}\right)=5$。老三分牛$S_3$头。$S_3=\frac{19}{5}+\frac{19}{5\times20}+\frac{19}{5\times20^2}+\frac{19}{5\times20^3}+\cdots=\lim\limits_{n\to\infty}4\left(1-\frac{1}{20^n}\right)=4$。

（二）程 序 设 计

1）设计思路

学生徐彼悦在处理上述算法有效数字位时，加上 0.005，徐彼悦分析算法后，认为可

以无限地分下去，不好处理，他假设分20次（分式中有20项）。整型量是无法分的，设三个儿子分得实型量分别为A、B、C，但分完后却应得整头牛，他在实型化整型处理有效数字位时，加上0.005，结果恰是答案的要求，学生们议论纷纷。徐彼悦请老师审阅，老师笑而不答。现将此程序设计让读者去批阅。

2）框图　见图3

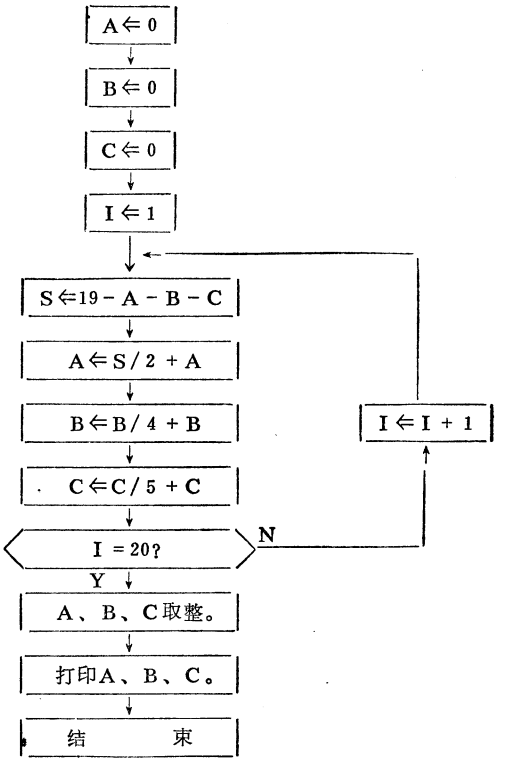


图 3

3）源程序及运行结果

```
      A=0.
      B=0.
      C=0.
      DO  10  I=1, 20
      S=19. -A-B-C
      A=S/2. +A
      B=S/4. +B
10    C=S/5. +C
      IA=IFIX ( (A*1000.+0.005) /1000.)
      IB=IFIX ( (B*1000.+0.005) /1000.)
      IC=IFIX ( (C*1000.+0.005) /1000.)
      WRITE (10, 20) IA, IB, IC
20    FORMAT (10X, 3HIA=, I2, 5X, 3HIB=, I2, 5X, 3HIC=, I2)
      STOP
```

```
      END
IA=10      IB=5      IC=4
```

（三）思 考 题

1）老师没批改，学生问："若这题一定要上机计算，应如何处理？"请读者批阅该作业。

2）老师反问这个学生，假若有47头牛，老大、老二、老三依次分1/3、1/4、1/5头牛，还得借几头牛呢！

3）另一位老师对该学生说，不用循环语句限制分多少次，而用比较前后两次值的误差，若误差相当小时，便不再分，将三人分得量实型化整型。改写学生的源程序运行结果如下：

```
      A1=0.0
      B1=0.0          A1、B1、C1分别表示老大、老二、老三这一次分得数量。
      C1=0.0
      I=0
10    I=I+1           I表示分的次数
      A2=A1
      B2=B1           A2、B2、C2分别表示老大、老二、老三上一次分得数量。
      C2=C1
      S=19.0-A1-B1-C1          分后还剩下S，以下再分。
      A1=S/2.0+A1
      B1=S/4.0+B1
      C1=S/5.0+C1
      D=ABS(A1-A2)+ABS(B1-B2)+ABS(C1-C2)     这一次和上一次
                                             分后二者误差D
      IF (D.GT.0.000001) GOTO 10
      IA=IFIX(A1+0.000001)
      IB=IFIX(B1+0.000001)     实型化整型
      IC=IFIX(C1+0.000001)
      WRITE(5,20) I,IA,IB,IC
20    FORMAT(5X,2HI=,I3/10X,3HIA=,I2,5X,3HIB=,I2,5X,3HIC=,
      I2)
      STOP
      END
```

源程序运行后打印结果如下

```
I=7                          分了七次
   IA=10     IB=5     IC=4    老大、老二、老三分别得的数量。
```

4 出 售 金 鱼

出售金鱼者决定将缸里的金鱼分五次全部卖出：

第一次卖出全部金鱼的一半加二分之一条金鱼；

第二次卖出剩余金鱼的三分之一加三分之一条金鱼；

第三次卖出剩余金鱼的四分之一加四分之一条金鱼；

第四次卖出剩余金鱼的五分之一加五分之一条金鱼。

现在还剩下11条金鱼，当然出售金鱼时不能切开或者有任何破损的，问这鱼缸里原有多少条金鱼？

（一）笔算步骤和结果

设总数为N

第一次卖的 $\frac{N}{2}+\frac{1}{2}$

第一次剩下的 $N-\frac{N}{2}-\frac{1}{2}=\frac{N-1}{2}$

第二次卖的 $\frac{N-1}{6}+\frac{1}{3}=\frac{N+1}{6}$

第二次剩下的 $\frac{N-1}{2}-\frac{N+1}{6}=\frac{2N-4}{6}=\frac{N-2}{3}$

第三次卖的 $\frac{N-2}{12}+\frac{1}{4}=\frac{N+1}{12}$

第三次剩下的 $\frac{N-2}{3}-\frac{N+1}{12}=\frac{4N-8-N-1}{12}=\frac{3N-9}{12}=\frac{N-3}{4}$

第四次卖的 $\frac{N-3}{20}+\frac{1}{5}=\frac{N+1}{20}$

第四次剩下的 $\frac{N-3}{4}-\frac{N+1}{20}=\frac{5N-15-N-1}{20}=\frac{4N-16}{20}=\frac{N-4}{5}$

依题意有 $\frac{N-4}{5}=11$

所以 $N=59$（条）

（二）程 序 设 计

1）设计思路

不许用刀切金鱼，由第一次卖出一半再加1/2可知，原有金鱼一定是奇数，故设原有金鱼总数N的初始值为13，最多不过1000。利用循环语句，据算法分析来寻找原有总数N，若不符合条件，总数N再加2，保证N总是奇数，符合条件时，打印出N值。

2）框图 见图4

3）源程序及运行结果

```
      N=13
10    N=N+2
      IF (N. GT. 1000)   GOTO  100
      M=N
      DO  20  I=2, 5
      A1= (FLOAT (M) +1.0) /FLOAT (I)
```

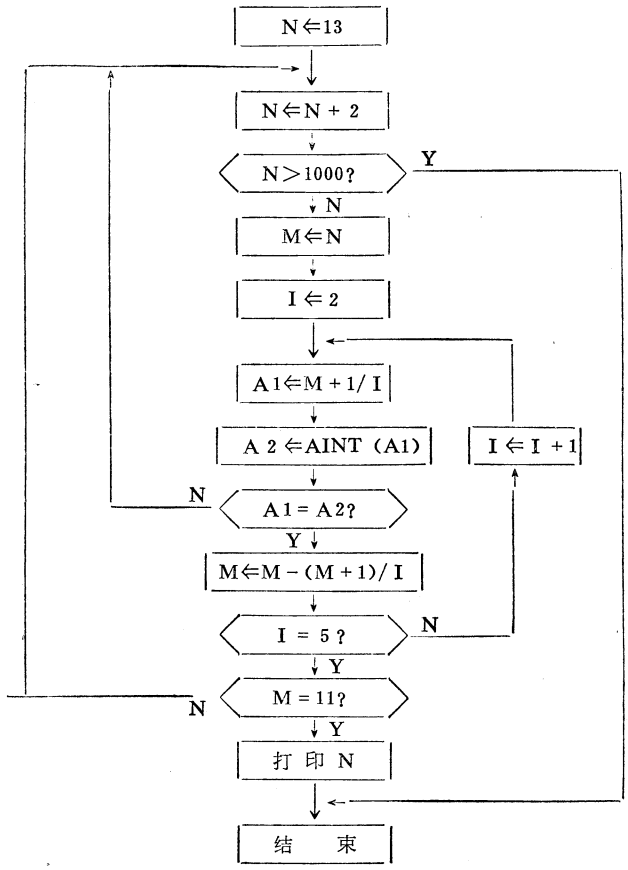


图 4

```
        A2 = AINT (A1)
        IF (A1. NE. A2)   GOTO  10
20      M = M -  (M + 1) /I
        IF (M. NE. 11)    GOTO  10
        WRITE (6, 30) N
30      FORMAT (1X, I 8 )
100     STOP
        END
```

59　　（结果为59条）

（三）思　考　题

郭乐迷把自己珍藏的几张唱片分别赠送给甲、乙、丙三人。他把总数的一半唱片和一张唱片的一半送给了甲，然后又把剩下的一半唱片和一张唱片的一半送给了乙，最后仅剩的一半唱片和半张唱片送给了丙，已全部赠完。问他珍藏几张唱片，甲、乙、丙三人每人各得几张唱片。

5 巧 分 桔 子

日本的中村义作教授是一位世界著名的数学游戏专家。他说一家农户以果园为生，有一次他们获得了空前大丰收，而且又恰逢春节，于是父亲便拿出一堆桔子，共2520只，分赏给他六个儿子，父亲按纸上算的数字分给儿子。分完后，父亲说："老大，把你分到的桔子分1/8给老二；老二拿到以后，连同原先的桔子分1/7给老三；老三拿到以后，连同原先桔子分1/6给老四；老四拿到以后，连同原先桔子分1/5给老五；老五拿到以后，连同原先桔子分1/4给老六；老六拿到以后，连同原先桔子分1/3给老大。结果，大家手里的桔子正好一样多，请问老大、老二、老三、老四、老五、老六每人原先分到的桔子各有多少只（桔子个数当然指整数）？

（一）笔算步骤和结果

本题有多种解法，仅介绍通过逆推法来找答案。这堆桔子共有2520只，由于六兄弟拿到桔子一样多，所以他们每人都分到2520/6＝420（只）。老六的420只是他分给老大后剩下的，在他分给老大之前应有420÷2/3＝630(只)，他给老大的桔子是630÷3＝210(只)。

老六未分桔子给老大之前，老大的桔子只有：420－210＝210（只），这些桔子是老大给老二1/8后剩下的，所以他原分到的桔子数应是：210÷7/8＝240（只）。他给老二的桔子是240÷8＝30（只）。

老二原有桔子加上老大给他的30只，他共有桔子：420÷6/7＝490（只），他原分到的桔子数应是490－30＝460（只），他给老三的是：490÷7＝70（只）。

老三原有桔子加上老二给他的70只，他共有桔子420÷5/6＝504（只），他原分到的桔子数为504－70＝434（只），他给老四的桔子数为504÷6＝84（只）。

老四原有桔子加上老三给他的84只，他共有420÷4/5＝525（只），所以他原来分到的桔子为525－84＝441（只）。他给老五105只。

老五原有桔子加上老四给他的105只，他共有420÷3/4＝560（只），所以他原来分到的桔子为560－105＝455（只），他给老六140只。

老六原有的桔子加上老五给的140只，他共有420÷2/3＝630（只），所以他原分到490（只）。

（二）程 序 设 计

1）设计思路

定义数组MA（6），MB（6），置放六人分的数量，一个数组做为变量，另一个数组暂存之，利用多重循环，据算法分析来寻找每人应得的数量，最后将老大至老六应分得的桔子数，依次分行打印出。

2）框图　见图5

3）源程序及运行结果

```
DIMENSION  MB(6), MA(6)
DO  10  MA1=236, 270
DO  20  MA2=455, 490
DO  30  MA3=430, 480
DO  40  MA4=431, 470
```

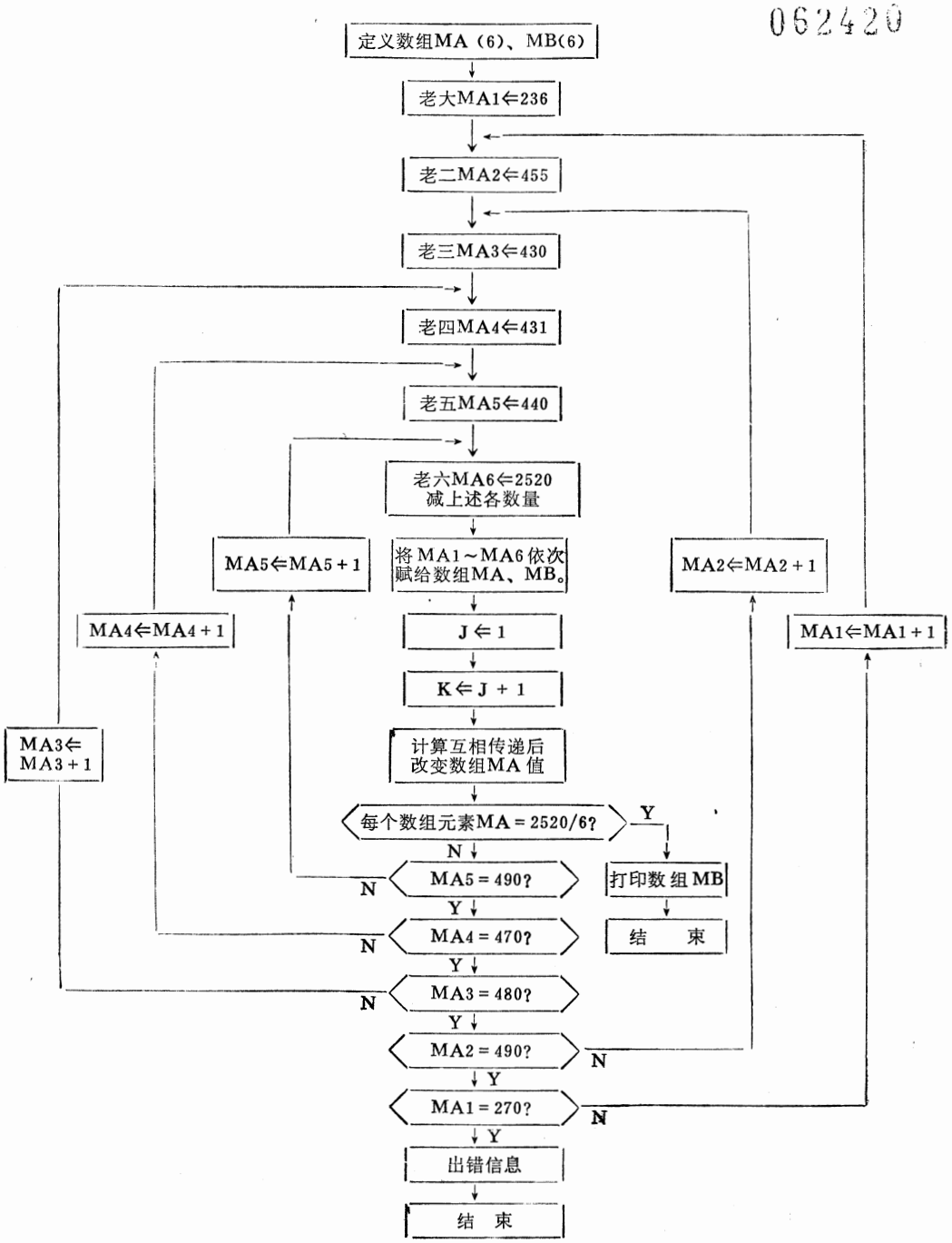


图 5

DO 50 MA5 = 440, 490

MA6 = 2520 − MA1 − MA2 − MA3 − MA4 − MA5

MA(1) = MA1

MA(2) = MA2

MA(3) = MA3

```
      MA (4) =MA4
      MA (5) =MA5
      MA (6) =MA6
      MB (1) =MA1
      MB (2) =MA2
      MB (3) =MA3
      MB (4) =MA4
      MB (5) =MA5
      MB (6) =MA6
      DO  60  J=1, 5
      K=J+1
      MA(K) =MA(K) +MA(J) / (9-J)          是老大给老二，老二给老
                                           三……老五给老六。
60    MA(J) =MA(J) * (9-J-1) / (9-J)       如上给后，自己还剩下的数量。
      MA(1) =MA(1) +MA(6) /3               是老六给老大的。
      MA(6) =MA(6) *2/3                    老六给后，自己还剩下的。
      DO  70  MT=1, 6
      IF (MA (MT) . NE. 2520/6)  GO  TO  50
70    CONTINUE
      GOTO  80
50    CONTINUE
40    CONTINUE
30    CONTINUE
20    CONTINUE
10    CONTINUE
      STOP  'ERR'
80    DO  90  MS=1, 6
      WRITE (6, 100)  MB (MS)
100   FORMAT  (1X, I8)
90    CONTINUE
      STOP
      END
```

依次打印出老大～老六分的桔子数

```
240
460
434
441
455
490
```

（三）思 考 题

新年晚会老师给大家分糖，手端着一盘糖，让第一个同学先拿 1 块糖，再把盘中的糖

分七分之一给他，然后让第二个同学拿2块糖，再从盘中的糖分七分之一给他。第三个同学拿3块糖后，仍从盘里的糖分七分之一给他，照这个办法分下去，最后一个同学自己拿完糖后，糖恰好分完，而且每人分到的糖块数相同，问共有几人？每人分几块？

6 整元兑换硬币

用一元人民币兑换一分，二分，五分的硬币，共有几种不同的分法？

（一）算法分析和结论

先从大分额的5分硬币着手，一元人民币可以兑换5分硬币个数的可能性有0，1，2，3，…，20。从这个排序一一进行分析2分的个数可能性有0，1，2，3，…。总钱数1元（100分）减去每一种排序中的5分和2分钱，剩下便是兑换1分钱的个数。如此计算，共有541种兑换方法。

（二）程序设计

1）设计思路

打印表头：每行打印四组，每组是由5分（5F）、2分（2F）、1分（1F）三种数字组成。5分币赋给M1，建立数组M2（4），L2（4），分别放入2分和1分硬币。各种兑换方法，详见源程序运行后打印的结果。

2）框图　见图6

3）源程序及运行结果

```
      DIMENSION  M2(4), L2(4)
      WRITE (6, 40)
      N=0
      DO  20  I=1, 21
      M1=I-1
      L=100-5*M1
      L1=L/2+1
      K=0
      DO  20  J=1, L1
      K=K+1
      M2(K) =J-1
      L2(K) =L-M2(K) *2
      N=N+1
      IF (K. LT. 4)  GO  TO  20
      WRITE (6, 30)  (M1, M2(K), L2(K), K=1, 4)
      K=0
20    CONTINUE
      WRITE (6, 30)  (M1, M2(I), L2(I), I=1, K)
30    FORMAT (3X, 4(3I5, 1H:) /)
      WRITE (6, 50) N
50    FORMAT (20X, I5)
40    FORMAT (3X, 4(3X, 2H5F, 3X, 2H2F, 3X, 2H1F, 1H:) /3X, 64
      (1H~) )
```

```
STOP
END
```

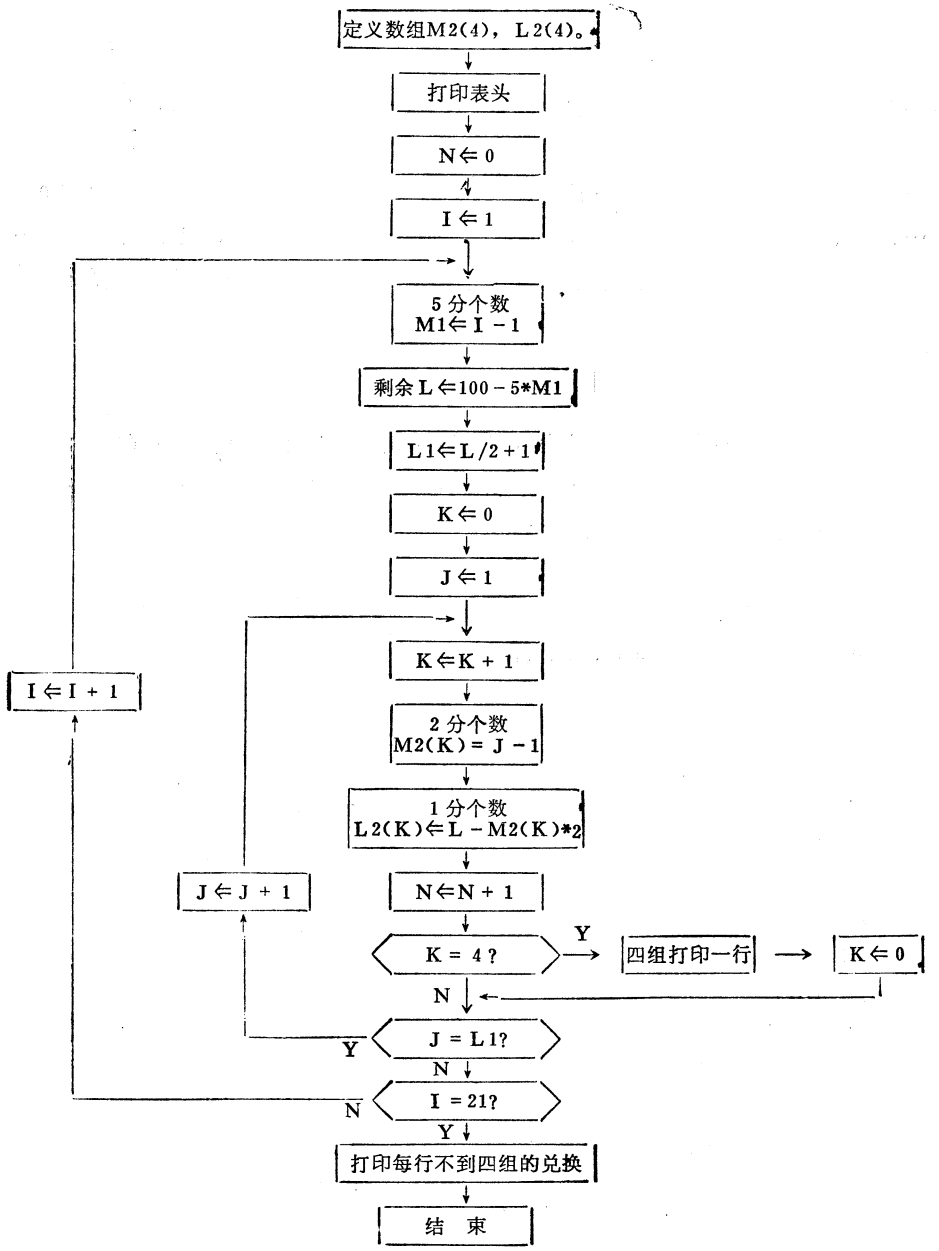


图 6

程序运行打印输出结果如下：

5F	2F	1F：	5F	2F	1F：	5F	2F	1F：	5F	2F	1F：
0	0	100：	0	1	98：	0	2	96：	0	3	94：
0	4	92：	0	5	90：	0	6	88：	0	7	86：
0	8	84：	0	9	82：	0	10	80：	0	11	78：

0	12	76 :	0	13	74 :	0	14	72 :	0	15	70 :
0	16	68 :	0	17	66 :	0	18	64 :	0	19	62 :
0	20	60 :	0	21	58 :	0	22	56 :	0	23	54 :
0	24	52 :	0	25	50 :	0	26	48 :	0	27	46 :
0	28	44 :	0	29	42 :	0	30	40 :	0	31	38 :
0	32	36 :	0	33	34 :	0	34	32 :	0	35	30 :
0	36	28 :	0	37	26 :	0	38	24 :	0	39	22 :
0	40	20 :	0	41	18 :	0	42	16 :	0	43	14 :
0	44	12 :	0	45	10 :	0	46	8 :	0	47	6 :
1	0	95 :	1	1	93 :	1	2	91 :	1	3	88 :
1	4	87 :	1	5	85 :	1	6	83 :	1	7	81 :
1	8	79 :	1	9	77 :	1	10	75 :	1	11	73 :
1	12	71 :	1	13	69 :	1	14	67 :	1	15	65 :
1	16	63 :	1	17	61 :	1	18	59 :	1	19	57 :
1	20	55 :	1	21	53 :	1	22	51 :	1	23	49 :
1	24	47 :	1	25	45 :	1	26	43 :	1	27	41 :
1	28	39 :	1	29	37 :	1	30	35 :	1	31	33 :
1	32	31 :	1	33	29 :	1	34	27 :	1	35	25 :
1	36	23 :	1	37	21 :	1	38	19 :	1	39	17 :
1	40	15 :	1	41	13 :	1	42	11 :	1	43	9 :
1	44	7 :	1	45	5 :	1	46	3 :	1	47	1 :
2	0	90 :	2	1	88 :	2	2	86 :	2	3	84 :
2	4	82 :	2	5	80 :	2	6	78 :	2	7	76 :
2	8	74 :	2	9	72 :	2	10	70 :	2	11	68 :
2	12	66 :	2	13	64 :	2	14	62 :	2	15	60 :
2	16	58 :	2	17	56 :	2	18	54 :	2	19	52 :
2	20	50 :	2	21	48 :	2	22	46 :	2	23	44 :
2	24	42 :	2	25	40 :	2	26	38 :	2	27	36 :
2	28	34 :	2	29	32 :	2	30	30 :	2	31	28 :
2	32	26 :	2	33	24 :	2	34	22 :	2	35	20 :
2	36	18 :	2	37	16 :	2	38	14 :	2	39	12 :
2	40	10 :	2	41	8 :	2	42	6 :	2	43	4 :
3	0	85 :	3	1	83 :	3	2	81 :	3	3	79 :
3	4	77 :	3	5	75 :	3	6	73 :	3	7	71 :
3	8	69 :	3	9	67 :	3	10	65 :	3	11	63 :
3	12	61 :	3	13	59 :	3	14	57 :	3	15	55 :
3	16	53 :	3	17	51 :	3	18	49 :	3	19	47 :
3	20	45 :	3	21	43 :	3	22	41 :	3	23	39 :
3	24	37 :	3	25	35 :	3	26	33 :	3	27	31 :
3	28	29 :	3	29	27 :	3	30	25 :	3	31	23 :
3	32	21 :	3	33	19 :	3	34	17 :	3	35	15 :
3	36	13 :	3	37	11 :	3	38	9 :	3	39	7 :
4	0	80 :	4	1	78 :	4	2	76 :	4	3	74 :

……　　（因较多，中间省略）

12	12	16；	12	13	14；	12	14	12；	12	15	10；
12	16	8；	12	17	6；	12	18	4；	12	19	2；
13	0	35；	13	1	33；	13	2	31；	13	3	29；
13	4	27；	13	5	25；	13	6	23；	13	7	21；
13	8	19；	13	9	17；	13	10	15；	13	11	13；
13	12	11；	13	13	9；	13	14	7；	13	15	5；
14	0	30；	14	1	28；	14	2	26；	14	3	24；
14	4	22；	14	5	20；	14	6	18；	14	7	16；
14	8	14；	14	9	12；	14	10	10；	14	11	8；
14	12	6；	14	13	4；	14	14	2；	14	15	0；
15	0	25；	15	1	23；	15	2	21；	15	3	19；
15	4	17；	15	5	15；	15	6	13；	15	7	11；
15	8	9；	15	9	7；	15	10	5；	15	11	3；
16	0	20；	16	1	18；	16	2	16；	16	3	14；
16	4	12；	16	5	10；	16	6	8；	16	7	6；
17	0	15；	17	1	13；	17	2	11；	17	3	9；
17	4	7；	17	5	5；	17	6	3；	17	7	1；
18	0	10；	18	1	8；	18	2	6；	18	3	4；
20	0	0；									

541　（种兑换方法）

（三）思　考　题

交给一个孩子一元钱，让他换成硬币。其中他应换若干个2分的，还应换一些1分的，1分的个数是2分的个数十倍。其余的都换成5分的。问：这个孩子每类硬币换成多少？

7 老人分马

一个老人有若干匹马，记为n，他把马分给三个儿子，大儿子得X匹，二儿子得Y匹，小儿子得Z匹，并且满足X| n+1，Y| n+1，Z| n+1，X>Y>Z，问老人的马的匹数（即n），有多少种可能？

（一）笔算步骤和结果

我们详细分析本题，这可以化成一个单位分数的问题，设

$$n+1=Xa,\quad n+1=Yb,\quad n+1=Zc。$$

这里a，b，c表示整数（用字母a、b、c…表示整数）。故由X+Y+Z=n得

$$\frac{1}{a}+\frac{1}{b}+\frac{1}{c}=\frac{n}{n+1},\quad a|n+1,\ b|n+1,\ c|n+1,\ a<b<c。\qquad(1)$$

现在，就来给出（1）中的n能够取多少个自然数值。

很明显，因为$\frac{n}{n+1}<1$，故$a\geqslant 2$。我们来证明不可能有$a>2$。在证明之前，先

介绍初等数论中一个重要定理，即算术基本定理：

设正整数 $n>1$，如果不计素因数的次序，则只有一种方法把 n 分解成素因数的连乘积。

由此定理可知，如果 $n>1$，把 n 的相同的素因数合并成幂数形式，则 n 只能分解成一种形式：

$$n=P_1^{a_1}P_2^{a_2}\cdots P_S^{a_S},\ S\geqslant 1,$$

这里 $P_1,\cdots P_S$ 是不同的素数，$P_1<\cdots<P_S$，$a_1,\cdots a_S$ 都是正整数，我们把 $n=p_1^{a_1}\cdots P_S^{a_S}$ 叫作 n 的标准分解式。例如1650的标准分解式为

$$1650=2\times 3\times 5^2\times 11.$$

如果（1）中 $a>2$，设 $M=abc$，分三种情形来讨论：

1）如果 $M=P_1^{a_1}\cdots P_S^{a_S}$ 是M的标准分解式，$P_1,\cdots,P_S$ 是奇素数，当 $S\geqslant 2$ 时，则由 $P_1P_2\mid M=abc$，$a\mid n+1$，$b\mid n+1$，$c\mid n+1$，总有 $P_1P_2\mid n+1$，而 $P_1\geqslant 3$，$P_2\geqslant 5$，故 $n+1\geqslant P_1P_2\geqslant 15$；当 $S=1$ 时，可设 $a=P_1^{e_1}$，$b=P_1^{e_2}$，$c=P_1^{e_3}$，由 $2<a<b<c$，故 $e_3>e_2>e_1\geqslant 1$ 是正整数，$e_3\geqslant 3$，故 $P_1^3\mid c$，而 $c\mid n+1$，于是 $n+1\geqslant P_1^3\geqslant 27$。

2）如果 $M=2^a$，类似1）中 $S=1$ 的讨论知 $2^4\mid c$，故有 $n+1\geqslant 16$。

3）如果 $M=2^aP_1^{a_1}\cdots P_S^{a_S}$ 是M的标准分解式，当 $S\geqslant 2$ 时，则由 $2P_1P_2\mid M=abc$，$a\mid n+1$，$b\mid n+1$，$c\mid n+1$，总有 $2P_1P_2\mid n+1$，而 $2P_1P_2\geqslant 2\times 3\times 5=30$，于是 $n+1\geqslant 2P_1P_2\geqslant 30$；当 $S=1$ 时，由 $2<b<c$，$a\mid n+1$，$b\mid n+1$，$c\mid n+1$，总有 $2P_1^2\mid n+1$ 或 $2^2P_1\mid n+1$，故有 $n+1\geqslant 12$。

总之，归纳以上三种情形，我们得出 $n+1\geqslant 12$，由（1）（得）

$$\frac{11}{12}\leqslant 1-\frac{1}{n+1}=\frac{n}{n+1}=\frac{1}{a}+\frac{1}{b}+\frac{1}{c}\leqslant\frac{1}{3}+\frac{1}{4}+\frac{1}{5}=\frac{47}{60},$$ 这是矛盾的。

因此，$a=2$ 代入（1）得

$\frac{1}{b}+\frac{1}{c}=\frac{1}{2}-\frac{1}{n+1}$，$b\mid n+1$，$c\mid n+1$，$2<b<c$。（2）如果 $b\geqslant 5$，当 $n+1\geqslant 8$ 时，由（2）得 $\frac{3}{8}=\frac{1}{2}-\frac{1}{8}\leqslant\frac{1}{2}-\frac{1}{n-1}=\frac{1}{b}+\frac{1}{c}\leqslant\frac{1}{5}+\frac{1}{6}=\frac{11}{30}$，这是矛盾的，故 $n+1<8$，由此和 $5\leqslant b<c\mid n+1$，得出

$$8>n+1\geqslant c\geqslant b+1\geqslant 6,$$

故有 $n+1=6$，$b=5$，但 $5\nmid n+1=6$（符号 $\nmid$ 表示不整除）或 $n+1=7$，$b=5$ 或 $n+1=7$，$b=6$，都与 $b\mid n+1$ 矛盾，故有 $b=3$ 或 $b=4$。当 $b=3$ 时，由（2）得

$$\frac{1}{c}=\frac{1}{6}-\frac{1}{n+1},\ c\mid n+1,\ c>3 \qquad (3)$$

再由（3）可得 $\frac{1}{c}=\frac{1}{6}-\frac{1}{n+1}\geqslant\frac{1}{6}-\frac{1}{c}$

和 $\frac{1}{6}-\frac{1}{c}=\frac{1}{n+1}>0$

故 $7\leqslant c\leqslant 12$

$c=7$，代入（3）得出 $n=41$，$a=2$，$b=3$，$c=7$；

c = 8，代入（3）得出 n = 23， a = 2， b = 3， c = 8；

c = 9，代入（3）得出 n = 17， a = 2， b = 3， c = 9；

c = 10，11，无满足条件的解；

c = 12，代入（3）得出 n = 11， a = 2， b = 3， c = 12；

当 b = 4 时，由（2）得 $\frac{1}{c}=\frac{1}{4}-\frac{1}{n+1}$，c| n+1，c>4；　　（4）

再由（4）得 $\frac{1}{c}=\frac{1}{4}-\frac{1}{n+1}\geqslant\frac{1}{4}-\frac{1}{c}$；

故　　$5\leqslant c\leqslant 8$。

c = 5，代入（4）得 n = 19， a = 2， b = 4； c = 5；

c = 6，代入（4）得 n = 11， a = 2， b = 4； c = 6；

c = 7，无解；

c = 8，代入（4）得 n = 7， a = 2， b = 4， c = 8。

通过以上论述，我们证明了这个问题中马的匹数共有六种可能，而分法共有七种：

	n	a	b	c	→	x	y	z
1	7	2	4	8		4	2	1
2	11	2	4	6		6	3	2
3	11	2	3	12		6	4	1
4	17	2	3	9	→	9	6	2
5	19	2	4	5		10	5	4
6	23	2	3	8		12	8	3
7	41	2	3	7		21	14	6

注：(1) 上表左半部要对照推导开始时$\frac{1}{a}+\frac{1}{b}+\frac{1}{c}=\frac{n}{n+1}$的公式；

(2) 表右半部是分析开始时n + 1 = Xa，n + 1 = Yb，n + 1 = Zc而推导的。

（二）程 序 设 计

1）设计思路

在算法分析过程中，运用了较多的数学知识——单位分数来解决，最后推导出N、a、b、c，进一步转换为老大、老二、老三分得的马匹数X、Y、Z。程序设计时，直接采用求出X、Y、Z的方法。

总数N必然为奇数，初始值令N = 5。利用三重循环寻找三人分得的头数，若不满足条件时，N再加2，保证N总为奇数，符合条件则打印出一组解，再继续循环寻找其他组解。当N大于100时，便停止。

2）框图　见图7

3）源程序及运行结果

```
C       PROGRAM  FM
        WRITE (6, 60)
60      FORMAT (10X, 1HN, 9X, 1HX, 9X, 1HY, 9X, 1HZ)
        N=5
```

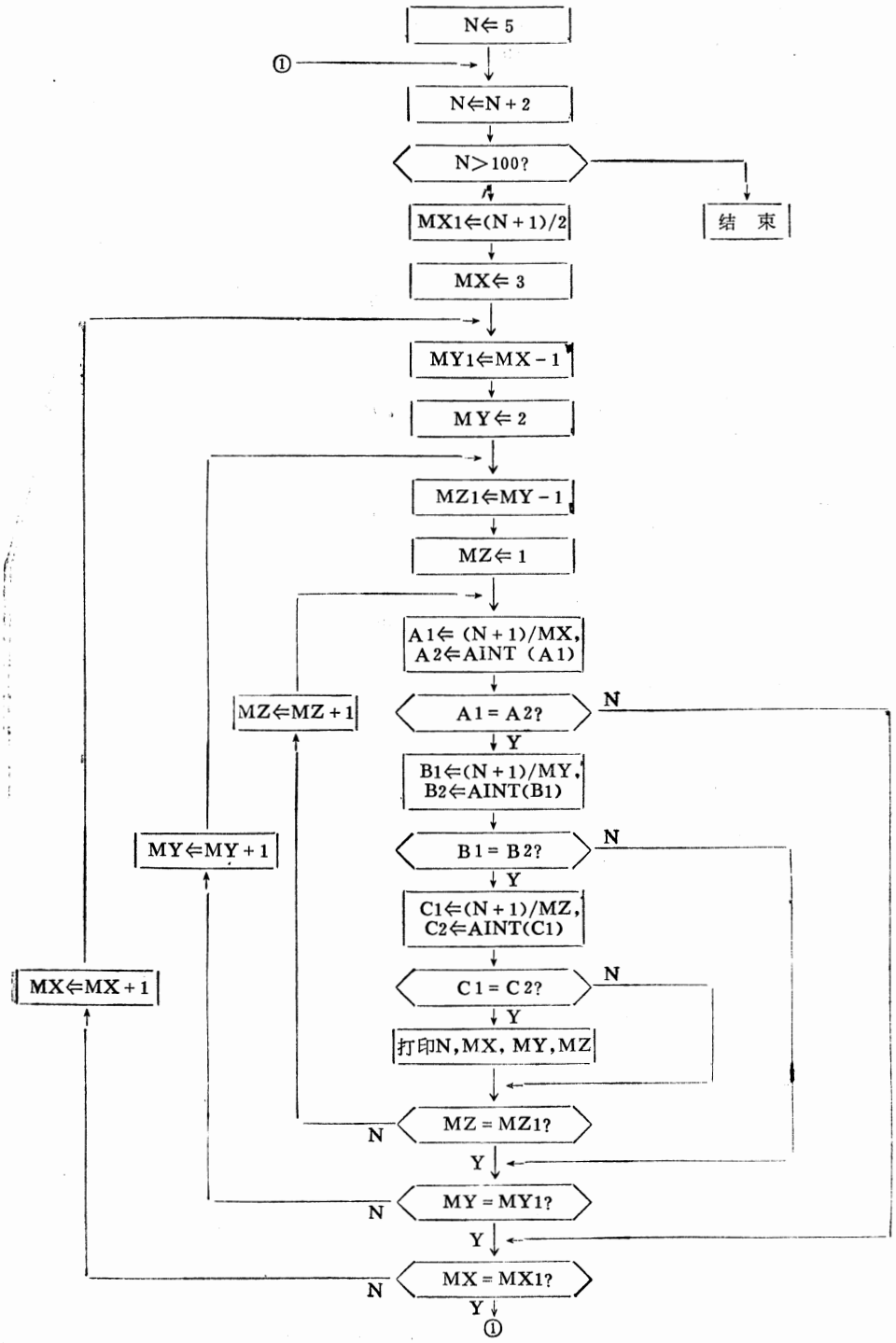

N⇐5
①
N⇐N+2
N>100?
结 束
MX1⇐(N+1)/2
MX⇐3
MY1⇐MX-1
MY⇐2
MZ1⇐MY-1
MZ⇐1
A1⇐(N+1)/MX,
A2⇐AINT(A1)
MZ⇐MZ+1
A1=A2?
N
Y
B1⇐(N+1)/MY,
B2⇐AINT(B1)
MY⇐MY+1
B1=B2?
N
Y
C1⇐(N+1)/MZ,
C2⇐AINT(C1)
MX⇐MX+1
C1=C2?
N
Y
打印N,MX,MY,MZ
MZ=MZ1?
N
Y
MY=MY1?
N
Y
MX=MX1?
N
Y
①

图 7

```
50      N=N+2
        IF (N. GT. 100)  GOTO  100
        MX1= (N+1) /2
        DO  10  MX=3, MX1
        MY1=MX-1
        DO  20  MY=2, MY1
        MZ1=MY-1
        DO  30  MZ=1, MZ1
        A1= (FLOAT (N) +1.0) /FLOAT (MX)
        A2=AINT (A1)
        IF (A1. NE. A2)  GOTO  10
        B1= (FLOAT (N) +1.0) /FLOAT (MY)
        B2=AINT (B1)
        IF (B1. NE. B2)  GOTO  20
        C1= (FLOAT (N) +1.0) /FLOAT (MZ)
        C2=AINT (C1)
        IF (C1. NE. C2)  GOTO  30
        IF (MX+MY+MZ. NE. N)  GOTO  30
        WRITE (6, 40)  N, MX, MY, MZ
40      FORMAT (1X, 4I10)
30      CONTINUE
20      CONTINUE
10      CONTINUE
        GOTO 50
100     STOP
        END
```

程序运行后打印结果如下

N	X	Y	Z
7	4	2	1
11	6	3	2
11	6	4	1
17	9	6	2
19	10	5	4
23	12	8	3
41	21	14	6

（三）思 考 题

1）设有23个苹果，分给四个孩子，老大、老二、老三、老四所得数分别不得超过9、8、7、6个苹果，问共有多少种不同的分法？并找出每一种具体分法。

2）欧拉出的一道题——父亲临终时，让按下列方式分配他的遗产：大儿子分得100克朗和剩下财产的1/10，二儿子分得200克朗和剩下财产的1/10，三儿子分得300克朗和剩下财产的1/10，…依次类推，最后发现这种分法好极了，因为所有儿子分得的钱数恰好相

等。问他共有几个儿子？每个儿子分得多少遗产？

8 左右衣袋钱数相等

当伯特走进家门时，玛利笑了起来说："怎么回事，你走起路来，叮哨叮哨作响？"

伯特回答说："那当然，我口袋里装满了一角与二角五分的硬币，但还不到二十元。"

"那可是个累赘。"玛利说，"你知道你到底有多少钱吗？"

伯特笑着说："你猜吧！我左右两个口袋中的钱数相同，左口袋中每种硬币的数目相等，而右口袋中的每种硬币的钱数相等，你能算出来吗？"

（一）笔算步骤和结果

设每个口袋中的钱数为2X分。因左口袋中两种硬币的个数相等，故1角与2角5分的硬币数均为$\frac{2X}{10+25}=\frac{2X}{35}$（个）。

因右口袋中两种硬币的钱数相等，都为X分。故1角的硬币数均为$\frac{X}{10}$个，2角5分的硬币为$\frac{X}{25}$个。

于是，两个口袋共有1角的硬币$\left(\frac{2X}{35}+\frac{X}{10}\right)$个，2角5分的硬币为$\left(\frac{2X}{35}+\frac{X}{25}\right)$个。

两种硬币数的比值为$\left(\frac{2X}{35}+\frac{X}{10}\right)/\left(\frac{2X}{35}+\frac{X}{25}\right)=\frac{11X}{70}/\frac{17X}{175}=\frac{55}{34}$

即一角硬币的总数是55的整倍数：55K

二角五分硬币的总数是34的整倍数：34K

K为某一正整数，这样，两个口袋中的总钱数为$10\times55K+25\times34K=1400K$（分）

因总钱数不到20元，故有$1400K<2000$。

从上式可以看出，K只能等于1（K = 1）。

所以，总钱数为14元，每个口袋中的钱数为7元，左口袋中一角和二角五分的硬币数均为20个，右口袋中一角的硬币数为35个，二角五分的硬币数为14个。

（二）程序设计

1）设计思路

把1角、2角5分的硬币分别用10、25分表示。用循环语句找右口袋钱的总数，控制变量I初值为10（1角），因题意右袋钱总数不到10元，两种硬币相等，所以1角钱的总数不过500（即为变量终值I）。该I能被25（分）整除，则2I便是右口袋总钱数M。转子程序，找左口袋两种硬币相等和个数JK，又一条件是左右钱数相等。将符合条件的结果打印出等式，以便观察。

2）框图　见图8

3）源程序及运行结果

```
JP=0
JS=0
DO  10  I=10, 500, 10                    右口袋1角（10分）总钱数I
```

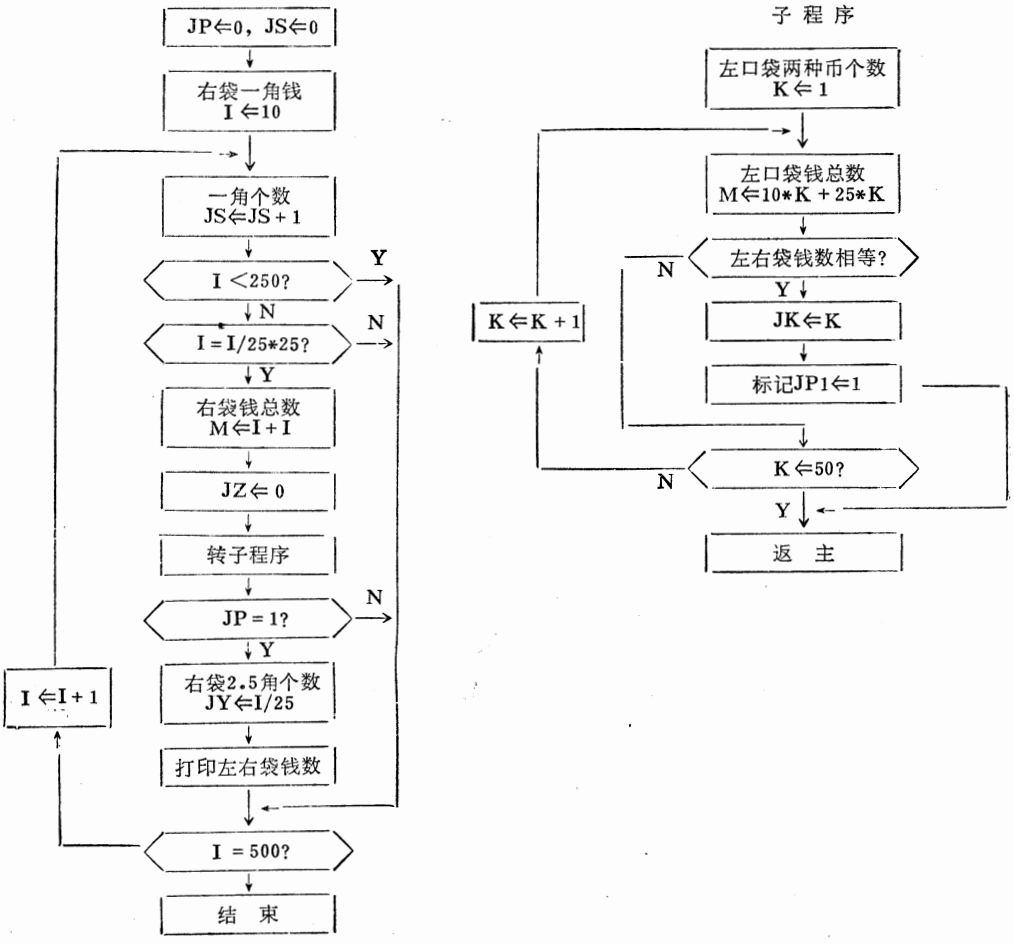


图 8

```
      JS = JS + 1                              1角（10分）钱的个数JS
      IF (I. LE. 250) GO  TO  10
      IF (I. NE. I/25*25) GO  TO  10
      M = I + I                                右口袋总钱数是2I
      JZ = 0                                   左口袋两种钱的个数JZ
      CALL  PS (M, JP, JZ)
      IF (JP. NE. 1) GO  TO  10
      JY = I/25                                右袋25分（2角5分）的个数JY
      JP = 0
      WRITE (6, 15) M, JS, JY, M, JZ, JZ
15    FORMAT (/5X, 6HRIGHT:, I4, 4H=10*, I2, 4H+25*, I2
     1 /6X, 5HLEFT:, I4, 4H=10*, I2, 4H+25*, I2)
10    CONTINUE
      STOP
      END
```

```
      SUBROUTINE  PS (N, JP1, JK)          子程序判左右口袋钱相等否
      DO  10  K=1, 50
      M=10*K+25*K
      IF (N. NE. M) GO  TO  10
      JK=K
      JP1=1                                左右口袋钱相等置标记JP1
      GO  TO  20
10    CONTINUE
20    RETURN
      END

      RIGHT：700=10*35+25*14               右口袋装的钱
      LEFT：700=10*20+25*20                左口袋装的钱
```

（三）思 考 题

资本主义国家，男女同工不同酬——巴勃拿着利润的钱，兴致勃勃地回到办公室，让女秘书把其它三个雇员找来，说："为了感谢大家，这是230元红利，将分给你们四人。我是按照你们跟我共事的年头比例分配的，但是每个男人每年的钱为每个女人每年的钱的一倍半"。接着他便把四个装好钱的纸袋分给他们四人。

这四个人分别跟巴勃一块工作 2 年、5 年、6 年、7 年，你能分辨出四人中的男人与女人的数目及他们每人应得的钱数吗？

9 传 递 小 球

有12堆小球，首尾相连围成一圈。每堆球数是 3 的倍数，如 3，6，9，…；36，将每堆分成三等份。三分之一留在本堆，另外二份分别给左、右邻堆。各堆准备好后同时传递。传送后再依上法进行。当某堆球数不是 3 的倍数时，可从布袋备用球补充 1 至 2 球。请论证若干次后，可以使各堆球相等。

（一）笔算步骤和结果

我们论证更强的结论，不限制具体堆数和每堆的个数。先分析调整一次后的情形。设在此次调整前，球数最多的堆中有球3m个，球数最少的堆中有球3n个，当然m＞n。经过一次调整，可以得出下面的结论：

（1）调整后每堆球数仍在3n与3m之间。

设某堆原有球3l个，与它左、右相邻的两堆中，原来分别有球3k和3h个，那么

$$n \leqslant l \leqslant m,\quad n \leqslant k \leqslant m,\quad n \leqslant h \leqslant m,$$

由此得

若l + k + h是 3 的整数倍，则该堆调整后有球l + k + h个，显然它在3n与3m之间。

若l + k + h不是 3 的整数倍，这时l + k + h＜3m，需补入一球或两球，使该堆球数仍为 3 的整数倍，容易想到，该堆调整后球的总数仍不会超过3m。

（2）原来有球3n个的堆中，调整后至少有一堆的球数要增多。

原来有球3n个的堆中，至少有一堆，它的左、右相邻的堆中有球3k＞3n或3h＞

3 n （否则原来各堆球数相等，用不着调整），于是，调整后这堆至少有球 n + k + h ＞3 n 个。

根据以上的结论，可知每调整一次，有球3 n 个的堆数就至少减少一个；经过有限次调整后，每堆球数就都大于3 n 了。又因有球最多的堆中，经过调整，球数不增加，这样，堆中最多球数与最少球数的差数就要减少，但此差数是一有限数，所以经过有限次调整后，此差数就等于零，也就是说，每堆小球个数相等了。

（二）程 序 设 计

1）设计思路

定义数组 JA(12)、JB(12)表示12堆球，其一做暂存。将 3 ， 6 ， 9 ，…，36 依次赋给数组 JA(12) 。用循环语句将12堆的每堆分成三等份，同时传递后，立即判该堆是否为 3 的倍数，若不是 3 的倍数时，补充 1 ～ 2 球。重复上述传递，直到各堆球数均等。每传递一次，打印出次数 J 和各堆的球数，形象地观察出传递、调整过程。程序运行过程见打印结果。

2）框图　见图9

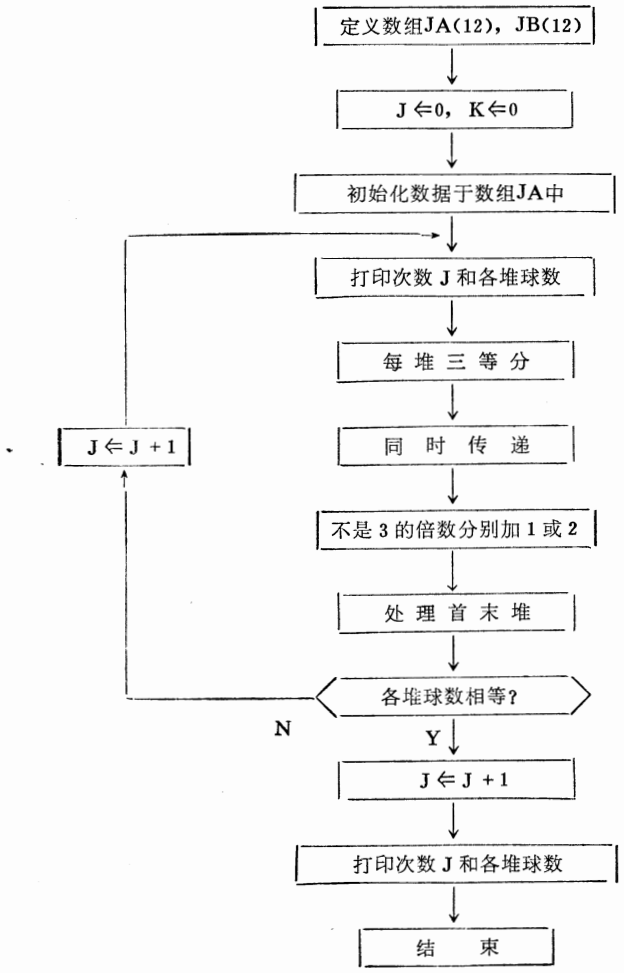


图 9

3) 源程序及运行结果

```
      DIMENSION  JA(12) , JB(12)
      J = 0                                          传递次数J
      K = 0
      DO  10  I = 1, 12                              数据初始化
      K = K + 3
      JA(I) = K
10    JB(I) = 0
25    WRITE (6, 20) J, (JA(KK), KK = 1, 12) 打印第J次：各堆球数。
20    FORMAT (5X, 2HJ = ,I2, 3H:    , 12 (I2, 3X)/)
      DO  30  I = 1, 12                        DO  30  将12堆各分三等分
      JA (I) = JA (I)/3
30    JB (I) = JA (I)
      DO  40  I = 2, 11                        DO  40  是传递和判断
      JA(I) = JA(I) + JB(I - 1) + JB(I + 1)
      IF (JA(I).EQ.JA(I)/3*3) GO  TO  40
      IF (JA(I).EQ.JA(I)/2*2) GO  TO  45
      JA(I) = JA(I) + 1
45    JA(I) = JA(I) + 2
40    CONTINUE
      JA (1) = JA(1) + JB(12) + JB(2)              处理首堆
      IF (JA(I).EQ.JA(I)/3*3)GO  TO  50
      IF (JA(I).EQ.JA(I)/2*2)GO  TO  47
      JA (I) = JA(I) + 1
47    JA(I) = JA(I) + 2
50    JA(12) = JA(12) + JB(11) + JB(1)             处理末堆
      IF (JA(12).EQ.JA(12)/3*3)GO  TO  60
      IF (JA(12).EQ.JA(12)/2*2)GO  TO  57
      JA(12) = JA(12) + 1
57    JA(12) = JA(12) + 2
60    DO  70  I = 1, 11
      IF(JA(I).NE.JA(I + 1))GO  TO  80          各堆球数不等，继续调整。
70    CONTINUE
      J = J + 1
      WRITE(6, 20) J, (JA(KK), KK = 1, 12)     各堆球数相等
      STOP
80    J = J + 1
      GO  TO  25
      END
```

程序运行打印结果如下

```
J =  0:    3   6   9  12  15  18  21  24  27  30  33  36
J =  1:   15   6   9  12  15  18  21  24  27  30  33  24
J =  2:   15  12   9  12  15  18  21  24  27  30  32  24
J =  3:   20  12  14  12  15  18  21  24  27  32  30  26
```

J = 4:	18	16	12	16	15	18	21	24	27	32	30	24
J = 5:	22	15	16	16	18	18	21	24	27	32	30	24
J = 6:	22	20	15	18	20	22	21	24	27	32	30	28
J = 7:	24	18	20	20	22	22	24	24	27	32	32	28
J = 8:	26	22	18	22	22	24	26	28	27	32	32	27
J = 9:	24	21	22	22	24	26	28	28	30	32	32	27
J =10:	24	24	21	24	26	28	28	30	32	30	32	27
J =11:	28	26	26	26	28	28	30	32	30	30	32	27
J =12:	28	28	24	28	28	30	32	30	30	30	32	30
J =13:	30	28	28	28	30	32	30	30	30	30	30	32
J =14:	32	30	27	30	32	30	30	30	30	30	30	30
J =15:	30	32	32	32	30	30	30	30	30	30	30	30
J =16:	30	30	30	30	30	30	30	30	30	30	30	30

（三）思 考 题

1）主题的程序是以12堆球、每堆球数以等差级数为例，请改为更多堆的球，每堆球数没有规律，只要原始数是3的倍数即可，编写程序，观察运行后是否也达到预期效果。

2）10个小孩围坐一圈分糖果，老师分给第一个小孩12块，第二个小孩2块，第三个小孩8块，第四个小孩22块，第五个小孩16块，第六个小孩4块，第七个小孩10块，第八个小孩6块，第九个小孩14块，第十个小孩20块。

然后按下列规则调整，所有的小孩同时把自己的糖分一半给右边的小孩。

糖的块数变为奇数的人，向老师要一块，问经过多少次调整，大家的糖的块数都一样？并且每人有多少块糖？

思考题2），笔算步骤和结果，以及程序设计等，详见《趣味程序设计100例》第24题。

10 分 酒 比 赛

在十九处名胜古迹游览区入口处，有座宽敞的啤酒饮料商店，还可供游人休息、娱乐，其中有个项目是酒店将8升的大容器盛满啤酒，附带5升和3升的两个空容器。三三两两的游人买酒后，比赛看谁能利用两个上述的空容器将这8升啤酒倒来倒去，分成两个4升的啤酒，而且还要比赛看谁分的方法多，相识的与不相识的游人共聚一起比赛，输者敬酒，优胜者喝，引起酒店里一阵阵欢笑声。

观看后，引起我们的深思，这是一道趣味图论题。但不是比赛倒换次数最少便将酒对半分开。而是不仅对半分，还要寻找有几种方法。我们进行算法分析、程序设计，在计算机上算出16种分法。当我们把分法带到该酒店时，使服务员大吃一惊，她们说只见到最多四、五种分法。亲爱的读者，你能找出几种方法？

（一）算 法 分 析

对于三个容器装的酒，用三元数组表示为（A_1，A_2，A_3），其中A_1、A_2、A_3为容器中实际装入的酒。

（1）对于一种状态（A_1，A_2，A_3），根据装酒的不同情况，可能有$B_1 \Rightarrow B_2$，$B_1 \Rightarrow B_3$，$B_2 \Rightarrow B_1$，$B_2 \Rightarrow B_3$，$B_3 \Rightarrow B_1$，$B_3 \Rightarrow B_2$（B_1、B_2、B_3分别表示三个容器，$B_1 \Rightarrow B_2$表示B_1中

的酒倒入B_2中一些），其中有几种可能。因此要计算出各种可能。

（2）每倒过一次后，首先要看是否为最终结果，若是最终结果，则转算法（3）。若不是最终结果，则要检查是否与以前状态相同？若不同，且次数小于20，则再倒一次，返回算法（2）。若与以前算法同，则返回以前的状态，重新倒一次。

（3）首先保存结果。然后检查是否有别的倒法，若有则继续倒，并转算法（2）；若没有则停止。

（二）程 序 设 计

1）设计思路

我们将程序编写成具有通用性，数组M（3）可由终端输入三个不同容量值。

程序中设置数组A（20,3），用于保存一次查找中的各状态；OUT（20，20，3）用于保存各次查找的结果；T（7）用于保存一个状态下的各种可能倒法的标志；TRACE（20，7）用于存放各个状态的T（7）。

子程序CALCU用于计算一种状态下各种可能的方法，并用数组T的相应元素标记，如T（1）表示$B_1 \Rightarrow B_2$，T（6）表示$B_3 \Rightarrow B_2$等等。

子程序PQR完成一次倒酒，首先检查T中的各元素，遇到不为0的元素后，则按该元素表示的方法改变BOT(I)的值，然后把该元素置为0。

回到本题，酒店是指定三个容器的容量分别为8、5、3，由终端输入，源程序运行后，请见打印出16种结果。

2）框图　见图10-1、图10-2、图10-3。

3）源程序及运行结果

```
C***  PROGRAM  NAME: WINE
C***
      INTEGER  N, NUM, C, I, J
      INTEGER  M (3) , A (20, 3) , BOT (3) , OUT (20, 20, 3) , TRACE (20,
      7) , T(7)
      DO  10  I = 1, 20                    ┐
      DO  10  J = 1, 3                     │
10    A(I, J) = 0                          │
      DO  15  I = 1, 20                    │ 各个数组清零
      DO  15  J = 1, 7                     │
15    TRACE(I, J) = 0                      │
      DO  25  I = 1, 3                     │
      M(I) = 0                             ┘
25    BOT(I) = 0
      READ (1, 30) M (1) , M (2) , M (3)   读入三个容器的容量
30    FORMAT (3I2)
      WRITE (1, 31) M (1) , M (2) , M (3)
31    FORMAT (3X, 'FIRST =', I3, 'SECONDE =', I3, 'THIRD =', I3 )
      N = 1
      NUM = 0
      BOT(1) = M(1)                        第一个容器满
```

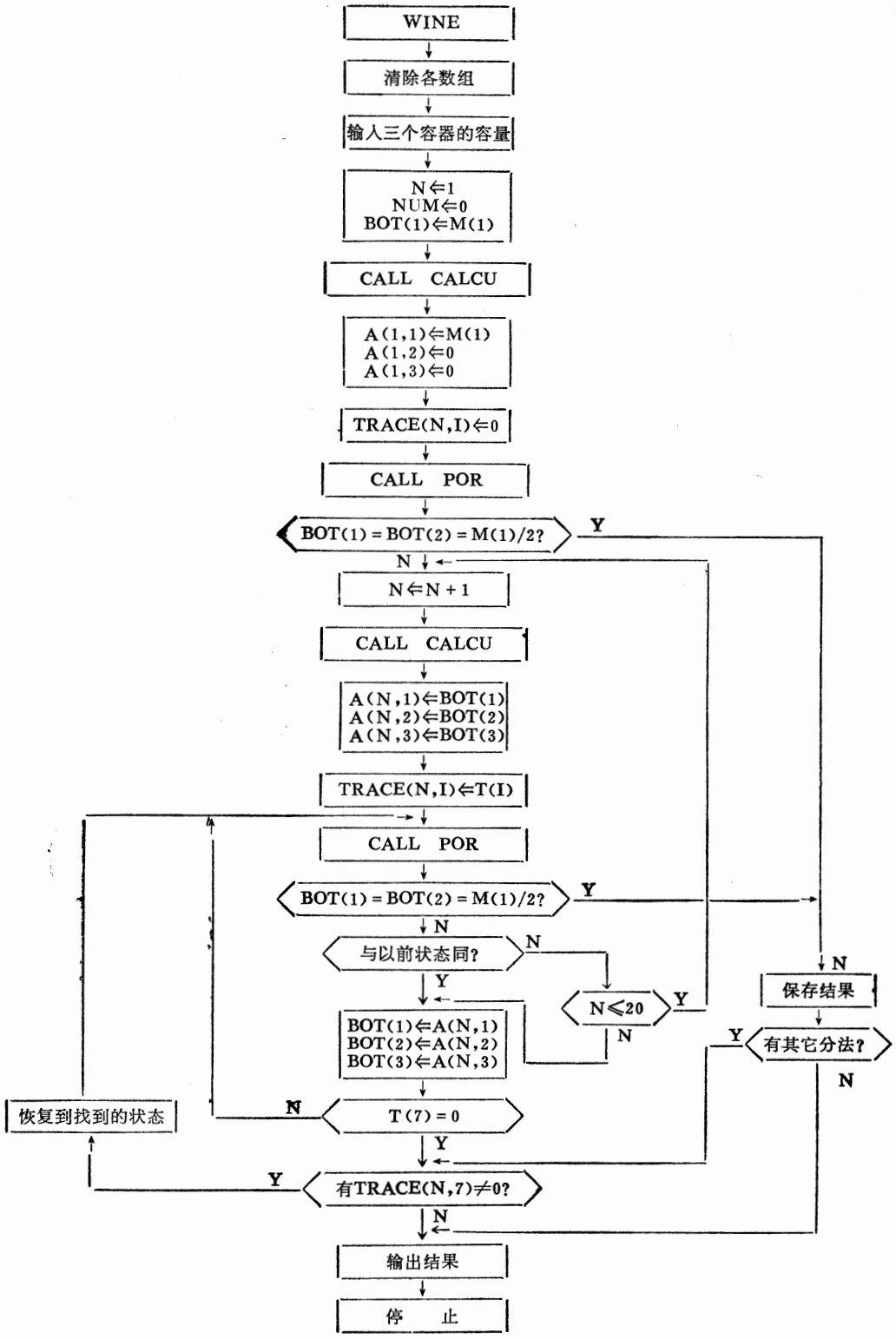

WINE
清除各数组
输入三个容器的容量
N⇐1
NUM⇐0
BOT(1)⇐M(1)
CALL CALCU
A(1,1)⇐M(1)
A(1,2)⇐0
A(1,3)⇐0
TRACE(N,I)⇐0
CALL POR
BOT(1) = BOT(2) = M(1)/2?
Y
N
N⇐N + 1
CALL CALCU
A(N,1)⇐BOT(1)
A(N,2)⇐BOT(2)
A(N,3)⇐BOT(3)
TRACE(N,I)⇐T(I)
CALL POR
BOT(1) = BOT(2) = M(1)/2?
Y
N
与以前状态同?
N
Y
N⩽20
Y
N
BOT(1)⇐A(N,1)
BOT(2)⇐A(N,2)
BOT(3)⇐A(N,3)
保存结果
Y
有其它分法?
N
恢复到找到的状态
N
T(7) = 0
Y
Y
有TRACE(N,7)≠0?
N
输出结果
停 止

图 10-1

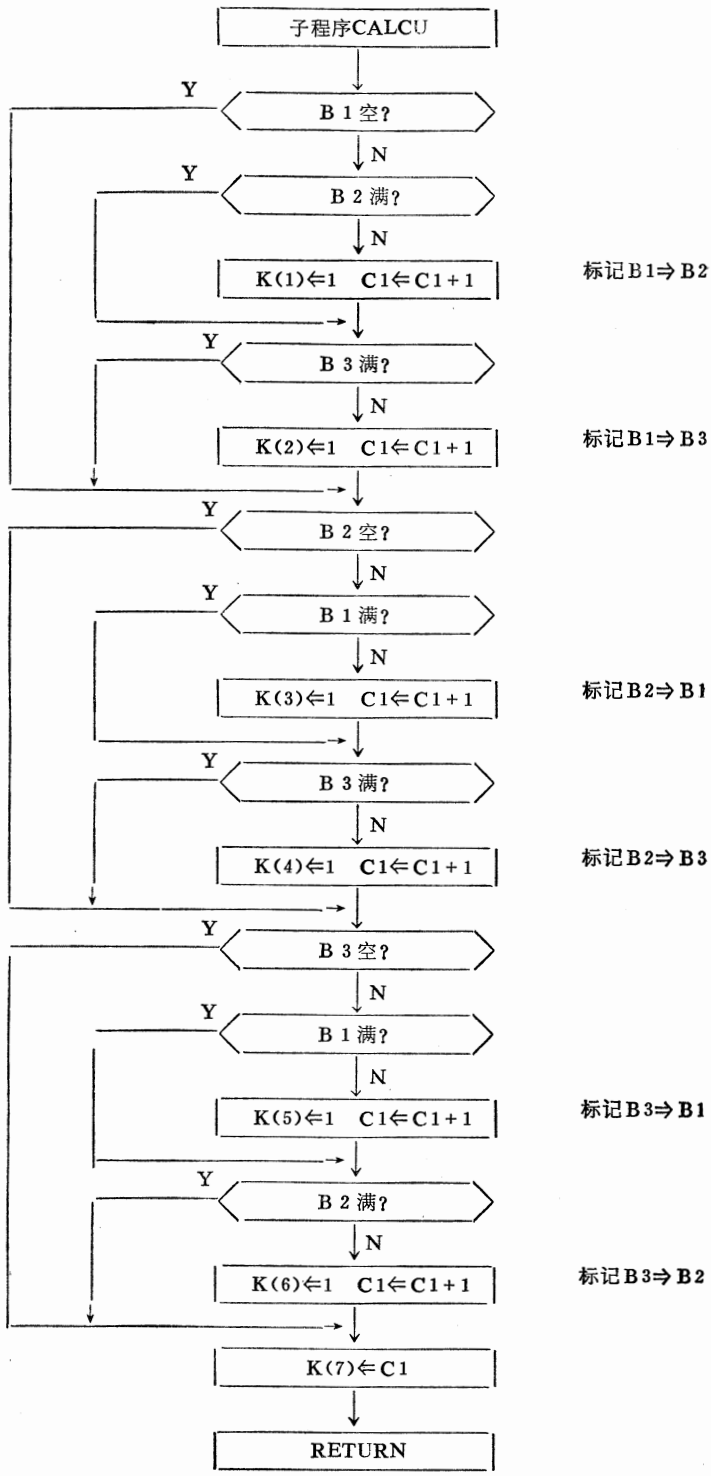

子程序CALCU
B 1 空?
Y
N
B 2 满?
K(1)⇐1 C1⇐C1+1
标记B1⇒B2
B 3 满?
K(2)⇐1 C1⇐C1+1
标记B1⇒B3
B 2 空?
B 1 满?
K(3)⇐1 C1⇐C1+1
标记B2⇒B1
B 3 满?
K(4)⇐1 C1⇐C1+1
标记B2⇒B3
B 3 空?
B 1 满?
K(5)⇐1 C1⇐C1+1
标记B3⇒B1
B 2 满?
K(6)⇐1 C1⇐C1+1
标记B3⇒B2
K(7)⇐C1
RETURN

图 10-2

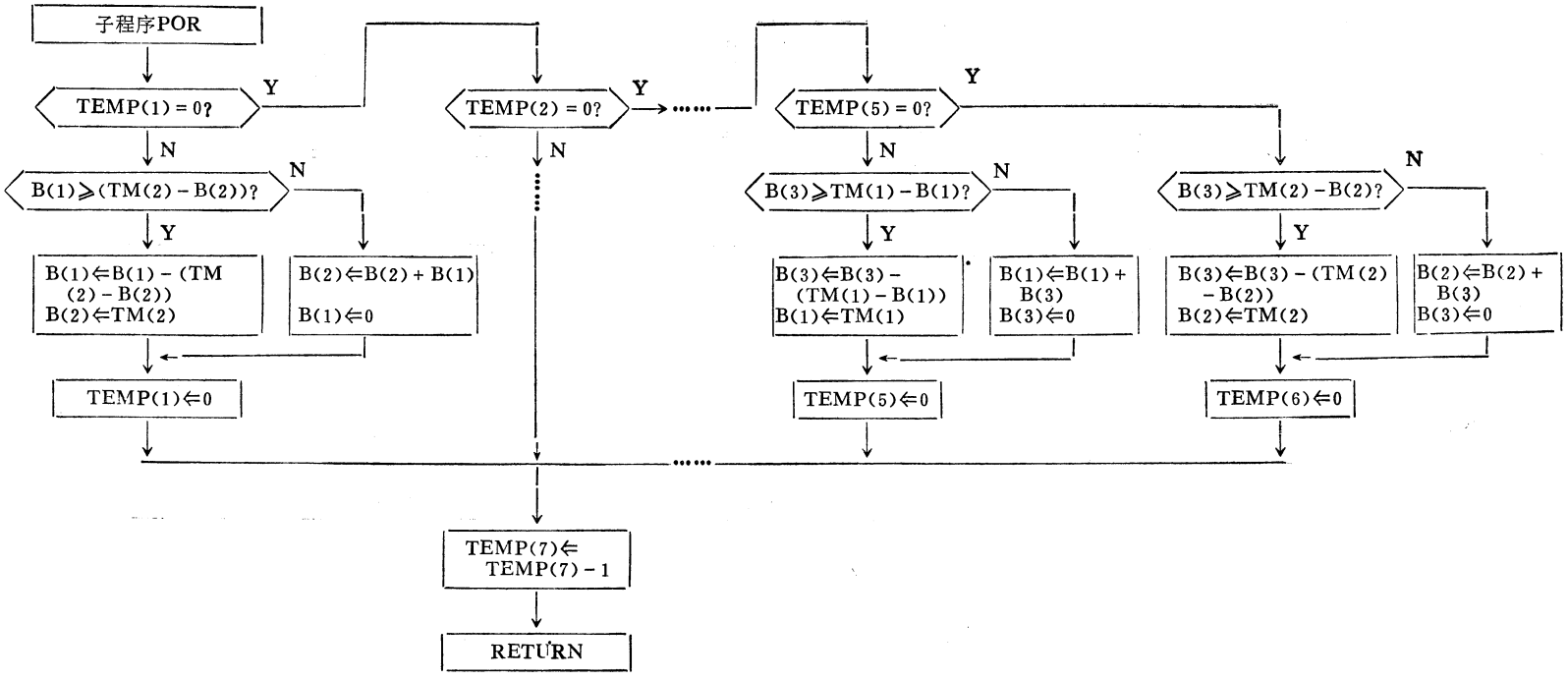

子程序POR
TEMP(1) = 0?
Y
N
B(1)≥(TM(2) - B(2))?
N
Y
B(1)⇐B(1) - (TM (2) - B(2))
B(2)⇐TM(2)
B(2)⇐B(2) + B(1)
B(1)⇐0
TEMP(1)⇐0
TEMP(2) = 0?
Y
N
TEMP(5) = 0?
Y
N
B(3)≥TM(1) - B(1)?
N
Y
B(3)⇐B(3) - (TM(1) - B(1))
B(1)⇐TM(1)
B(1)⇐B(1) + B(3)
B(3)⇐0
TEMP(5)⇐0
B(3)≥TM(2) - B(2)?
N
Y
B(3)⇐B(3) - (TM(2) - B(2))
B(2)⇐TM(2)
B(2)⇐B(2) + B(3)
B(3)⇐0
TEMP(6)⇐0
TEMP(7)⇐ TEMP(7) - 1
RETURN

图 10 — 3

```
C***
      CALL  CALCU (BOT, C, M, T)          计算各种可行的倒酒方法
      A (N, 1) = BOT (1)                  ┐
      A (N, 2) = BOT (2)                  ├保存初态
      A (N, 3) = BOT (3)                  ┘
C***  SAVE  THE  TRACE
      DO  40  I = 1, 7                    ┐
40    TRACE (N, I) = T(I)                 ├保存可倒的方法
      CALL  POR (BOT, T, M)               ┘倒一次
      DO  45  I = 1, 7
45    TRACE (N, I) = T(I)                 保存余下的可倒的方法
      IF (BOT (1) . EQ. BOT (2) . AND. BOT (1) . EQ. M(1)/2) GO  TO 200
                                          检查是否得到最后结果
C***  TRY  NEXT                           继续倒
50    N = N + 1
      CALL  CALCU (BOT, C, M, T)          ┐
      A (N, 1) = BOT(1)                   ├计算可行的方法并保存已有的状态
      A (N, 2) = BOT(2)                   ┘
      A (N, 3) = BOT(3)
C***  SAVE  THE  TRACE
      DO  90  I = 1, 7
90    TRACE (N, I) = T(I)
80    CALL  POR (BOT, T, M)               倒一次
      DO  85  I = 1, 7
85    TRACE (N, I) = T(I)
      IF (BOT(1). EQ. BOT(2). AND. BOT (1). EQ. M (1)/2) GO TO 200
                                          检查是否得到最后结果
      DO  100  I = 1, N
C***  COMPARE  WITH  BEFORE
      IF (BOT (1) . EQ. A (I, 1). AND. BOT (2) . EQ. A (I, 2). AND. BOT
     1 (3). EQ. A (I, 3)) GO  TO  150     比较是否与前面的状态相同

100   CONTINUE
      IF (N. LE. 20) GO  TO  50
C***  SAME  AS  BEFORE  OR  TIMES>20
150   BOT(1) = A(N, 1)                    ┐
      BOT(2) = A(N, 2)                    ├与前面的相同，则恢复最后一次的
      BOT(3) = A(N, 3)                    ┤状态。
      IF (T(7). NE. 0) GO  TO  80         ┘

160   DO  170  I = 1, N-1                 ┐
      IF (TRACE (N-I, 7). EQ. 0) GO  TO  170  ├检查是否有其它可倒的方法
      N = N-I                             ┤
      GO  TO  400                         ┘
```

```
170       CONTINUE
C***      TERMINATE
          GO  TO  350                                  若没有，则终止。

400       BOT ( 1 )A = (N, 1)
          BOT( 2 ) = A (N, 2)
          BOT( 3 ) = A (N, 3)
          DO  420  I = 1.7
420       T(I) = TRACE (N, I)                          若有，则在找到的状态起继续进行。
          DO  430  I = N + 1, 20
          DO  430  J = 1, 7
430       TRACE (I, J) = 0
          GO  TO  80
C***      FOUND  A  WAY,  SAVE  IT
200       N = N + 1
          A(N, 1) = BOT(1)
          A(N, 2) = BOT(2)                             找到一种方法，保存之。
          A(N, 3) = BOT(3)
          NUM = NUM + 1
          DO  220  I = 1, N
          DO  220  J = 1, 3
220       OUT (NUM, I, J) = A(I, J)
          IF (NUM. EQ. 20) GO  TO  350
          GO  TO  160
350       WRITE (1, 330) NUM                           输出最后结果
330       FORMAT (2X, 'TATOL  METHEODS = ', I3)
          DO  360  N = 1, NUM
          WRITE (1, 340) N
360       WRITE (1, 370) ((OUT(N, I, J), J = 1, 3), I = 1, 20)
340       FORMAT (/////, 3X,'METHEOD', I3, //)
370       FORMAT (3X, 3I5)
          STOP
          END
C***      SUBROUTINE  WHICH  CALCULATE  THE  NUMBER  FROM  X
          TO  Y                                        子程序：计算在某状态下可行的
                                                       倒酒方法。
          SUBROUTINE  CALCU (B, C1, M1, K)
          INTEGER  B(3) , C1, M1 (3) , K(7)
          DO  700  I = 1, 7
700       K(I) = 0                                     BX⇒BY 表示第X个容器可 倒入第Y个
          C1 = 0                                       容器。
          IF (B( 1 ). EQ. 0) GO  TO  730

          IF (B( 2 ). EQ. M1( 2 )) GO  TO  710
```

```
C***  MARK THE POR B1⇒B2                          ┐
      K (1) =1                                    │
      C1 = C1+1                                   ├ B1⇒B2
710   IF (B(3). EQ. M1(3)) GO TO 730              ┘
C***  MARK THE POR B1⇒B3                          ┐
      K(2) =1                                     │
      C1 = C1+1                                   ├ B1⇒B3
730   IF (B(2). EQ. 0) GO TO 750                  ┘
      IF (B (1) . EQ. M1 (1)) GO TO 740
C***  MARK THE POR B2⇒B1                          ┐
      K(3) =1                                     ├ B2⇒B1
      C1 = C1+1                                   ┘
740   IF (B(3). EQ. M1(3)) GO TO 750
C***  MARK THE POR B2⇒B3                          ┐
      K(4) =1                                     ├ B2⇒B3
      C1 = C1+1                                   ┘
750   IF (B(3). EQ. 0) GO TO 780                  ┐
      IF (B(1). EQ. M1 (1)) GO TO 760             ├ B3⇒B1
C***  MARK THE POR B3⇒B1                          ┘
      K(5) = 1                                    ┐
      C1 = C1+1                                   ┘
760   IF (B(2). EQ. M1(2)) GO TO 780              ┐
C***  MARK THE POR B3⇒B2                          ├ B3⇒B2
      K(6) =1                                     ┘
      C1 = C1+1
780   K(7) = C1
      RETURN
      END
C***  SUBROUTINE POR (B, TEMP, TM)        子程序：分一次，即从一个容器倒入另一个容器。
      INTEGER B (3), TEMP (7), TM (3)
      IF (TEMP (1). EQ. O) GO TO 840
C***  BOT1⇒BOT2
      IF (B(1). GE. TM (2)-B(2)) GO TO 800        ┐
      B(2) = B(2)+ B(1)                           │
      B(1) = 0                                    │
      GO TO 820                                   ├ B1⇒B2
800   B(1) = B(1) - (TM(2) - B(2))                │
      B(2) = TM(2)                                │
820   TEMP(1) = 0                                 ┘
      GO TO 1100
840   IF (TEMP (2). EQ. 0) GO TO 880
C***  BOT1⇒BOT3                                   ┐
      IF (B(1). GE. TM (3)-B (3)) GO TO 850       ┘
```

```
          B(3) = B(3) + B(1)
          B(1) = 0                                          } B1⇒B3
          GO  TO  860
850       B(1) = B(1) - (TM(3) - B(3))
          B(3) = TM(3)
860       TEMP(2) = 0
          GO  TO  1100
880       IF (TEMP(3). EQ. 0) GO  TO  920
C***      BOT2⇒BOT1
          IF (B(2). GE. TM(1) - B(1)) GO  TO  890           }
          B(1) = B(1) + B(2)                                }
          B(2) = 0                                          }
          GO  TO  900                                       }
890       B(2) = B(2) - (TM(1) - B(1))                      } B2⇒B1
          B(1) = TM(1)                                      }
900       TEMP(3) = 0                                       }
          GO  TO  1100                                      }
920       IF (TEMP(4). EQ. 0) GO  TO  970
C***      BOT2⇒BOT3
          IF (B(2). GE. TM(3) - B(3)) GO  TO  940           }
          B(3) = B(3) + B(2)                                }
          B(2) = 0                                          }
          GO  TO  960                                       }
940       B(2) = B(2) - (TM(3) - B(3))                      } B2⇒B3
          B(3) = TM(3)                                      }
960       TEMP (4) = 0                                      }
          GO  TO  1100                                      }
970       IF (TEMP(5). EQ. O) GO  TO  1000
C***      BOT3⇒BOT1
          IF (B(3). GE. TM(1) - B(1)) GO  TO  980           }
          B(1) = B(1) + B(3)                                }
          B(3) = 0                                          }
          GO  TO  990                                       }  B3⇒B1
980       B(3) = B(3) - (TM(1) - B(1))                      }
          B(1) = TM(1)                                      }
990       TEMP(5) = 0                                       }
C***      GO  TO  1100
C***      BOT3⇒BOT2
1000      IF (B(3). GE. TM(2) - B(2)) GO  TO  1040          }
          B(2) = B(2) + B(3)                                }
          B(3) = 0                                          } B3⇒B2
          GO  TO  1020                                      }
1040      B(3) = B(3) - (TM(2) - B(2))                      }
          B(2) = TM(2)                                      }
```

```
1020      TEMP(6) = 0
1100      TEMP(7) = TEMP(7) - 1
          RETURN
          END
 853      输入，三个容器的容量分别为 8 ， 5 ， 3 。
          FIRST = 8, SECONDE = 5, THIRD = 3
          TATOL  METHODS = 16  程序运行后打印出共找到16种不同分法，分别 列出 如下：
          METHOD  1                方法 1
          8       0       0
          3       5       0
          0       5       3
          5       0       3
          5       3       0
          2       3       3
          2       5       1
          7       0       1
          7       1       0
          4       1       3
          4       4       0   ←—到此分完（以下同）
          0       0       0
          0       0       0
          0       0       0
          0       0       0
          0       0       0
          0       0       0
          0       0       0
          0       0       0
          0       0       0
          METHOD  2          方法 2
          8       0       0
          3       5       0
          3       2       3
          0       5       3
          5       0       3
          5       3       0
          2       3       3
          2       5       1
          7       0       1
          7       1       0
          4       1       3
          4       4       0
          0       0       0
          0       0       0
```

0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
METHOD 3 方法 3
8 0 0
3 5 0
3 2 3
5 0 3
5 3 0
2 3 3
2 5 1
7 0 1
7 1 0
4 1 3
4 4 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
METHOD 4 方法 4
8 0 0
3 5 0
3 2 3
6 2 0
6 0 2
1 5 2
0 5 3
5 0 3
5 3 0
2 3 3
2 5 1
7 0 1
7 1 0
4 1 3
4 4 0

0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
METHOD 5 方法 5
8 0 0
3 5 0
3 2 3
6 2 0
6 0 2
1 5 2
1 4 3
0 5 3
5 0 3
5 3 0
2 3 3
2 5 1
7 0 1
7 1 0
4 1 3
4 4 0
0 0 0
0 0 0
0 0 0
0 0 0
METHOD 6 方法 6
8 0 0
3 5 0
3 2 3
6 2 0
6 0 2
1 5 2
1 4 3
5 0 3
5 3 0
2 3 3
2 5 1
7 0 1
7 1 0
4 1 3
4 4 0
0 0 0

0 0 0
0 0 0
0 0 0
0 0 0

METHOD 7 方法 7

8 0 0
3 5 0
3 2 3
6 2 0
6 0 2
1 5 2
1 4 3
4 4 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0

METHOD 8 方法 8

8 0 0
3 5 0
3 2 3
6 2 0
6 0 2
5 0 3
5 3 0
2 3 3
2 5 1
7 0 1
7 1 0
4 1 3
4 4 0
0 0 0
0 0 0
0 0 0
0 0 0

0 0 0
0 0 0
0 0 0
METHOD 9 方法 9
8 0 0
5 0 3
0 5 3
3 5 0
3 2 3
6 2 0
6 0 2
1 5 2
1 4 3
4 4 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
METHOD 10 方法10
8 0 0
5 0 3
5 3 0
3 5 0
3 2 3
6 2 0
6 0 2
1 5 2
1 4 3
4 4 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0

0 0 0
0 0 0
METHOD 11 方法11
8 0 0
5 0 3
5 3 0
2 3 3
0 5 3
3 5 0
3 2 3
6 2 0
6 0 2
1 5 2
1 4 3
4 4 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
METHOD 12 方法12
8 0 0
5 0 3
5 3 0
2 3 3
2 5 1
0 5 3
3 5 0
3 2 3
6 2 0
6 0 2
1 5 2
1 4 3
4 4 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0

0 0 0

METHOD 13 方法13

8 0 0
5 0 3
5 3 0
2 3 3
2 5 1
7 0 1
7 1 0
3 5 0
3 2 3
6 2 0
6 0 2
1 5 2
1 4 3
4 4 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0

METHOD 14 方法14

8 0 0
5 0 3
5 3 0
2 3 3
2 5 1
7 0 1
7 1 0
4 1 3
0 5 3
3 5 0
3 2 3
6 2 0
6 0 2
1 5 2
1 4 3
4 4 0
0 0 0
0 0 0
0 0 0
0 0 0

METHOD 15 方法15

8	0	0
5	0	3
5	3	0
2	3	3
2	5	1
7	0	1
7	1	0
4	1	3
4	4	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

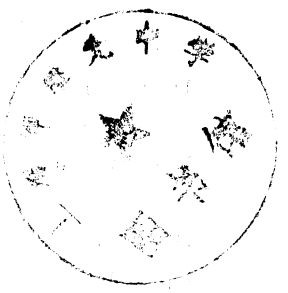
METHOD 16 方法16

8	0	0
5	0	3
5	3	0
2	3	3
2	5	1
3	5	0
3	2	3
6	2	0
6	0	2
1	5	2
1	4	3
4	4	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

（三）思 考 题

1）法国著名数学家波瓦松在青年时代研究了一个有趣的数学游戏——某人有12品脱（容量单位，1品脱＝0.568升）啤酒一瓶，想从中倒出6品脱，但他没有6品脱的容器，仅有一个8品脱和一个5品脱的容器，怎样倒法才能使啤酒分成两个6品脱呢？出乎意料的是，对这个数学游戏的研究竟决定了他一生的道路，从此以后竟成了法国一位著名的数学家。请你试试能找出几种分法？

2）这类题只要按一定比例的三个容器的容量都能分吗？



二　辨别四对夫妻关系

11　分辨阅览室的桌子和灯

一个图书馆，分东西两个阅览室。东阅览室里，每张桌子上有2盏灯；西阅览室里，每张桌子上有3盏灯。现在知道两个阅览室总的桌数和灯数都是奇数，你能否知道两个阅览室的桌子哪边是奇数，哪边是偶数？

（一）算法分析和结论

假设东边的桌子数是x，西边的桌子数是y，两边桌子上总的灯数是N，于是：N = 2x + 3y。

由于两边桌子的总数是奇数，所以x和y必然其中有一个是偶数，另一个是奇数。

1）先假设x为奇数，y为偶数，那么等式右边，2x和3y都是偶数，因此N也是偶数，与题意不合，故不成立。

2）如果x为偶数，y为奇数，那么2x为偶数，3y为奇数，2x + 3y也必为奇数。所以满足N为奇数的要求。

（二）程序设计

1）设计思路

由于两个阅览室的桌子数和灯数都是两种状态，不是偶数，便是奇数，没必要算出具体总数是多少，据题要求，只要判出奇、偶数便可。定义四个一维数组，L(2)，N(2)表示东阅览室的灯和桌子。LL(2)，NN(2)表示西阅览室的灯和桌子。令1表示奇数，0表示偶数。据题意可分别赋值N(1) = 0，L(1) = 0；N(2) = 1，L(2) = 1。NN(1) = 0，LL(1) = 0；NN(2) = 1，LL(2) = 0。（因为东阅览室的灯始终是偶数）。

条件是L(I) + LL(I) = 1和N(I) + NN(I) = 1同时成立。

2）框图　见图11

3）源程序及运行结果

```
      DIMENSION  L(2), N(2), LL(2), NN(2)
      DO  11  I=1, 2
      L (I) =I-1
      N (I) =I-1
      LL (I) =I-1
11    NN (I) = 0
      K = 1
      DO  100  J=1, 2
      DO  200  I=1, 2
      MS=N(J)+NN(I)
      KS=L(J)+LL(I)
      IF (KS. NE. K)  GO  TO  200
```

```
      IF (MS. NE. K)  GO TO 200
      WRITE (6, 111)  L (J) . N (J) , LL (I) , NN (I)
111   FORMAT (1X, 2HL=, I3, 3X, 2HN=, I2/1X, 3HLL=, I2, 2X, 2HNN=,I2)
      STOP
200   CONTINUE
100   CONTINUE
      END
```

L= 1 N= 1

LL= 0 NN= 0

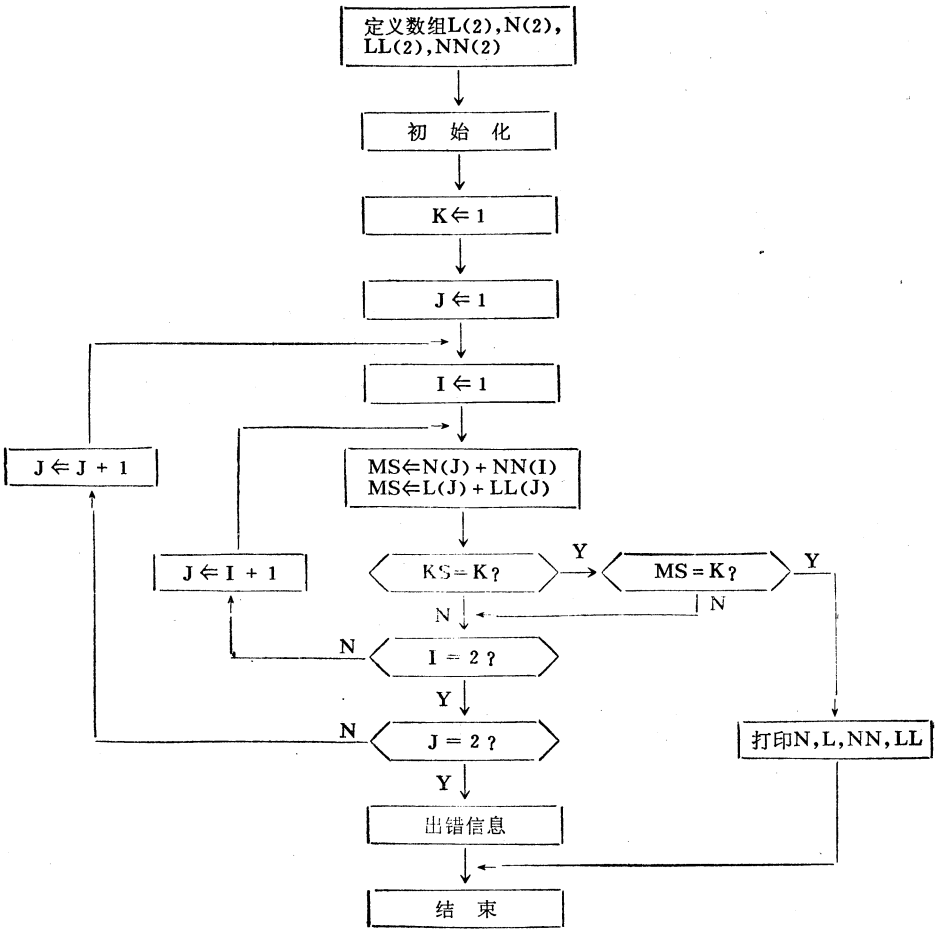


图 11

(三) 思 考 题

有三对情侣即将举行婚礼，三个新郎分别是A、B、C，三个新娘分别是X、Y、Z。有人不知底细，想弄清到底是谁和谁结婚，于是他就去询问了这六位新人中的三位，但听到的回答是：A说，他将和X结婚。X说，她的未婚夫是C。C说，他将和Z结婚。这个人听了后知道他们是在和他开玩笑，说的全是假话。请你编写程序找出到底是谁和谁结婚。

12 挑选乒乓球

13个乒乓球里有一个次品，外表和正品一样，但重量不同，不知轻重。若用不带砝码的天平，问称量几次能将次品挑出来，使正品出厂。

（一）笔算步骤和结果

为了叙述方便，把13个球编成1～13号。第一次称，天平两边各放4个，不妨设1～4号在左边，5～8号在右边。有三种可能：

第一种——平衡。说明1～8号是正品，次品必在9～13号之中。第二次称，取9～13号中的3个。不妨取9、10、11号，放在天平左边，再把正品1～3号放在右边。这时，又会有三种可能：

（1）平衡。说明9、10、11号是正品，次品在12、13号之中。第三次称，取12、13号之一，不妨取12号与1号并称。若平衡，可肯定13号是次品；若不平衡，则12号是次品。

（2）左边重。说明9、10、11号之中必有一个是次品。第三次称，取9、10、11号中的2个，不妨取9、10号并称。若平衡，11号必是次品；若不平衡，那末重的一个必是次品。

（3）右边重。可用类似（2）的方法称第三次，注意，这时次品是较轻的一个。

第二种——左边重。说明次品必在1～8号之中，而9～13号都是正品。第二次称。把左端的4号搬到右端，右端的5号搬到左端，再将6、7、8号三个换成三个正品。9、10、11号，这时又有三种可能：

（1）平衡。说明次品必在6、7、8号之中，而且次品一定是较轻的。第三次称，将6、7并称。若平衡，则8号是次品；若不平衡，轻的一个是次品。

（2）左边重。说明次品在1、2、3号之中，而且重的一个是次品。第三次称，取1、2号并称。若平衡，则3号是次品；若不平衡，则重的一个是次品。

（3）右边重。说明次品是4号或5号，这时只要把其中之一与一个正品称第三次，就可判别其中哪一个是次品。

第三种——右边重。类似地用第二种情况的处理方法。

（二）程序设计

1）设计思路

我们还是按题意将13个球找出1个次品球，但稍做改变：次品球重量比正品球较轻，为了减化程序，用天平称量次数有限，找出次品即停。

打印程序名字，定义数组A（13），置放已编号的1～13个球。设正品球重量为1，次品球重量为0。先将数组A（13）依次全冲入1，随机输入次品所在位置N（由键盘敲入），第6语句A（N）⇐0，便是次品球所在的位置赋0。

用DO 10循环语句可把全部球分成三组，即C1、C2、C3，分不到组里的球（如第一次分时，第13个球）单放。出循环体比较三组重量算作称量次数，用条件语句判断轻重，决定下一次分组变换。此法最多分组（进入循环体）三次，每次出循环体最多称量三次。故称“三三称量法”，便指出次品所在位置。举六个例子，详见运行后输出结果。

2）框图　见图12

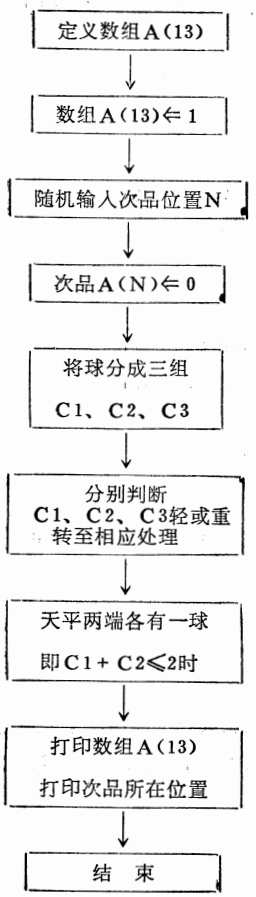


图 12

3）源程序及运行结果

```
C       PROGRAM  FQ
        INTEGER  A(13) , C1, C2, C3
        DO  5  I=1, 13
5       A(I) = 1
        WRITE  (10)  "INPUT  N:"
        READ  (11)  N
6       A(N) = 0
        M = 4
        K = 1
        N = 1
8       C1 = 0
        C2 = 0
        C3 = 0
        DO  10  I = 1 , M
        C1 = C1 + A(K + I - 1)
        C2 = C2 + A(K + M + I - 1)
```

```
      IF   (N. EQ. 1)   C3 = C3 + A(K + 2*M + I - 1)
10    CONTINUE
      IF   (N. NE. 1)   GOTO  20
      IF   (C1. EQ. C2)   GOTO  14
      GOTO  20
14    IF   (C1. EQ. C3)   GOTO  16
      GOTO  20
16    K = 13
      GOTO  200
20    IF   (C1. EQ. C2)   GOTO  130
      IF   (C1. LE. C2)   GOTO  140
      K = K + M
      GOTO  140
130   K = K + 2*M
140   IF   ( (C1 + C2) . LE. 2)   GOTO  200
      N = N + 1
      IF   (N. EQ. 2)   M = 2
      IF   (N. EQ .3)   M = 1
      GOTO  8
200   WRITE   (10, 260)  (A (I) , I = 1 , 13) , K
260   FORMAT   (1X, 2HA : , 13I4/1X, 3HNO : , I5)
      STOP
      END
```

```
INPUT  N :
1                                                    输入第 1 个位置为 0
A :   0   1   1   1   1   1   1   1   1   1   1   1   1
NO :   1                                             找出第 1 个位置是次品
INPUT  N :
4                                                    输入第 4 个位置为 0
A :   1   1   1   0   1   1   1   1   1   1   1   1   1
NO :   4                                             找出第 4 个位置为次品
INPUT  N :
6                                                    输入第 6 个位置为 0
A :   1   1   1   1   1   0   1   1   1   1   1   1   1
NO :   6                                             找出第 6 个位置为次品
INPUT  N :
9
A :   1   1   1   1   1   1   1   1   0   1   1   1   1
```

```
NO:  9
INPUT  N:
10
A:  1  1  1  1  1  1  1  1  1  0  1  1  1
NO:  10
INPUT  N:
13
A:  1  1  1  1  1  1  1  1  1  1  1  1  0
NO:  13
STOP
```

（三）思 考 题

1）若次品球比正品球可能轻或重，改写程序使轻、重全能找出。

2）若按笔算步骤限制三次称量，找出次品，应怎样编写程序。

13 这样区分全钢和半钢表壳

现有十箱手表壳。已知九箱是全钢表壳，一箱是半钢表壳。假如全钢表壳和半钢表壳的外形和颜色一模一样，只是它们的重量不一样，半钢表壳每只重 90 克，全钢表壳重100克，你能否只称一次，就把这一箱半钢表壳从十箱表壳中区分开来。

（一）笔算步骤和结果

先将十箱手表壳从 1 到10编号，然后从第 1 箱中取 1 只，第 2 箱中取 2 只，第 3 箱中取 3 只……第10箱中取10只，总共取出55只表壳，如果这55只表壳全是全钢，则它们的总重量应为5500克。因此，若称量结果比5500克少10克，这就说明这 55 只表壳中有一只是半钢，所以，第一箱就是半钢手表壳；如果少20克，则有二只半钢手表壳，第二箱就是半钢手表壳。依次类推，就能区别出哪一箱是半钢手表壳。

（二）程 序 设 计

1）设计思路

定义数组K（10）用作十个箱子编号用，用键盘敲入半钢箱号N（1≤N≤10）。按计算分析用循环语句赋值给数组K（10）中 9 个数组元素为全钢的（每 只 100），而第N箱，输入N个90（半钢重每只为90）。然后将这十箱中 每 箱 抽 出的相应个数乘以重量相加，按笔算方法便可验证出果然第N箱是半箱表壳。程序运行时敲入N = 1 、 6 、10三种箱号，验证全对。

2）框图　见图13

3）源程序及运行结果

```
      DIMENSION  K(10)
2     WRITE  (10, 1)
1     FORMAT  (20H  INPUT  N  (1<=N<=10):,10X, 14H(N=-1  TO  STOP))
      READ  (11, 10)  N
10    FORMAT  (I2)
      IF  (N. EQ. -1)  STOP
      DO  30  I=1,  10
```

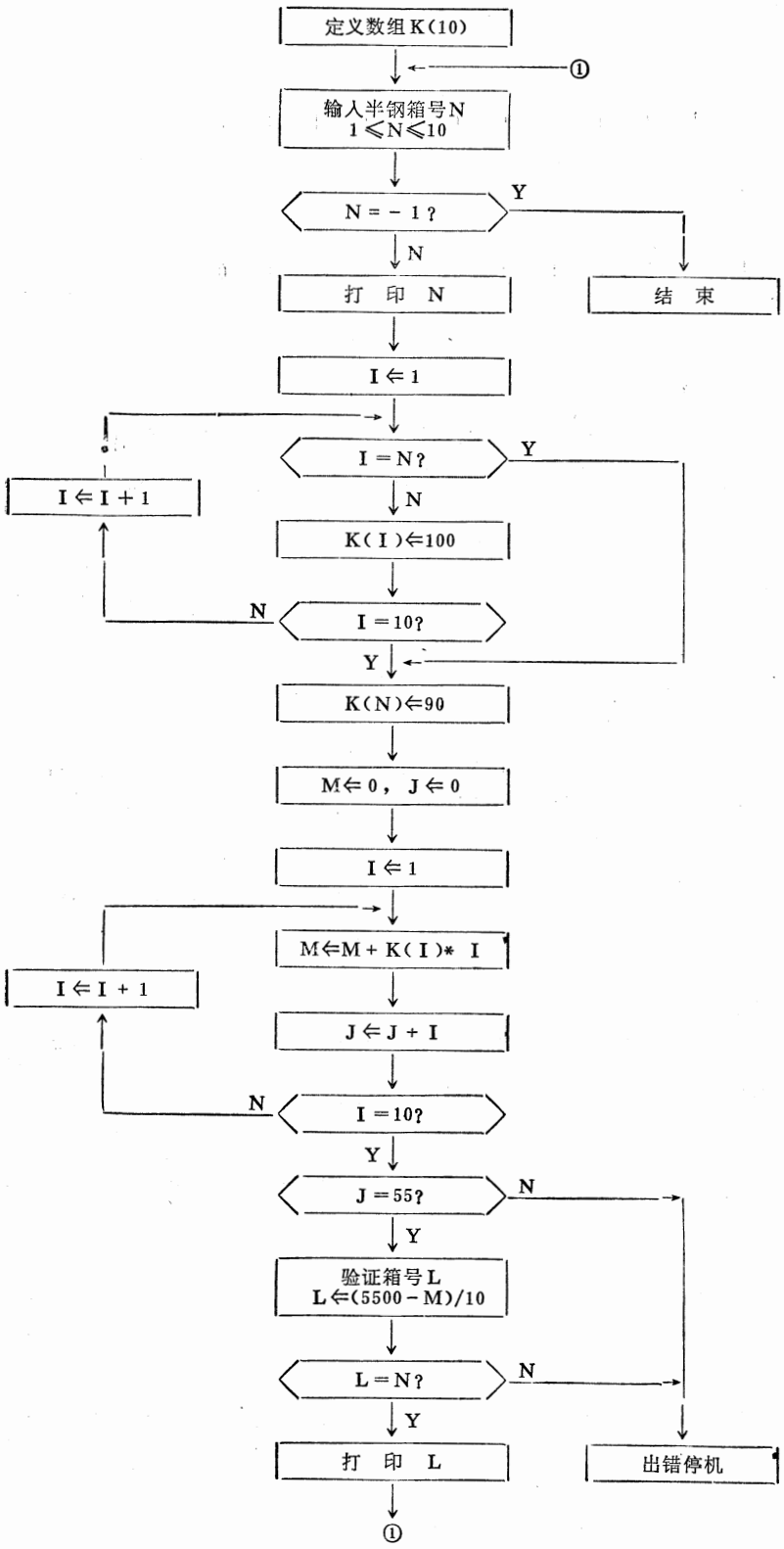

定义数组K(10)
①
输入半钢箱号N
1≤N≤10
N = - 1 ?
Y
N
结 束
打 印 N
I⇐1
I = N?
Y
N
I⇐I + 1
K(I)⇐100
N
I = 10?
Y
K(N)⇐90
M⇐0, J⇐0
I⇐1
M⇐M + K(I)* I
I⇐I + 1
J⇐J + I
N
I = 10?
Y
J = 55?
N
Y
验证箱号L
L⇐(5500 - M)/10
L = N?
N
Y
打 印 L
出错停机
①

图 13

```
      IF   (I. EQ. N)   GOTO  30
      K(I) = 100
30    CONTINUE
      K(N) = 90
      M= 0
      J= 0
      DO  40  I=1, 10
      M=M+K(I)*I
40    J=J+I
      IF   (J. NE. 55)   STOP  'ERR'
      L=(5500-M)/10
      IF   (L. NE. N)   STOP  'ERR1'
      WRITE   (10, 50)   L
50    FORMAT   (10X, 2H**, I2, 2H**)
      GOTO  2
      END
```

```
INPUT  N   (1<=N<=10) :          (N=-1  TO  STOP)
  1
            **  1**
INPUT  N   (1<=N<=10) :          (N=-1  TO  STOP)
  6
            **  6**
INPUT  N   (1<=N<=10) :          (N=-1  TO  STOP)
  10
            **  10**
INPUT  N   (1<=N<=10) :          (N=-1  TO  STOP)
 -1
STOP
```

（三）思　考　题

你需称几次——一袋一袋的洗衣粉堆成10堆，八堆洗衣粉是合格产品，每袋1斤。唯独有两堆分量不足，每袋只有9两。从外形上看，看不出哪一堆是9两的。用台称一堆一堆的去称吧，称的次数比较多。有人找到一个办法，只称了数次，就找到9两的那一堆。这是个什么办法呢？

如果有四十堆洗衣粉，其中有一堆是9两一袋的，那末要称几次，才能找出这一堆？

14　求婚者的智慧

酋长国王的女儿达西要出嫁了，按以往的风俗习惯，是搭个高台，台下是众多的求婚者。达西在台上扔朵花，扔在台下谁身上，达西便得嫁给谁。但她怕落不到心爱的克俊身上。达西私约克俊商量如何是好。克俊出了如此主意…。达西便和父亲说："我不愿搭台撒花，这么多人来，挤在一起乱哄哄的，没秩序。"父亲说，"不这样也可以，但结婚时当

场在人群中决定嫁给谁，不许指名，由你决定”。达西高兴地告诉主持人如何行事。婚日来临，人群拥挤，主持人叫求婚者排成一行，克俊先在队外数了数队列共有100人还多一个，于是自己找了个适当位置也站在队列中，主持人喊叫1、2，1、2报数，报单数的退出场外，余下的靠拢又成一行，再重新1、2，1、2…报数，报单数的退场，如此下去最后只剩一人，达西便嫁给谁。众人惊奇地看着最后剩下的竟是克俊，请问克俊站在原队形的什么位置？

（一）笔算步骤和结果

据达西在队外数队形中人数是101人，他站进队形便是102人了。

第一遍报数后，退出场外共有102÷2=51（人），退出者是第1，3，5，7，9……（人）。也就是$2^0\times1$，$2^0\times3$，$2^0\times5$，$2^0\times7$，……。

第二遍报数后，退出场外共有（51+1）÷2=26（人），退出者是原队形中第2，6，10，14（人），…。也就是$2^1\times1$，$2^1\times3$，$2^1\times5$，$2^1\times7$，……。

第三遍报数后，退出场外共有26÷2=13（人），退出者是原队形中第4，12，20（人），…。也就是$2^2\times1$，$2^2\times3$，$2^2\times5$，$2^2\times7$，……。

如此报数报到第六遍时，按上述规律退场后，仅剩下站在原队形第2^6位置，即智慧的克俊站在第64个位置上。

（二）程序设计

1）设计思路

定义数组JA（102），依次存放102人。队形一行102人，利用循环语句从1开始数，当数到奇数时便将该数组元素冲成零，直到数至102为止，相当于第一遍报名退出场的人。再次进行循环体，仍从1开始计数。上次被冲成零的数组元素就不考虑，又将奇数的数组元素冲为零。这相当于第二遍报数后退出场的人。如此一遍遍数下去，直到最后剩下的一个人为止，此人便是达西的心上人克俊预先选中的位置，每报完一遍名，计数器IP加1，最后也将IP打印出。

2）框图　见图14

3）源程序及运行结果

```
      DIMENSION  JA(102)
      DO  10  I=1, 102
10    JA(I) =I
      IP= 0
      N = 0
15    K = 0
      DO  20  I=1, 102
      IF (JA(I). EQ. O) GOTO  20
      K = K + 1
      IF (K. EQ. K/2*2) GOTO  20
      M=JA(I)
      JA(I) = 0
      N = N + 1
      IF (N. EQ. 102) GOTO  40
```

```
20        CONTINUE
          IP=IP+1
          GOTO  15
40        WRITE (10, 50) IP, M
50        FORMAT (5X, 3HIP=, I2, 5X, 2HM=, I2)
          STOP
          END
```

IP= 6　　M=64　　报了6遍数，克俊站在原队形第64个位置

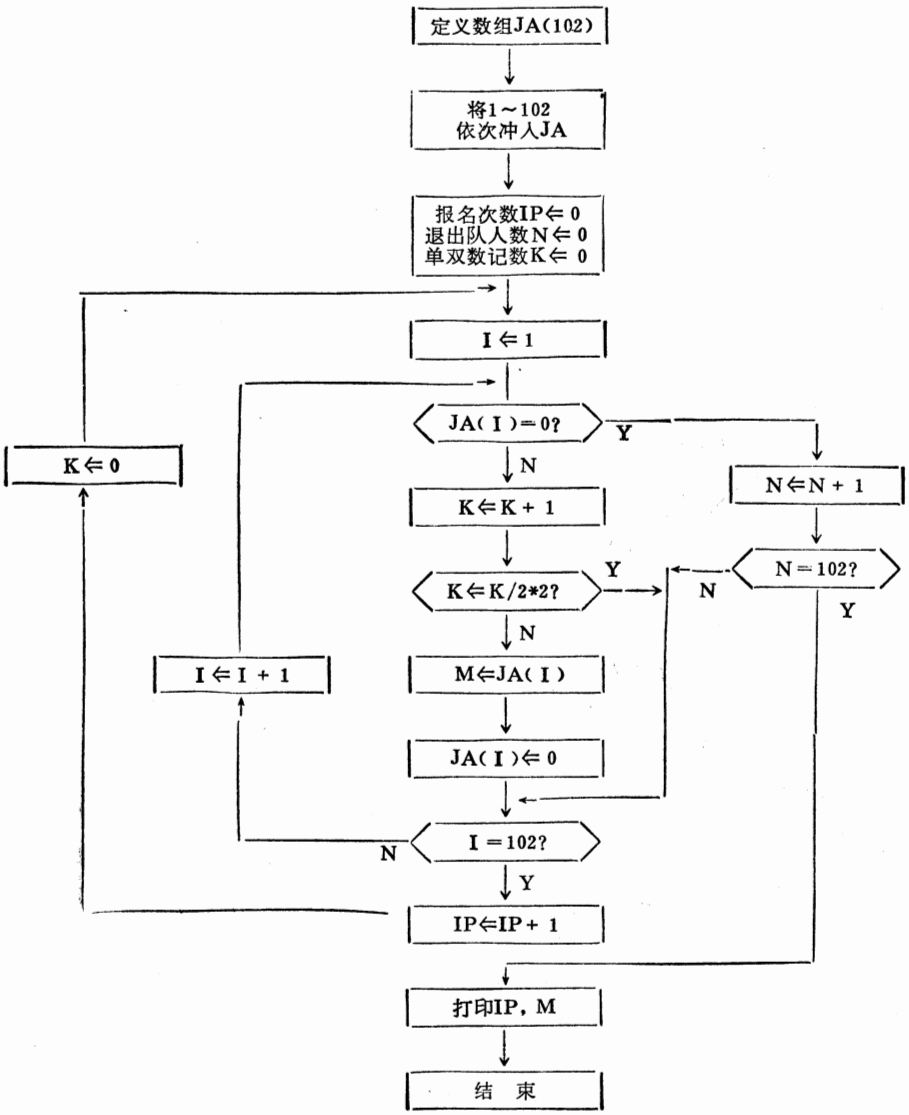


图 14

（三）思 考 题

积极献血——某单位有1000人积极报名献血，负责人说只需要一人献血就够了，但不好决定究竟让谁去。为了不打消报名者的积极性，便决定让报名者排成一行。从第一名开始，1至3报数，凡报到3数的人就退出队伍，余下的向前靠拢，就地排成一行，再按此规定报数，重复进行，直到第P次报数后，只剩下三个人为止，这三人中规定是第3个人便是献血者。请问共报了几次数？这位光荣献血者站在最初队伍的什么位置？

笔算步骤、结果和程序设计等，详见《趣味程序设计100例》第17题。

15 白兔智斗狡猾的狐狸

有十个空洞，沿着凤凰山顶围成一大圈，每洞之间相距较远，有一只狐狸和一只白兔，住在各自洞里。狡猾的狐狸总想捕捉吃掉这只兔子，一天二者远远相望，白兔对狐狸说："你用不着总打我的主意，把十个洞依次编成1～10号，我藏在其中一个洞里，你从自己住的第10洞出发，第一次走一个间隔（两洞之间为一间隔），到该洞寻找我，第二次走两个间隔，进洞找我，第三次走三个间隔进洞寻找，依次类推，次数不限，你若能找到我，你可饱餐一顿。"狐狸心想，不限次数，他总可以找到这只白兔。结果，从早晨一直找到傍晚，也未找到白兔，请算算白兔可能藏在哪几个洞里？

（一）笔算步骤和结果

如图15-1所示，狐狸从第⑩洞出发，第一次到①洞，第二次到③洞，第三次到⑥洞，第四次到⑩洞，第五次到⑤洞……。

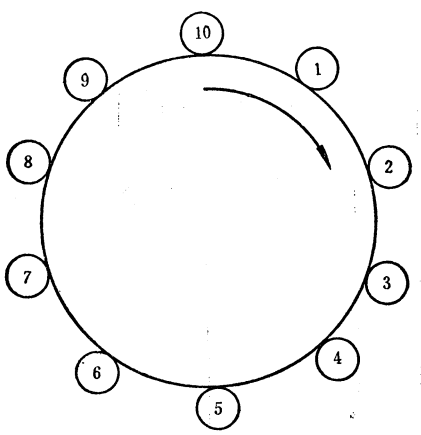


图 15-1

实际上有几个洞，狐狸永远也无法进去。这是一个有关余数的问题，我们设狐狸找小白兔的次数为n，那么在这n次中所走过的间隔数就是$S = 1 + 2 + 3\cdots + n = \frac{n(n+1)}{2}$，因为走完10个间隔就是绕山顶转了一圈，所以，找第n次时进入了哪一个洞，就相当于求S除以10所得的余数。我们不妨把10个洞编上0、1、2…、9这十个号码。题中第10洞便用第0洞表示。

我们知道所有的整数除以10得到的余数只可能是0，1，2…9这十个数，因此，次数n除以10的余数也就只可能是这十个数。用式子表示，就是$n \equiv x \pmod{10}$，$0 \leqslant x \leqslant 9$。

下面我们再叙述一下余数的一些性质：

（1）设a、b、c均为整数，若a、b分别除以正整数m所得的余数相同，那么a + c和b + c分别除以m所得的余数也相同。用式子表示就是若$a \equiv b \pmod{m}$，则$a + c \equiv b + c \pmod{m}$。

（2）设a、b、c、d均为整数，若a、b分别除以正整数m所得的余数相同，c、d分别除以正整数m所得余数相同，那么ac和bd分别除以m所得的余数也相同。即：若$a \equiv b \pmod{m}$，$c \equiv d \pmod{m}$，则$ac \equiv bd \pmod{m}$。

把上面的性质用于 n(n+1) 和x(x+1) 分别除以 10 所得的余数上来，我们可判断出它们的余数是相同的。即：由 n ≡x(mod10) 可得n+1≡x+1(mod10),n(n+1)≡x(x+1)(mod10)，$\frac{n(n+1)}{2}\equiv\frac{x(x+1)}{2}$(mod10)。

其中的 x 只可能等于 0 、 1 、 2 、… 9 这十个数。将这十个数分别代入 x(x+1) 中去，我们得出 x (x +1) 除以10的余数，和在图15-1中分析第 2 、 4 、 7 、 9 洞为安全洞是一致的。

（二）程 序 设 计

1）设计思路

定义数组M(10) ，表示10个洞。首先将这10个数组元素全冲成零。由键盘敲入需要狐狸寻找次数N，利用循环语句，由 1次到N次，按题意规则去找。每进入一个该进的洞时，代表该洞的数组元素便加 1 ，依次循环下去，当找到第N次结束时，打印出数组M（10），数组元素所赋的值，表示狐狸进入该洞的次数，若数组元素值为零，表示狐狸从未进入该洞，便是白兔可能躲藏在这洞里。

程序中示例由键盘敲入寻找次数N = 90和N = 255，运行后打印出数组元素 M(2)、M(4) 、M(7) 、M(9) 为零，表示第 2 、 4 、 7 、 9 洞狐狸从未进去过，与笔算推导一致。

2）框图　见图15-2

3）源程序及运行结果

```
      DIMENSION  M(10)
      WRITE (10) 'NUMBER  OF  SEARCHING'
      READ (11, 5) N
5     FORMAT (I3)
      DO  10  I=1, 10
10    M(I) =0
      K = 0
      DO  20  I=1, N
      K = K + I
      IF (K. LE. 10) GOTO  30
      L=K-K/10*10
      IF (L. EQ. O) GOTO  15
      M(L) = M(L) +1
      GOTO  20
15    M(10) = M(10) +1
      GOTO  20
30    M(K) = M(K) +1
20    CONTINUE
      WRITE (10, 40)  (K1, M(K1), K1=1, 10)
40    FORMAT (10 (3X, 2HM (, I2, 2H) =, I4/) )
      J = 0
      DO  50  I=1, 10
```

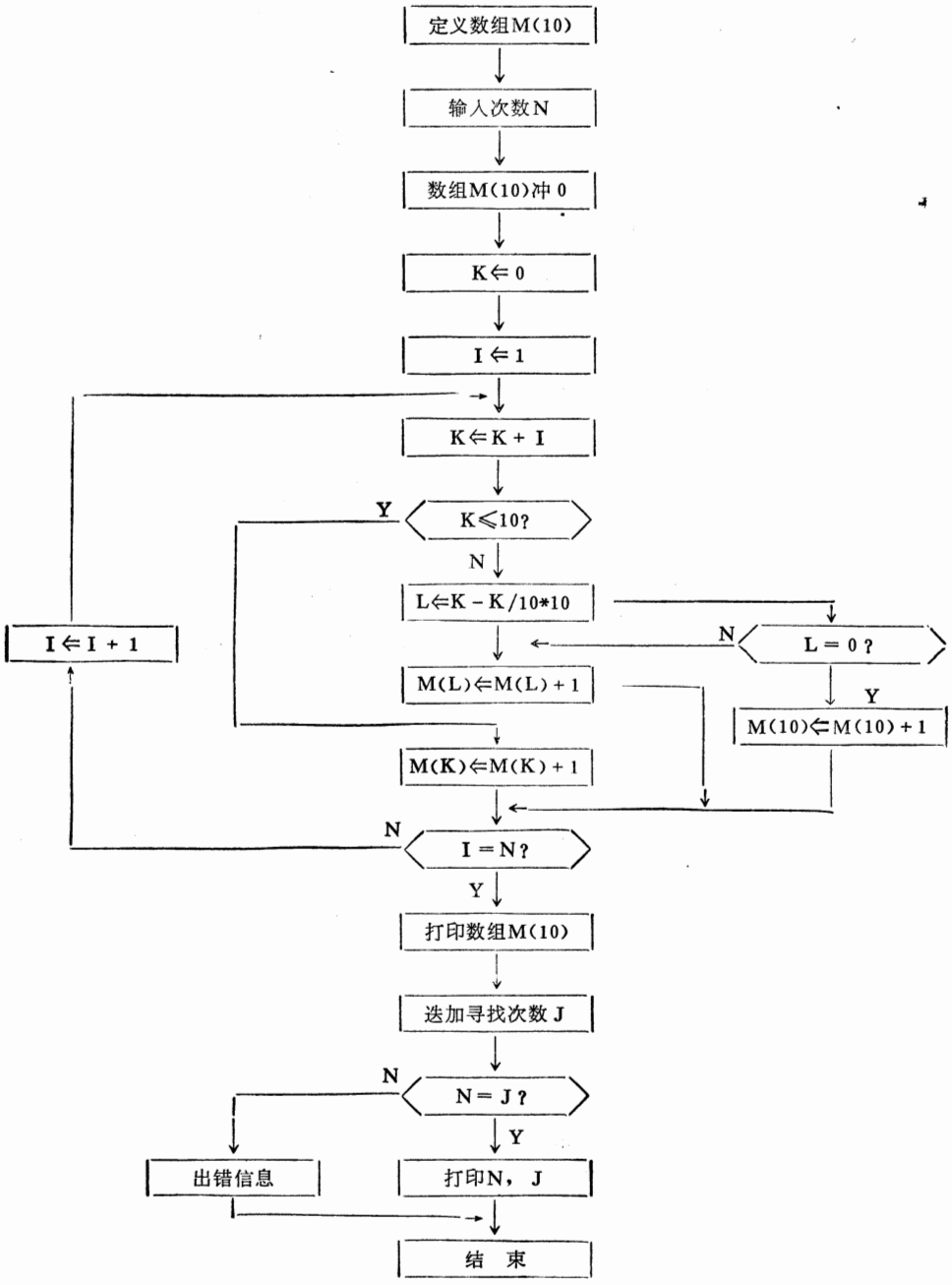


图 15-2

```
50      J=M(I)+J
        IF (J. NE. N) STOP  'ERR'
        WRITE (10, 60) N, J
60      FORMAT (10X, 2HN=, I3, 5X, 2HJ=, I3)
        STOP
        END
```

```
NUMBER  OF  SEARCHING
 90                                  输入寻找90遍
   M ( 1) =18
   M ( 2) = 0                        数组元素值为 0 是未寻找过的安全洞。
   M ( 3) = 9
   M ( 4) = 0                        数组元素的值，是进入该洞的次数。
   M ( 5) =19
   M ( 6) =18
   M ( 7) = 0
   M ( 8) = 9
   M ( 9) = 0
   M (10) =17
          N = 90      J = 90
STOP
NUMBER  OF  SEARCHING
255                                  输入寻找255遍
   M ( 1) =51
   M ( 2) = 0
   M ( 3) =25
   M ( 4) = 0
   M ( 5) =52
   M ( 6) =51
   M ( 7) = 0
   M ( 8) =26
   M ( 9) = 0
   M (10) =50
          N = 255     J = 255
STOP
```

（三）思 考 题

假若沿山顶圆周依次编号有50个洞，狐狸从自己的第50个洞出发，第一次走两个间隔（两洞之间为一个间隔），进洞去找白兔，第二次走四个间隔进洞寻找白兔，第三次走六个间隔，依次类推。问白兔能有安全洞藏身吗？该洞号是几？

16 找出次品钮扣

贝美钮扣厂在成品包装时不小心，将一个次品钮扣掉进成品盒里，恰好该盒里共80个钮扣，其中有一个次品，其外表形状、颜色等与79个成品钮扣完全一样，只是次品重量比成品钮扣重量轻些。质量检验科科长拿出一台不带砝码的天平，趁机考核检查员们如何用最少的称量次数将次品查找出来，次品不准出厂，保证质量。

（一）笔算步骤和结果

称量四次便可找出较轻的次品钮扣，其过程如下。

（1）将钮扣分为三组：其中两组为27个，另一组为26个。先将27个的两组，置于天

平的两臂上，若不平衡，则次品钮扣必在轻的那组中，若达平衡，则次品必在26个那组中（可再将该组加一个正品钮扣，成为27个）。

（2）27个量取3次，每组9个，量一次，即可得出次品在哪一组。

（3）将9个分为3组，量一次，可定次品在何组。

（4）3个任取2个称量，便可定出次品（至此仅称量比较4次）。

（二）程序设计

1）设计思路

定义数组N（80），置放80个钮扣，已知成品钮扣比次品钮扣较轻。可设79个成品钮扣用数字2表示，一个次品钮扣用1表示。

用READ语句输入80个钮扣并打印出，其中一个次品“1”随机置放任一位置，程序判别四次（相当天平称量四次），便指出次品所在位置，并打印出。

据笔算分析，每判别一次便有几种可能情况，每种可能决定下次再判时从哪个位置（I）开始查找。M个钮扣放天平一端，总重为K 1。另一端从L = I + M开始数至M个为止，也称量M个钮扣，总重为K 2，利用算术条件语句判断K 1与K 2哪个轻或重，转至相应情况，再作如此处理，这样经四次称重判断，便可找出次品钮扣在数组中所藏的位置，并打印出。

程序举例将次品藏在第46个位置，判断四次将其找出，并打印出所在位置N（46），见打印结果最后一行。

读者可任意将次品变换在另一位置。若不继续布置寻找时，将N（1）输入0时便停止。

2）框图 见图16

3）源程序及运行结果

```
      DIMENSION  N (80)
1111  READ (5, 50)  (N(I) , I=1, 80)
 50   FORMAT (10I1)
      IF (N ( 1 ) . EQ.  0 ) STOP
      WRITE (6, 60)  (I, N (I) , I=1, 80)
 60   FORMAT ( (3X, 4 (2HN (, I2, 2H) =, I1, 8X) /) )
      I = 1
 10   M = 27
      CALL  SUB1 (N(I) , M, K1)
      L = I +M
      CALL  SUB1 (N(L) , M, K2)
      IF (K1-K2) 111, 122, 133
111   J = 1
      GO  TO  11
122   J = 2
      I =55
      GO  TO  11
133   J = 3
```

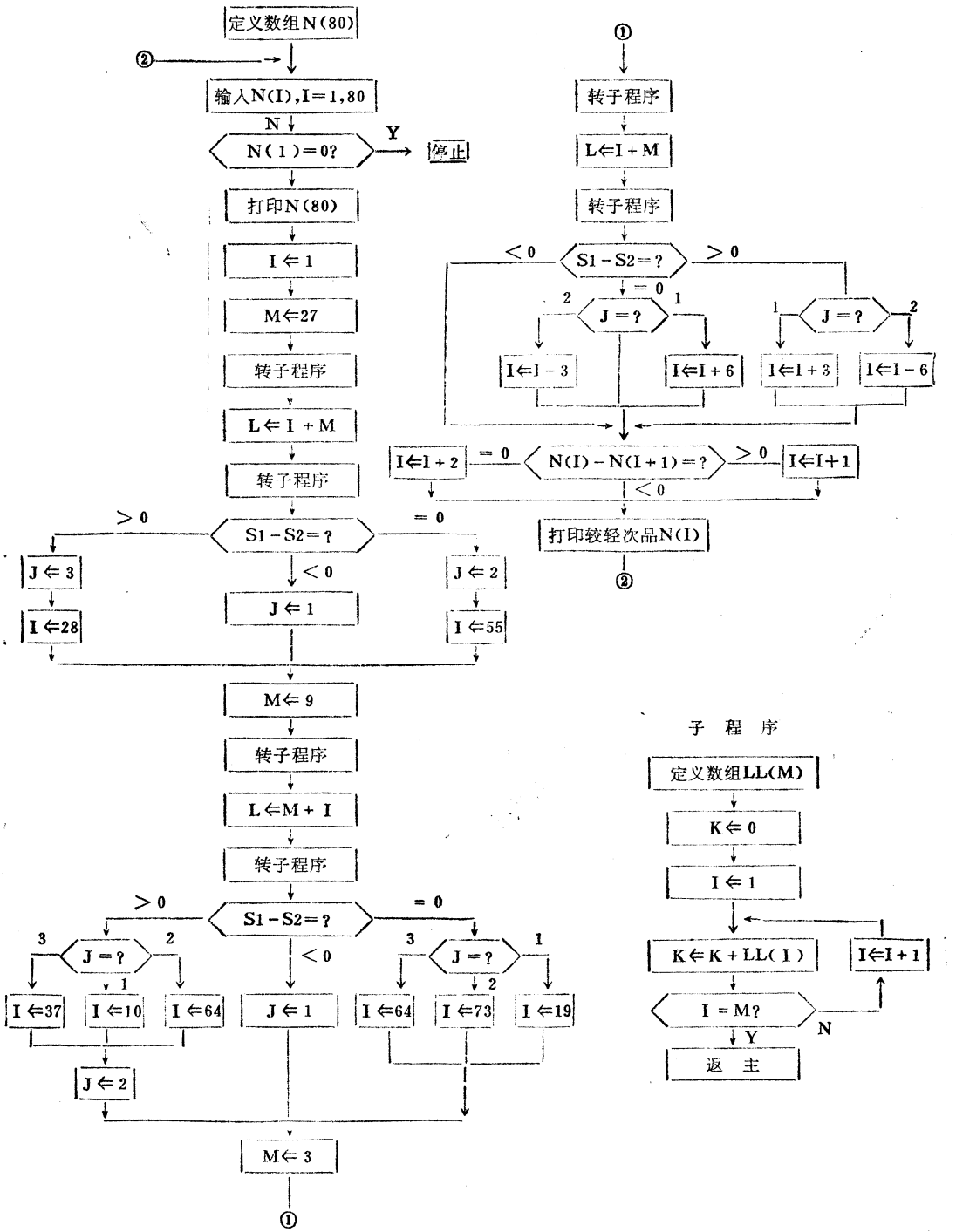

定义数组N(80)
②
输入N(I),I=1,80
N
N(1)=0?
Y
停止
打印N(80)
I⇐1
M⇐27
转子程序
L⇐I+M
转子程序
S1-S2=?
>0
=0
<0
J⇐3
I⇐28
J⇐2
I⇐55
J⇐1
M⇐9
转子程序
L⇐M+I
转子程序
S1-S2=?
J=?
I⇐37
I⇐10
I⇐64
J⇐2
J⇐1
I⇐64
I⇐73
I⇐19
M⇐3
①
①
转子程序
L⇐I+M
转子程序
S1-S2=?
J=?
I⇐I-3
I⇐I+6
J=?
I⇐I+3
I⇐I-6
I⇐I+2
N(I)-N(I+1)=?
I⇐I+1
打印较轻次品N(I)
②
子 程 序
定义数组LL(M)
K⇐0
I⇐1
K⇐K+LL(I)
I⇐I+1
I=M?
N
Y
返 主

图 16

```
      I = 28
11    M = 9
      CALL  SUB1 (N(I) , M, K1)
      L = M + I
      CALL  SUB1 (N(L) , M, K2)
      IF (K1 - K2) 144, 155, 166
144   J = 1
      GO  TO  14
155   GO  TO (151, 152, 153) , J
151   I = 19
      GO  TO  144
152   I = 73
      GO  TO  144
153   I = 46
      GO  TO  144
166   GO  TO (161, 162, 163) , J
161   I = 10
160   J = 2
      GO  TO  14
162   I = 64
      GO  TO  160
163   I = 37
      J = 2
14    M = 3
      CALL  SUB1(N(I) , M, K1)
      L = M + I
      CALL  SUB1(N(L) , M, K2)
      IF (K1 - K2) 17, 188, 199
188   GO  TO (1, 2) , J
1     I = I + 6
      GO  TO  17
2     I = I - 3
      GO  TO  17
199   GO  TO (3, 4) , J
3     I = I + 3
      GO  TO  17
4     I = I - 6
17    IF (N(I) - N(I + 1) ) 20, 21, 22
20    WRITE (6, 15) I, N (I)
15    FORMAT (11X, 2HN (, I2, 2H) =, I1)
      GO  TO  1111
21    I = I + 2
      GO  TO  20
```

```
22        I = I + 1
          GO  TO  20
          END
          SUBROUTINE  SUB1 (LL, M, K)
          DIMENSION  LL (M)
          K = 0
          DO  23  I=1, M
23        K=K+LL (I)
          RETURN
          END
```

```
N ( 1) = 2      N ( 2) = 2      N ( 3) = 2      N ( 4) = 2
N ( 5) = 2      N ( 6) = 2      N ( 7) = 2      N ( 8) = 2
N ( 9) = 2      N (10) = 2      N (11) = 2      N (12) = 2
N (13) = 2      N (14) = 2      N (15) = 2      N (16) = 2
N (17) = 2      N (18) = 2      N (19) = 2      N (20) = 2
N (21) = 2      N (22) = 2      N (23) = 2      N (24) = 2
N (25) = 2      N (26) = 2      N (27) = 2      N (28) = 2
N (29) = 2      N (30) = 2      N (31) = 2      N (32) = 2
N (33) = 2      N (34) = 2      N (35) = 2      N (36) = 2
N (37) = 2      N (38) = 2      N (39) = 2      N (40) = 2
N (41) = 2      N (42) = 2      N (43) = 2      N (44) = 2
N (45) = 2      N (46) = 1      N (47) = 2      N (48) = 2
N (49) = 2      N (50) = 2      N (51) = 2      N (52) = 2
N (53) = 2      N (54) = 2      N (55) = 2      N (56) = 2
N (57) = 2      N (58) = 2      N (59) = 2      N (60) = 2
N (61) = 2      N (62) = 2      N (63) = 2      N (64) = 2
N (65) = 2      N (66) = 2      N (67) = 2      N (68) = 2
N (69) = 2      N (70) = 2      N (71) = 2      N (72) = 2
N (73) = 2      N (74) = 2      N (75) = 2      N (76) = 2
N (77) = 2      N (78) = 2      N (79) = 2      N (80) = 2
```

N (46) = 1　　(第46个是次品)

(三) 思 考 题

1) 若80个钮扣中有一个次品，但不知该次品重量轻或重，只要能指出次品即可，你需要称量几次，如何称量？

2) 若40个钮扣中有一个次品，也不知该次品重量轻或重，只要指出次品即可，如果称量四次以下可以吗？为什么？如何称量？

3) 附注：要在 $(3^N-1)/2$ 个物件中找出一个假的，需称量N次，而且少于这个次数是不充分的，你说对吗？

17 生者与死者

十七世纪法国数学家加斯帕·巴克的一本《数目之游戏问题》，其中有不少有趣的数学

游戏，约瑟游戏是其中之一：

“15个基督教徒和15个异教徒同乘一船航行。途中风浪大作，危险万状，领航人告诉大家，只有将全船人的一半投入海中，其余人始能幸免，大家赞成这个办法，并议定30人围成一圈，由第一人数起，挨次报数，每数至第9个人，便把他投入海中，循环进行，直到仅剩15个乘客为止，问如何排法，方可使每次投海者都是异教徒？”

（一）笔算步骤和结果

这个问题具有历史局限性，但由于它流传极广，还是编入。有人曾把它的算法隐示于下列诗句之中：

From numbers' aid and art,

借助数字的技巧和帮助，

Never will fame depart!

名声就决不会离去！

这句诗中元音字母依次为 o u e a i a a e e i a e e a，

分别用1，2，3，4，5代替a，e，i，o，u便得一排数（带圈的数字表示基督教徒，不带圈的表示异教徒）：

④5②1③1①2②3①2②1 故所求排法是④个基督教徒，5个异教徒，再②个基督教徒，1个异教徒，③个基督教徒…。

具体笔算方法是约瑟计数问题，省略推导过程。

（二）程序设计

1）设计思路

定义数组K（30）置放30人，数组M（15）置放入海者，数组N（15）置放留在船上的人。循环语句控置变量I由1～30，每数到9，便将该数组元素K（I）冲成零，表示投海者，投海者记数器L。再依次继续数。当数到30，再从1重复数，每数到9，判该数组元素K（I）是否为零。若是零，则不再计数，跳过它，继续数。当投海者L＝15时，便完成任务，打印出投海者数组M和留在船上的人（数组N）。即生、死者的序号，见打印结果。

2）框图 见图17

3）源程序及运行结果

```
      DIMENSION  K(30) , M(15) , N(15)
      DO  10  I=1, 30
10    K(I)=I
      L = 0                                  记入海者人数L
      JS= 0                                  JS是数9个人的记数器
15    DO  20  I=1, 30
      IF (K(I) . EQ. 0) GO  TO  20           已入海
      JS=JS+1
      IF (JS. NE. 9) GO  TO  20
      JS= 0
      L = L + 1
      M(L)=K(I)
```

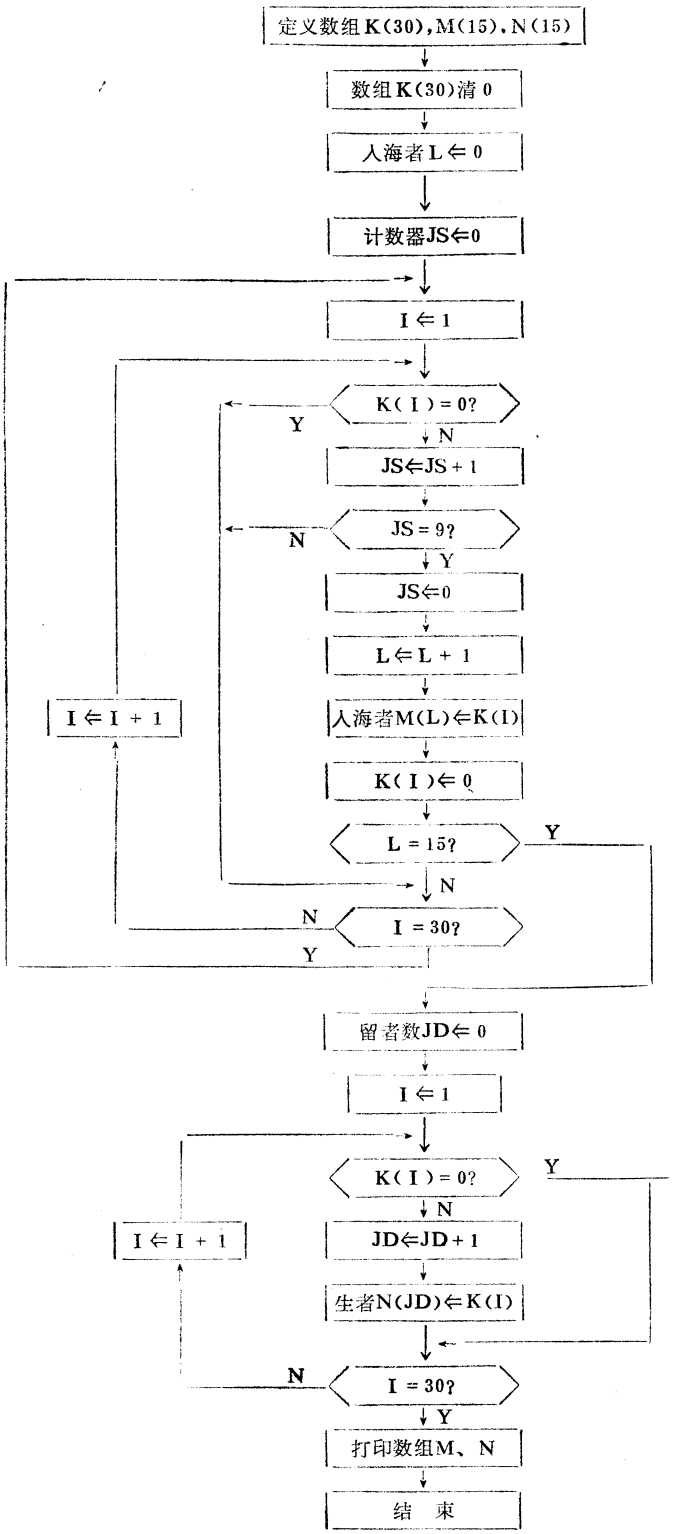

定义数组K(30),M(15),N(15)
数组K(30)清0
人海者L⇐0
计数器JS⇐0
I⇐1
K(I)=0?
Y
N
JS⇐JS+1
JS=9?
N
Y
JS⇐0
L⇐L+1
人海者M(L)⇐K(I)
K(I)⇐0
L=15?
Y
N
I=30?
N
Y
I⇐I+1
留者数JD⇐0
I⇐1
K(I)=0?
Y
N
JD⇐JD+1
生者N(JD)⇐K(I)
I⇐I+1
N
I=30?
Y
打印数组M、N
结 束

图 17

```
      K(I) = 0
      IF (L. EQ. 15) GO TO 30
20    CONTINUE
      GO TO 15
30    JD = 0                                  记留在船上人数JD
      DO 40 I = 1, 30
      IF (K(I) . EQ. 0) GO TO 40
      JD = JD + 1
      N(JD) = K(I)
40    CONTINUE
      WRITE (6, 50) (M(J), J = 1, 15), (N(J), J = 1, 15)
50    FORMAT (2X, 9HDROP SEA:, 15 (I4)/2X, 9HON SHIP:, 15 (I4)/)
      STOP
      ENO
```

打印结果

```
DROP SEA:   9  18  27   6  16  26   7  19  30  12  24   8  22   5  23  投海者
ON   SHIP:  1   2   3   4  10  11  13  14  15  17  20  21  25  28  29 留在船上
```

（三）思 考 题

如果倾刻间风平浪静，呼救投海者还可上船，但救上来的人有两种情况：（1）投海顺序为偶数者还活着；（2）投海顺序为奇数者已死去。

问救活的人和死去者，在原队形的位置各是几？

18 如此辨认伪币

12个相同的钱币，其中有一个是伪币，而不知此伪币比真币轻或重（真币皆为等重），用天平仅限量取三次，便要找出该伪币，并指出该伪币比真币轻或重。

（一）笔算步骤和结果

将币分为3组，每组4个，第一次将两组分别置于天平之两端，可出现两种情形，分别讨论。

（1）当天平达平衡时，伪币必在尚未称的那组中，达平衡的8个钱币皆为真币，再将可疑4个币标以1，2，3，4。再进行第二次称量，将1，2，3，三个可疑币置天平之一端，另一端置真币3个，可出现两种情况：

（a）天平达平衡，此时钱币4即是伪币。第三次称量即可指出伪币较真币轻或重。

（b）一边较重时，此时伪币位于钱币1，2，3中，若置真币那端较重，则伪币就较轻，再称一次，即可显示钱币1，2，3哪个较轻，即是伪币。若可疑币1，2，3较重时，即伪币较重，同理再称一次，亦可找出伪币。

（2）天平不平衡时，即其余的钱币皆为真币。现将较重的四个标号1，2，3，4（若其中有一为伪币，则伪币较重），较轻的四个标号为1′，2′，3′，4′（若其中有一为伪币，则伪币较轻）。第二次称取时，将1，2，1′三个币置一端，3，4，2′置另一端，可能出现下列三种情况：

（a）天平达平衡时，此时伪币必定为3′和4′（而较真币轻），第三次称取时，将3′与4′分别置天平两端，较轻的即为伪币。

（b）钱币1，2，1′，一端较重，此时可得知3，4，1′为真币，因此伪币必在钱币1，2中（而比真币重）或者钱币2′（比真币轻）。第三次称取时，将钱币1与2，分别置于天平之两端，若平衡，则伪币为2′。若不平衡，则伪币为较重的那个。

（c）钱币3，4，2′那端较重，与（b）同理，可得出币1，2，2′为真币。若3，4中有一是伪币，则伪币较重。若1′是伪币，则伪币轻。第三次称取时，将钱币3与4置于天平两端，若平衡，则伪币为1′，若不平衡，则较重的是伪币。

（二）程序设计

1）设计思路

定义三个数组A1(4)，A2(4)，A3(4)，每个数组可容四个钱币，A1(4)表示钱币1～4的位置，A2(4)表示钱币5～8的位置，A3(4)表示9～12的位置。先用循环语句将12个真币（用数字1表示）置放这三个数组中，随后便用赋值语句置放一个伪币，取而代之真币的位置，若伪币轻，用数字0表示。若伪币重，用数字2表示。

据笔算分析三次称量的方法，便可指出伪币是轻或重以及它藏的位置。每次称量时可能出现几种不同情况中的一种情况，用条件转移语句转至相应情况分别处理。用M表示伪币所在的位置，打印时打出N为0表示伪币轻，N为2表示伪币重，最后还打印出伪币是轻（LIGHT）还是重（HEAVY）的字样。

2）框图　见图18

3）源程序及运行结果

```
      DIMENSION  A1(4), A2(4), A3(4)
      DO  10  I=1, 4
      A1(I) =1.0
      A2(I) =1.0
10    A3(I) =1.0
14    A2(3) =0.
      X1=0.
      X2=0.
      DO  15  I=1,4
      X1=X1+A1(I)
15    X2=X2+A2(I)
      IF (X1-X2) 20, 25, 30
30    E=A1(1) +A1(2) +A2(1)
      BB=A1(3) +A1(4) +A2(2)
      IF (B-BB) 35, 40, 45
35    IF (A1(3) -A1(4)) 50, 55, 60
50    M=4
      N=1
      GO  TO  100
55    M=5
      N=0
```

此循环语句是将12个钱币全安放同样重量（1.0）的真币。

语句14藏伪币，用它取而代之真币。

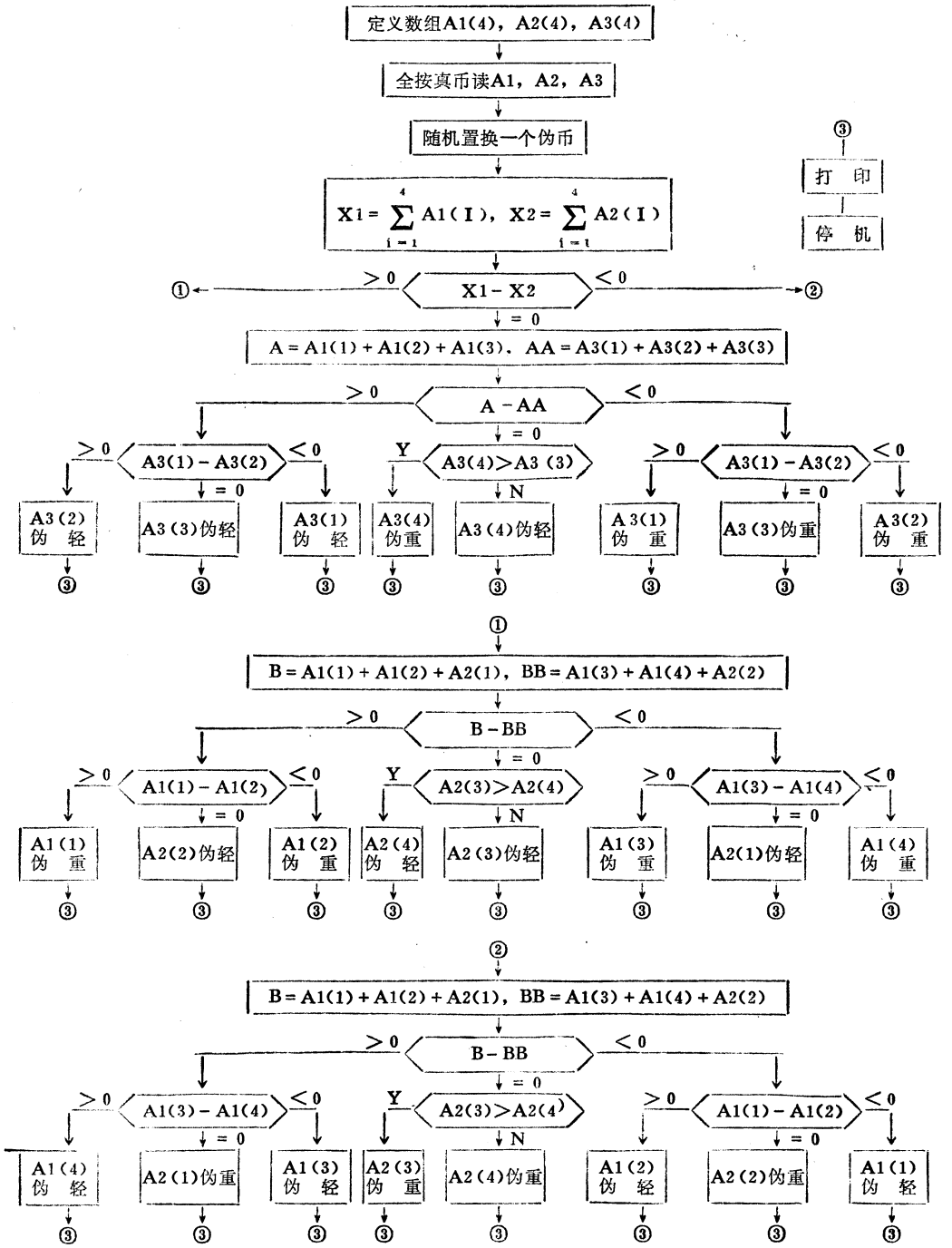


图 18

```
      GO  TO  100
60    M = 3
      N = 1
      GO  TO  100
40    IF (A2(3) . GT. A2(4) )GO  TO  42
```

```
      M = 7
      N = 0
      GO  TO  100
42    M = 8
      N = 0
      GO  TO  100
45    IF (A1(1) - A1(2) ) 65, 70, 75
75    M = 1
      N = 1
      GO  TO  100
70    M = 6
      N = 0
      GO  TO  100
65    M = 2
      N = 1
      GO  TO  100
20    B = A1(1) + A1(2) + A2(1)
      BB = A1(3) + A1(4) + A2(2)
      IF (B - BB) 80, 85, 90
80    IF (A1(1) - A1(2) ) 1, 2, 3
 1    M = 1
      N = 0
      GO  TO  100
 2    M = 6
      N = 1
      GO  TO  100
 3    M = 2
      N = 0
      GO  TO  100
85    IF (A2(3) . GT. A2(4) ) GO  TO  4
      M = 8
      N = 1
      GO  TO  100
 4    M = 7
      N = 1
      GO  TO  100
90    IF (A1(3) - A1(4) ) 51, 6, 7
51    M = 3
      N = 0
      GO  TO  100
 6    M = 5
      N = 1
      GO  TO  100
```

```
 7      M = 4
        N = 0
        GO  TO  100
25      A = A1(1) + A1(2) + A1(3)
        AA = A3(1) + A3(2) + A3(3)
        IF (A - AA) 9, 16, 11
 9      IF (A3(1) - A3(2) ) 12, 13, 14
12      M = 10
        N = 1
        GO  TO  100
13      M = 11
        N = 1
        GO  TO  100
14      M = 9
        N = 1
        GO  TO  100
16      IF (A3(4) . GT. A3(3) )GO  TO  150
        M = 12
        N = 0
        GO  TO  100
150     M = 12
        N = 1
        GO  TO  100
11      IF (A3(1) - A3(2) )26, 21, 22
26      M = 9
        N = 0
        GO  TO  100
21      M = 11
        N = 0
        GO  TO  100
22      M = 10
        N = 0
100     IF (N. EQ. 0) GO  TO  101
        GO  TO  102
101     WRITE (6, 201) A1, A2, A3
        WRITE (6, 54) M
        STOP
54      FORMAT (1X, 2HM=, I2, 5X, 12HIT  IS  LIGHT. )
102     WRITE (6, 201) A1, A2, A3
        WRITE (6, 56) M
56      FORMAT(1X, 2HM=. I2, 5X, 12HIT  IS  HEAVY.)
201     FORMAT (1X, 12F5.1)
        STOP
```

```
END
```

```
 1.0  1.0    1.0  1.0     1.0   1.0  0.0  1.0  1.0  1.0  1.0  1.0
M=7          IT   IS  LIGHT.
```

程序运行后打印出伪币在第7个位置，并且较轻。

```
 1.0  1.0  1.0  1.0  2.0   1.0  1.0  1.0  1.0  1.0  1.0  1.0
M=5          IT   IS HEAVY.
```

将程序语句14改为 A2(1)=2.0即在第五个位置上按放重币，程序搜寻后也打印出。

（三）思 考 题

商人对顾客说："我不慎，有一颗假珍珠掉进这个盒里，因外表、形状、颜色和大小都和那11颗真的珍珠完全一样，我也找不出哪一颗是假的，如果你买，我按11颗计价。"

顾客说："那好办，真、假珍珠重量不同。"于是顾客便在天秤上只称三次便对商人说："你骗我，这12颗珍珠至少有两个是假的。"

请问顾客是怎样称量的？

19 谁 在 说 谎

张三说李四在说谎，李四说王五在说谎，王五说张三和李四都在说谎。现在问：张三、李四、王五三个人，到底谁说的是真话，谁说的是假话。

（一）笔算步骤和结果

张三说的话是A，A真当且仅当B假；

李四说的话是B，B真当且仅当C假；

王五说的话是C，C真当且仅当A与B同时为假。

分两种情况进行讨论：第一种情况A真；第二种情况A假。

第一种情况下，A真，"即李四在说谎"。因此B假，即并非"王五正在说谎"。这样，C真，因此，A与B皆假，这与原来假设A真相矛盾。于是第一种情况不成立。

第二种情况下，A假，即并非"王五正在说谎"。这样B真，于是C假，这时没有矛盾。

结论：A假，B真，C假。

（二）程 序 设 计

1）设计思路

分析得出两种状态，用0表示假，1表示真。甲乙丙三人分别用A、B、C表示。没有下标为自己说的，如A、B、C。有下标是别人说自己的，如A1是表示别人说甲，同理B1、C1分别表示别人说乙、丙。由于题中有二人说到乙，所以B用到了两个下标——B1和B2。

利用循环语句分别判A、B、C说的是真话或假话，真话者置1，假话者置0，最后判三个人说法都符合时，便判断正确，请见程序运行后打印结果。

2）框图 见图19

3）源程序及运行结果

```
      INTEGER  A, A1, B, B1, B2, C, C1
      DO  10  A=0,1
```

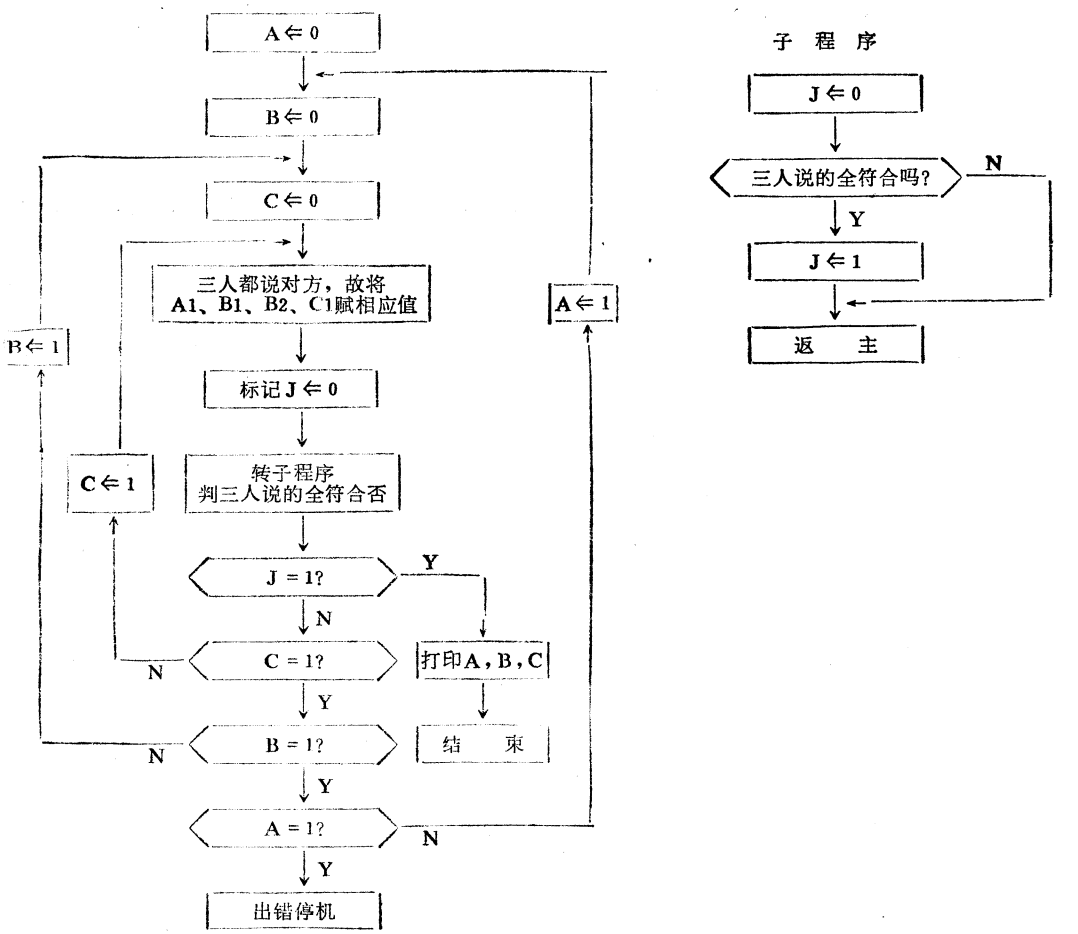


图 19

```
      DO  20  B = 0, 1
      DO  30  C = 0, 1
      IF (A. EQ. 0) GO  TO  13
      B1 = 0
      GO  TO  15
13    B1 = 1
15    IF (B. EQ. 0) GO  TO  23
      C1 = 0
      GO  TO  25
23    C1 = 1
25    IF (C. EQ. 1) GO  TO  50
      A1 = 1
      B2 = 1
      J = 0
      CALL  PZJ (A, A1, B, B1, B2, C, C1, J)
      IF (J. EQ. 1) GO  TO  60
```

```
      A1=0
      B2=1
      CALL  PZJ (A, A1, B, B1, B2, C, C1, J)
      IF (J. EQ. 1) GO  TO  60
      GO  TO  30
50    B2=0
      A1=0
      CALL  PZJ (A, A1, B, B1, B2, C, C1, J)
      IF (J. EQ. 1) GO  TO  60
30    CONTINUE
20    CONTINUE
10    CONTINUE
      STOP'ERR'
60    WRITE (6, 70) A, B, C
70    FORMAT (10X, 1HA, 3X, 1HB, 3X, 1HC/10X, 3 (I1, 3X))
      STOP
      END
      SUBROUTINE  PZJ (A, A1, B, B1, B2, C, C1, J)
      INTEGER  A, A1, B, B1, B2, C, C1
      J=0
      IF (A1. EQ. A. AND. B.EQ.B1.AND.B.EQ.B2.AND.C.EQ.C1) GO  TO  10
      RETURN
10    J=1
      RETURN
      END
```

程序运行后打印结果

```
A    B    C
0    1    0
```

A B C下面对应的数字表示真与假

（三）思 考 题

传说古代有一个“说谎国”和一个“老实国”。说谎国的人，口是心非，非说谎话不可，从来不说一句真话。老实国的人实事求是，一句假话也不说。有一天，两个说谎国的人混在老实国人中间，想偷偷进入老实国。他们俩和一个老实国的人进城的时候，哨兵嘛问他们三个：“你们是哪个国家的人？”第一个回答说：“我是老实国人”。第二个声音轻，哨兵没听清楚，于是指着第二个问第三个，“他是哪一国人，你又是哪一国人？”第三个回答说，“他说他是老实国人，我也是老实国人。”哨兵只知道三个人中间只有一个是老实国的人，可不知道是谁。他面对这样的回答，应该做如何分析？请你进行程序设计，协助哨兵判断。

提示：第二个人的回答哨兵没有听清楚。假如他是老实国人，他回答一定是“老实国人”。如果他是说谎国人，他要说谎，回答一定也是“老实国人”。第三个人如果是说谎国人，他在转述第二个人的回答的时候必定要说谎，就会说成“他说他是说谎国人”。可是

第三个人并不这样说，可见他没有说谎，他是老实国的人。

20 辨别鸡蛋和鸭蛋

三个鸡蛋和三个鸭蛋，分放在三个盒子里，每个盒子里放两个；一个盒子里放的是两个鸡蛋，一个盒子里放两个鸭蛋，另一个盒子里放一个鸭蛋和一个鸡蛋。而每个盒子上都贴有纸条，上面分别写着："两个鸭蛋"、"两个鸡蛋"、"一个鸡蛋和一个鸭蛋"。但这些纸条上写的都与盒中实际情况不符。要求从一个盒子里取出一个蛋，看一看，但不能看另外一个蛋。就区别出哪个盒子里是两个鸡蛋，哪个盒子里是两个鸭蛋，哪个盒子里是一个鸡蛋和一个鸭蛋。

（一）笔算步骤和结果

因为每一个盒子上贴的标记都是错误的，所以在贴有"一个鸡蛋和一个鸭蛋的盒子里拿出一个蛋看一下，就可以分晓。

若拿出的是鸡蛋，那么这个盒子里肯定两个都是鸡蛋（因不是一个鸡蛋和一个鸭蛋），其余两个盒子里只有一个鸡蛋了，所以贴有两个鸡蛋的标记的盒子里肯定是两个鸭蛋。而贴有两个鸭蛋的盒子里肯定是一个鸡蛋和一个鸭蛋。

若拿出来的是鸭蛋，那么这个盒子里肯定是两个鸭蛋，贴有两个鸭蛋的标记的盒子里，肯定是两个鸡蛋，贴有两个鸡蛋标记的盒子里肯定是一个鸡蛋和一个鸭蛋。

（二）程序设计

1）设计思路

道理很简单，如何进行程序设计，需仔细思索，我们将问题转化为另一形式，将六个蛋装在三个盒子里，互相搭配有几种装法，然后找出哪种装法是说谎，（即与标记不一样）。用1表示鸡蛋，用0表示鸭蛋，则三个盒子即数组A（3）分别装00、10、11，用十进制1、2、3分别表示00、10、11，即如下形式

鸡鸭蛋	十进制数	冲入数组
00⇒	1	⇒A（1）
10⇒	2	⇒A（2）
11⇒	3	⇒A（3）

数组A(3) 如上装法，是实际情况，不说谎，（若如此标签贴上，则全是真话），我们让标签是这样贴，但盒里的内容（装什么蛋）与标签不符。另设数组B（3），让B(3)内容与A（3)不一致，相当贴假标签（说谎）。如

A（1）⇐1　则让B（1）⇐2　或B（1）⇐3
A（2）⇐2　则让B（2）⇐3　或B（2）⇐1
A（3）⇐3　则让B（3）⇐1　或B（3）⇐2

这样使得三个盒子上的标签全在说谎，试编写程序找出上述数组B（3)的两种情况完全与数组A（3)内容相矛盾，请见程序运行后打印结果。

2）框图　见图20

3）源程序及运行结果

```
INTEGER  A(3), B(3), B1, B2, B3
DO  10  I=1, 3
```

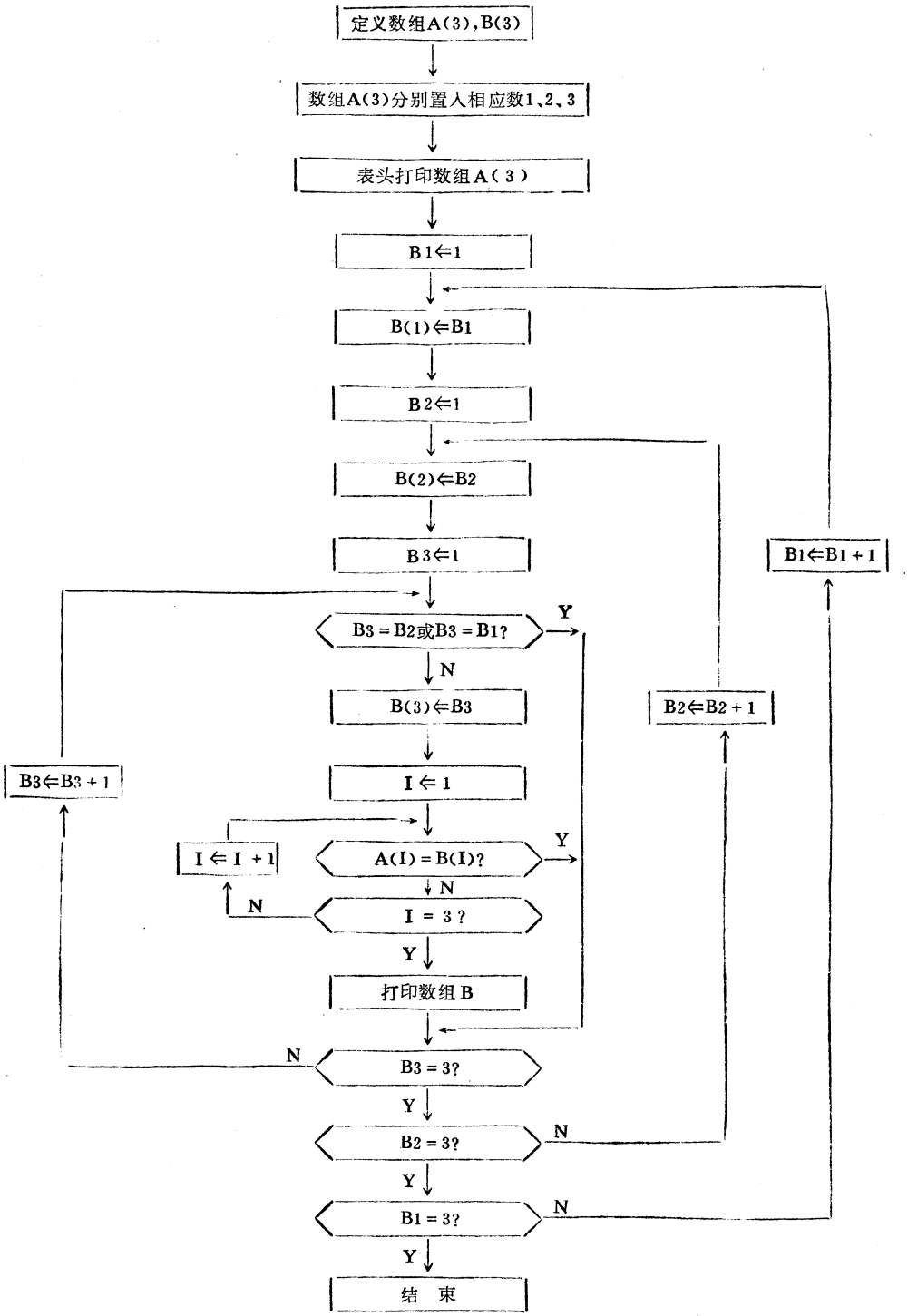

定义数组A(3),B(3)
数组A(3)分别置入相应数1、2、3
表头打印数组A(3)
B1⇐1
B(1)⇐B1
B2⇐1
B(2)⇐B2
B3⇐1
B3=B2或B3=B1?
Y
N
B(3)⇐B3
I⇐1
A(I)=B(I)?
Y
N
I=3?
N
I⇐I+1
Y
打印数组B
B3=3?
N
Y
B2=3?
N
Y
B1=3?
N
Y
结 束
B3⇐B3+1
B2⇐B2+1
B1⇐B1+1

图 20

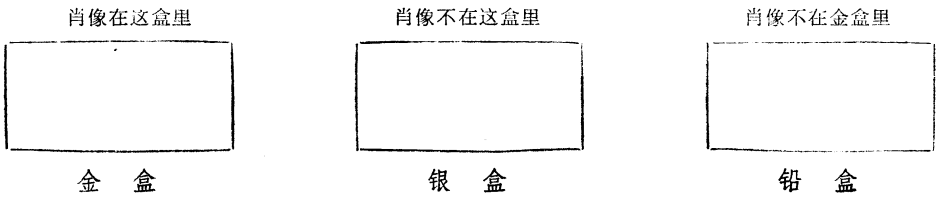
```
10      A (I) =I
        WRITE (6, 20)  (A(K) , K=1, 3)
20      FORMAT (10X, 4HA( 1), 3X, 4HA( 2), 3X, 4HA( 3), 3X/10X, 3(I3, 3X),
        C/8X, 21H……)
        DO  30  B1=1, 3
        B ( 1 ) = B1
        DO  40  B2=1, 3
        IF (B1. EQ. B2) GO  TO  40
        B ( 2 ) = B 2
        DO  50  B3=1, 3
        IF (B3. EQ. B2. OR. B3. EQ. B1) GO  TO  50
        B ( 3 ) = B 3
        DO  60  I=1, 3
        IF (A(I) . EQ. B(I) ) GO  TO  50
60      CONTINUE
        WRITE (6, 65)  (B (K) , K=1, 3)
65      FORMAT (10X, 3 (I3, 3X) /)
50      CONTINUE
40      CONTINUE
30      CONTINUE
        STOP
        END
```

A（1）	A（2）	A（3）	程序运行后打印结果
1	2	3	盒上标签
……	……	……	……
2	3	1	盒里装的与标签不符的装法
3	1	2	盒里装的与标签不符的另一装法

（三）思　考　题

数学家斯摩林根据莎士比亚的名剧《威尼斯商人》中的情节编成一道数学题：

女主角鲍西娅对求婚者说：“这里有三只盒子，一只是金盒子，一只是银盒子，一只是铅盒子，每只盒子上的铭牌上各写有一句话。三句话中，只有一句是真话，谁能猜中我的肖像放在哪一只盒子里，谁就能做我的丈夫”。盒子上的话见下图示意。



请你编写程序，看看鲍西娅的未婚夫怎样猜的。

21 公安人员断案

公安人员审问甲、乙、丙、丁四个嫌疑犯，已确知，这四个人当中仅有一人是偷窃者，还知道这四个人的说话，不是完全诚实者，就是完全说谎者。在回答公安人员的问话中：

甲说："乙没有偷，是丁偷的。"

乙说："我没有偷，是丙偷的。"

丙说："甲没有偷，是乙偷的。"

丁说："我没有偷，我用的那东西是我家里的。"

据上述四人答话，请判断谁是偷窃者。

（一）笔算步骤和结果

从集合论角度，由各人的陈述，可写出式子：

$B'D+BD'$，$B'C+BC'$，$A'B+AB'$，$D'+D$。其中A、B、C、D各代表甲、乙、丙、丁偷东西的陈述，而A′、B′、C′、D′分别代表甲、乙、丙、丁不曾偷东西的陈述。根据上述分析，在此省略将其联合并进一步简化，最后得出乙是偷窃者。

（二）程序设计

1）设计思路

可设四种状态：6表示说谎，7表示说真话；0表示偷了，1表示没偷。数组MA(4)的角标1～4，分别表示甲、乙、丙、丁四人。有的因几个人对同一个人不同说法，故又设数组MB(4)、MC(4)，含意同数组MA(4)。这些数组元素不全都用到，如数组元素MA(3)便没用到，最后将用到的数组元素MB(3)赋值给MA(3)，以便MA全有内容，便于输出。

利用循环语句控制变量6和7，表示四个人的说谎与否两种状态，对每种状态用0或1表示偷与否来对应。最后判断四人说话都互相一致时，打印出数组元素MA为0者，其下标便是偷窃者，如打印出MA(2)，便判出偷者是乙。

2）框图　见图21

3）源程序及运行结果

```
      DIMENSION  MA(4), MB(4), MC(4)
      DO  10  MD1=6,   7
      DO  20  MD2=6,   7
      DO  30  MD3=6,   7
      DO  40  MD4=6,   7
      IF (MD4. EQ. 6) GOTO  50
      MA(4) =1
      GOTO  60
50    MA(4) =0
60    IF (MD3. EQ. 6)  GOTO  70
      MA(1) =1
      MA(2) =0
      GOTO  80
```

甲、乙、丙、丁依次答话

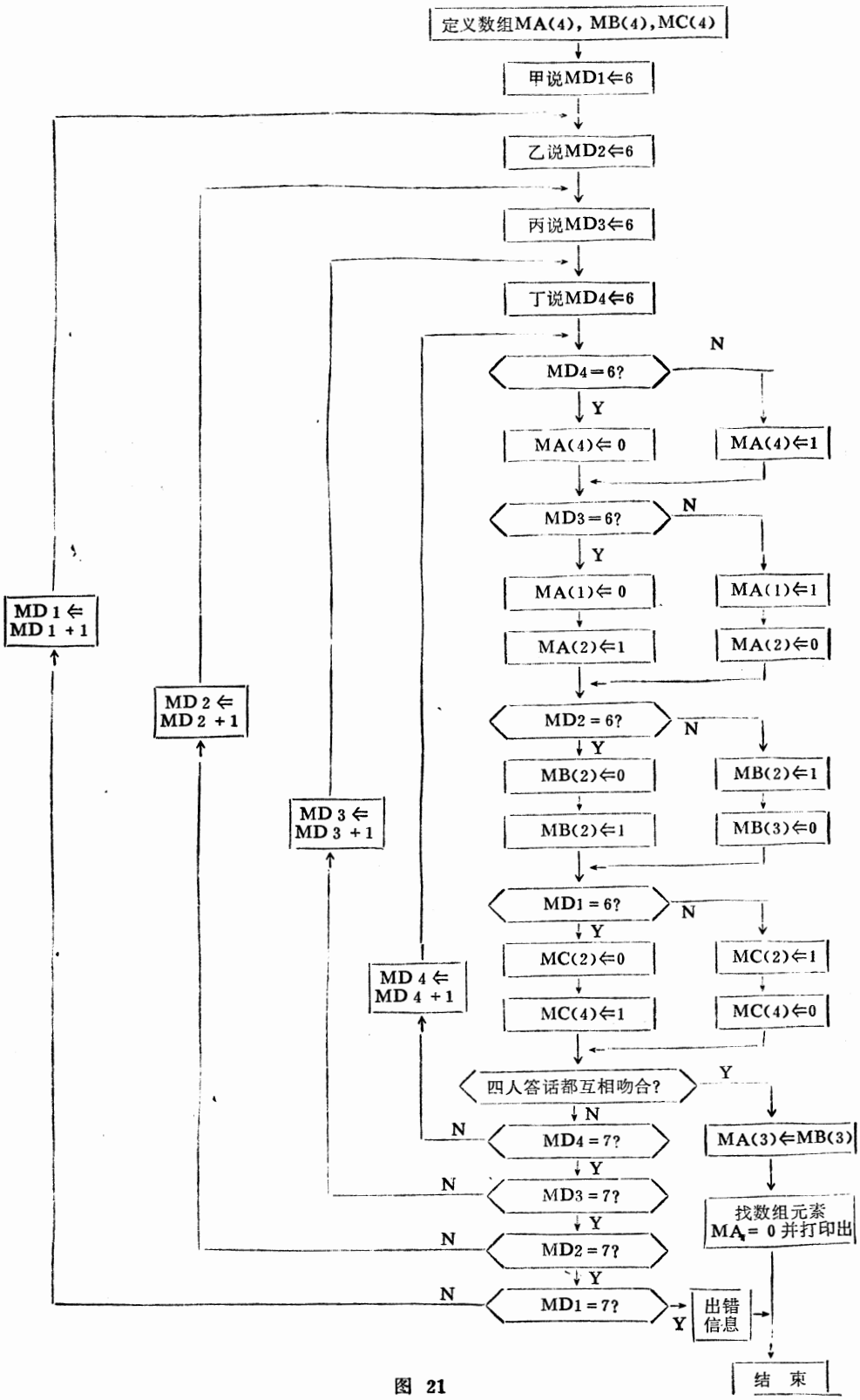

定义数组MA(4), MB(4),MC(4)
甲说MD1⇐6
乙说MD2⇐6
丙说MD3⇐6
丁说MD4⇐6
MD4=6?
N
Y
MA(4)⇐0
MA(4)⇐1
MD3=6?
N
Y
MA(1)⇐0
MA(1)⇐1
MA(2)⇐1
MA(2)⇐0
MD2=6?
N
Y
MB(2)⇐0
MB(2)⇐1
MB(2)⇐1
MB(3)⇐0
MD1=6?
N
Y
MC(2)⇐0
MC(2)⇐1
MC(4)⇐1
MC(4)⇐0
四人答话都互相吻合?
Y
N
MD4=7?
N
Y
MD3=7?
N
Y
MD2=7?
N
Y
MD1=7?
Y
MA(3)⇐MB(3)
找数组元素
MA=0 并打印出
出错
信息
结 束
MD1⇐
MD1+1
MD2⇐
MD2+1
MD3⇐
MD3+1
MD4⇐
MD4+1

图 21

```
 70      MA(1) = 0
         MA(2) = 1
 80      IF (MD2. EQ. 6) GOTO  90
         MB(2) = 1
         MB(3) = 0
         GOTO  100
 90      MB(2) = 0
         MB(3) = 1
100      IF (MD1. EQ. 6) GOTO  110
         MC(2) = 1
         MC(4) = 0
         GOTO  120
110      MC(2) = 0
         MC(4) = 1
120      IF (MA(2) . EQ. MB(2) )GOTO  130
         GOTO  40
130      IF (MB(2) . EQ. MC(2) ) GOTO  140
         GOTO  40
140      IF (MC(4) . NE. MA(4) ) GOTO  40
         GOTO  150
 40      CONTINUE
 30      CONTINUE
 20      CONTINUE
 10      CONTINUE
         STOP'ERR1'
150      MA(3) = MB(3)
         DO  160  I = 1, 4
         IF (MA(I) . EQ. 0) GOTO  180
160      CONTINUE
         STOP'ERR2'
180      WRITE (6, 170) I
170      FORMAT (9X, 2HA(,I1, 1H) )
         STOP
         END
```

判四人答话都互相吻合吗

A（2）　　程序运行后打印出A（2），找出乙是偷窃者。

（三）思　考　题

怎样辨认是男孩还是女孩。

最近，有位朋友外出到克罗特地区，碰到了一件使他为难的事：那个地方有一种奇特的习惯；男人和男孩说真话；而女人和女孩从不连续地说真话或假话，如果前句是真话，则她下句便是假话；反之亦然。

有一次，他遇到一对带着孩子的克罗特夫妇，朋友问这个孩子，“你是男孩吗？”孩

子用克罗特语回答，朋友听不懂他的话。恰好，这对夫妇都会说朋友的语言。其中之一说："孩子说，我是男孩'。"另一个说："孩子是女孩，孩子说谎了。"根据这对夫妇的话，编写程序判断出这孩子是男孩还是女孩？

22 确定值班大夫何日值班

某医院内科有A、B、C、D、E、F、G七位大夫一星期内每人要轮流值班一天。现在已知：

A大夫值班日比C大夫晚一天；

D大夫值班日又比E大夫的前一天要晚三天；

B大夫值班日比G大夫早三天；

F大夫值班日在B、C值班日中间，且正好是星期四。

请你确定星期一至星期日，究竟是那位大夫在值班？

（一）笔算步骤和结果

据已知条件，不难得出如下结论：

A大夫是星期日值班，B大夫是星期二值班，C大夫是星期六值班，D大夫是星期三值班，E大夫是星期一值班，F大夫是星期四值班，G大夫是星期五值班。

（二）程序设计

1）设计思路

定义数组MA（7）依次表示A～G七位大夫。利用循环语句，据已给条件、分别确定数组元素的数值，该数值便是该大夫的值班日（星期几）。先找B大夫的值班日，再依次找G、C、A、D、E、F大夫的值班日，符合条件的数值1～7，分别表示星期一至星期日。最后打印出数组MA。

2）框图 见图22

3）源程序及运行结果

```
DIMENSION  MA(7)
DO  10  MA2=1, 3
MA7=MA2+3
DO  20  MA3=5, 7
MA1=MA3+1
DO  40  MA4=1, 7
MA5=MA4-2
MA(6) = 4
MA(1) =MA1
MA(2) =MA2
MA(3) = MA3
MA(4) =MA4
MA(5) = MA5
MA(7) =MA7
DO  50  I=1, 7
IF (MA (I) . GT. 7) GOTO  40
```

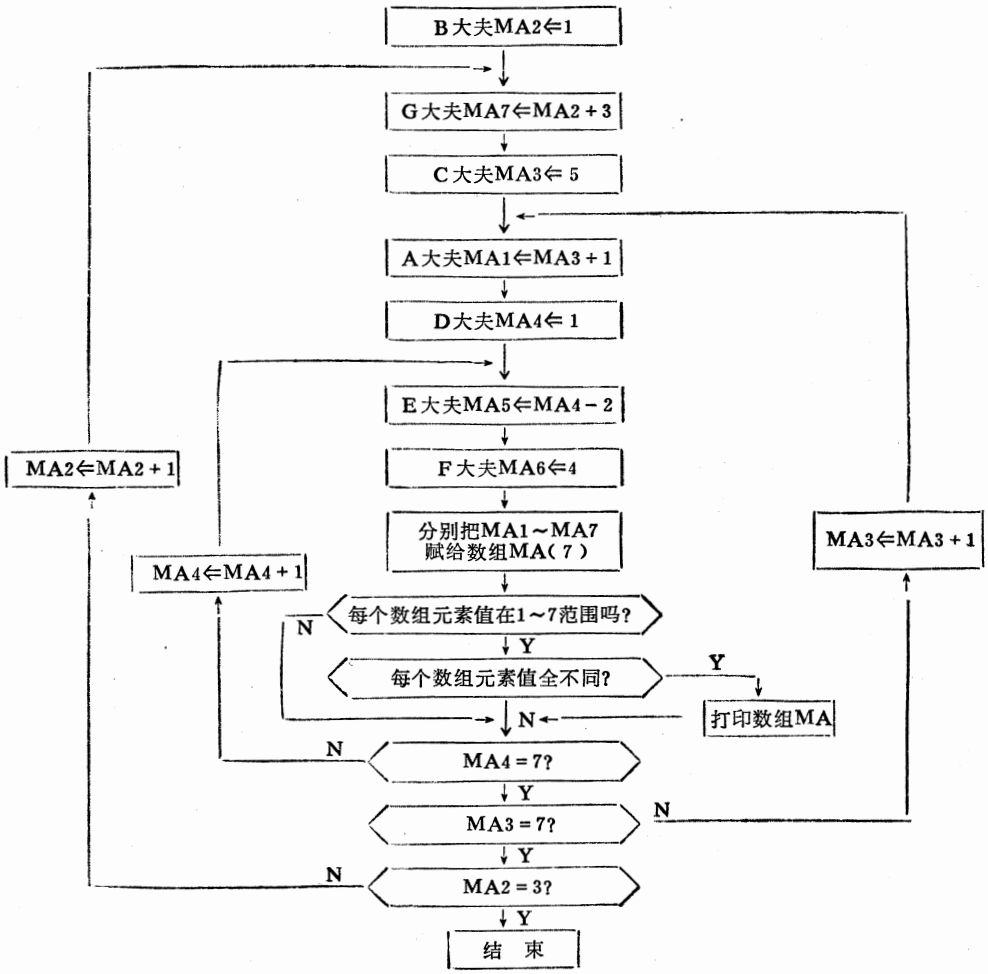


图 22

```
      IF (MA(I) . LT. 1) GOTO 40
50    CONTINUE
      DO 60 J=1, 6
      J1=J+1
      DO 70 K=J1, 7
      IF (MA(J) EQ. MA(K)) GOTO40
70    CONTINUE
60    CONTINUE
      DO 80 MT=1, 7
      WRITE (6, 90) MT, MA(MT)
90    FORMAT (1X, 2HA (, I3, 2H)=, I3)
80    CONTINUE
40    CONTINUE
20    CONTINUE
10    CONTINUE
```

```
STOP
END
```

```
下面是打印结果。等号左边数组依次是A～G大夫，等号右边数字是星期几值班
A(1) = 7
A(2) = 2
A(3) = 6
A(4) = 3
A(5) = 1
A(6) = 4
A(7) = 5
```

（三）思 考 题

这一天是星期几——曹、钱、刘、洪四人出差，同住一个招待所。一天下午，他们分别要找一个单位去办事。甲单位星期一不接待；乙单位星期二不接待；丙单位星期四不接待；丁单位只在星期一、三、五这三天接待。当然星期日四个单位都不接待。曹："两天前，我已经去谈了一次，今天再去一次，还可以与老洪同走一段路。"钱："今天我一定要去，要不明天人家就不接待。"刘："这星期，前几天和今天我去都能办成事。"洪："我今天和明天去，对方都接待。"请你推算这一天是星期几，他们四人各自要去哪个单位办事。

23 辨别四对夫妻关系

有甲、乙、丙、丁四个男子和梅、兰、竹、菊四个女子，分别在四个机构工作，这四个机构都有同样的一条怪规则，绝不同时雇用一对夫妇，亦不允许兼职，现已知四男、四女恰是四对夫妇，而四机构的每一机构，分别雇用了其中的一男一女，已知条件如下：

（1）梅和兰的丈夫在一机构工作；

（2）甲和兰在一个机构工作；

（3）乙的妻子和丙在一个机构工作；

（4）竹的丈夫和甲的妻子在一个机构工作；

（5）假若菊是甲或丙的妻子；

（6）则丁是梅或兰的丈夫。

（一）笔算步骤和结果

这类题运用逻辑推理，根据已知条件列成表23-1。

表 23-1　　工作关系　　表A　　夫妻关系　　表B

男＼女	梅	兰	竹	菊	梅	兰	竹	菊
甲	0	1	0	0	0	0	0	1
乙	0	0	0	1	0	0	1	0
丙	0	0	1	0	1	0	0	0
丁	1	0	0	0	0	1	0	0

答案是甲与菊、丙与梅、乙与竹、丁与兰是夫妻关系。

注：A表中“1”表示在同一机关中工作的人，“0”表示空白。

B表中“1”表示是夫妻关系，“0”表示它的空白。

（二）程 序 设 计

1）设计思路

四男四女的工作关系和夫妻关系，分别可用4×4矩阵的二维数组表示，行顺序表示甲、乙、丙、丁，列顺序表示梅、兰、竹、菊。定义数组A（4，4）、B（4，4）分别表示工作关系、夫妻关系，并分别暂存于数组E1（4，4，5）、E2（4，4，5）。首先将这些数组全赋值5，表示未用过的空白。根据题设已知的六个条件来改变A、B数组元素的值，即：有工作关系（同一机构工作）的，A（I，J）⇐1；否则，A（I，J）⇐0。同理，是夫妻关系，B（I，J）⇐1；否则，B（I，J）⇐0。

子程序ZX是填A、B数组，子程序ZY是判满足条件否，子程序ZZ是保存A、B或恢复A、B。在主程序的循环语句里，据已知条件作为实元，分别调用这几个子程序，对二人关系反复寻找、判断，直到把初始化数组A、B的元素表示空白的“5”，被“1”和“0”所置换，而且置换后符合条件，才打印出正确结果。

2）框图 见图23

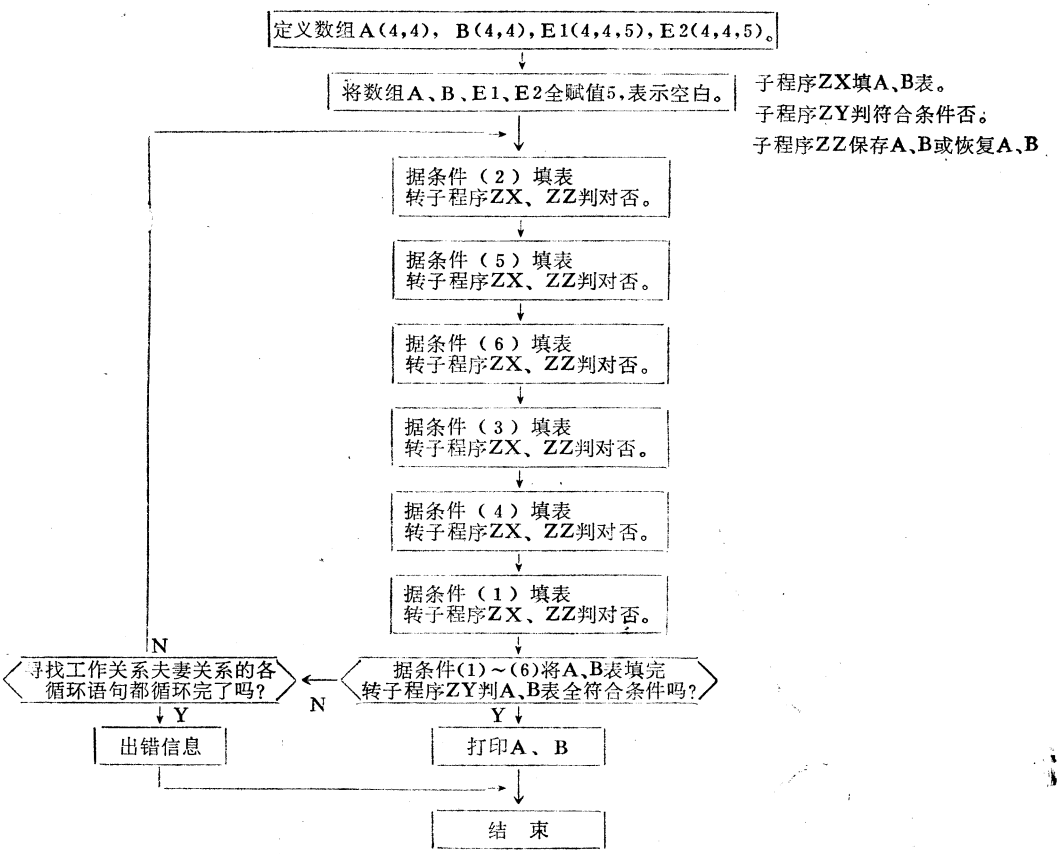


图 23

3）源程序及运行结果

```
      INTEGER A(4, 4) , B(4, 4) , E1(4,4,5),E2(4,4,5)  E1保存A表，E2保存B表。
      DO  1  I=1, 4
      DO  1  J=1, 4
      A (I, J) =5
      B (I, J) =5
      DO  1  K=1, 5
      E1 (I, J, K) = 5
1     E2 (I, J, K) = 5
      CALL  ZX (A, B, 1, 2)
      CALL  ZZ (1, E1, E2, A, B, 1)
      DO  100  I1=1, 3, 2
      CALL  ZX(B, A, I1, 4)
      CALL  ZZ(1, E1, E2, A, B, 2)
      DO  90  I2=1, 2

      CALL  ZX (B, A, 4, I2)
      CALL  ZZ (1, E1, E2, A, B, 3)
      DO  80  I3=1, 3
      CALL  ZX (A, B, 3, I3)
      CALL  ZX (B, A, 2, I3)
      CALL  ZZ (1, E1, E2, A, B, 4)
      DO  70  I4=1, 4, 3
      DO  70  I5=2, 3
      CALL  ZX (A, B, I5, I4)
      CALL  ZX (B, A, 1, 14)
      CALL  ZX (B, A, I5, 3)
      CALL  ZZ (1, E1, E2, A, B, 5)
      GOTO  (50, 40), I2
40    DO  45  I7=1, 3
      CALL  ZX (B, A, I7, 1)
      CALL  ZX (A, B, 4, 1)
      CALL  ZY (A, B, IT)
      IF (IT. EQ. 0) GOTO  110
      CALL  ZZ(2,E1,E2,A,B,5)
45    CONTINUE
      GOTO  65
50    DO  60  I6=1, 3
      CALL  ZX (A, B, I6, 1)
      CALL  ZY (A, B, IT)
      IF(IT.EQ.0)  GOTO  110
      CALL  ZZ(2,E1,E2,A,B,5)
60    CONTINUE
```

初始化：将数组元素全赋5，相当习惯清零，表示空白。

按条件（2）填表，甲、兰同单位。

按条件（5）填表：菊为甲（I1=1）或丙（I1=3）的妻子。

按条件（6）填表：I2为1时丁为梅的丈夫，为2时丁是兰的丈夫。

按条件（3）填表：I3为乙妻

菊（I4=4）或梅（I4=1）为甲妻
竹的丈夫不能为甲（条件4）或丁（条件6），所以I5为竹的丈夫。
由DO70至此，按条件（4）填表。

判条件（6）

I2=2时丁是兰的丈夫，I7为梅的丈夫。丁与梅同单位在此找。

I2=1时丁为梅丈夫
I6为兰的丈夫

按条件（1）填表

```
65      CALL  ZZ (2, E1, E2, A, B, 4)
70      CONTINUE
        CALL  ZZ (2, E1, E2, A, B, 3)
80      CONTINUE
        CALL  ZZ (2, E1, E2, A, B, 2)
90      CONTINUE
        CALL  ZZ (2, E1, E2, A, B, 1)
100     CONTINUE
        STOP "ERR"
110     WRITE (10, 500)((A(I, J), J=1,4), (B(I, J), J=1, 4),I=1,4)

500     FORMAT (10X, 4I3, 10X, 4I3)
        STOP
        END
        SUBROUTINE  ZZ (N, E1, E2, A, B, M)
        INTEGER  E1(4,4,5) , E2(4,4,5) , A(4, 4) , B(4, 4)
        GOTO (10, 20) , N
10      DO  15  I=1, 4
        DO  15  J=1, 4
        E1(I, J, M) =A(I, J)
15      E2(I, J, M) =B(I, J)
        GOTO  30
20      DO  25  I=1, 4
        DO  25  J=1, 4
        A(I, J) =E1(I, J, M)
25      B(I, J) =E2(I, J, M)
30      RETURN
        END
        SUBROUTINE  ZY (IA, IB, IT)

        DIMENSION  IA(4, 4) , IB(4, 4)
        IT=0
        DO  1  K0=1, 2
        DO  1  K1=1, 4
        I S=0
        J S=0

        DO  2  K2=1, 4
        GOTO (10, 20) , K0
10      K3=K1
        K4=K2
        GOTO  30
20      K3=K2
```

恢复A、B。

未找到 找到，打印。

子程序ZZ，当N=1时，保存A、B，N=2时恢复A、B。

子程序ZY判断满足条件否，当IT=0时满足条件，IT=1时不满足条件。

K0 = 1，按行计算；2，按列计算

工作单元

求一行或一列的元素和

```
      K4 = K1
30    JS = JS + IB (K3, K4)
      IS = IS + IA (K3,K4)
      IF (IA(K3, K4) . EQ. 1. AND. IB(K3, K4) . EQ. 0) GOTO 2   判同一对夫
      IF (IA(K3, K4) . EQ. 0. AND. IB(K3, K4) . EQ. 1) GOTO 2   妻是否不同
      IF (IA(K3, K4) . EQ. 0. AND. IB(K3, K4) . EQ. 0) GOTO 2   单位
      GOTO  3
2     CONTINUE
      IF (IS. EQ. JS. AND. IS. EQ. 1) GOTO  1   判是否一行或一列只有一对夫妻或
                                                两人同单位
      GOTO  3
1     CONTINUE
      IF (IA(1, 2) . NE. 1) GOTO  3                          条件（2）
      IF (IB(1, 4) . NE. 1. AND. IB(3, 4) . NE. 1) GOTO  3   条件（5）
      IF (IB(4, 1) . NE. 1. AND. IB(4, 2) . NE. 1) GOTO  3   条件（6）
      JT = 0
      DO  7  M1 = 1, 4
      IF (IB(M1, 2) , EQ. 0) GOTO  7
      JT = M1                                                找兰的丈夫JT
      GOTO  8
7     CONTINUE
      GOTO  3
8     IF (IA(JT, 1) . NE. 1) GOTO  3                         条件（1）
      JT = 0
      DO  9  M1 = 1, 4
      IF (IB(2, M1). EQ. 0) GOTO 9
      JT = M1                                                找乙的妻子JT
      GOTO  11
9     CONTINUE
      GOTO  3
11    IF (IA(3, JT) . NE. 1) GOTO  3                         条件（3）
      JT = 0
      JT1 = 0
      DO  12  M1 = 1, 4                                      找甲妻 J T
      IF (IB(1, M1) . EQ. 1) JT = M1                         找竹的丈夫 J T 1
      IF (IB(M1, 3) . EQ. 1) JT1 = M1
12    CONTINUE
      IF (IA(JT1, JT) . NE. 1) GOTO  3                       条件（4）
      GOTO  4                                                满足全部条件
3     IT = 1                                                 不满足条件
4     RETURN
      END
      SUBROUTINE  ZX (IA, IB, KI, KJ)                        子程序ZX填A、B表：若相同位置，
                                                             在A表为1，B表必然为0，反之，亦然。
```

```
      DIMENSION  IA(4, 4) , JB(4, 4)
      DO   5   K1=1, 4
      IF (K1. EQ. KI) GOTO   4
      IA (K1, KJ) =0
      GOTO   5
4     IA(K1, KJ) =1
      IB(K1, KJ) =0
5     CONTINUE
      DO  10  K1=1, 4
      IF (K1. EQ. KJ) GOTO  9
      IA (KI, K1) =0
      GOTO  10
9     IA(KI, K1) =1
      IB(KI, K1) =0
10    CONTINUE
      RETURN
      END
```

程序运行打印结果（中文是注 解），左表(A表)是工作关系，右 表（B表）是夫妻关系，行是甲乙丙丁，列是梅 兰竹菊。

梅	兰	竹	菊	梅	兰	竹	菊	
0	1	0	0	0	0	0	1	甲
0	0	0	1	0	0	1	0	乙
0	0	1	0	1	0	0	0	丙
1	0	0	0	0	1	0	0	丁

（三）思　考　题

谁养斑马——有五个不同国籍的人，居住着五幢不同颜色的房子，他们各有不同的心爱动物（如斑马、狗等），喝不同的饮料（如水、茶等）和抽不同的香烟，现在已知：

（1）英国人住在红房子里；

（2）西班牙人有条狗；

（3）绿房子的主人喝咖啡；

（4）乌克兰人喝茶；

（5）绿房子在白色房子的右边（从读者的方向看）；

（6）抽“马宝路”牌香烟的人养蜗牛；

（7）黄房子的主人抽“可乐”牌香烟；

（8）当中那幢房子的主人喝牛奶；

（9）挪威人住左边第一幢房子；

（10）抽“本生牌”香烟的人和养狐狸的人是隔壁邻居；

（11）抽“可乐”牌香烟的人和养马者是隔壁邻居；

（12）抽“肯特”牌香烟的人喝桔子水；

（13）日本人抽“摩尔”牌香烟；

(14) 挪威人和蓝房子的主人是隔壁邻居。

请你编写程序，让计算机找一找，谁是喝水的人？谁养斑马？（编写程序时，不要局限于“辨别四对夫妻”的思路，要探索算法）。

提示：解此题时，最好列一表格，运用逻辑推理方法，将已知条件逐个填入表中（见表23-2）：

1) 从条件(9)、(14)、(8)可知：挪威人住左边第一幢，蓝房子在左二，左三（居中）房主喝牛奶。

2) 从条件（5）推断，白、绿两色房子相邻，其位置有两种可能：（a）白左三、则绿左四；（b）白左四、则绿左五。按第（a）种可能推演，中途发觉与已知条件有矛盾，说明此路不通。再按第（b）种可能推断。从条件(1)和(7)可知：英国人住的红房子在左三，剩下的左一应是黄房子，并且主人是抽“可乐”牌烟。

3) 从条件(3)、(11)可知左五（绿）房主喝咖啡，左二（蓝）房主养马。

4) 条件（4）说“乌克兰人喝茶”。已知左一和左三房主为挪威人和英国人，左五房主喝咖啡，可以推知乌克兰人只能是左二或左四房主之一。条件(12)：“抽肯特牌烟者喝桔子水。”排除了左一、左三、左五等已知条件，也只能是左二或左四。再看条件(13)，抽摩尔牌烟的日本人就只能住绿房子了。

5) 条件(2)说“西班牙人养狗”。已知左一、左三、左五房主的国籍，左二房主养马。西班牙人只能是左四。剩下养马的蓝房主人即乌克兰人，且是喝茶者。

6) 上面提到“抽肯特牌烟者喝桔子水”的人，非左二即左四，现在已知左二为乌克兰人，那他就只能是左四房主（西班牙人）了。

7) 条件(6)“抽马宝路牌烟的人养蜗牛”。排除其他条件，只能是英国人了。

8) 据条件(10)，可知抽本生牌的是乌克兰人。挪威人养狐狸。

推理到此，可以得出结论：挪威人喝水，养斑马者是日本人。

表 23-2

住房颜色	黄	蓝	红	白	绿
国　籍	挪　威	乌克兰	英　国	西班牙	日　本
动　物	狐　狸	马	蜗　牛	狗	斑　马
饮　料	水	茶	牛　奶	桔子水	咖　啡
香　烟	可　乐	本　生	马宝路	肯　特	摩　尔

注：《谁养斑马》，在国外曾风靡一时，被称为“世界难题”之一。在六十年代从美国传至世界各地，引起各方人士的兴趣。为了讨论这道难题，有不少大学生把题目贴在宿舍房门上，有废寝忘食的气氛；有些素不相识的人，因为讨论本题而一见如故；还有不少人利用电话展开讨论，使电话局应接不暇。

三　哥德巴赫和欧拉书信往来

24　哥德巴赫和欧拉书信往来

两位著名数学家哥德巴赫和欧拉是多年的老朋友。1742年哥德巴赫写信给欧拉，提出了一个命题。他在信中写道："我的问题如此，随便取某一个奇数，比如77，它可以写成三个素数之和：77 = 53 + 17 + 7，再任取一个奇数461，那么461 = 449 + 7 + 5，也是三个素数之和。461还可以写成257 + 199 + 5，仍为三个素数之和，这样，我发现：

任何大于 5 的奇数都是三个素数之和。

但这怎样证明呢？虽然任何一次试验都可得到上述结果，但不可能把所有奇数都拿出来检验，需要的是一般的证明，而不是个别的检验。"

欧拉回信说，这个命题看来是正确的，但他也给不出严格的证明。同时欧拉又提出了另一个命题：任何一个大于 2 的偶数都是两个素数之和。

但这个命题他也没有能够给予证明。

请你验证这两位数学家在书信中提出的问题。

（一）算 法 分 析

很容易证明哥德巴赫的猜想是欧拉命题的推论。事实上，任何一个大于 5 的奇数都可以写成如下形式：$2N+1=3+2(N-1)$，其中$2(N-1)\geqslant 4$。若欧拉命题是正确的，则偶数$2(N-1)$可以写成二个素数之和。故对于大于 5 的奇数，哥德巴赫命题是正确的。

但若哥德巴赫的命题是正确的，并不能保证欧拉命题的正确，故欧拉的命题是基本的。现在通常把这两个命题统称哥德巴赫猜想。

在过去二百多年中，尽管许多数学家为解决这个猜想付出了艰辛的劳动，但迄今它们仍是一个没有被证明也没有被推翻的定理。

他们的猜想是合理的，例如哥德巴赫所提出的任何大于 5 的奇数总可分为三个素数之和：

$$11=3+3+5;$$

$$55=3+5+47=3+11+41=\cdots;$$

$$99=3+7+89=5+7+87=\cdots;$$

例如欧拉提出任何一个大于 2 的偶数总可以分解为两个素数之和：

$$6=3+3;$$

$$52=5+47=11+41=\cdots;$$

$$100=3+97=41+59=\cdots。$$

有人曾对33×10^6以下的每一个不小于 6 的偶数一一进行验算，都表明它是正确的。然而，验算还不能代替证明，一部分数还不能代替一切数。问题在于大于 6 的一切偶数是不是都是成立的，好象看来十分明显的问题，证明起来又是多么困难啊！

近八十年来，哥德巴赫猜想吸引着世界上许多著名数学家，并在证明上取得了很大的进展。中国数学家在此取得了十分重大的成就。早在30年代数学家华罗庚便开始研究这一问题，得到了很好的成果。他证明了对于"几乎所有的"偶数猜想都是正确的。解放后不

久，他就倡议并指导他的一些学生研究这一问题，取得了许多成果，获得国内外高度评价。

1966年我国数学家陈景润证明了任何一个充分大的偶数都可以表示成两个数之和，其中一个是素数，另一个或是素数，或为两个素数的乘积。这是迄今世界上关于哥德巴赫猜想研究的最好成果。在世界数学界引起了强烈反响，被誉为“陈氏定理”。

我们利用埃拉托色尼筛选算法在计算机上作有限次的验证。

如100以内的自然数是素数者，必然不能被不大于$\sqrt{100}=10$的素数所整除。我们只要把10以内的素数的倍数一个一个地筛掉，就可以求出100以内的所有素数。同理，若求N以内的所有自然数中的素数，可用$\sqrt{N}$以内的素数倍数逐个筛掉，余下便是所求的素数。求素数的方法很多，这仅是其中常用的一种方法。

（二）程序设计

1）设计思路

为了减少机时，我们用100以内的奇偶数做有限次验证，将11～99间所有奇数，看是否能由三个素数之和表示。所有素数唯有2是偶素数（我们不用素数2），所有奇数，用三个奇素数之和来表示（三个奇素数之和当然仍是一个奇数）。一个奇数可能用多种形式的三个素数之和表示，我们仅用其中一种——用两个较小的奇素数和一个较大的奇素数来表示。为了使程序具有通用性，用终端输入奇数N。定义数组MA(3)、MB(3)、MC(3)、MS(3)，分别放三个素数和一个奇数。每三组打印在一行上。

当验证完上述哥德巴赫命题后，用赋值语句N⇐N+1，则N便为偶数，再用6～100间所有偶数来验证欧拉命题：偶数可以用两个素数之和来表示。

2）框图　见图24

3）源程序及运行结果

```
      DIMENSION MA(3), MB(3), MC(3), MS(3)
      INTEGER A, B, C, D, S
      READ (5, 5) N          终端输入N为99奇数
5     FORMAT (I4)
      JS=0
      DO  10  S=11, N, 2                    S为奇数（11～99），找它的三个素数。
      DO  15  A=3, S, 2
      J1=SQRT (FLOAT(A))
      DO  20  K1=2, J1                      先找S中最小一个素数A
      IF (A. NE. A/K1*K1) GO  TO  20
      GO  TO  15
20    CONTINUE                              出循环，肯定A是一个素数。S减A余D。
      D=S-A
      DO  25  B=3, D, 2
      J2=SQRT (FLOAT (B))
      DO  30  K2=2, J2                      再找D中一个素数B
      IF (B. NE. B/K2*K2) GO  TO  30
      GO  TO  25
30    CONTINUE
```

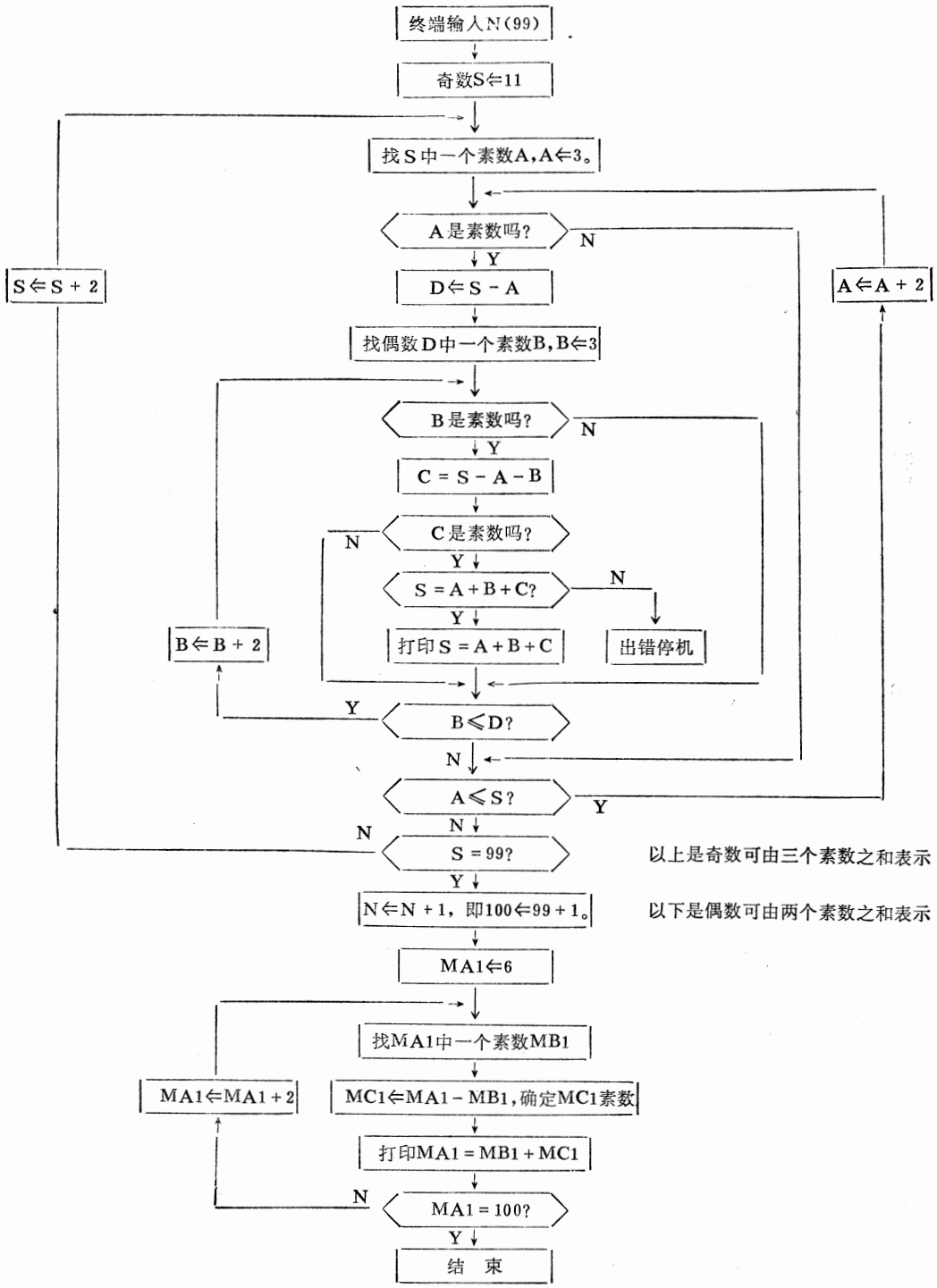

终端输入N(99)
奇数S⇐11
找S中一个素数A,A⇐3。
A是素数吗?
N
Y
S⇐S + 2
D⇐S - A
A⇐A + 2
找偶数D中一个素数B,B⇐3
B是素数吗?
N
Y
C = S - A - B
C是素数吗?
N
Y
S = A + B + C?
N
Y
B⇐B + 2
打印S = A + B + C
出错停机
Y
B≤D?
N
A≤S?
Y
N
N
S = 99?
以上是奇数可由三个素数之和表示
Y
N⇐N + 1，即100⇐99 + 1。
以下是偶数可由两个素数之和表示
MA1⇐6
找MA1中一个素数MB1
MA1⇐MA1 + 2
MC1⇐MA1 - MB1,确定MC1素数
打印MA1 = MB1 + MC1
N
MA1 = 100?
Y
结 束

图 24

```
      C=S-A-B                                    出循环，肯定B是素数。S减去A、B余C。
      J3=SQRT (FLOAT(C))
      DO  35  I3=2, J3
      IF(C. NE. C/I3*I3) GO  TO  35              判C是素数否
      GO  TO  25
35    CONTINUE
      IF (A+B+C. NE. S) STOP  'ERR1'
      JS=JS+1
      MS(JS)=S
      MA(JS)=A
      MB(JS)=B                                   将S、A、B、C分别放入相应数组，以备打印。
      MC(JS)=C
      IF (JS. LT. 3) GO  TO  10
      WRITE (5, 40) (MS(L), MA(L), MB(L), MC(L), L=1, 3)
40    FORMAT (2X, 3(I3, 1H=, I2, 1H+, I2, 1H+, I2, 5X)/)
      JS=0
      GO  TO  10
25    CONTINUE
15    CONTINUE
10    CONTINUE
      WRITE (5, 8)                               以上是哥德巴赫命题：奇数用三个素数表示。
8     FORMAT (3X, 3(10H-------, 6X))             以下是欧拉命题：偶数可用两个素数表示。
      N=N+1
      JS=0
      DO  110  MA1=6, N, 2
      DO  120  MB1=3, MA1, 2
      M2=SQRT (FLOAT (MB1))
      DO  130  M=2, M2
      IF (MB1. NE. MB1/M*M) GO  TO  130
      GO  TO  120
130   CONTINUE
      MC1=MA1-MB1
      KW=SQRT (FLOAT (MC1))
      DO  140  KY=2, KW
      IF (MC1. NE. MC1/KY*KY)  GO  TO  140
      GO  TO  120
140   CONTINUE
      JS=JS+1
      MA(JS)=MA1
      MB(JS)=MB1
      MC(JS)=MC1
      IF(JS. LT. 3) GO  TO  110
      WRITE (5, 150)  (MA(L), MB(L), MC(L), L=1, 3)
```

```
150     FORMAT (2X, 3 (I3, 1H=, I2, 1H+, I2, 5X)/)
        JS=0
        GO TO 110
120     CONTINUE
110     CONTINUE
        STOP
        END
```

程序运行时打印结果如下：

99

11=3+3+5	13=3+3+7	15=3+5+7
17=3+3+11	19=3+3+13	21=3+5+13
23=3+3+17	25=3+3+19	27=3+5+19
29=3+3+23	31=3+5+23	33=3+7+23
35=3+3+29	37=3+3+31	39=3+5+31
41=3+7+31	43=3+3+37	45=3+5+37
47=3+3+41	49=3+3+43	51=3+5+43
53=3+3+47	55=3+5+47	57=3+7+47
59=3+3+53	61=3+5+53	63=3+7+53
65=3+3+59	67=3+3+61	69=3+5+61
71=3+7+61	73=3+3+67	75=3+5+67
77=3+3+71	79=3+3+73	81=3+5+73
83=3+7+73	85=3+3+79	87=3+5+79
89=3+3+83	91=3+5+83	93=3+7+83
95=3+3+89	97=3+5+89	99=3+7+89

以上是哥德巴赫给欧拉信中提出的命题

以下是欧拉回信提出的命题

6=3+3	8=3+5	10=3+7
12=5+7	14=3+11	16=3+13
18=5+13	20=3+17	22=3+19
24=5+19	26=3+23	28=5+23
30=7+23	32=3+29	34=3+31
36=5+31	38=7+31	40=3+37
42=5+37	44=3+41	46=3+43
48=5+43	50=3+47	52=5+47
54=7+47	56=3+53	58=5+53
60=7+53	62=3+59	64=3+61
66=5+61	68=7+61	70=3+67
72=5+67	74=3+71	76=3+73
78=5+73	80=7+73	82=3+79

84 = 5 + 79　　86 = 3 + 83　　88 = 5 + 83

90 = 7 + 83　　92 = 3 + 89　　94 = 5 + 89

96 = 7 + 89　　98 = 19 + 79　　100 = 3 + 97

（三）思　考　题

1）请编写程序验证每一个形如p = 4k + 1的素数都可以表示成两个自然数的平方和（k为自然数）。

2）每个素数可以表示成四个平方数之和。

25　互质二数加减得1

任意两个互质的自然数，经过若干次该二数加减后，总可获得结果为 1 的数值。

（一）笔算步骤和结果

所谓两个互质（即互素）的数，是指该二数除 1 外再没有其它公因数。如14，9为互质数，又如187，79也是两个互质的数。试算该二数经若干次加减后便可得到1。

如14和 9 互质二数。14 + 14 = 28　9 + 9 + 9 = 27　则28 − 27 = 1（目的达到）。

（二）程序设计

1）设计思路

可由键盘输入两个互质的数M和N。判M和N是否互质，不互质，立即打印出错信息WRONG，重新输入。若是互质，将二数反复加减后，若得数为 1，则论证结果正确，打印出。再重新输入另外互质二数。若输入 1 时，则停。程序中举了四个例子，详见运行结果。

2）框图　见图25

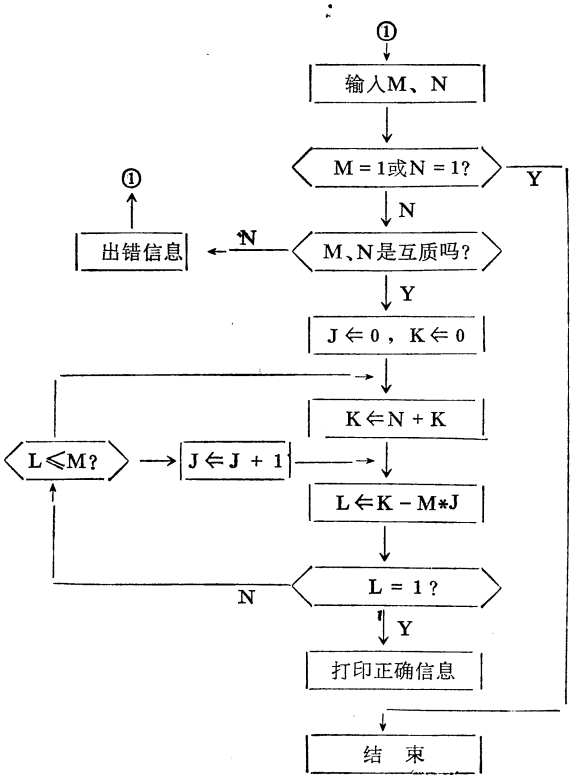


图 25

3）源程序及运行结果

```
 3        READ (5, 5) M, N
 5        FORMAT (2I3)
          IF (M. EQ. 1. OR. N. EQ. 1) STOP
          IF (M. GT. N) GO TO 10
          IF (N/M. EQ. FLOAT(N) /FLOAT(M))GO TO 30
          GO TO 15
10        IF (M/N. EQ. FLOAT(M)/FLOAT(N)) GO TO 30
15        DO 20 I=2, N/2
          IF (M/I. EQ. FLOAT (M)/FLOAT (I). AND. N/I. EQ. FLOAT(N)/
           1  FLOAT(I))GO TO 30
20        CONTINUE
          J=0
          GO TO 40
30        WRITE (6, 35)
35        FORMAT (10X, 5HWRONG)
          GO TO 3
40        K=0
45        J=J+1
          GO TO 60
50        K=N+K
60        L=K-M*J
          IF (L. EQ. 1) GO TO 70
          IF (L. LT. 1) GO TO 50
          IF (L. LE. M)GO TO 50
          GO TO 45
70        WRITE (6, 80) M, N
80        FORMAT (5X, 5HRIGHT, 3X, I3, 3HAND, I3, 5H--->1)
          GO TO 3
          END
```

打印结果如下

```
RIGHT      9 AND  7--->1     9 和 7 互质，变化为1。
WRONG     75AND  5           75和 5 不互质。
RIGHT    187AND 79--->1      187和79互质，变化为1。
RIGHT    199AND133--->1      199和133互质，变化为1。
```

（三）思 考 题

假设a－1和a＋1是大于10的素数(这对素数称为孪生素数)，证明a^3-4a可被120整除。

26 找相邻二数之和为素数

将1，2，3，…18，19，20这20个连续的自然数排成一圈，使每两个相邻数之和都为素数。

（一）笔算步骤和结果

1～20这20个连续自然数排成一圈，每两个相邻数相加必然都大于2，所以，每两个数相加若构成素数，必是奇素数，是由一个奇数加上一个偶数。因此，这20个数，必是奇偶相间排成1圈。

试算后，满足题意，可以排成以下形式：

1，18，19，12，17，20，11，8，15，16，13，10，9，14，5，6，7，4，3，2。

（二）程序设计

1）设计思路

定义数组IA（10）置放1～19十个奇数，数组IB(10）置放2～20十个偶数，数组IC（20）置放排列的结果。先从IA里取出一个奇数放入M单元，从IB里取出一个偶数放入N单元，转子程序，判M＋N是否为素数，不是素数，返主程序重取另一数，二数之和若是素数，置标记L⇐1，返主程序后，主程序判L为1得知暂成，再取一个IA，与前一个IA构成素数的IB相加，其和若仍为素数，依此法再取，否则，将IA与IB互换位置，重新另行选取，直到指针K为20时，说明20个数已安排就绪。打印输出数组IC(20)。

2）框图　见图26

3）源程序及运行结果

```
      DIMENSION  IA(10), IB(10), IC(20)
      DO  1  I=1, 10
      IA(I)=2*I—1
1     IB(I)=2*I
      K=1
      IC(K)=IA(1)
      IA(1)=0
2     CALL  KK (IC, IB, IA, K, N1)
      CALL  KK (IC, IA, IB, K, N1)
      GO  TO  2
      END
      SUBROUTINE  KK (IC, IB, IA, K, N1)
      DIMENSION  IA(10), IB(10), IC(20)
100   DO  150  I=1, 10
      IF (IB(I). EQ. 0) GO  TO  150
      N=IB(I)
      M=IC(K)
      CALL  SS (N, M, L)
      IF (L. NE. 1) GO  TO  150
      N1=I
      DO  10  II=1, 10
```

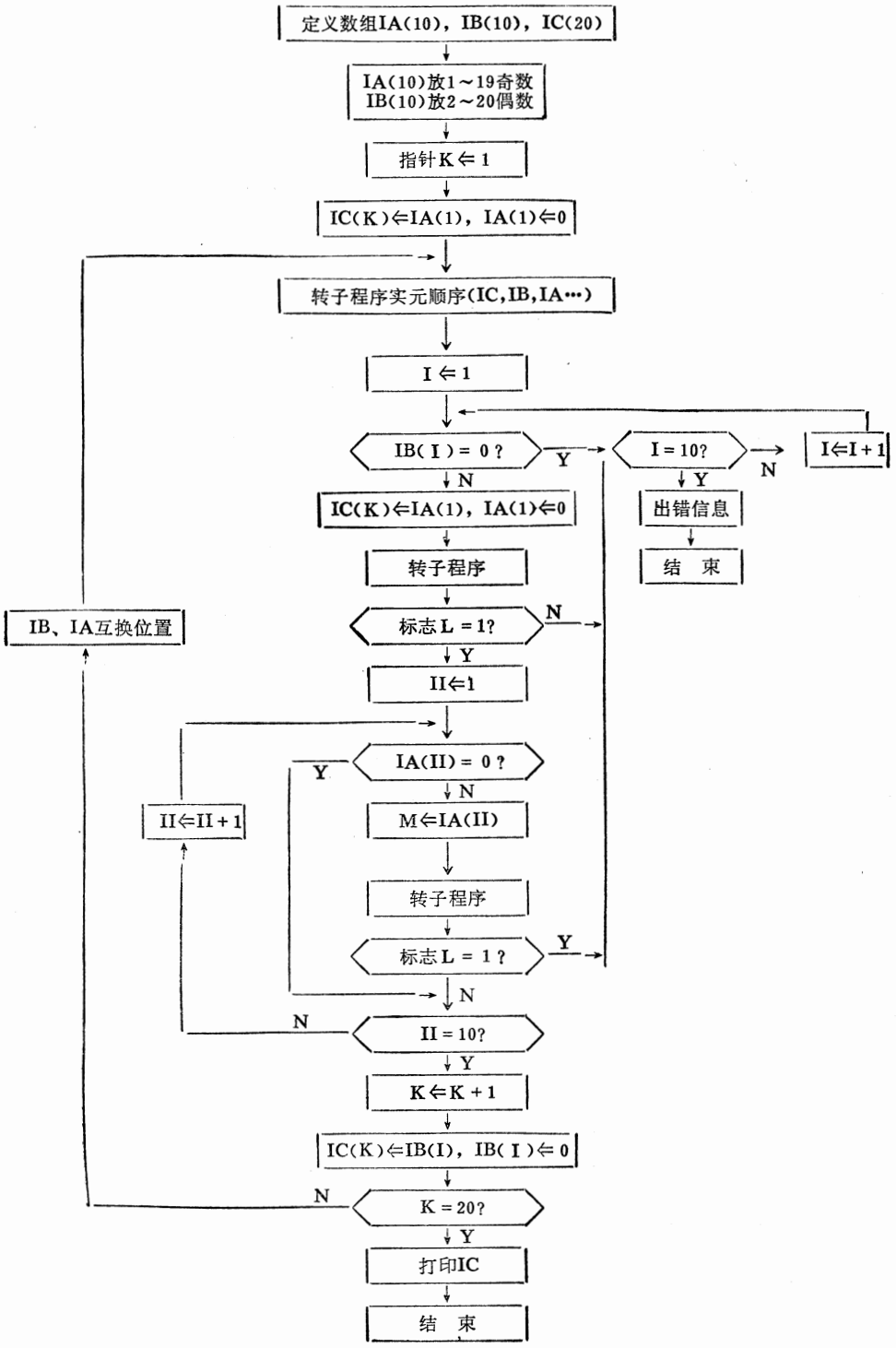

定义数组IA(10)，IB(10)，IC(20)
IA(10)放1～19奇数
IB(10)放2～20偶数
指针K⇐1
IC(K)⇐IA(1)，IA(1)⇐0
转子程序实元顺序(IC,IB,IA…)
I⇐1
IB(I)=0？
Y
I=10?
N
I⇐I+1
Y
出错信息
结 束
N
IC(K)⇐IA(1)，IA(1)⇐0
转子程序
标志L=1?
N
IB、IA互换位置
Y
II⇐1
IA(II)=0？
Y
N
II⇐II+1
M⇐IA(II)
转子程序
标志L=1？
Y
N
N
II=10?
Y
K⇐K+1
IC(K)⇐IB(I)，IB(I)⇐0
N
K=20?
Y
打印IC
结 束

图 26

```
      IF (IA(I I). EQ. 0) GO  TO  10
      M=IA (I I)
      CALL  SS (N, M, L)
      IF (L. EQ. 1) GO  TO  150
10    CONTINUE
      K=K+1
      IC(K)=IB( I )
      IB( I )=0
      IF (K. EQ. 20) GO  TO  300
      RETURN
150   CONTINUE
      IF (N1. EQ. 0) GO  TO  310
      K=K+1
      IC( K )=IB(N1)
      IB(N1)=0
      N1=0
      RETURN
300   WRITE ( 6 , 303) IC
303   FORMAT (1X, 20 (I3))
      STOP
310   STOP'ERR'
      END
      SUBROUTINE  SS (N, M, L)
      DIMENSION  LL(11)
      DATA  LL/3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37/
      L=0
      NM=N+M
      DO  8  IK=1, 11
 8    IF (LL (IK). EQ. NM) GO  TO  88
      RETURN
88    L=1
      RETURN
      END
```

程序运行后打印结果

```
1  18  19  12  17  20  11   8  15  16  13  10   9  14   5   6   7   4
3   2
```

（三）思 考 题

1）如果自然数是1～30是否也有这结论？自然数1～40呢？若不可能，那么你能把自然数由20扩大到几，也具有每相邻二数之和都是素数？

2）试编写程序验证：任何二相邻自然数必互质。

四 魔术师玩扑克牌

27 魔术师玩扑克牌

魔术师利用一副扑克牌中的十三张黑桃，预先将它排好后，迭在一起，拿在一只手里，牌面朝下。对观众说：我不看牌，我数数就能猜到每张是什么牌，我大声数数，你们听，不信？你们就看。把上面那张牌数为1，把它翻过来正好是黑桃A，给观众看后，把这黑桃A放在桌子上，然后按顺序从上到下数手中的余牌，第二次数1、2，把第一张放在这迭牌的下面，第二张牌翻过来，给观众看，正好是黑桃2，也把它放在桌子上。第三次数1、2、3，把前面两张依次放在这迭牌的下面，再翻过第三张牌给观众看，正好是黑桃3，把它也放在桌子上，以后照这样进行下去，将牌每次一张张地从上面传到底部，直到把十三张全翻完，从A到K依次抽出，准确无误，观众心想魔术师手中牌的原始次序是怎样安排的？

（一）笔算步骤和结果

为了得到直观认识，算法省略。请你拿13张黑桃牌，翻来复去多试几遍，便可知道魔术师先把牌排成下面序列：

A，8，2，5，10，3，Q，J，9，4，7，6，K。

（二）程序设计

1）设计思路

把13张牌依次排成圆圈，相当于沿圆周数数。我们可以用循环计数法，来模仿这个过程，设数组A(13)表示这一序列。开始先把A(1)置1，其它元素置0，然后数0的个数，数到第N（2≤N≤13）个0时，便把N置在这个位置，数到13的位置后再返回来，从1开始继续数，直到把13个数全填满为止。如果每翻出一张牌给观众看，叫做遍数，把13张牌全给观众看，需要数13遍，用循环控制变量I表示遍数。L是位置计数器，N是0的个数计数器。排完的序列置入数组A(13)，将数组A(13)再转换成具有A、J、Q、K名字的牌，置入数组B(13)。程序运行后，将A、B数组全打印出，对照看。

2）框图　见图27

3）源程序及运行结果

```
C       PROGRAM PUKE
        INTEGER  A(13), B(13)
        DO  15  J=1, 13
15      A(J)=0
        A(1)=1
        L=0
        DO  50  I=2, 13
        N=0
20      L=L+1
        IF (L. EQ. 14) L=1
        IF (A(L). NE. 0) GO  TO  20
```

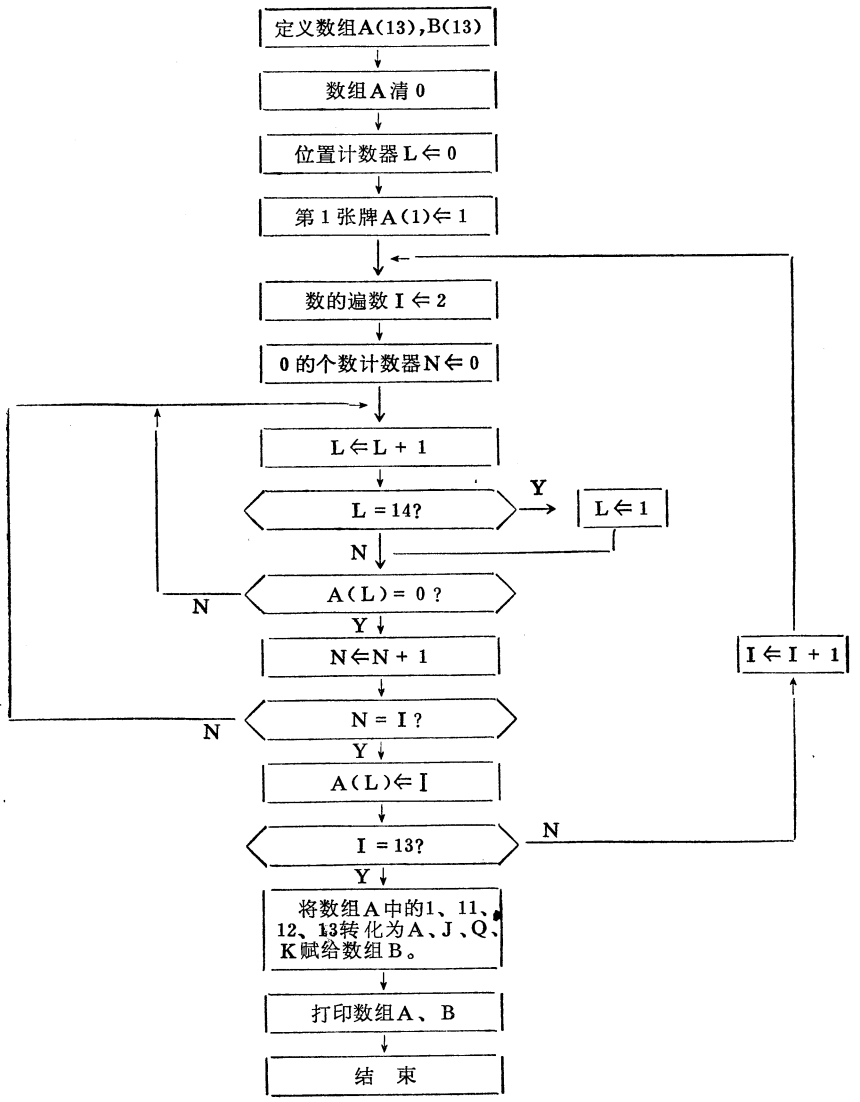


图 27

```
      N=N+1
      IF (N. NE. I) GO  TO  20
      A(L)=I
50    CONTINUE
      WRITE (10, 61) A
61    FORMAT (1X, 13I3)
      DO  55  J=1, 13
      M=A(J)
      GO  TO (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13) M
1     B(J)=''A''
```

```
          GO  TO  55
     2    B(J) = ''2''
          GO  TO  55
     3    B(J) = ''3''
          GO  TO  55
     4    B(J) = ''4''
          GO  TO  55
     5    B(J) = ''5''
          GO  TO  55
     6    B(J) = ''6''
          GO  TO  55
     7    B(J) = ''7''
          GO  TO  55
     8    B(J) = ''8''
          GO  TO  55
     9    B(J) = ''9''
          GO  TO  55
    10    B(J) = ''10''
          GO  TO  55
    11    B(J) = ''J''
          GO  TO  55
    12    B(J) = ''Q''
          GO  TO  55
    13    B(J) = ''K''
    55    CONTINUE
          WRITE (10, 60) B
    60    FORMAT (1X, 13 (1X, A2))
          STOP
          END
```

程序运行后打印结果

1	8	2	5	10	3	12	11	9	4	7	6	13
A	8	2	5	10	3	Q	J	9	4	7	6	K

（三）思 考 题

一副完整的54张扑克牌，一字排开，按一定规则有规律地重复多次去取，当取完53张牌时，还是拿不到大王。请设计取牌的规则，按你设计的规则取牌，大王应藏在第几号位置上，最后仅剩下它没被取到。

28 魔术师又玩扑克牌

魔术师在第27题里表演13张扑克牌，应观众要求，他又出来表演，这次拿半副扑克牌（黑桃全部和红桃全部），迭在一起，牌面朝下，拿在手里，对观众说，最上面一张是黑桃A，翻开给观众看，放在桌上。以后，从上至下每数两张全依次放在最底下，第三张给观

众看，便是黑桃 2，放在桌上。第三次又数两张依次放在最底下，第三张给观众看，便是黑桃 3，放在桌上。如此继续数下去，观众可以看到取出的牌放在桌上的顺序是：

黑桃A，2，3，4，5，6，7，8，9，10，J，Q，K，

红桃A，2，3，4，5，6，7，8，9，10，J，Q，K。

你知道这是怎么安排的吗？

（一）笔算步骤和结果

你可能说，那没什么奥妙的，牌是预先排好的。的确，牌是预先排好的，但要排成这样的顺序也十分不容易。为了不影响你独立思考的能力，我们省略算法分析，你可以拿这样的26张牌试一试，能否自己排成功，记下所花费的时间，如果你还排不出来，请参看下面的序列：

A，6，10，2，Q，3，3，J，9，4，7，Q，5，4，K，6，K，J，7，5，A，8，8，10，9，2。其中下面划线的表示黑桃，下面没有划线的表示红桃。

（二）程 序 设 计

1）设计思路

我们可以完全模仿魔术师变幻过程，但程序比较复杂，这里我们把这些牌看做一个数组A(26)，有26个元素，1～13代表黑桃A～K，14～26代表红桃A～K。数组初始状态为0，我们每次找出三个邻近的零元素，并把次数N填到第三个元素的位置，直到全填满为止。注意开始时，先把1填入第一个数组元素中，相当数第一遍。第二遍数两个零元素后，将2冲入第K个数组元素（K是指针），以后各遍依此类推用循环控制变量N由2～26做为遍数。计数器M记零的个数。将26张牌围成一圈，依次数到27时便是1位，这由指针K控制。排完次序后，打印数组A(26)。

2）框图　见图28

3）源程序及运行结果

```
      PROGRAM  PLPK
      INTEGER  A
      DIMENSION  A(26)
      DO  10  I=1, 26
10    A(I)=0
      A(1)=1
      K=1
      N=0
      DO  50  N=2, 26
      M=0
20    K=K+1
      IF (K .EQ. 27) K=1
      IF (A(K). NE. 0) GO  TO  20
      M=M+1
      IF (M. NE. 3) GO  TO  20
      A (K)=N
50    CONTINUE
```

```
      WRITE (6, 60) A
60    FORMAT (1X, 12I4)
      STOP
      END
```

程序运行后打印结果。1～13是黑桃A～K，14～26是红桃A～K。

1	19	10	2	25	16	3	11	22	4	20	12
5	17	26	6	13	24	7	18	14	8	21	23
9	15										

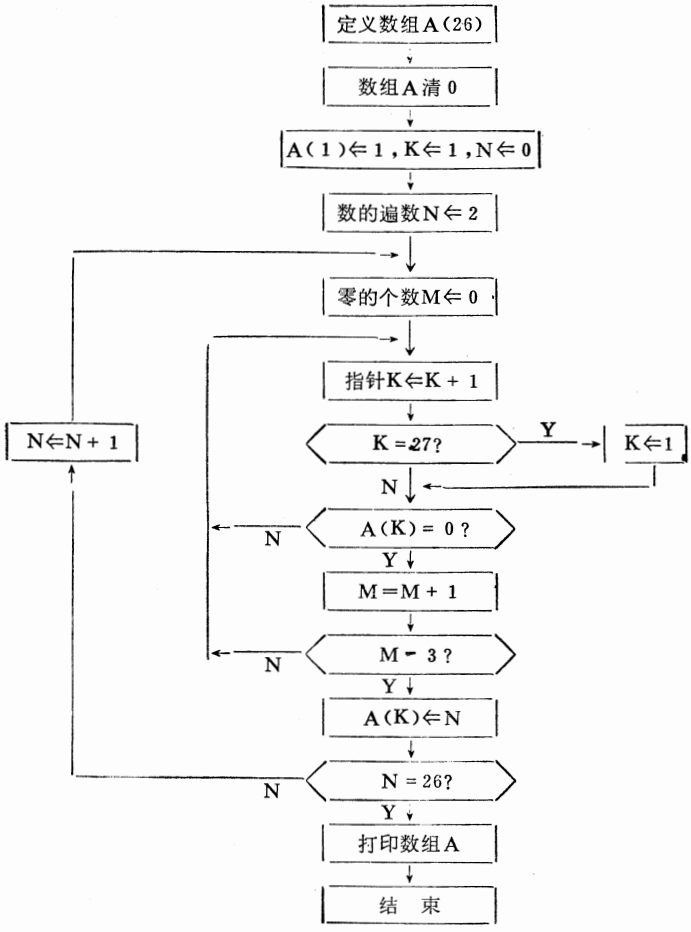


图 28

（三）思 考 题

请编写程序将一副完整的54张扑克牌，将其中丢失的张数找出来，并指出丢失的是什么牌。

其程序设计思路、框图以及源程序等，详见《趣味程序设计100例》第23题。

29 钟形牌游戏

用一付取出大小王的52张扑克牌可以做时钟单人纸牌游戏。牌分为四张一堆，按时钟的样子排列，第13堆放在中心。移动是从一堆上取出顶上的一张牌并把它放在它应归入那一堆的下面，叫归队。（Jack = 11，Queen = 12，King在中心），而这一堆为下一次移动提供牌。当四张 King 已放在中心的一堆上，游戏结束。如果所有其它的牌已放在正确的位置上（都已归队），就认为游戏成功。若中心一堆已归队，而其余的某堆有未全部归队时，认为游戏是半成功。写一个程序模拟时钟单人纸牌游戏。

（一）笔算步骤和结果

可以按题目所给条件和规则，将52张扑克牌洗好后，牌面朝下，按钟表盘上数字1～12和中心一堆摆成13堆，每堆放一张牌，重复四次便将52张牌分完。然后从任一堆（如从第一堆）最上面取一张翻开看是几便放到第几堆的最下面（牌面朝上），再从该堆的最上面翻开一张，如此继续下去。直到第13堆全部归队时，该局便结束，查看是全胜还是半胜。也可能是失败，当某堆已取尽，而第13堆未全归队时，便定义为失败。可用手摊牌，反复试验几局观看游戏的成功率。不过，这样速度慢，还是让我们编写程序，让计算机来做这种游戏吧！

（二）程序设计

1）设计思路

按题意将一副完整的扑克牌，拿去大王和小王，剩52张牌，由红桃、黑桃、梅花、方块各1～13张构成。编程序时可不计牌具体颜色和形状，统一由1～13数字重复四次组成。

定义数组K(13，4)，随机将52张牌分放在13堆上，每堆四张，即置入数组K(13，4)。从堆上每取走一张，暂存后便将该张牌的位置（数组元素）冲成零，并将这张牌归队。再从刚归队的堆里取出一张（数组元素），重复上述规则。直到取出一张（数组元素）为零时止，说明该堆已取走四次（已取尽），无法再进行了，便判1～12堆是否全归队，若12堆全归队，当然第13堆也必然归队，打印全胜（SUCCESS）；第1～12堆其中有某堆未归队，而第13堆已归队，打印半胜（RIGHT）；若第13堆未全归队，打印失败(FAIL)。

52张牌随机布置堆放在13堆上，有多种方法，我们采用半固定半随机相结合的方法。先用循环语句和赋值语句布置某些堆，并将某些堆留有空位，上机时由键盘敲数字1～5，这五个数字不得重复，但敲入次序不同，也即随机布置不同，可获得不同的结果，在程序中我们作了三局游戏，获得三种结局，敲入“54231”，打印出成功（SUCCESS）；敲入“32541”，打印半成功（RIGHT）；敲入“45132”打印失败（FAIL）。

当随机布置后，这样进行，先从某堆取一张牌，如先取K(1，4)，暂存L和L1单元，将K(1，4)冲零，判L = 0吗？是0，则该堆已取尽，判胜、半胜或失败而结束。若L不是0，则寻找应归的堆，再从该堆取一张放入L中……，直至L是0为止。

2）框图　见图29

3）源程序及运行结果

```
DIMENSION  K(13, 4)
L=1
```

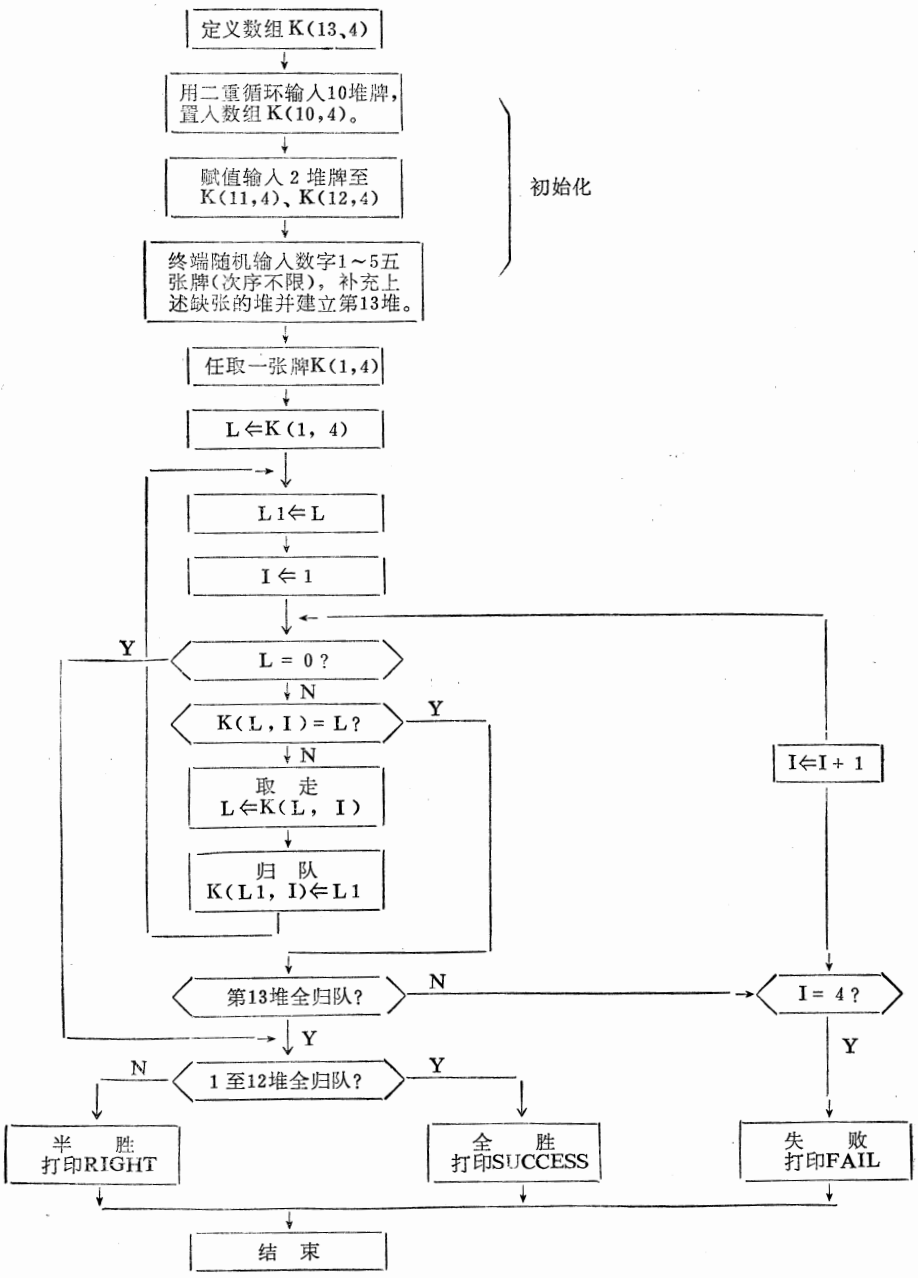


图 29

```
      DO  1  I=1, 10
      DO  1  J=1, 4
1     K (I, J) = L
      IF (L. EQ. 13) L=0
      L=L+1
      K(10, 4) = 13
      K(11, 1) = 11
```

```
      K(11, 2) = 12
      K(11, 3) = 8
      K(11, 4) = 9
      K(12, 1) = 7
      K(12, 2) = 10
      K(12, 3) = 6
10    WRITE(5, 20)
      WRITE(2, 20)
20    FORMAT (/10X, 22HINPUT DATA (5):1<=N<=5, 2X)
      READ (5, 50) K(12, 4), K (13, 1), K(13, 2), K(13, 3), K(13, 4)
      WRITE (2, 50) K(12, 4), K(13, 1), K(13, 2), K(13, 3), K(13, 4)
50    FORMAT (4X, 5I2)
      L=K(1, 4)
      K(1, 4) = 0
80    L1 = L
      DO 90 I=1, 4
      IF (L. EQ. O) GOTO 110
      IF (K(L, I). EQ. L) GOTO 100
      L=K(L, I)
      K(L1, I) = L1
      GOTO 80
100   IF ((L1. EQ. 13). AND. (I. EQ. 4)) GOTO 110
90    CONTINUE
110   DO 120 I=1, 12
      DO 120 II=1, 3
      L=II+1
      DO 120 J=L, 4
      IF (K(I, II). NE. K(I, J)) GOTO 130
120   CONTINUE
      WRITE (2, 150)
      STOP SUCCESS
130   L = 0
      DO 140 I=1, 4
140   L=L+K(13, I)
      IF(L. EQ. 52) GOTO 180
      WRITE (2, 160)
      STOP FAIL!
180   WRITE (2, 170)
170   FORMAT (5X, 11HYOU RIGHT! )
160   FORMAT (5X, 10HYOU FAIL! )
150   FORMAT (5X, 9HYOU WIN! )
      END
```

```
      INPUT  DATA (5):1<=N<=5
      INPUT  DATA (5):1<=N<=5          54231
YOU   WIN! STOP  SUCCESS

      INPUT  DATA (5):1<=N<=5
      INPUT  DATA (5):1<=N<=5          32541
YOU   RIGHT

      INPUT  DATA (5):1<=N<=5
      INPUT  DATA (5):1<=N<=5          45132
YOU   FAIL
```

（三）思　考　题

请你设计一种随机洗牌的方法，进行程序设计。

五 数的趣谈

30 回文数猜想

一个有名的尚未得到解答的数字问题之一，叫做“回文数猜想”。取一个任意的十进制数，将其倒过来，并将这两个数相加。然后把这个和数倒过来，与原来的和数相加……。重复这个过程直到获得一个回文式数为止。（所谓回文式数又称回文数，它是一串十进制数，其左右数字对称：即相同。若位数为奇数个，则除中间数字外都对称，显然需三位数字才有意义。若位数是偶数个，左右对称时，四位数字才有意义。）

例如，只要三步就可由68获得回文式数：

```
    6 8
+   8 6
-------
  1 5 4
+ 4 5 1
-------
  6 0 5
+ 5 0 6
-------
1 1 1 1
```

“回数猜想”就是：不论开始时采用什么数，在经过有限步骤之后都会得到一个回文式数。

（一）笔算步骤和结果

还没有人能确定这个猜想是对的还是错的。已经知道这个猜想对于二进制记数法，或以2的幂为基础所表示的数是不成立的。对于用其它记数法表示的数则尚未证明。

可能成为这个猜想的反例的最小十进制数是196。我们利用计算机已对这个数进行了几十万步计算，没有获得回文式数。但是，你能证明它永远也不会产生回文式数？

数学家已经对同时也是素数（除1和该数本身以外没有其它因数的数）的回文式数进行了研究。数学家们相信有无穷个回文式素数，但尚无法证明这一点。数学家还猜想存在无穷个回文式素数对。例如30103和30203，这些数中除了中间的数字以外其它数字都是对称的，这些数是依次连续的。

回文式素数必须有奇数个数字。因为每个有偶数个数字的回文式数都是11的倍数，所以不是素数，（如题目中算出的回文式数字1111）。你能证明这样的数总可被11除尽吗？（提示：若一个数中所有偶数位数字之和与其奇数位数字之和的差是11的倍数的话，这个数就能被11除尽。）

在回文式数中平方数是枚不胜举的，例如$11\times11=121$。平方数中回文数的比例，比随意选取的整数中回文式数的比例大得多。立方数也是这样。而且，回文式立方数几乎肯定有一个三次方根也是回文式数（例如，$11\times11\times11=1331$）。用计算机对回文式四次方数进行的研究，至今还没能找到一个四次方根不是回文式数的四次方数。也没有人发现存在回文式五次方数。数学家们猜想不存在X^k（k大于4）形式的回文式数。

（二）程 序 设 计

1）设计思路

用几位数字进行猜想，要据计算机字节长度而定，我们不妨先从不多的数字位数着手，如从五位数字着手，定义数组k（5），置放五位数字的每一位。

从键盘输入不超过五位正整数的N。然后分离出N的每一位数字，由低位至高位依次置入数组k（5）。再将数组k（5）高低位颠倒，将原数与颠倒的原数相加，存入ISUM单元，每加一次，计数器JS加1。若变换次数达到1000次时，令其暂停（PAUSE），考虑是否还令其继续变化。

回文式数字至少是三位数才有意义，所以判ISUM是否是三位数，少于三位数，标记IBZ=0，令其继续变化，标记IBZ=1是变化后成为三位以上的数，再判该数是否是回文式数字。

每输入一个原数N时，将输出N和回文数字ISUM，以及变化次数TIME。

可多次输入，进行猜想，输入－1时便结束。

2）框图　见图30

3）源程序及运行结果

```
      INTEGER  K(5)
10    READ (5, 20)  N
20    FORMAT (I5)
      IF (N. EQ. -1) STOP
      JS= 0
      IBZ= 0
      ISUM=N
30    M=ISUM
      DO  40  I=1, 6
      K(I) =MOD(M, 10)
      M= (M-K (I) ) /10
      IF (M. EQ. 0)  GOTO  50
      IF (I. EQ. 5)  GOTO  50
40    CONTINUE
50    IF (IBZ. EQ. 1)  GOTO  70
      DO  60  J=1, I
60    M=M+K(J) *10** (I-J)
      ISUM=ISUM+M
      JS=JS+1
      IF (JS. EQ. 1000)  GOTO  100
      IF (ISUM. LT. 100)  GOTO  30
      IBZ=1
      GOTO  30
70    IBZ=0
      L=I/2
      DO  80  J=1, L
```

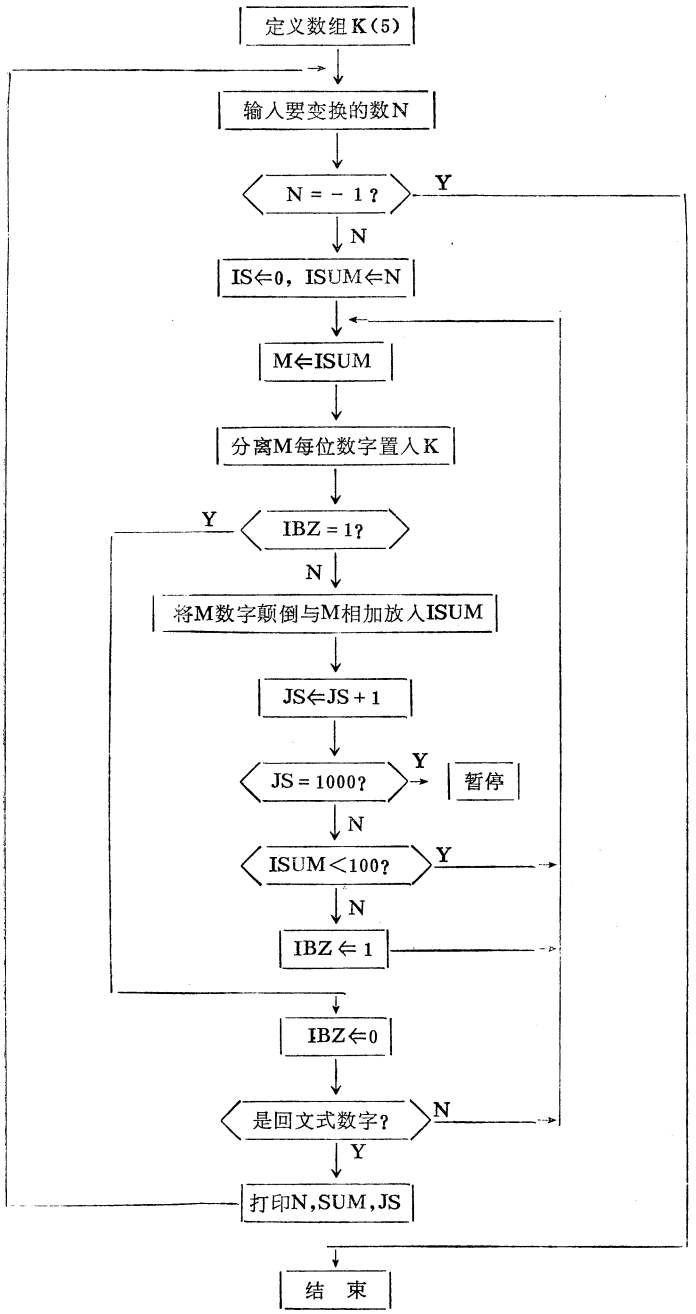


图 30

```
      JJ = I - J + 1
      IF (K(J) . NE. K(JJ) )  GOTO  30
80    CONTINUE
      WRITE ( 5 , 90) N, ISUM, JS
90    FORMAT (15X, 2HN = , I5, 5X, 4HSUM = , I5, 5X, 5HTIME = , I5)
```

```
        STOP  OK!
100     PAUSE  J=1000
        IBZ=1
        GOTO  30
        END
```

```
N=12501      SUM=28082      TIME=  6
N=    6      SUM=  363      TIME=  5
N=    9      SUM= 7447      TIME=  9
N= 1423      SUM= 4664      TIME=  1
N=  876      SUM=23232      TIME=  4
N=    5      SUM= 3003      TIME= 15
```

（三）思 考 题

1）主题目已谈到数学家已经对同时也是素数（除1和该数本身以外没有其它因数的数）的回文式数进行了研究。数学家们相信有无穷个回文式素数，但尚无法证明这一点。数学家还猜想存在无穷个回文式素数对。例如30103和30203，这些数中除了中间的数字以外其它数字都是对称的，这些数是依次连续的。请编写程序验证。

2）主题目中说在回文式数中平方数是枚不胜举的，验证平方数中回文数的比例，比随意选取的整数中回文式数的比例大得多。立方数也是这样。而且，回文式立方数几乎肯定有一个三次方根也是回文式数（例如$11\times11\times11=1331$）。请编写程序用计算机对回文式四次方数进行研究，是否能找到一个四次方根不是回文式数的四次方数。

31 数 的 趣 谈

随意写一个十进制的自然数（如2583），然后求这个数目字的平方和（$2^2+5^2+8^2+3^2=102$），对得出来的数（102）再用此法进行处理（$1^2+0^2+2^2=5$），并一直照此进行下去（$5^2=25$，$2^2+5^2=29$，$2^2+9^2=85$，…）。

证明这个过程不把原来数变化成1（因为1的平方还是1，所以以后的数永远是1），就必然变成145，然后便出现42、20、4、16、37、58、89这样一个周而复始的循环。

（一）笔算步骤和结果

我们用

$$L=10^{n-1}a_n+10^{n-2}a_{n-1}+\cdots+10^2a_3+10a_2+a_1$$

表示任意一个n位自然数，用

$$L_1、L_2、L_3\cdots \qquad (1)$$

表示出L产生的数字平方和的数列，其中L_1是L的数字平方和，其余各项都是各自的前

一项的数字平方和

$$L_1 = a_n^2 + a_{n-1}^2 + \cdots + a_3^2 + a_2^2 + a_1^2$$

我们有

$$L - L_1 = (10^{n-1} - a_n)\, a_n + (10^{n-2} - a_{n-1})\, a_{n-1} + \cdots + (10^3 - a_4)\, a_4 + (10^2 - a_3)\, a_3 + (10 - a_2)\, a_2 - (a_1 - 1)\, a_1$$

不难看出

$$(a_1 - 1)\, a_1 \leqslant 72$$

当 $n \geqslant 3$ ($a_n \neq 0$时)

$$(10^{n-1} - a_n)\, a_n \geqslant 99$$

同时 $(10^{i-1} - a_i)\, a_i \geqslant 0$, $i = 2, 3, \cdots, n-1$。

所以 $L > L_1$。

由这个不等式可知，当数列（1）各项还是大于或等于三位数之前，后面的项恒小于前面的项，即数列这一部分是递减数列，所以任一个大于三位数的L，在经过若干步平方和以后必定都变化成一个不多于三位的数，因此本题的论点是否正确，只要研究一下三位数就够了。

因此，我们完全可以认为L就是一个三位的数，即 $n = 3$，这时$a_3 \neq 0$并得：

$$L - L_1 = (100 - a_3)\, a_3 + (10 - a_2)\, a_2 - (a_1 - 1)\, a_1 \geqslant 99 - 72 = 27$$

或 $L_1 \leqslant L - 27$

由这个不定方程可知，数列（1）的某个项是一个两位数，设这个项是：$L_q = 10j + k$。由于把L_q改为$10k + j$数列L_{q+1}、L_{q+2}、L_{q+3}…的各项不变，所以用检查L_q在$j \geqslant k \geqslant 0$，$j \geqslant 1$时的各数字平方和的方法，可以完全证明本题的论点。

当$L_q = 10j + k$，$j \geqslant k \geqslant 0$，$j \geqslant 1$时，$L_{q+1}$必然是下表中某个数字。各列各数是$j^2 + k^2$的值

j \ k	0	1	2	3	4	5	6	7	8	9
1	1	2								
2	4	5	8							
3	9	10	13							
4	16	17	20	25	32					
5	25	26	29	34	41	50				
6	36	37	40	45	52	61	72			
7	49	50	53	58	65	74	85	98		
8	64	65	68	73	80	89	100	113	128	
9	81	82	85	90	97	106	117	130	145	162

我们可以把表中的1，10，100三个数以及题中指出的145，42，20，4，16，37，58，89八个数舍掉，因为这些数都是这个定理规定的数，此外还可以舍掉：

2，40，50，52，61，73，80，81，85，90，98，130这些数，因为这些数和前边那些数或表中的另外一些数的区别只是数码的排列，或只多一个数码0而已。这样，用来检验这个定理是否正确的数就只剩下28个了，它们是：

5，8，9，13，17，18，25，26，29，32，34，36，41，45，49，53，64，65，68，

72，74，82，97，106，113，117，128，162。

我们把检验的结果列表如下，表的头一栏所列的是应该检验的数，第二栏内依次写的是由这个数产生的数列（1）的几个项，填写时，要边写边查，一直写到出现了能肯定这个定理是否正确的数为止。

由于用来检查的各数，最终都能变化成为1或145，42，20，4，16，37，58，89当中的一个数，而这些数又都周期地重复，因此本题所说的定理也就得到了证明。

5	25 29、85	72	53、34、25
8	64、52	74	65
9	81	82	68、100
18	65、61	106	37
32	13、10	113	11、2
36	45、41、17、50	128	69、117、51、26、40
49	97、130	162	41

（二）程 序 设 计

1）设计思路

随意输入十进制的自然数N。N是几位，视计算机字节长度而定，我们不妨设N是五位数字，按题意进行变换。每变化一次，计数器J加1，当变化成数字145时便成功，因为145再进行变化便是42，……，89，而后又是145，周而复始的循环。

变化过程要判是否变为1，是则停。可输入多个自然数进行练习，看经过几次J便达目的。打印出输入的自然数N和变化后的数字145以及次数TIME。输入 N≤1时便停止。

2）框图　见图31

3）源程序及运行结果

```
10        WRITE (5, 20)
20        FORMAT (15X, 11HINPUT DATA :)
          READ (5, 30) N
30        FORMAT (I5)
          J = 0
          IF (N. LE. 1)   GOTO  70
          M = 0
          L = N
40        DO  50  I = 1, 5
          K = MOD (L, 10)
          M = M + K*K
          L = (L - K) /10
          IF (L. EQ. 0)   GOTO  55
50        CONTINUE
55        IF (M. EQ. 1)   GOTO  70
          J = J + 1
          L = M
          M = 0
          IF (L. NE. 145)   GOTO  40
```

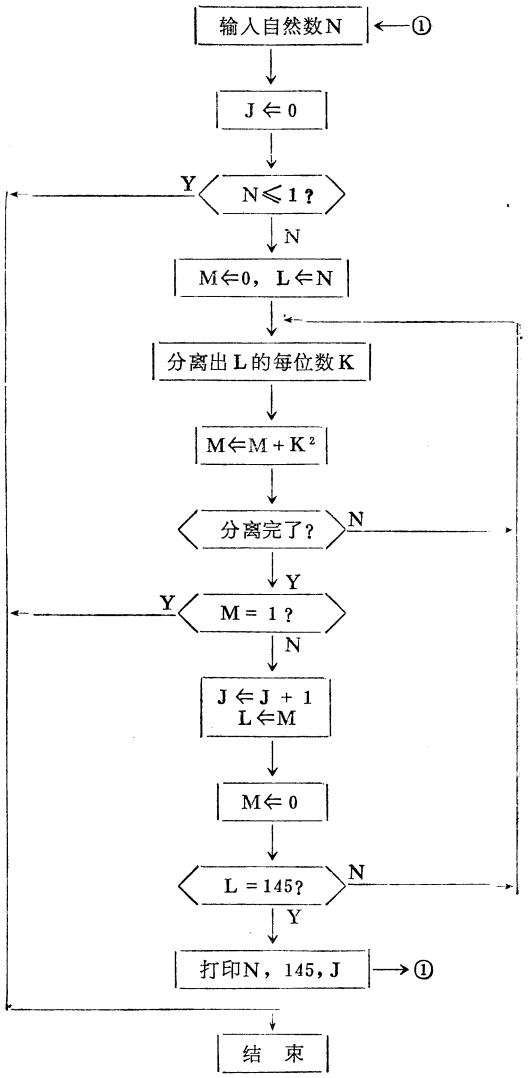


图 31

```
        WRITE (5, 60)   N, J
60      FORMAT (10X, 2HN=, I5, 10H- - - >145, 5HTIME=, I5/)
        GOTO  10
70      STOP  OK !
        END

        INPUT  DATA : 6
   N =  6 - - - >  145  TIME =      10

        INPUT  DATA : 9
   N =  9 - - - >  145  TIME =       7
```

```
      INPUT  DATA：1324
N=   1324 - - - >  145  TIME=   9

      INPUT  DATA：7695
N=   7695 - - - >  145  TIME=   6
```

（三）思 考 题

我知道你抹去的是什么数字：

（1）请你任意写一个四位数。

（2）把其中四个数字的次序全部颠倒过来，得到另一个四位数（如abcd→dcba）。

（3）把大的四位数减去小的四位数，得到一个新的四位数。

（4）在这个新的四位数中，由你任意抹去一个不是0的数字。

（5）现在只要你告诉我剩下的三个数字，我就能知道你抹去的是哪一个数字。如你告诉我剩下的是3，8，4，我即知道你抹去的是3。什么道理？请编写程序验证。

32 立方和恰好等于和的平方

$1^3+2^3=9$　　$(1+2)^2=9$

$1^3+2^3+3^3=36$　　$(1+2+3)^2=36$

……　　……

请分析上述几组式子，能否找出规律，并迅速算出下式：

$1^3+2^3+3^3+4^3+5^3+6^3+7^3+8^3+9^3+10^3=?$

（一）笔算步骤和结果

求几个数的立方和，一般总是先算出各数的立方再相加。但对于从1开始若干个连续自然数的立方和，我们可以从题中的几组式子受到启发，找出规律，迅速算出答案。

$1^3+2^3+3^2+\cdots+9^3+10^3=(1+2+3+\cdots+9+10)^2=(55)^2=3025$

用数学归纳法可以证明：$1^3+2^3+3^3+\cdots+(n-1)^3+n^3=[1+2+3+\cdots+(n-1)+n]^2$

（二）程 序 设 计

1）设计思路

我们以从1开始的自然数为两位数为例，由键盘输入。估计数字较大，设双精度变量A、B、B1置放计算结果。举了九个例子，详见运行后打印结果。读者可据机器字节长度，从1开始的自然数扩大至四位数等。

2）框图　见图32

```
      DOUBLE  PRECISION  A, B, B1
2     WRITE (10, 1000)
1000  FORMAT (3H  N：)
      READ  (11, 5)  N
5     FORMAT (I2)
      IF  (N. LE. 0)  GOTO  50
      IF (N. GE. 100)  GO TO  50
      A=0. 0D1
      B=0. 0D1
```

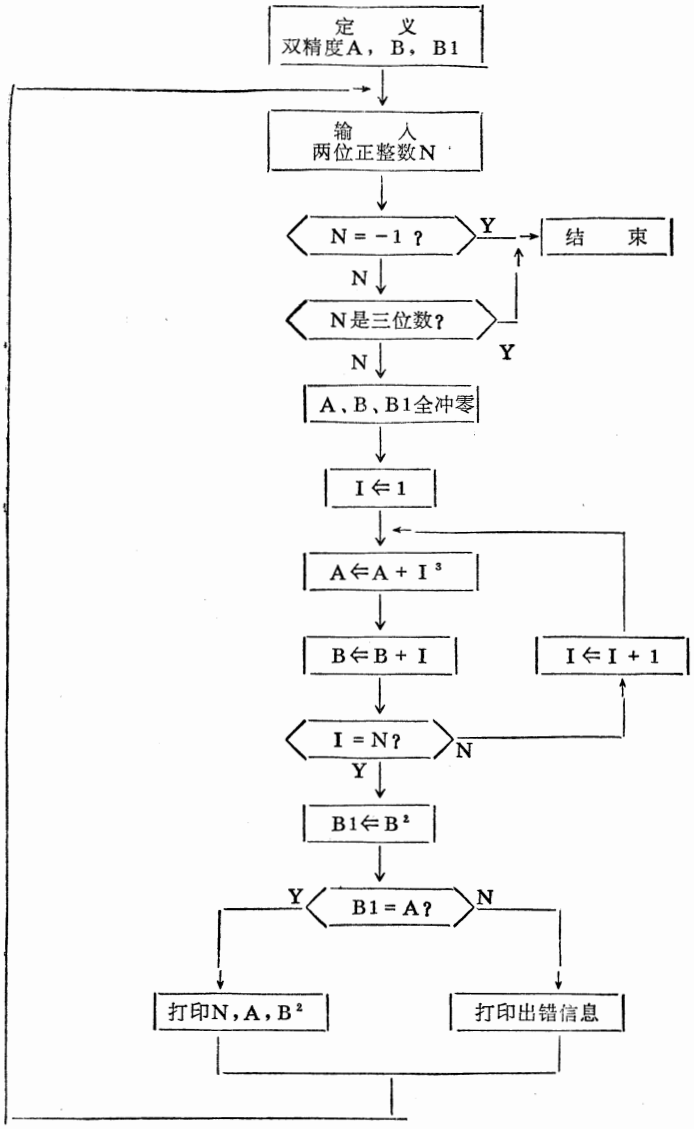


图　32

```
      DO  10  I=1, N
      AI=FLOAT (I)
      A=A+DBLE (AI**3)
10    B=B+DBLE (AI)
      B1=B*B
      B1=DABS (A-B1)
      IF  (B1. GT. 1.0D0)  GOTO  30
      WRITE  (10, 20)  N, A, B, B
20    FORMAT (5X, 4H1---, I2, 1H:, F9.0, 1H=, F5.0, 1H*, F5.0)
      GOTO  2
```

```
30        WRITE   (10, 40)
40        FORMAT   (5X, 5HERROR)
          GOTO  2
50        STOP
          END
```

```
N:
01
    1---1:        1.  =     1. *     1.   1～1 每个数立方和=每个数之和的平方
N:
05
    1---5:      225.  =    15. *    15.   1～5 每个数立方和=每个数之和的平方
N:
09
    1---9:     2025.  =    45. *    45.   1～9 每个数立方和=每个数之和的平方
N:
10
    1--10:     3025.  =    55. *    55.   1～10每个数立方和=每个数之和的平方
N:
50
    1--50:  1625625.  =  1275. *  1275.   1～50每个数立方和=每个数之和的平方
N:
90
    1--90: 16769025.  =  4095. *  4095.   1～90每个数立方和=每个数之和的平方
N:
99
    1--99: 24502500.  =  4950. *  4950.   1～99每个数立方和=每个数之和的平方
N:
35
    1--35:   396900.  =   630. *   630.   1～35每个数立方和=每个数之和的平方
N:
81
    1--81: 11029041.  =  3321. *  3321.   1～81每个数立方和=每个数之和的平方
N:
00
STOP
```

（三）思 考 题

求四个相邻正奇数，使它们的平方和，比夹在它们之间的偶数平方和大112。其算法分析、结论和程序设计等，详见《趣味程序设计100例》第85题。

33 胸 中 有 数

请你任意写一个数字不相同的三位数，如abc，对调a与c的位置，得到另一个三位数cba，把大的一个三位数减去小的一个三位数，设所得的差是efd。

最后，再将efd加上dfe。不用你告诉我，我就知道你所得的和是1089。为什么？

（一）笔算步骤和结果

设abc＞cba（即a＞c），根据题意先求它们的差：

$$abc-cba=100a+10b+c-(100c+10b+a)=99(a-c)$$

将这个差写成十进制的形式：

$$99(a-c)=100(a-c)-(a-c)=100(a-c)-100$$
$$+100-10+10-a+c=100(a-c-1)+90+(10-a+c) \quad (1)$$

所以这个差的百位数字是(a－c－1)，十位数字是9，个位数字是(10－a＋c)。将这三个数字的次序颠倒一下，得到一个新的数：

$100(10-a+c)+90+(a-c-1)$ （2）

把这两个数相加，就得到1089，即（1）+（2）得：

$100(a-c-1)+90+(10-a+c)+100(10-a+c)+90+(a-c-1)=1089$

（二）程 序 设 计

1）设计思路

此程序用来验证。由键盘输入数字不相同的三位数字，判该三位数字是否符合规定，不符合规定时，程序指示重新输入。子程序PP是将三位数分离出个、十、百位数，然后颠倒这三位数，主程序再做减或加，经减、加后，判结果是否为1089，结果正确则打印出演算过程，否则，打印出错信息，可多次输入，当输入－1时则停止。

2）框图 见图33

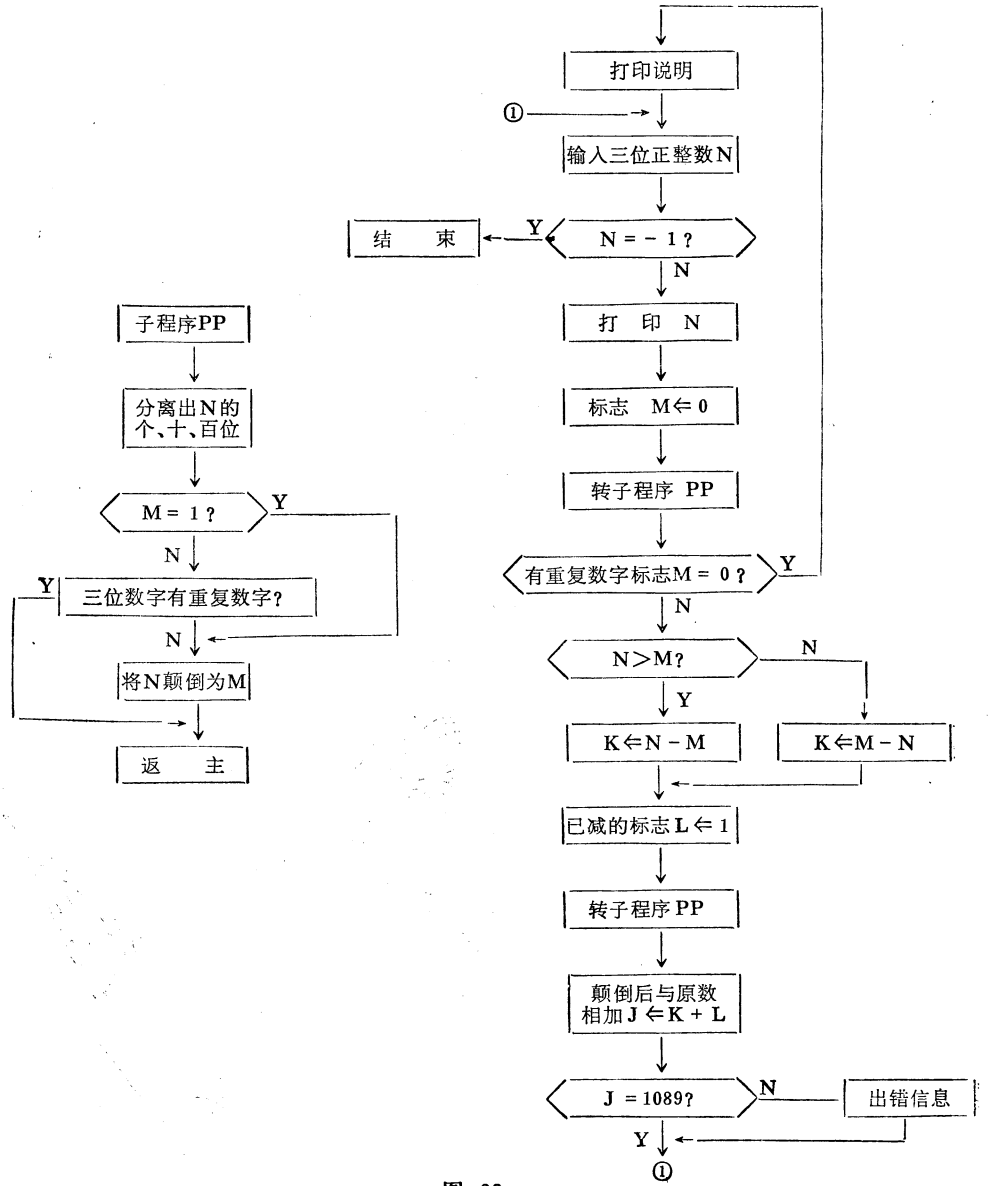


图 33

3) 源程序及运行结果

```
5       WRITE (10)  (/20H  INPUT  N  (STOP : N = -1) )
        READ (11, 10)  N
10      FORMAT (I3)
        IF (N. EQ. -1) STOP
        WRITE (10, 20) N
20      FORMAT (5X, 2HN =, I3)
        M = 0
        CALL  PP (N, M)
        IF (M. EQ. 0) GOTO  5          输入的三位数字有重复数字
        IF (N. GT. M) GOTO  40
        K = M - N
        GOTO  45
40      K = N - M
45      L = 1
        CALL  PP (K, L)
        J = K + L
        IF (J. EQ. 1089) GOTO  60
        WRITE (10, 50)
50      FORMAT (5X, 5HERROR)
        GOTO  5
60      WRITE (10, 70) N, M, K, K, L, J
70      FORMAT  (5X, 8HRIGHT  , I3, 1H-, I3, 1H=, I3, 5X, I3, 1H+, I3,
          1H=, I4)
        GOTO  5
        END
        SUBROUTINE  PP (N, M)
        IA = N - N/10*10                分离出N的个、十、百位
        IB = (N - N/100*100) /10
        IC = N/100
        IF (M. EQ. 1) GOTO  100         判需要再判三位数字重复吗
        IF (IA. EQ. IB. OR. IA. EQ. IC. OR. IB. EQ. IC) GOTO  105
                                        判三位数字有重复数字否将数字N颠倒为M
100     M = 100*IA + 10*IB + IC
105     RETURN
        END

        INPUT  N  (STOP : N = -1)
012
   N = 12
   RIGHT      12 - 210 = 198      198 + 891 = 1089
```

```
        INPUT  N   (STOP : N = - 1)
123
   N = 123
   RIGHT      123 - 321 = 198       198 + 891 = 1089

        INPUT  N   (STOP : N = - 1)
305
   N = 305
   RIGHT       305 - 503 = 198          198 + 891 = 1089

        INPUT  N   (STOP : N = - 1)
456
   N = 456
   RIGHT      456 - 654 = 198       198 + 891 = 1089

        INPUT  N   (STOP : N = - 1)
987
   N = 987
   RIGHT      987 - 789 = 198       198 + 891 = 1089

        INPUT  N   (STOP : N = - 1)
- 1
        STOP
```

（三）思 考 题

有这样一个三位数，其尾数不大于 2 。若将尾数移到首位后（其余二个数字次序不变，依次后推），新三位数将是原三位数的两倍多。请你求出这个三位数。

34 北京市中学生智力竞赛题

将一个整数写成两个整数的平方和

如 $5 = 1^2 + 2^2$。

求25 = ？，85 = ？。

（一）笔算步骤和结果

$25 = 3^2 + 4^2$，$85 = 2^2 + 9^2$。

我们编写程序再计算些一个整数由两个整数的平方和构成。

（二）程 序 设 计

1）设计思路

由题意知只要找到25或85是由哪两个数平方之和构成便可，我们给以扩大。随机输入几个二位数（其中有25和85），看是否能找到符合题意的要求。例如找 5，17，13，25，45，65，85等是由哪两个不同数字的平方相加构成的，每随机输入 1 ～ 2 位数时用二重循环寻找，找到后便打印。找不到时便打印 “NOT FIND。”

2）框图 见图34

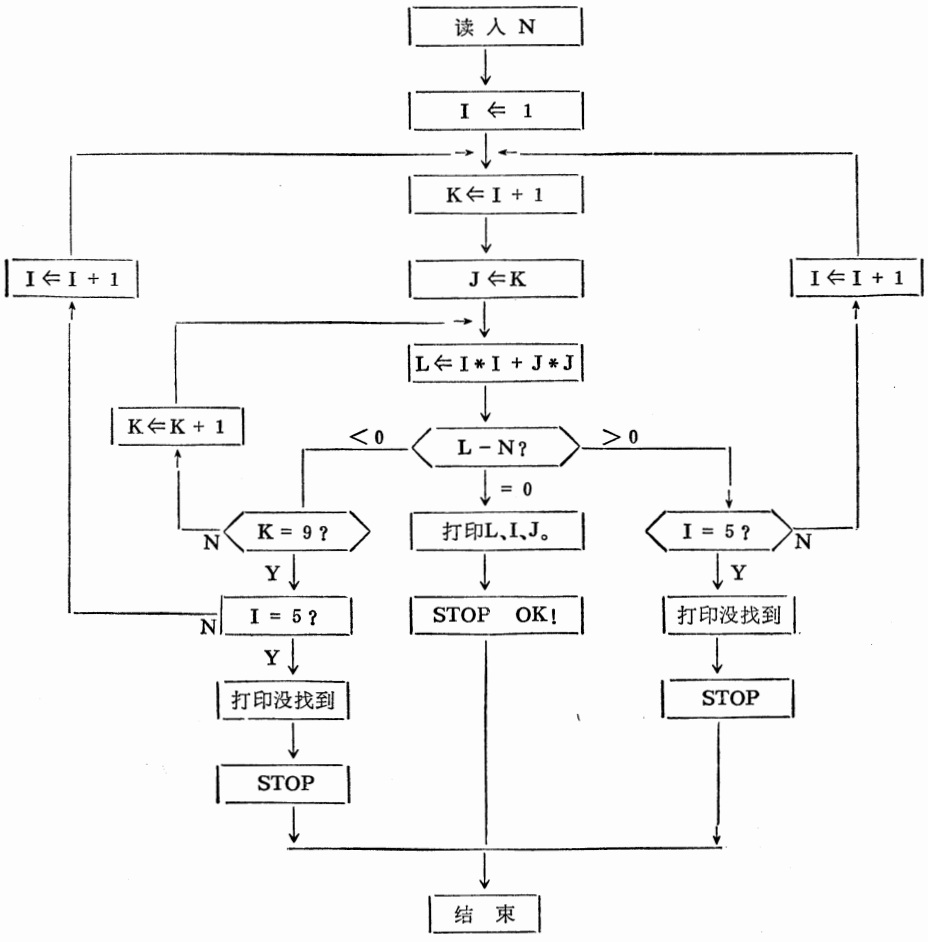


图 34

3) 源程序及运行结果

```
      WRITE (5, 1)
1     FORMAT (5X, 'INPUT A NUMBER N=? ')
      READ (5, 2) N
2     FORMAT (I2)
      DO 10 I=1, 5
      K=I+1
      DO 20 J=K, 9
      L=I*I+J*J
      IF (L-N) 20, 30, 10
30    WRITE (5, 40) L, I, J
40    FORMAT (5X, I2, 1H=, I1, 4H**2+, I1, 3H**2, //)
      STOP 'OK! '
20    CONTINUE
10    CONTINUE
```

```
      WRITE (5, 3)
3     FORMAT (//5X, 'NOT FIND! ')
      STOP
      END
```

```
5 = 1**2 + 2**2
17= 1**2 + 4**2
13= 2**2 + 3**2
25= 3**2 + 4**2
45= 3**2 + 6**2
65= 1**2 + 8**2
85= 2**2 + 9**2
```

（三）思 考 题

编写程序求三位自然数是由哪两个不同数字的平方之和构成。如$916 = 4^2 + 30^2$，……。

35 刘徽《九章算术》中的勾股数

若A、B、C为满足$A^2 + B^2 = C^2$的正整数，则A、B、C为勾股数。我国古代数学书《周髀算经》曾提到“勾广三，股修四，径隅五”这三个边都是正整数的直角三角形。在公元263年时，我国数学家：刘徽写了一本数学书，书名叫作《九章算术》，其中有

$$3^2 + 4^2 = 5^2$$
$$5^2 + 12^2 = 13^2$$
$$7^2 + 24^2 = 25^2$$
$$8^2 + 15^2 = 17^2$$
$$20^2 + 21^2 = 29^2$$

由此看来我国古代数学家已经研究出很多组勾股数。

（一）笔算从略，见程序设计思路

请你编写程序，求出100之内的所有组勾股数，并打印出全部结果。

（二）程 序 设 计

1）设计思路

由不定方程：

$$A^2 + B^2 = C^2 \qquad (1)$$

有定理：不定方程（1）的适合条件

$$A > 0,\ B > 0,\ C > 0,\ (A,\ B) = 1,\ 2 \mid A$$

的一切正整数解，可以用下列公式表示出来：

$$A = 2XY,\ B = X^2 - Y^2,\ C = X^2 + Y^2$$

这里的X和Y都是正整数，而且$X > Y$，$(X、Y) = 1$，$2 \nmid (X + Y)$

如果按照此定理编写出源程序当然是可以的。但对不了解此定理的读者这样编写就比较困难。所以这里使用一般的方法。首先设法得到从3到100之间的数的两组合。利用二重循环可以达到这一目的。令外循环变量为A，A从1到99。令内循环的循环变量为B，B从A + 1到100。然后在循环体内判断A和B是否满足等式（1）。

将满足等式的A和B及C打印出来。为了缩短机器运算时间，我们可以利用勾股数的奇偶特性。即在A和B中一个是奇数，另一个必定是偶数。那么可以让B从A + 1开始，每次增加步长为2。因为A若是奇数，A + 1就是偶数。以后步长是2，B总为偶数。如果A是偶数，A + 1就是奇数。以后步长是2，B总为奇数。我们用整型变量I、J、L分别代表A、B、C。

2）框图　见图35

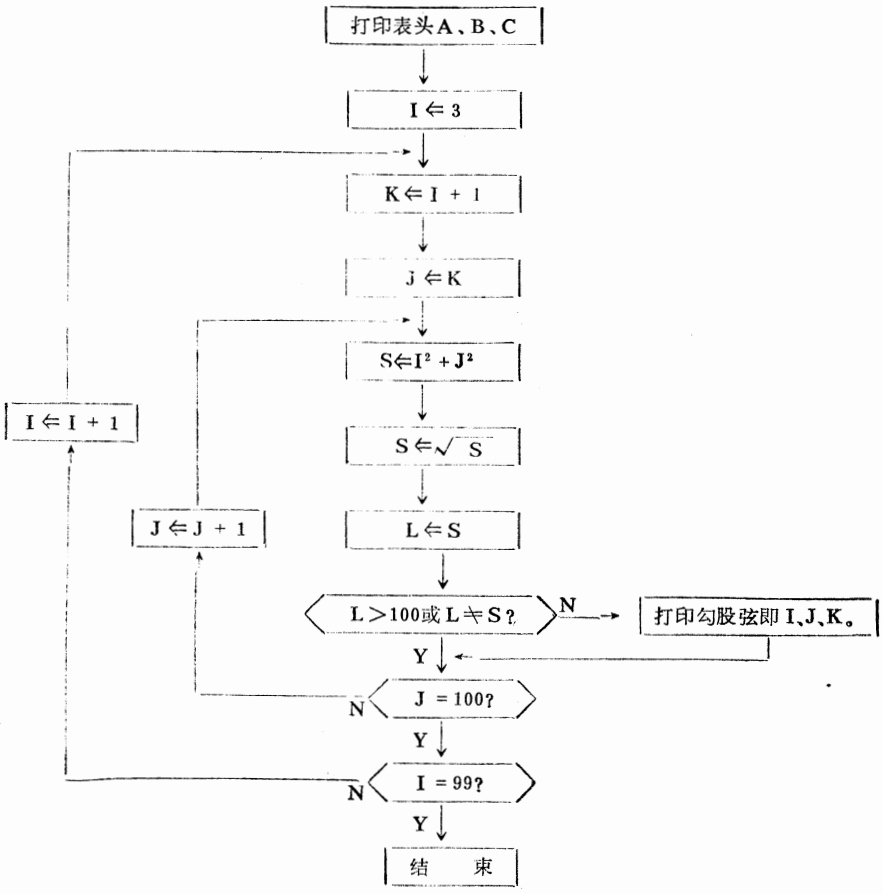


图 35

3）源程序及运行结果

```
WRITE (5,30)
DO 10 I=3, 99
K=I+1
DO 10 J=K, 100, 2
S = I * I + J * J
S = SQRT(S)
L = S
IF (L.GT.100.OR.ABS (S - L) .GT.0.1E-06) GOTO 10
WRITE(5,20) I, J, L
```

```
10      CONTINUE
20      FORMAT(21X,3I5)
30      FORMAT(25X,'A    B    C'/20X,'……')
        END
```

A	B	C
3	4	5
5	12	13
7	24	25
8	15	17
9	12	15
9	40	41
11	60	61
12	35	37
13	84	85
15	20	25
15	36	39
16	63	65
20	21	29
21	28	35
21	72	75
24	45	51
25	60	65
27	36	45
28	45	53
33	44	55
33	56	65
35	84	91
36	77	85
39	52	65
39	80	89
40	75	85
45	60	75
48	55	73
51	68	85
57	76	95
60	63	87
65	72	97

（三）思 考 题

求完全平方数的平方根——为了求出一个数的平方根，只需把这个数减去从1开始的奇数列，并记住所能作的减法次数，这个次数就是这个数的平方根。例如求$\sqrt{25}$的值，从25依

次减去从1开始的奇数1、3、5、7、9，减去5次使25变为零，所以：

$$\sqrt{25}=5$$

36 水 仙 花 数

水仙花数是指一个三位数，其各个数之立方和等于该数。例如153，即为一水仙花数。因为$153=1^3+5^3+3^3$

（一）笔算步骤和结果

尚有$370=3^3+7^3+0^3$等，让我们编写程序再继续搜寻，观看是否还有更多的水仙花数。

（二）程序设计

1）设计思路

从三位数字中找水仙花数，用循环语句来寻找，循环控制变量初值K为100，终值为999，步长为1，每个循环变量K分离出个、十、百位三个数字，将这三个数字各个立方相加，判相加和是否等于K，若相等，则找到一个水仙花数。若不相等，再换一个三位数K按上述方法寻找。每找到一个水仙花数，打印该数，并在其后打印其百、十、个位数字。详见源程序后的打印结果。

2）框图 见图36

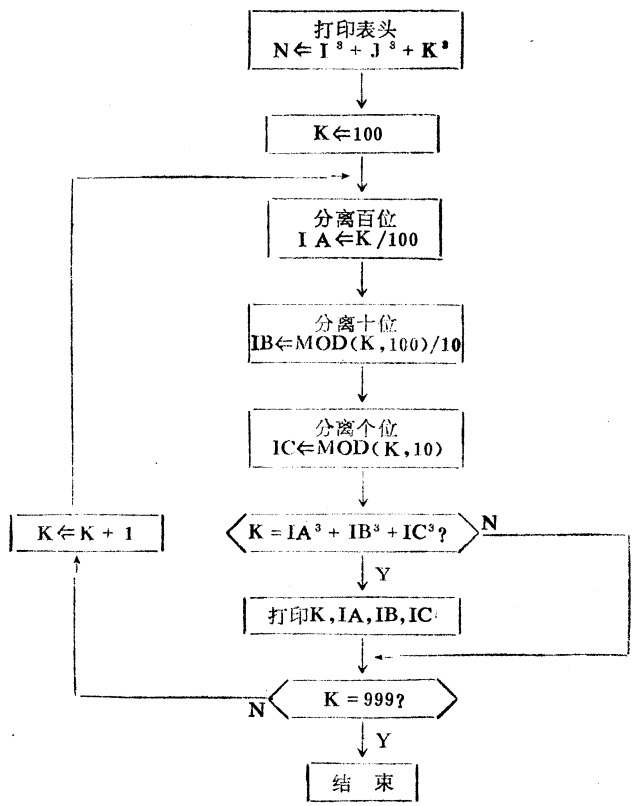


图 36

3）源程序及运行结果

```
      WRITE(5,30)
      DO 10 K=100,999
      IA=K/100
      IB=MOD(K,100)/10
      IC=MOD(K,10)
      IF(K.NE.IA**3+IB**3+IC**3)GOTO10
      WRITE(5,20)K, IA, IB, IC
10    CONTINUE
20    FORMAT(5X,4I4)
30    FORMAT(5X,18HN=I**3+J**3+K**3)
      STOP
      END
```

N = I**3 + J**3 + K**3

153	1	5	3
370	3	7	0
371	3	7	1
407	4	0	7

（三）思 考 题

奇数列与数的立方有密切的联系，表明这一性质的规律是：

$1^3 = 1$

$2^3 = 8 = 3 + 5$

$3^3 = 27 = 7 + 9 + 11$

$4^3 = 64 = 13 + 15 + 17 + 19$

……

请编写程序验证。

37 巧妙的算法

$1^2 = 1$，$2^2 = 1 + 3$，$3^2 = 1 + 3 + 5$

请仔细研究上面这些式子，看能否从中找出某种规律，然后利用这种规律迅速算出下面式子。

（1） $1 + 3 + 5 + 7 + 9 + 11 = ?$

（2） $1 + 3 + 5 + 7 + 9 + 11 + 13 + 15 + 17 + 19 + 21 + 23 + 25 = ?$

（一）笔算步骤和结果

继续写下去，规律就更明显了。

$4^2 = \underbrace{1 + 3 + 5 + 7}_{4\text{项}} = 16$，　　$5^2 = \underbrace{1 + 3 + 5 + 7 + 9}_{5\text{项}} = 25$

$6^2 = \underbrace{1 + 3 + 5 + 7 + 9 + 11}_{6\text{项}} = 36$，　　$7^2 = \underbrace{1 + 3 + 5 + 7 + 9 + 11 + 13}_{7\text{项}} = 49$。

我们还可用等差级数前几项求和公式$S_n=\frac{(a_1+a_n)n}{2}$证明

$$n^2=\underbrace{1+3+5+7+\cdots+(2n-3)+(2n-1)}_{n\text{项}}$$

（二）程 序 设 计

1）设计思路

由笔算找出规律，从而可以快速计算。我们编写程序用来验证，由自然数1依次的连续奇数至N。将程序编写为通用性，用循环语句，由奇数1开始，步长为2，保证每次都是奇数，因而连续。并将结果打印出，当遇－1时便停，令N为三位数字。

2）框图 见图37

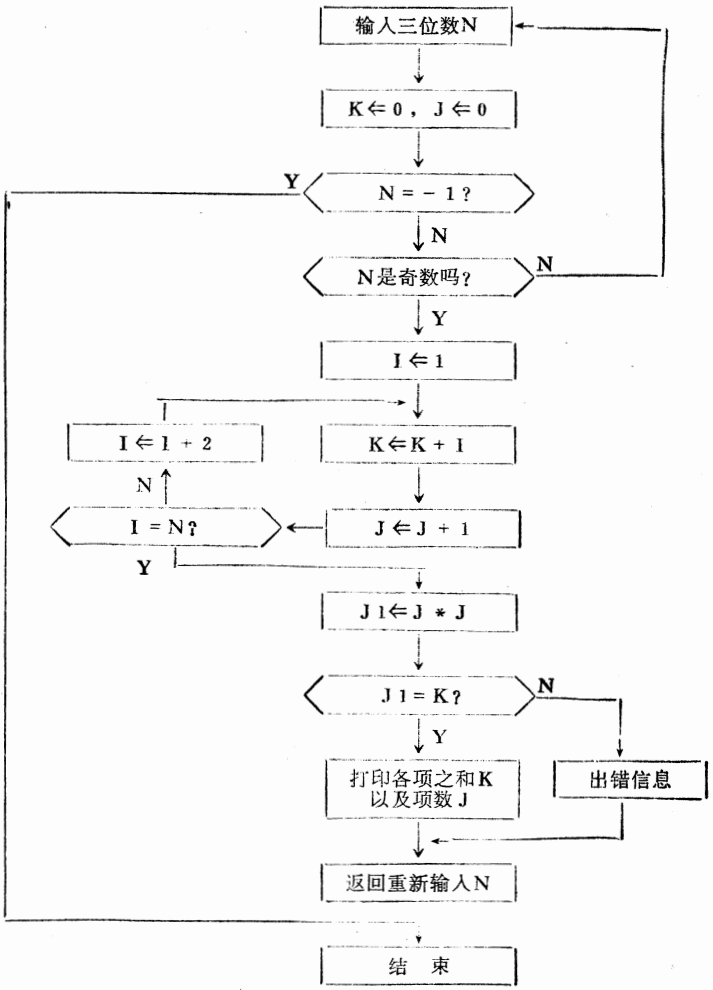


图 37

3）源程序及运行结果

```
5       WRITE(5,60)
        READ(5,10)N
10      FORMAT(I3)
        K = 0
        J = 0
        IF(N. EQ. -1)GOTO50
        IF(N/2*2. EQ. N)GOTO5
        DO 20 I=1, N, 2
        K = K + I
20      J = J + 1
        J1 = J*J
        IF(J1. NE. K)GOTO 40
        WRITE(5,30)N, K, J, J
30      FORMAT(5X,1H1,2H--,I3, 3H:       , I6, 1H=,I3,1H*, I3)
        GOTO 5
40      WRITE(5,45)
45      FORMAT(5X,5HERROR)
60      FORMAT(5X,12HINFUT  N  =  ?  )
        GOTO  5
50      STOP
        END
```

```
1- -9:      25 = 5* 5
1- -5:       9 = 3* 3
1- -7:      16 = 4* 4
1- -11:     36 = 6* 6
1- -13:     49 = 7* 7
1- -21:    121 = 11*11
1- -33:    289 = 17*17
```

（三）思 考 题

请找出十个连续的自然数，个个都是合数(非素数)。程序运行后，把结果打印输出。

38 完 全 数

古希腊人认为某些数是完全的数，所谓完全数，就是指该数的所有因数之和等于它自身的数，数6就是这样一个数，因为6 = 1 + 2 + 3，另一个完全数是28，因为28 = 1 + 2 + 4 + 7 + 14，28之后下一个完全数相当大，不易通过实验求得，所有这些数都是偶数，从来没有人找到过一个奇数完全数，但是没有人能够证明：每一个完全数必然是偶数。也从未有人能够找到计算完全数的公式，你能用试验的办法再找到几个完全数吗？

（一）算法分析

还没有计算完全数的公式。题中谈到至今发现的完全数都不是奇数。已知大于2的素

数都是奇数，可见素数与完全数是无缘的。还可概括简述如下。

1）很明显，素数不可能是完全数。1和P是素数P的所有因子，而P本身还不能当作因子。因此所有因子的和就是1，但1当然不等于P。

2）除了注意到这个极简单的情形，我们可以考察一种略为复杂的情形。数9不是素数，但它是一个素数的平方。试验由1到8的数，我们看到1和3是可以考虑的所有因子。由于1 + 3 = 4，小于9，所以9不是完全数。

3）一般来说，我们可以用同样的方法证明，素数的任何次幂都不是完全数。

不仅没有计算完全数的公式，也没有人说出完全数是否有无限多个或者是否有最后一个。

我们让程序去搜寻完全数。

（二）程序设计

1）设计思路

用循环语句来搜寻2～10000的自然数里究竟有多少个完全数。每个自然数N判它有几个因子I，置放数组K(100)，将每个I累加，置入X单元，并记下因子I的个数。若N = X，说明N是一个完全数，打印出N、N的因数的个数J、数组K。再继续搜寻另外的完全数。能寻找出几个呢？请见程序运行后打印结果。

2）框图　见图38

3）源程序及运行结果

```
      INTEGER X, Y
      DIMENSION K(100)
      DO 10 N= 2, 10000
      X = 0
      Y = 0
      J = 1
      N1 = N/2
      DO 20 I = 1, N1
      IF (N. NE. N/I*I) GO TO 20
      X = X + I
      Y = Y + 1
      K(J) = I
      J = J + 1
20    GONTINUE
      IF(X. NE. N)  GO TO 10
      WRITE(6,30)  N, Y, (K(M), M= 1, Y)
30    FORMAT(/1X, 4HSUM:I4, 3X, 7HNUMBER:I3, 1H:3X, 20I5)
10    CONTINUE
      STOP
      END
```

程序运行后打印结果如下

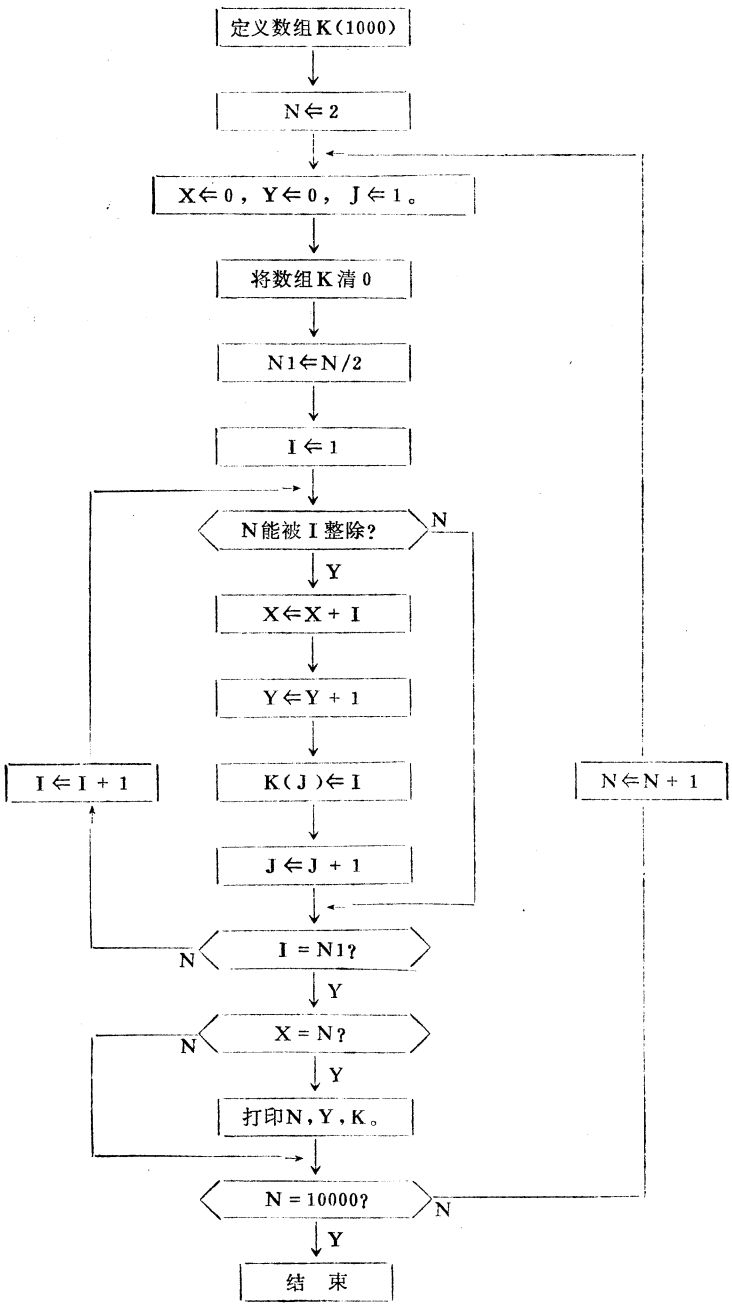


图 38

	完全数		因子个数	因子								
SUM:	6	NUMBER:	3:	1	2	3						
SUM:	28	NUMBER:	5:	1	2	4	7	14				
SUM:	496	NUMBER:	9:	1	2	4	8	16	31	62	124	248
SUM:	8128	NUMBER:	13:	1	2	4	8	16	32	64	127	254
				508	1016	2032	4064					

（三）思 考 题

请你将自然数扩大，继续搜寻完全数，看你还能再找出几个来。

39 自然数可用不超过四个平方数之和来表示

任意大于 2 的自然数都可用不超过四个平方数之和来表示。

试写出大于 2 且小于96的并能够写成从 1 到10的各个整数的平方的两个或三个、或四个的和的数。

（一）笔算步骤和结果

不难找出3～95的每个自然数，可用1～10中二、三、四个数的平方和表示，例如：

$$13 = 3^2 + 2^2,$$

$$33 = 5^2 + 2^2 + 2^2 = 4^2 + 4^2 + 1^2,$$

$$80 = 8^2 + 4^2 = \cdots\cdots$$

$$\cdots\cdots。$$

（二）程序设计

1）设计思路

在程序中找3～95的数，可能有多种表示法，但只要找到其中一种便可，一般是先找到一个大数的平方和，余下的数用小的数的平方和填补凑成。定义数组MA(4)，置放四个数的平方，空位时用 0 代替。外循环控制变量M为3～95，内循环控制变量 I 为1～10，寻找每个M由几个I^2构成，每找到一个答案，便打印输出一行，以便观察。

2）框图 见图39

3）源程序及运行结果

```
      DIMENSION MA( 4 )
      DO 10 M = 3, 95
      N = M
      DO 20 I = 1,10
      IF (N. LE. I*I) GO TO 30
 20   CONTINUE
 30   L = I - 1
110   MT = 0
      DO 40 J = 1, L
      J1 = L - J + 1
      DO 50 K = 1, 4
      IF (N. LE. J1*J1) GO TO 40
      N = N - J1* J1
      MT = MT + 1
      IF (MT. EQ. 5) GO TO 100
      MA(MT) = J1*J1
      IF(N.EQ.O) GO TO 60
 50   CONTINUE
 40   CONTINUE
```

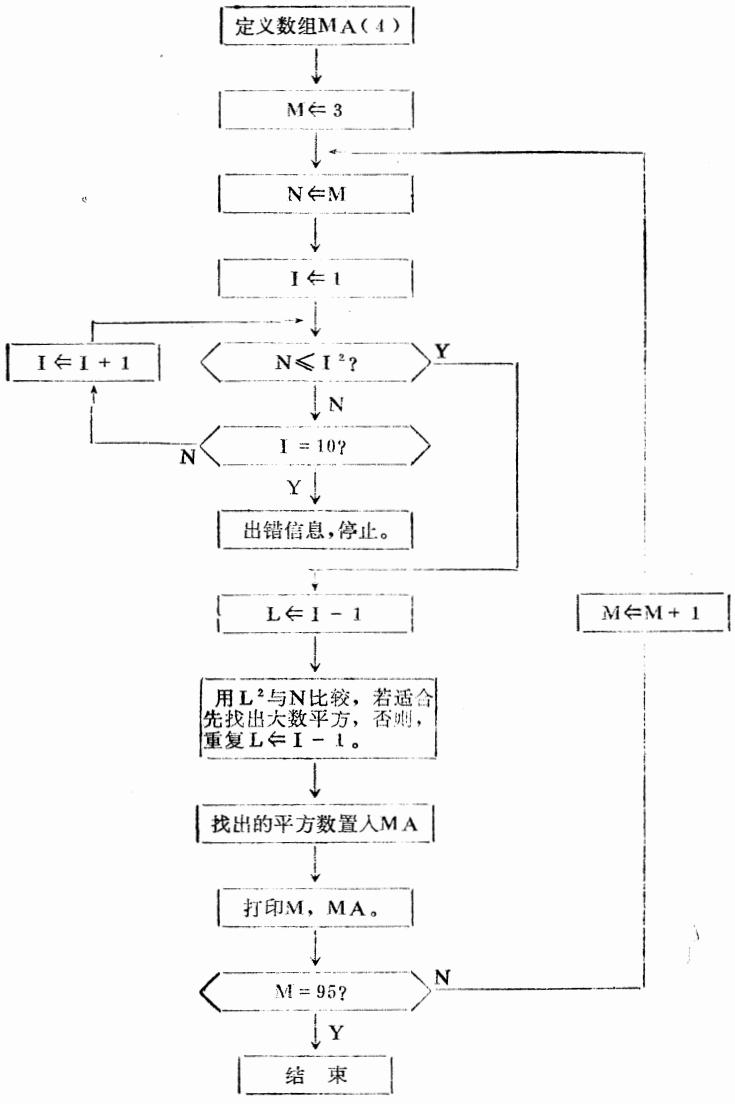


图 39

```
60      WRITE(6,70)  M, MA(1) , MA(2) , MA(3) MA(4)
70      FORMBT (1X, I3, 1H=, I3, 1H+, I3, 1H+, I3, 1H+, I3)
        DO 80 MP=1 , 4
80      MA(MP)=0
10      CONTINUE
        STOP
100     L=L-1
        N=M
        DO 90 MP=1 , 4
90      MA(MP)=0
        GO TO 110
```

END

3 =	1 +	1 +	1 +	0
4 =	4 +	0 +	0 +	0
5 =	4 +	1 +	0 +	0
6 =	4 +	1 +	1 +	0
7 =	4 +	1 +	1 +	1
8 =	4 +	4 +	0 +	0
9 =	9 +	0 +	0 +	0
10 =	9 +	1 +	0 +	0
11 =	9 +	1 +	1 +	0
12 =	9 +	1 +	1 +	1
13 =	9 +	4 +	0 +	0
14 =	9 +	4 +	1 +	0
15 =	9 +	4 +	1 +	1
16 =	16 +	0 +	0 +	0
17 =	16 +	1 +	0 +	0
18 =	16 +	1 +	1 +	0
19 =	16 +	1 +	1 +	1
20 =	16 +	4 +	0 +	0
21 =	16 +	4 +	1 +	0
22 =	16 +	4 +	1 +	1
23 =	9 +	9 +	4 +	1
24 =	16 +	4 +	4 +	0
25 =	25 +	0 +	0 +	0
26 =	25 +	1 +	0 +	0
27 =	25 +	1 +	1 +	0
28 =	25 +	1 +	1 +	1
29 =	25 +	4 +	0 +	0
30 =	25 +	4 +	1 +	0
31 =	25 +	4 +	1 +	1
32 =	16 +	16 +	0 +	0
33 =	25 +	4 +	4 +	0
34 =	25 +	9 +	0 +	0
35 =	25 +	9 +	1 +	0
36 =	36 +	0 +	0 +	0
37 =	36 +	1 +	0 +	0
38 =	36 +	1 +	1 +	0
39 =	36 +	1 +	1 +	1
40 =	36 +	4 +	0 +	0
41 =	36 +	4 +	1 +	0
42 =	36 +	4 +	1 +	1
43 =	25 +	16 +	1 +	1

44 =	36 +	4 +	4 +	0
45 =	36 +	9 +	0 +	0
46 =	36 +	9 +	1 +	0
47 =	36 +	9 +	1 +	1
48 =	16 +	16 +	16 +	0
49 =	49 +	0 +	0 +	0
50 =	49 +	1 +	0 +	0
51 =	49 +	1 +	1 +	0
52 =	49 +	1 +	1 +	1
53 =	49 +	4 +	0 +	0
54 =	49 +	4 +	1 +	0
55 =	49 +	4 +	1 +	1
56 =	36 +	16 +	4 +	0
57 =	49 +	4 +	4 +	0
58 =	49 +	9 +	0 +	0
59 =	49 +	9 +	1 +	0
60 =	49 +	9 +	1 +	1
61 =	36 +	25 +	0 +	0
62 =	49 +	9 +	4 +	0
63 =	49 +	9 +	4 +	1
64 =	64 +	0 +	0 +	0
65 =	64 +	1 +	0 +	0
66 =	64 +	1 +	1 +	0
67 =	64 +	1 +	1 +	1
68 =	64 +	4 +	0 +	0
69 =	64 +	4 +	1 +	0
70 =	64 +	4 +	1 +	1
71 =	36 +	25 +	9 +	1
72 =	64 +	4 +	4 +	0
73 =	64 +	9 +	0 +	0
74 =	64 +	9 +	1 +	0
75 =	64 +	9 +	1 +	1
76 =	49 +	25 +	1 +	1
77 =	64 +	9 +	4 +	0
78 =	64 +	9 +	4 +	1
79 =	49 +	25 +	4 +	1
80 =	64 +	16 +	0 +	0
81 =	81 +	0 +	0 +	0
82 =	81 +	1 +	0 +	0
83 =	81 +	1 +	1 +	0
84 =	81 +	1 +	1 +	1
85 =	81 +	4 +	0 +	0
86 =	81 +	4 +	1 +	0

87 = 81 + 4 + 1 + 1
88 = 64 + 16 + 4 + 4
89 = 81 + 4 + 4 + 0
90 = 81 + 9 + 0 + 0
91 = 81 + 9 + 1 + 0
92 = 81 + 9 + 1 + 1
93 = 64 + 25 + 4 + 0
94 = 81 + 9 + 4 + 0
95 = 81 + 9 + 4 + 1

（三）思 考 题

请验证每个正整数可以表示成最多是四个平方数之和。要改写本程序，因本程序是据3～95的正整数编写的。

40 惊人的记忆力

魔术演员有时能以异于常人的记忆力使观众惊叹不止。他们会牢牢记住大串大串的词语、数目等等。你也可以用这样的魔术让同事们吃一惊的。我告诉你怎样表演。

预备50张卡片，照图40-1那样写上数目和符号。每张卡片上都有一长串数字，左侧有一个英文字母和一个数字组成的编号。把这些卡片分发给同事们，然后告诉他们说，每张卡片上写的所有数字你记得很牢，他们只要说出编号，你就能说出数目来。例如他们说“E_4”，你立即说出10128224。

因为数字很长，总数又有50个之多，所以你的表演必然会使在场的人大为惊奇。然而你却根本没有背熟这些长数，事情到底如何，秘密在哪里？

A_1	34212	B_1	46223	C_1	58234	D_1	610245	E_1	712256
A_2	44404	B_2	56416	C_2	68428	D_2	7104310	E_2	3124412
A_3	54616	B_3	66609	C_3	786112	D_3	8106215	E_3	9126318
A_4	64828	B_4	768112	C_4	888016	D_4	9108120	E_4	10128224
A_5	750310	B_5	870215	C_5	990120	D_5	10110025	E_5	11130130
A_6	852412	B_6	972318	C_6	1092224	D_6	11112130	E_6	12132036
A_7	954514	B_7	1074421	C_7	1194328	D_7	12114235	E_7	13134142
A_8	1056616	B_8	1176524	C_8	1296432	D_8	13116340	E_8	14136248
A_9	1158718	B_9	1278627	C_9	1398536	D_9	14118445	E_9	15138354

图 40-1

（一）笔算步骤和结果

这里的秘密是卡片上的编号（字母和数字）能告诉你卡片上的数字是什么。首先要记住，A代表20，B代表30，C代表40，D代表50，E代表60。因此字母和它右下角的数字，按一定规则来编制卡片上的长串数。举例说：

假如人家说了“E_4”，这就是64，你做如下处理：

第一，把两个数字相加，6 + 4 = 10；

第二，把该数乘以2，64 × 2 = 128；

第三，大数减去小的数，6 − 4 = 2；

第四，两个数字相乘，$6 \times 4 = 24$。

把全部得数连续写在一起，成为10128224，这就是事先写在“E_4”这张卡片上的数。

这些运算可以这样简单表示：|+，2，−，×|，即相加、乘2、大减小、相乘。

（二）程序设计

1）设计思路

我们索性将魔术师记忆全亮出来，并打印出卡片，与笔算的卡片对照。卡片中最小5位数，最大是8位数。

定义数组N(8)，置放8位数字的每一位，先从高位开始放置，依次低位至N(1)，用计数器L计数，每放入一数，L = L + 1，置入N(L)中。

卡片共有9行5列，定义实型数组S(5，9)，置放5 × 9 = 45个元素，因最大元素是8位数，取实型数，打印出带小数点。

由上分析，利用二重循环，组成行列数，又可利用循环参数，互相搭配成两位数（即观众任意说给魔术师的两位数）。

由笔算顺序，程序依次计算两位数的加、乘2、相减、相乘。每变换一次，将数由高位至低位依次冲入数组N(8)里，最后将这单个数相加，充入数组S(5，9)里，打印数组S(5，9)，便成卡片。

2）框图　见图40-2

3）源程序及运行结果

```
      DIMENSION N(8), S(5, 9)
      DO 10 I= 1, 9
      DO 10 J= 1, 5
      SUM= 0
      L= 1
      M= I + J + 1
      IF (M. LT. 10) GOTO 1
      N (L) =M/10
      L= L + 1
1     N (L) =MOD(M, 10)
      L= L + 1
      M= 2 * (10*(J + 1) + I)
      IF (M. LT. 100) GOTO 2
      N (L) =M/100
      L= L + 1
2     N (L) =MOD(M, 100) /10
      L= L + 1
      N (L) =MOD(M, 10)
      L= L + 1
      N(L) =IABS(J - I + 1)
      L= L + 1
      M= I * (J + 1)
```

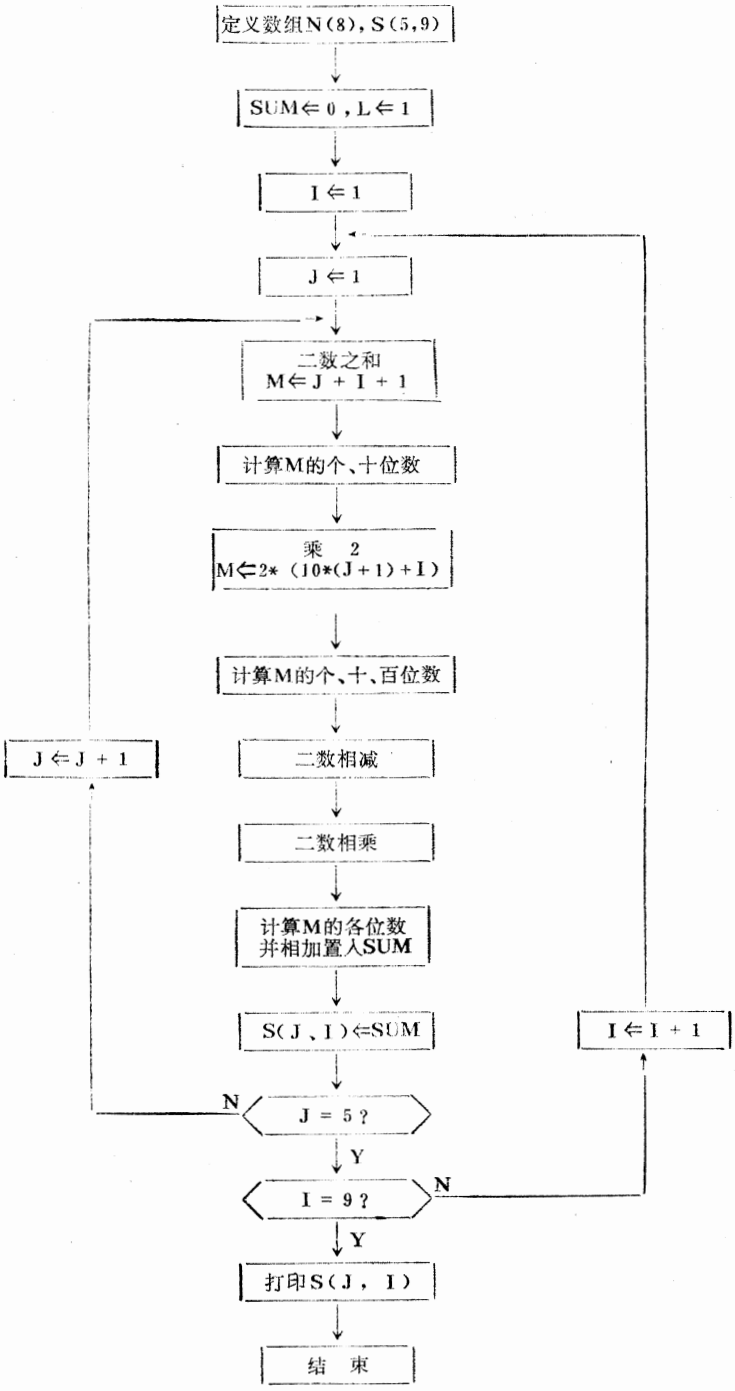

定义数组N(8)，S(5,9)
SUM⇐0，L⇐1
I⇐1
J⇐1
二数之和
M⇐J＋I＋1
计算M的个、十位数
乘 2
M⇐2*（10*(J+1)+I）
计算M的个、十、百位数
二数相减
二数相乘
计算M的各位数
并相加置入SUM
S(J、I)⇐SUM
J = 5？
N
Y
I = 9？
N
Y
打印S(J，I)
结 束
J⇐J＋1
I⇐I＋1

图 40-2

```
      IF (M. LT. 10) GOTO 3
      N(L) =M/10
      L = L + 1
3     N(L) =MOD(M, 10)
      DO 20 K= 1, L
      SUM = SUM + 10. **(L-K) *N(K)
20    CONTINUE
      S (J, I) =SUM
10    CONTINUE
      WRITE (5,50)
      DO 40 I = 1, 9
40    WRITE (5,30) (I, S(J, I), J = 1, 5)
30    FORMAT (3X, 3H1  A, I1, F10.0, 3H1  B, I1, F10.0, 3H1  C, I1,
     1 F10.0, 3H1  D, I1, F10.0, 3H1  E, I1, F10.0,2H  1)
      WRITE (5, 50)
      STOP OK!
50    FORMAT (3X, 35H-----------------------------------,
     1 37H-------------------------------------)
      END
```

程序运行后打印结果

A1	34212.	B1	46223.	C1	58234.	D1	610245.	E1	712256.
A2	44404.	B2	56416.	C2	68428.	D2	7104310.	E2	8124412.
A3	54616.	B3	66609.	C3	786112.	D3	8106215.	E3	9126318.
A4	64828.	B4	768112.	C4	888016.	D4	9108120.	E4	10128224.
A5	750310.	B5	870215.	C5	990120.	D5	10110025.	E5	11130130.
A6	852412.	B6	972318.	C6	1092224.	D6	11112130.	E6	12132036.
A7	954514.	B7	1074421.	C7	1194328.	D7	12114235.	E7	13134142.
A8	1056616.	B8	1176524.	C8	1296432.	D8	13116340.	E8	14136248.
A9	1158718.	B9	1278627.	C9	1398536.	D9	14118445.	E9	15138254.

（三）思　考　题

将程序改写，观众提出任一张号码，如E_4（由键盘输入64），如B_2（由键盘输入32），……每提问一张号码，程序运行后打印出相应结果。

六 漫步迷宫

41 漫步迷宫

试编写一个程序寻找一条通过迷宫的路径。一个迷宫可看成一个矩阵，它包含若干个黑色单元和若干个白色单元，并且被指定了一个入口单元和一个出口单元。通过迷宫要从入口开始，在每次移动中只能从一个白色单元移到一个相邻的（上下相邻的或左右相邻的）白色单元，移至出口单元为止。

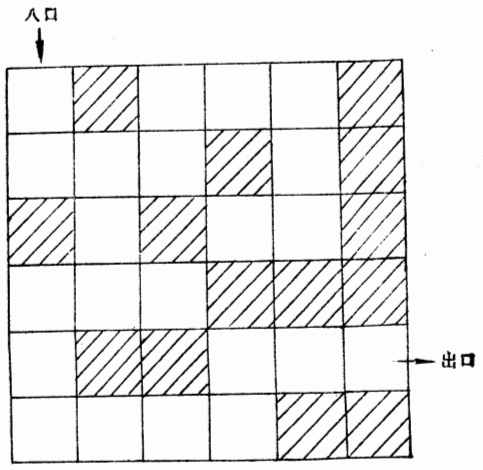


图 41-1

（一）算法分析

我们暂指定该迷宫为 6 × 6 的二维数组，不失一般性。出入口及障碍布置如图41-1所示。阴影表示黑格子（障碍物），其余白格子可通行。

（1）设入口位置〔1，1〕为－2，出口位置〔5，6〕为－1，空格为0，障碍物为1。则 6 × 6 二维数组按列写成如图41-2所示。

```
－2，0，1，0，0，0
 1，0，0，0，1，0
 0，0，1，0，1，0
 0，1，0，1，0，0
 0，0，0，0，0，1
 1，1，1，1，－1，1
```

图 41-2

（2）设指针 k 表示走的方向，按题意只许上下左右走空位。故 k 由 1 至 4，其方向如图41-3所示。

（3）将图41-3赋值便是图41-4的位移增量，1 列是行 I 的增量，2 列是列 J 的增量。如 k = 1 时，c（k，1）= 1，c（k，2）= 0。

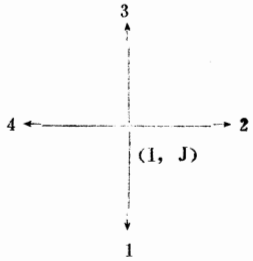


图 41-3

（4）在每个位置试探时，先从方向 1 开始，走不通时，依次 k ⇐ k + 1，当 k 为尽头 4 时，下次又恢复方向 1。

	1	2
K	I	J
1	1	0
2	0	1
3	-1	0
4	0	-1

图 41-4

（二）程序设计

1）设计思路

（1）定义数组A(6，6)赋图41-2值，数组B(6，6)先清零，每走通一步：(a)打印A(M，N)。(b)将A(M，N)保留于B(M，N)里，但这时B(M，N)置5，表示已走过。(c)保留位置座标I⇐M，J⇐N。(d)重新开始K⇐1。

（2）试探时：(a)先判行M、列N是否出界，若出界，则改变方向。(b)判走的位置是否为-1出口，是，则打印结果。否则，继续判、走，直到出口为止。

2）框图 见图41-5

3）源程序及运行结果

```
C       PROGRAM  MBMG
        INTEGER  A(6, 6), B(6, 6), C(4, 2)
        DATA  C/1, 0, -1, 0, 0, 1, 0, -1/
        READ(5, 5)A                          输入图41-2于A数组
5       FORMAT (6I2)
        WRITE (5, 10) ((A (I, J), J=1, 6), I=1, 6) 打印数组A，观察输入对否。
10      FORMAT (5X, 6I3)
        DO  20  I=1, 6
        DO  20  J=1, 6                       将数组B清0
20      B (I, J) =0
        I = 1
        J = 1
        K = 1                                初始化，从第1个位置开始。
        M = 0
        N = 0
        JP = 0
30      M = I + C(K, 1)                      行=行+指针值
        N = J + C(K, 2)                      列=列+指针值
        IF (M. EQ. 0. OR. M. EQ. 7. OR. N. EQ. 0. OR. N. EQ. 7) GO  TO  70
                                             已过边界不能试探
        IF (A (M, N). EQ. -1) GO  TO  80     已到出口，转打印结束。
        IF (A (M, N). NE. 0) GO  TO  60      不是空白，不通。
        IF (B (M, N). NE. 5) GO  TO  40      通，又不是走过的位置，故可走。
        I1=M
        J1=N                                 虽然通，但已走过该位置，暂保存该位置于I1、J1。
        JP = 1                               置标记JP
        GO  TO  60                           再继续找其它位置有否可行
40      I = M
        J = N                                来此，走到M、N位置，保留行、列于I、J中。
        WRITE (5, 50) I, J
50      FORMAT (5X, 2HA (, I1, 1H,, I1, 1H) /) 打印走通的位置
```

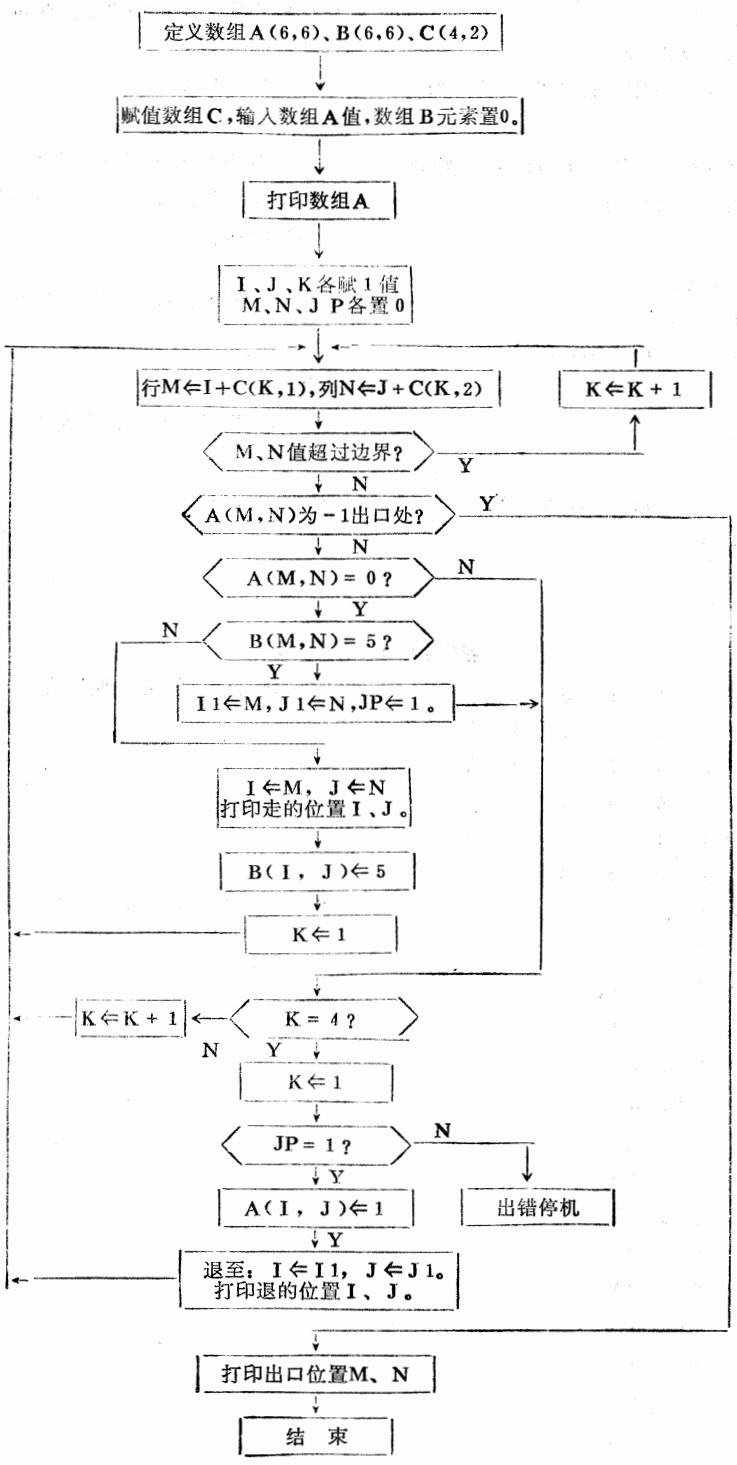

定义数组A(6,6)、B(6,6)、C(4,2)
赋值数组C，输入数组A值，数组B元素置0。
打印数组A
I、J、K各赋1值
M、N、J P各置0
行M⇐I+C(K,1)，列N⇐J+C(K,2)
K⇐K+1
M、N值超过边界？
Y
N
A(M,N)为-1出口处？
Y
N
A(M,N)=0？
N
Y
N
B(M,N)=5？
Y
I1⇐M，J1⇐N，JP⇐1。
I⇐M，J⇐N
打印走的位置I、J。
B(I，J)⇐5
K⇐1
K⇐K+1
K=4？
N
Y
K⇐1
JP=1？
N
Y
出错停机
A(I，J)⇐1
Y
退至：I⇐I1，J⇐J1。
打印退的位置I、J。
打印出口位置M、N
结 束

图 41-5

```
      B(I, J) = 5                    走过的位置树标记 5
      K = 1                          恢复指针从 1 开始继续寻找通路
      GO TO 30                       指针须到尽头 4
60    IF (K. NE. 4) GO TO 70         指针已到尽头 4
      K = 1
      IF (JP. NE. 1) STOP 'ERR'      该位置四个方向都不通，故错。
      A(I, J)=1                      需退回原位置，先将本位置 1 设障碍。
                                     避免死循环。
      I = I1
      J = J1                         退一步，退至原位（I1, J1）。
      WRITE (5, 50) I, J
      GO TO 30                       再找
70    K = K + 1                      已过界或指针没到 4，再换个方向。
      GO TO 30
80    WRITE (5, 50) M, N             到出口处，转此打印。
      STOP
      END
```

程序运行时打印如下

```
 -2   1   0   0   0   1      打印输入的A数组，观察出入口和设置的障碍。
  0   0   0   1   0   1
  1   0   1   0   0   1
  0   0   0   1   0   1
  0   1   1   0   0  -1
  0   0   0   0   1   1
A (2, 1)                     以下是打印所走的顺序位置，由入口至出口。
A (2, 2)
A (3, 2)
A (4, 2)
A (4, 3)
A (4, 2)
A (4, 1)
A (5, 1)
A (6, 1)
A (6, 2)
A (6, 3)
A (6, 4)
A (5, 4)
A (5, 5)
A (5, 6)
```

（三）思 考 题

请将本程序改写为通用形式，由终端输入二维数组M×N，自选障碍物和出入口，观察是否全能走通？为什么？

42 怪兽的踪迹

据说某国农业科学家，在地下用水泥砌成一格格方块，贮存各种粮食做实验。当地有一种嗅觉很灵敏的类似鼠的怪兽，啃破水泥隔段，进入粮“仓”吃粮食，吃几口后，又啃破隔段进入相邻的粮“仓”盗食。如此觅食，破坏力极大，几乎将所有粮“仓”全吃遍，才懒洋洋地顺着来路退出洞口。

为了捕捉这种怪兽，请绘制出怪兽觅食的踪迹。

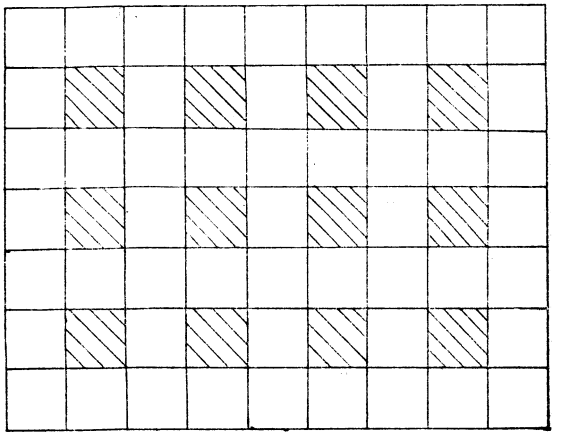


图 42-1

（一）算法分析

设农业科学家在地下用水泥砌成N×M矩阵（N、M为奇数），令位于奇行或奇列的矩阵元素为墙，位于偶行且偶列的矩阵元素为四周是墙的装粮食的方块，如图42-1所示。

试在边墙任选一个入口并按下列规则构成迷宫：

（1）如周围有可啃破的墙（即非边墙，并且该墙的另一边的方块是未经走过的方块），则任选一面啃破之，并前进一步。

（2）如周围没有可啃破的墙，则后退一步。

重复上述步骤，直至退到入口处为止。

（3）进入某一位置后，试探方向设为逆时针方向，依次为下、右、上、左。先试探一步，出界否，若出界则改变方向，若没出界，再沿该方向试探下一步，是粮“仓”否，若是进入目标的粮“仓”则连续走两步——经过墙一步，再进入粮“仓”为第二步。若墙的那边不是进入目标，则改变方向再试探，直到四个方向全试探完，无路可走时，便顺来路后退至入口。每前进两步，数组行列值加或减2。

（二）程序设计

1）设计思路

为了使程序具有通用性，可用终端输入二维数组的阶数。本程序定义最大数组K(23，25)，在这范围内可任选行、列各为奇数的数组。

（1）将奇行奇列元素赋值－1表示墙，偶行偶列元素赋值0表示粮“仓”。

（2）由终端随意指定入口位置，为第1步。由此进入目标为第二步。

（3）当进入新位置时，按算法分析给定的方向试探，连走两步，进入目标。

（4）用计数器I累加走的步数。每前进一步，便将经过的位置（K数组元素）赋于步数值，最后打印出数组K，可观察到每一步走的位置。

（5）当在某位置试探无路可走时，将该位置打印出，是后退的第一步。每后退一步，打印一次后退的位置。直至退到入口为止。

（6）我们选用数组K（7，9）和K（15，15）二例做表演，请见程序运行后打印结果。

2）框图　见图42-2

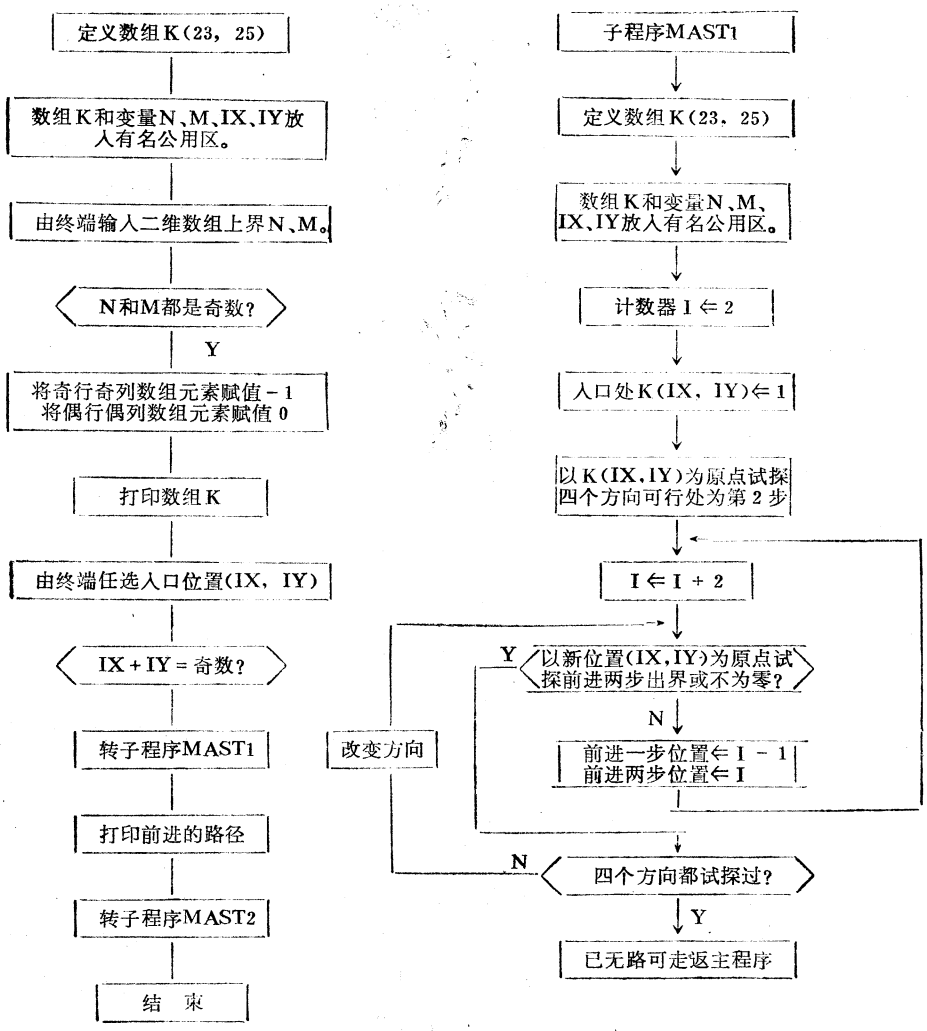


图 42-2(1)

3) 源程序及运行结果

```
      INTEGER  K(23, 25)                BCM-2机允许的数组K (23, 25)
      COMMON /KNM/K,N,M,IX,IY           将数组和变量存放在有名公用区里
10    WRITE (5, 20)
20    FORMAT (20X, 3HN=?, 3HM=?, )
      READ (5, 30) N, M
30    FORMAT (2I3)
      IF (N/2.. EQ. INT (N/2. ) . OR. M/2.. EQ. INT (M/2. )) GOTO 10
      DO 35 I=1, N                      初始化
      DO 35 J=1, M                      奇行奇列置-1值
35    K(I, J) = -1                      偶行偶列置0值
      DO 38 I=2, N, 2
      DO 38 J=2, M, 2
38    K(I, J) = 0
```

```
      DO  1  I=1, N
1     WRITE (5, 5)(K(I, J), J=1,M)  打印初始化的数组K，以便观察。
40    WRITE (5, 50)
50    FORMAT (20X, 3HX=？, 3HY=？)
      READ (5, 30) IX, IY           终端输入选择的入口位置IX、IY。
      IF ((IX+IY) /2.. EQ. INT ((IX+IY) /2. )) GOTO  40
                                    IX+IY不是奇数时重选
      CALL  MAST1                   转子程序前进
      DO  2  I=1, N                 打印数组K里前进路径顺序号
2     WRITE (5, 5) (K(I, J), J=1, M)
5     FORMAT (1X, 25I3)
      CALL  MAST2
      STOP  OK!                     转子程序后退
      END                           退至入口处则停
      SUBROUTINE  MAST1             子程序试探前进的路径，通则进。
      INTEGER  K(23, 25)
      COMMON  /KNM/K, N, M, IX, IY
      I = 2                         步数计数器I
      K(IX, IY) = 1                 入口处为第1个位置
```

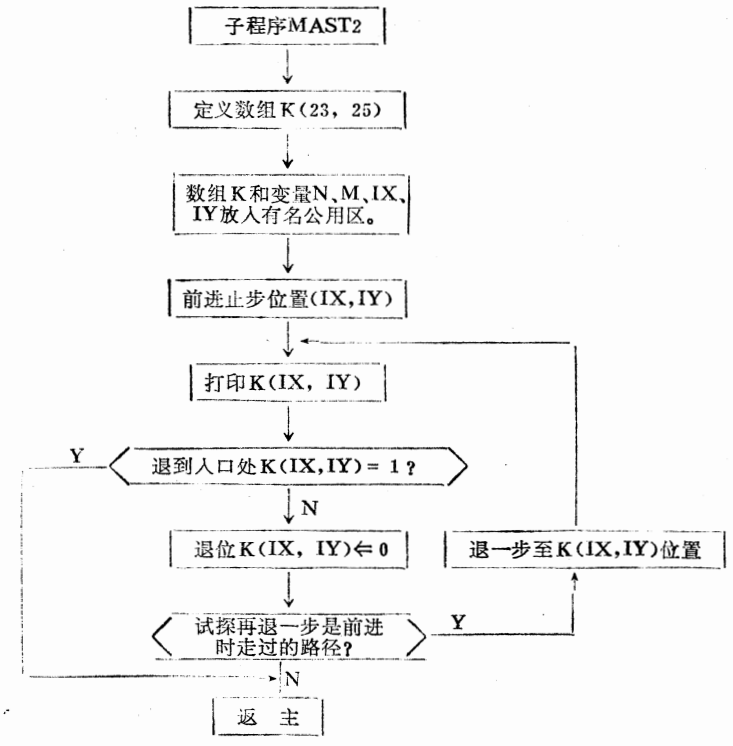


图 42-2(2)

```
      IF (IX+1. GT. N) GOTO  10      判X出界否
      IF(K(IX+1,IY).EQ.-1)GOTO 10  试探往下走是墙否
      IX=IX+1
      GOTO  40
10    IF (IY+1. GT. M) GOTO  20      判Y出界否
      IF(K(IX,IY+1).EQ.-1)GOTO 20  试探往右走是墙否
      IY=IY+1
      GOTO  40
20    IF (IX-1. LT. 1) GOTO  30      判X出界否
      IF(K(IX-1,IY).EQ.-1)GOTO 30  试探往上走是墙否
      IX=IX-1
      GOTO  40
30    IY=IY-1
40    K (IX, IY)=2                   40句是第2步
50    I=I+2                          步数计数器加2
      IF (IX+2. GT. N) GOTO  60      判往下走两步出界否
      IF(K(IX+2,IY).NE.0)GOTO 60   试探往下走第2个位置是食物仓否
      IX=IX+2                        是食物仓
      K (IX-1, IY)=I-1
      K (IX, IY)=I                   连续走两步(推倒墙一步，进入食物仓一步)
      GOTO  50
60    IF (IY+2. GT. M) GOTO  70      判往右走两步出界否
      IF(K(IX,IY+2).NE. 0)GOTO 70  试探往右走第2个位置是食物仓否
      IY=IY+2                        是食物仓
      K (IX, IY-1)=I-1
      K (IX, IY)=I                   连续走两步(推倒墙进入食物仓)
      GOTO  50
70    IF (IX-2. LT. 1) GOTO  80
      IF(K(IX-2,IY).NE.0)GOTO 80   试探往上走两步有食物仓则进入
      IX=IX-2
      K (IX+1, IY)=I-1
      K (IX, IY)=I
      GOTO 50
80    IF (IY-2. LT. 1) RETURN
      IF(K(IX,IY-2).NE.0)RETURN    试探往左走两步有食物仓则进入
      IY=IY-2
      K (IX, IY+1)=I-1
      K (IX, IY)=I
      GOTO  50
      END
      SUBROUTINE  MAST2              子程序试探后退路径，通则退。
      INTEGER  K (23, 25)
      COMMON  /KNM/K, N, M, IX, IY
```

```
10      WRITE (5, 20) IX, IY
20      FORMAT(20X,2HA(,I2,1H,,1H),)
        IF(K(IX,IY). EQ. 1) RETURN
        K (IX, IY) = 0
        IE (IX+1. GT. N) GOTO 30
        IF(K(IX+1,IY).LE.0)GOTO 30
        IX = IX+1
        GOTO 10
30      IF (IY+1. GT. M)GOTO 40
        IF(K(IX,IY+1).LE.0)GOTO 40
        IY = IY+1
        GOTO 10
40      IF (IX-1. LT. 1) GOTO 50
        IF(K(IX-1,IY).LE.0)GOTO 50
        IX = IX-1
        GOTO 10
50      IF (IY-1. LE. 0) RETURN
        IF(K(IX,IY-1).LE.0)RETURN
        IY = IY-1
        GOTO 10
        END
```

先从上述无路可走时的位置IX、IY后退每后退一步打印一次。

退到入口(即前进时第1个位置)则停

要从前进的路径上逐步退出。因每前进一步时将路径中的0和－1已被走的步数置换，故后退时若元素值小于等于0时，不能后退。

```
                N = ? M = ? 7, 9
-1 -1 -1 -1 -1 -1 -1 -1 -1
-1  0 -1  0 -1  0 -1  0 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1
-1  0 -1  0 -1  0 -1  0 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1
-1  0 -1  0 -1  0 -1  0 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1
```

输入数组的行列值上界

左表是输入的原始数组

```
                X = ? Y = ? 2, 1
-1 -1 -1 -1 -1 -1 -1 -1 -1
 1  2 -1 24 -1 18 17 16 -1
-1  3 -1 23 -1 19 -1 15 -1
-1  4 -1 22 21 20 -1 14 -1
-1  5 -1 -1 -1 -1 -1 13 -1
-1  6  7  8  9 10 11 12 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1
                    A (2, 4)
                    A (3, 4)
                    A (4, 4)
                    A (4, 5)
                    A (4, 6)
```

输入选择的入口位置

左表是程序运行时打印的路径

表中数字1～24是依次走过的顺序位置

以下是程序运行时打印的后退顺序位置

```
                    A (3, 6)
                    A (2, 6)
                    A (2, 7)
                    A (2, 8)
                    A (3, 8)
                    A (4, 8)
                    A (5, 8)
                    A (6, 8)
                    A (6, 7)
                    A (6, 6)
                    A (6, 5)
                    A (6, 4)
                    A (6, 3)
                    A (6, 2)
                    A (5, 2)
                    A (4, 2)
                    A (3, 2)
                    A (2, 2)
                    A (2, 1) STOP  OK!
```

```
              N = ? M = ? 15, 15      输入数组的行列值上界
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1  0 -1  0 -1  0 -1  0 -1  0 -1  0 -1  0 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1  0 -1  0 -1  0 -1  0 -1  0 -1  0 -1  0 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1  0 -1  0 -1  0 -1  0 -1  0 -1  0 -1  0 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1  0 -1  0 -1  0 -1  0 -1  0 -1  0 -1  0 -1   左表是输入的原始数据
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1  0 -1  0 -1  0 -1  0 -1  0 -1  0 -1  0 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1  0 -1  0 -1  0 -1  0 -1  0 -1  0 -1  0 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1  0 -1  0 -1  0 -1  0 -1  0 -1  0 -1  0 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

```
              X = ? Y = ? 15, 12      输入选择的入口位置
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1  0 -1 82 81 80 -1 50 49 48 -1 18 17 16 -1
-1 -1 -1 83 -1 79 -1 51 -1 47 -1 19 -1 15 -1
-1 86 85 84 -1 78 -1 52 -1 46 -1 20 -1 14 -1
-1 87 -1 -1 -1 77 -1 53 -1 45 -1 21 -1 13 -1
```

```
-1  88 -1  74  75  76 -1  54 -1  44 -1  22 -1  12 -1
-1  89 -1  73 -1 -1 -1  55 -1  43 -1  23 -1  11 -1
-1  90 -1  72 -1  58  57  56 -1  42 -1  24 -1  10 -1
-1  91 -1  71 -1  59 -1 -1 -1  41 -1  25 -1   9 -1
-1  92 -1  70 -1  60 -1  38  39  40 -1  26 -1   8 -1
-1  93 -1  69 -1  61 -1  37 -1 -1 -1  27 -1   7 -1
-1  94 -1  68 -1  62 -1  36 -1  30  29  28 -1   6 -1
-1  95 -1  67 -1  63 -1  35 -1  31 -1 -1 -1   5 -1
-1  96 -1  66  65  64 -1  34  33  32 -1   2   3   4 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1   1 -1 -1 -1
```

左表是程序运行时打印的路径，表中数字1～96是依次走过的顺序位置。

```
A (14, 2)     以下是程序运行时打印的后退顺序位置
A (13, 2)
A (12, 2)
A (11, 2)
A (10, 2)
A ( 9, 2)
A ( 8, 2)
A ( 7, 2)
A ( 6, 2)
A ( 5, 2)
A ( 4, 2)
A ( 4, 3)
A ( 4, 4)
A ( 3, 4)
A ( 2, 4)
……(省略)
A (14, 12)
A (15, 12) STOP  OK!
```

（三）思 考 题

除边墙外，其余的也不一定能走遍全距阵，如我们选用数组K（15，15）时，在数组元素K（2，2）的位置，便没有走到。这和选择入口位置有关。请你自选数组上界和入口位置，观察行走路径，总结规律，如何能走遍全矩阵？如何才能使走的步数最少？

43 四通八达的迷宫

在一个M×N个格子的迷宫中，随机地在格子布设路障（记为1）。能否编写一个程序，要求从指定的始点起，找出一条能通向指定终点的道路。规定在每一位置，不仅可以在上下左右四个方向走，而且还可在通过该位置的两个对角线方向走。如图43-1中a位置示意。若所设路障无法走出，则可退回，直到退回入口为止。

（一）算 法 分 析

可用M×N的二维数组表示迷宫，元素值为0时表示通路，元素值为1表示有路障走不通。并设入口点的坐标为〔1，1〕，出口为〔M，N〕。

用试探法找出可以走通的下一步。当处于某个位置〔I，J〕时，有八个方向可以逐一

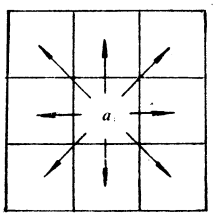


图 43-1

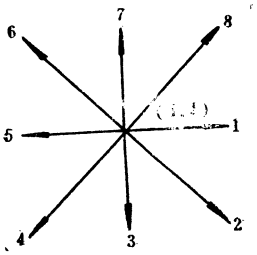


图 43-2

试探，对各方向安顺时针顺序依次编号，编号为1，2，……，8。（图43-2）。且规定每次都从方向 1 开始试探，当方向 1 走不通时，换判方向 i，依此顺序改换试探的方向。

（1）从入口点〔1，1〕处开始试探。

（2）对每个位置〔I，J〕，在试探下一步时，首先从方向 1 开始。若某个方向 i（$1 \leqslant i \leqslant 8$）所指的位置处的元素值为 0，则表示此路可行，于是记住目前所处的位置〔I，J〕和试探的方向 i，并前进判已找到的下一个位置，然后返回本步骤的开始处。

（3）若在某个位置〔I，J〕处试探时，七个方向都试探过了，但是都不通，则只好退回一步，即回到刚才记下的位置〔I，J〕，并恢复上一步进来时的方向 i，然后从方向 i +1（$i+1 \leqslant 8$）处开始（2）中的试探。

（4）若一直退到位置〔1，1〕，且〔1，1〕所邻的方向都试探过了，均行不通，则说明此迷宫无解——找不到出口。

（5）若一直找到位置〔M，N〕，且〔M，N〕的元素值为 0，则说明已经找到一条从入口到出口的位置，至此试探过程结束。每次记下的〔I，J〕和方向 i 则是该路径。

（二）程 序 设 计

1）设计思路

设迷宫为5×15的二维数组，为了与试探算法一致，程序中设置 7×17的二维数组 MAZE，其第一行、第七行、第一列、第十七列的元素均为 1。

当处于某位置〔I，J〕时，可能试探的八个方向的位移增量用8×2的二维数组MOVE表示，其行号表示其方向编号（图43-3），第一列为行方向的增量，第二列表示列方向的增量。

为了记住所走过的路径，特设置一个130×3的二维数组 STACK，作为栈使用。变量 TOP是栈顶的指针。当处于〔I，J〕位置时，若找到某个方向 i 是可以通过的，则把 I、J 和 i 的值分别放入TOP所指的STACK的相应位置中，并把 TOP 的值加1（图43-4）。若要退栈，则取出TOP-1所指位置的 I、J 和 i 的值，即为应退回的位置。下一次从 i + 1 的方向试探。

MA为21×61的二维数组，作为迷宫的输出之用。当找到一条通路之后，则在该路途上分别用 a 符号连上。

MA中的“X”表示该位置设置了路障，无“X”则表示无路障。

用户的输入数据放在文件“INMAZE”中，可通过修改INMAZE文件的内容，随机地设置迷宫中的路障。

	I	J
1	0	1
2	1	1
3	1	0
4	1	-1
5	0	-1
6	-1	-1
7	-1	0
8	-1	1

图 43-3

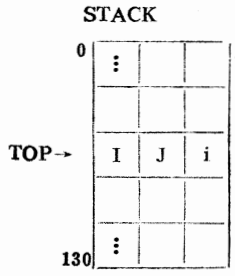


图 43-4

程序中的主要变量的意义如下：

G：下一次要试探的位置的行号；

H：下一次要试探的位置的列号；

V：下一次要试探的方向；

I：当前位置所在的行；

J：当前位置所在的列；

TOP：栈顶指针。

当用户输入的迷宫中路障设置如下所示（即文件INMAZE）时：

```
0 0 1 1 1 0 0 1 0 1 0 1 0 1 0
1 1 0 1 0 1 1 0 1 0 1 0 0 1 1
1 0 1 0 1 1 0 1 1 0 0 1 0 0 1
1 1 0 0 1 0 1 1 0 1 1 0 1 1 0
1 0 1 1 0 0 1 0 1 0 1 1 0 1 0
```

源程序运行后，形象地打印出所走路径正确的结果（见图43-6）。

2）框图　见图43-5

3）源程序及运行结果

```
C***  MAZE  PROGRAM
      INTEGER  MAZE(7, 17), MARK(7, 17)
      INTEGER  MOVE(8, 2), STACK(130, 3)
      CHARACTER  MA (21, 61)
      INTEGER  I, J, V, G, H, TOP
C***  INITIAT  MOVE, ALL  1'S  AROUND  MOVE
      DATA  MOVE (1, 1), MOVE (2, 1), MOVE (3, 1), MOVE (4, 1),
         MOVE (5, 1), MOVE (6, 1),
     1   MOVE (7, 1), MOVE (8, 1), MOVE (1, 2), MOVE (2, 2), MOVE (3, 2) ,
         MOVE (4, 2),
     2   MOVE (5, 2), MOVE (6, 2), MOVE (7, 2), MOVE (8, 2) /0, 3*1, 0.
         3*-1, 1, 1, 0, 3*-1, 0, 1/
```

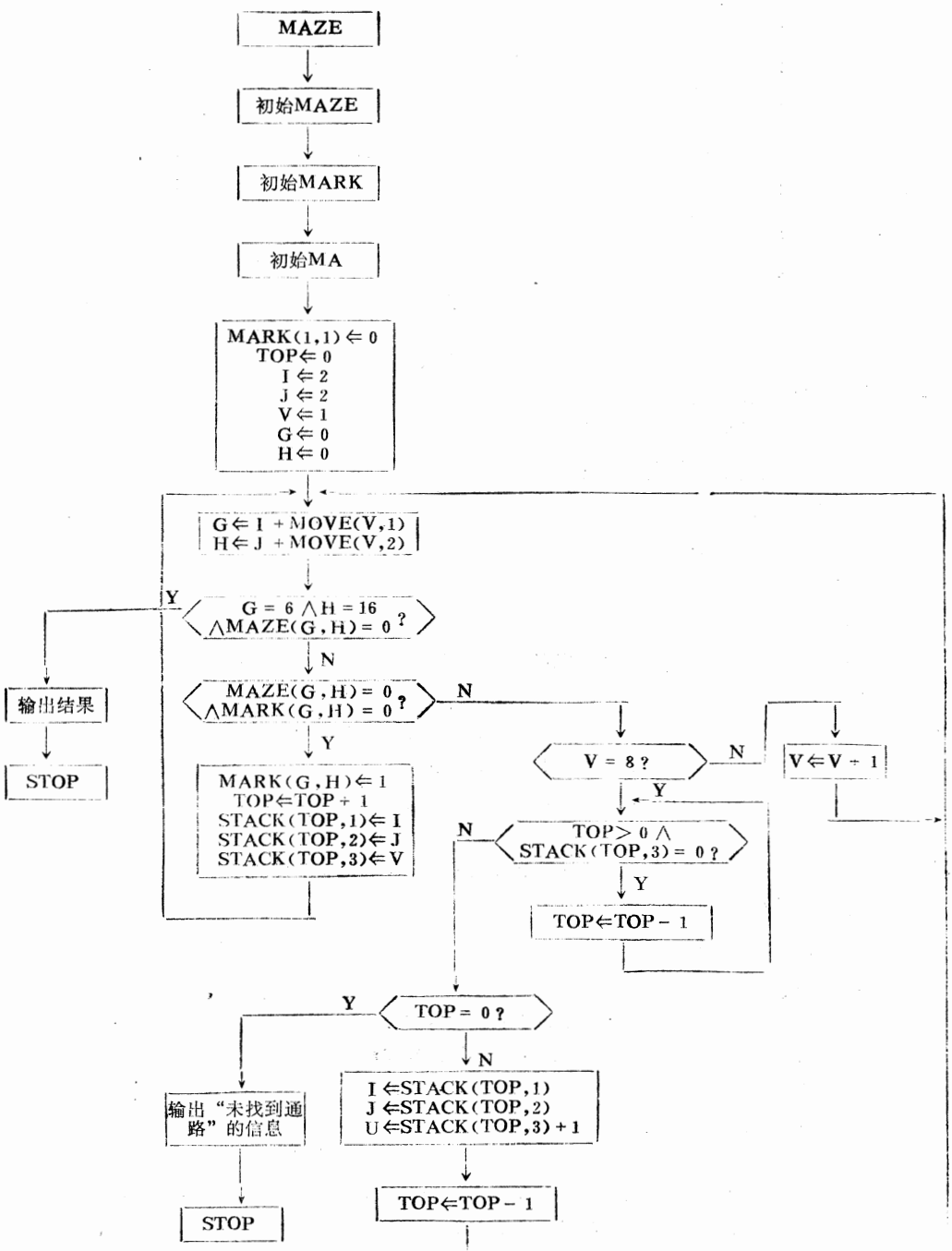

MAZE
初始MAZE
初始MARK
初始MA
MARK(1,1)⇐0
TOP⇐0
I⇐2
J⇐2
V⇐1
G⇐0
H⇐0
G⇐I+MOVE(V,1)
H⇐J+MOVE(V,2)
Y
G=6∧H=16
∧MAZE(G,H)=0?
N
输出结果
STOP
MAZE(G,H)=0
∧MARK(G,H)=0?
N
Y
MARK(G,H)⇐1
TOP⇐TOP+1
STACK(TOP,1)⇐I
STACK(TOP,2)⇐J
STACK(TOP,3)⇐V
V=8?
N
V⇐V+1
Y
N
TOP>0∧
STACK(TOP,3)=0?
Y
TOP⇐TOP-1
Y
TOP=0?
N
输出“未找到通路”的信息
STOP
I⇐STACK(TOP,1)
J⇐STACK(TOP,2)
U⇐STACK(TOP,3)+1
TOP⇐TOP-1

图 43-5

```
      DO  5  I=1, 21
      DO  5  J=1, 61
5     MA(I, J)=' '
      DO  10  I=1, 17
      MAZE(1, I)=1
      MAZE(7, I)=1
      MARK(1, I)=1
10    MARK(7, I)=1
      DO  15  I=1, 7
      MAZE(I, 1)=1
      MAZE(I, 17)=1
      MARK(I, 1)=1
15    MARK(I, 17)=1
      DO  20  I=2, 6
      DO  20  J=2, 16
20    MARK(I, J)=0
C***  INITIATE  MA
      DO  25  J=1, 61, 1
      DO  25  I=1, 21, 4
25    MA(I, J)='-'
      DO  30  I=2, 20, 1
      DO  30  J=1, 61, 4
30    MA(I, J)='I'
      DO  35  I=5, 17, 4
      DO  35  J=5, 57, 4
35    MA(I, J)='+'
```

初始MAZE, MARK和MA

```
C***  INPUT  INITIATED  DATA  OF  MAZE
      OPEN (4, FILE='MAP', ACCESS='WRITE')
      OPEN (5, FILE='INMAZE', ACCESS='READ')
      READ(5, 60) ((MAZE(I, J), J=2, 16), I=2, 6)
60    FORMAT (15I1)
      DO  40  I=2, 6
      DO  40  J=2, 16
      IF(MAZE(I, J). NE. 1) GO  TO  40
      MA((I-1)*4-1, (J-1)*4-1)='X'
40    CONTINUE
C***  END  OF  INITIATION
C***  STARTING  SEARCH  FROM  FIRST  POSITION
      MARK(2, 2)=1
      TOP=0
      I = 2
      J = 2
      V = 1
      G = 0
      H = 0
```

读入用户给出的迷宫设置

从第一个位置开始寻找，

```
C***  SEARCHING  FOR  NEXT
70    G = I + MOVE(V, 1)
      H = J + MOVE(V, 2)
      IF (G. EQ. 6. AND. H. EQ. 16. AND. MAZE(G, H) . EQ. 0) GO TO 130
      IF (MAZE(G, H). EQ. 0. AND. MARK(G, H). EQ. 0) GO  TO  90
      IF (V. EQ. 8) GO  TO  85
      V = V + 1                                          此路不通，换一个方向
      GO  TO  70
80    TOP = TOP - 1
85    IF (TOP. GT. 0. AND. STACK(TOP, 3). EQ. 8) GO  TO  80
      IF (TOP. EQ. 0) GO  TO  110
      I = STACK(TOP, 1)
      J = STACK(TOP, 2)                                  退回一步，并在该位 置的未
      V = STACK(TOP, 3) +1                               探索过的方向起查找
      TOP = TOP - 1
      GO  TO  70
C***  FOUND  A  POSIBLE  WAY  TO  MOVE  AHEAD
90    MARK (G, H) = 1
      TOP = TOP + 1
      STACK(TOP, 1) = I
      STACK(TOP, 2) = J                                  该方向可行，进栈，并下一步
      STACK(TOP, 3) = V                                  的探寻方向定为 1
      I = G
      J = H
      V = 1
      GO  TO  70
110   WRITE (1, 120)
120   FORMAT (3X, 14HNO  PATH  FOUND!)                   输出无路可通的信息
      GO  TO  320
C***  DRAW  THE  MAZE
130   MARK (G, H) = 1                                    以下是绘制迷宫的部分
      TOP = TOP + 1
      STACK(TOP, 1) = I
      STACK(TOP, 2) = J
      STACK(TOP, 3) = V
      DO  140  I = 1, 3
      MA(3, I) = 'a'
140   MA(19, 58 + I) = 'a'
150   DO  240  I = 1, TOP
      G = STACK(I, 1) - 1
      H = STAC(KI, 2) - 1
      GO  TO (160, 170, 180, 190, 200, 210, 220, 230), STACK (I, 3)
160   DO  165  J = H*4, H*4 + 3
```

```
165       MA(G*4 - 1, J) = 'a'
          GO TO 240
170       J = 0
          DO 175 V = G*4, G*4 + 3
          MA(V, H*4 + J) = 'a'
175       J = J + 1
          GO TO 240
180       DO 185 V = G*4, G*4 + 3
185       MA(V, H*4 - 1) = 'a'
          GO TO 240
190       J = 2
          DO 195 V = G*4, G*4 + 3
          MA(V, (H - 1) *4 + J) = 'a'
195       J = J - 1
          GO TO 240
200       DO 205 J = (H - 1) *4 - 1, H*4 - 2
205       MA(G*4 - 1, J) = 'a'
          GO TO 240
210       J = 0
          DO 215 V = (G - 1) *4 - 1, G*4 - 2
          MA(V, (H - 1) *4 - 1 + J) = 'a'
215       J = J + 1
          GO TO 240
220       DO 225 V = (G - 1) *4 - 1, G*4 - 2
225       MA(V, H*4 - 1) = 'a'
          GO TO 240
230       J = 3
          DO 235 V = (G - 1) *4 - 1, G*4 - 2
          MA(V, H*4 + J) = 'a'
235       J = J - 1
240       CONTINUE
C***      OUTPUT MAZE
          DO 310 I = 1, 21
          IF (I. EQ. 2) GO TO 270
          IF (I. EQ. 3) GO TO 280
          IF (I. EQ. 18) GO TO 290
          IF (I. EQ. 19) GO TO 300
          WRITE (4, 260) (MA (I, J), J = 1, 61)
260       FORMAT (13X, 61A1)
          GO TO 310
270       WRITE (4, 275) (MA(I, J), J = 1, 61)
275       FORMAT (4X, 5HENTER, 4X, 61A1)
          GO TO 310
```

```
280       WRITE (4, 285) (MA(I, J), J=1, 61)
285       FORMAT (10X, 3H⇒a, 61A1)
          GO TO 310
290       WRITE (4, 295) (MA(I, J), J=1,61)
295       FORMAT (13X, 61A1, 2X, 4HEXIT)
          GO TO 310
300       WRITE (4, 305) (MA(I, J), J=1, 61)
305       FORMAT (13X, 61A1, 3Ha⇒)
310       CONTINUE
320       STOP
          END
```

```
0 0 1 1 1 0 0 1 0 1 0 1 0 1 0
1 1 0 1 0 1 1 0 1 0 1 0 0 1 1
1 0 1 0 1 1 0 1 1 0 0 1 0 0 1
1 1 0 0 1 0 1 1 0 1 1 0 1 1 0
1 0 1 1 0 0 1 0 1 0 1 1 0 1 0
```

这是用户给出的一种迷宫设置

图43-6为本程序所找到的通路，便是用户给出的上述一种迷宫设置，程序运行打印结果。

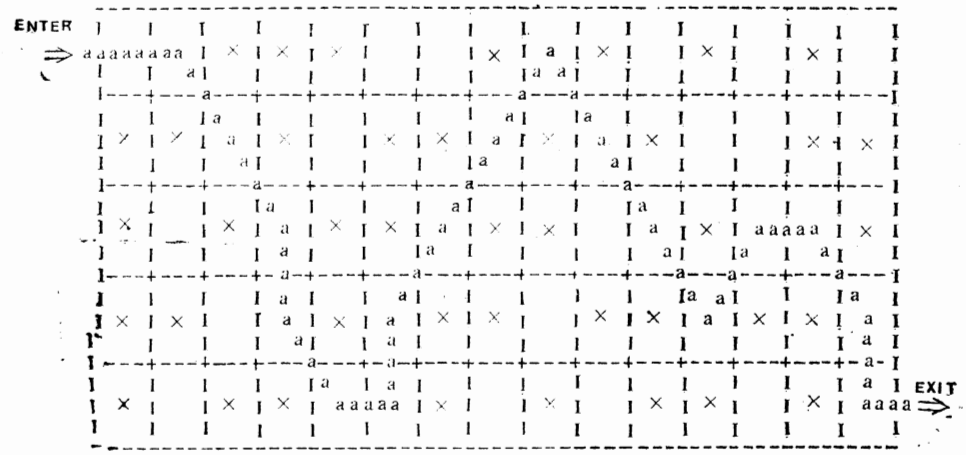


图 43-6

（三）思 考 题

该程序找出的途径不一定是最短的，应该怎样改进，以便在可能有几条路径的情况下找出一条最短的路途。

七 计算机会告诉你怎样度过年华

44 日历编排的规律

日历的编排是每400年一个大周期。如今年的月、日、星期几，和400年前的完全一样。那么请你计算出公元1901年至2300年这400年间每月的28号星期几的机会多？

（一）算法分析

现行天文历法根据天体运行的规律，取每年是365.2425天。这样，每400年共有365.2425×400天，如果以365天作为一年，每400年就少了0.2425×400＝97天。这97天要靠设置润年（每年366）天来凑齐，所以每400年中应该设置97个润年。按下列方法可确定哪年是润年：

（1）年份的末尾二位数能被4整除的年份是润年。如1972的72，1976的76，1980的80都能被4整除，所以这些年份是润年。

（2）年份末二位如果是00，那么，该年份数能被400整除的才是润年。

由此导出计算万年历的公式：

$$S = X - 1 + \frac{X-1}{4} - \frac{X-1}{100} + \frac{X-1}{400} + C。$$

式中：X为公元年数（如1984年，则X＝1984）；

C为从元旦起，到要算的那天总数（如1984年2月28日，C＝31＋28＝59），括号里31为1月份天数。若算3月份，要加2月份天数，若润年，2月份为29天，平年为28天。

S/7余数为星期几（如余数为0，1，2，……，6。分别为星期天，星期一，星期二，……，星期六）。

本题具体推算方法很多，可先算出1901年至2300年何年是润年，以润年为界，分段推算，累加起便知这400年间每月的28号星期一至星期日各占多少天，从而比较出星期几的天数最多。（留给读者详细推算。）

结果是：星期一，……，星期六，星期日，分别为687天，685天，685天，687天，684天，688天，684天。可见，星期六的天数最多，请看下述程序设计部分，在电子计算机上计算的结果。

（二）程序设计

1）设计思路

定义数组IS(7)，置放星期一至星期日，在外循环语句里，由1901（年）至2300（年）逐个输入，内循环是1～12月，先判该年是否润年，若润年，2月为29天，然后转子程序计算，该年每月的28号是星期几，置放在相应数组元素里，累加起来。出了外循环，打印出数组IS（7），再比较数组元素IS哪个大，将大者MAX打印出便是400年间每月28号星期几的机会多。

2）框图　见图44

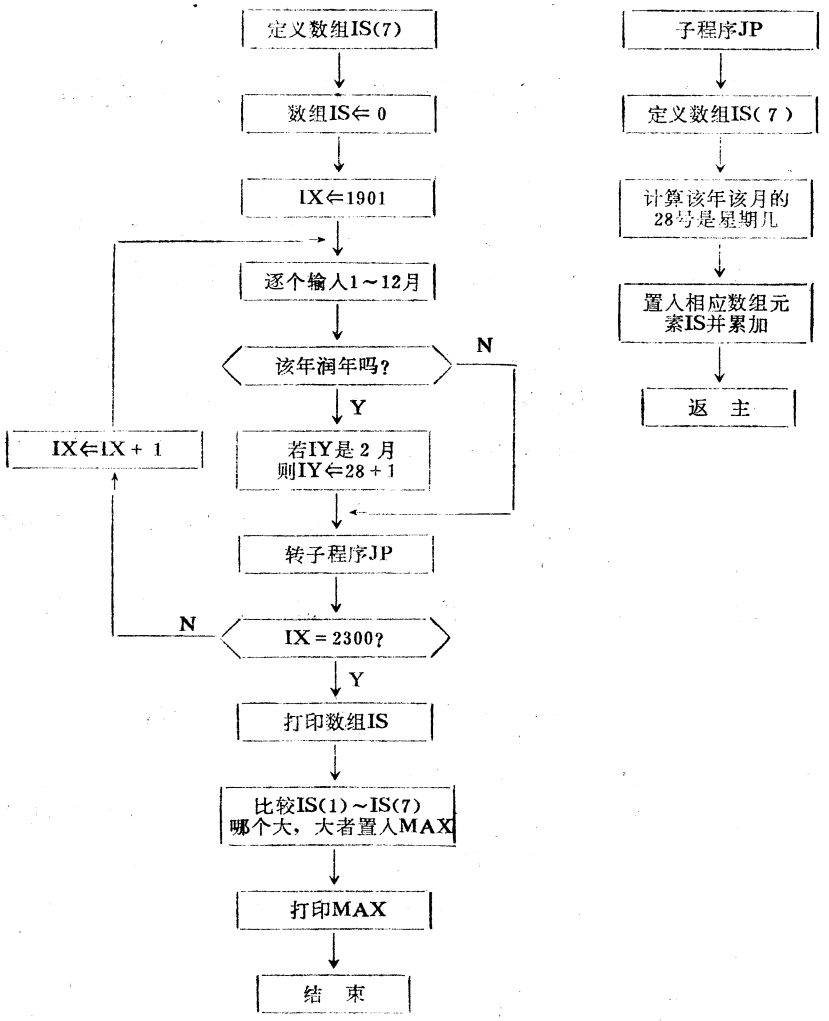


图 44

3）源程序及运行结果

```
      DIMENSION  IS(7)
      DO  10  J=1, 7
10    IS(J) =0
      IZ=28
      DO  30  IX=1901, 2300
      IC=IZ
      DO  30  IY=1, 12
      GOTO (101, 40, 50, 40, 60, 40, 60, 40, 40, 60, 40, 60, 40) IY
40    K=31
      GOTO  100
60    K=30
      GOTO  100
50    K=28
```

```
      IF (IX. NE. IX/4*4) GOTO  100
      IF (IX. NE. IX/100*100) GOTO  90
      IF (IX. NE. IX/100*400) GOTO  100
 90   K = K + 1
100   IC = IC+K
101   CALL  JP (IC, IX, IS)
 30   CONTINUE
      WRITE (10, 150) (J, IS (J) , J=1, 7)
150   FORMAT ((5X, 3HIS (, I1, 2H) =, I6))
      M = IS( 1 )
      J1 = 1
      DO  70  J = 2, 7
      IF (M. GE. IS(J)) GOTO  70
      J1 = J
      M = IS (J)
 70   CONTINUE
      WRITE (10, 80) J1, M
 80   FORMAT (5X, 7HMAX : KS (, I1, 2H) =, I6)
      STOP
      END
      SUBROUTINE  JP (IC, IX, IS)
      DIMENSION  IS( 7 )
      X = FLOAT (IX - 1)
      S = X +  (IX - 1) /4 -  (IX - 1) /100 +  (IX - 1) /400 + IC
      P = AMOD (S, 7. 0)
      M = IFIX (P)
      IF (M. EQ. 0) GOTO  7
      GOTO (1, 2, 3, 4, 5, 6) M
 1    IS( 1 ) = IS( 1 ) + 1
      GOTO  8
 2    IS( 2 ) = IS( 2 ) + 1
      GOTO  8
 3    IS( 3 ) = IS( 3 ) + 1
      GOTO  8
 4    IS( 4 ) = IS( 4 ) + 1
      GOTO  8
 5    IS( 5 ) = IS( 5 ) + 1
      GOTO  8
 6    IS( 6 ) = IS( 6 ) + 1
      GOTO  8
 7    IS( 7 ) = IS( 7 ) + 1
 8    RETURN
      END
```

以下是程序运行后打印结果

```
IS（1）=  687
IS（2）=  685
IS（3）=  685
IS（4）=  687
IS（5）=  684
IS（6）=  688
IS（7）=  684
MAX：KS(6）=  688
```

（三）思 考 题

请你编写程序打印出任何一年的日历。（详见《趣味程序设计100例》第79题。）

45 是否有每月此日都不是星期天

是否有哪一日，全年每月的此日都不是星期天。

（一）算 法 分 析

在平年，x日在各月的全年日序数被7除的余数依次等于下列各数被7除的余数：

〔x〕，（x＋3），〔x＋3〕，（x＋6），〔x＋1〕，（x＋4），〔x＋6〕，〔x＋2〕，（x＋5），〔x〕，（x＋3），〔x＋5〕。

不论x为1到30中的哪一个，其中余数0，1，……，6都出现了，所以当$1 \leqslant x \leqslant 30$时，不论该年元旦为星期几，总有某月的几号是星期天。当x＝31时，x号仅在大月出现，并且这些月x号在全年的日序数被7除时的余数，可由上列加方括号的各数被7除时得到，其中不会出现x＋4＝35被7除的余数为零。

（1）因此，在平年只要余数零对应星期天，即该年元旦是星期一，则各月31号都不是星期天。

（2）在润年，同样讨论可知，仅有31号，且仅在该年元旦为星期天时，各月31号都不是星期天。

（二）程 序 设 计

1）设计思路

我们据算法分析来推算20世纪这一百年，是否有哪几年，全年每月的31号都不是星期天。

定义数组Y(12)，置放12个月每月的天数，用DATA语句输入数组Y(12)时，调试的机器要求在有名公用区输入，令其名为YJI。外循环输入20世纪的每一年，转子程序计算该年元旦是星期几，返主后，判若是星期一，据算法分析（1），再判该年是平年吗，若是平年，则验算该年每月的31号都应不是星期天；返主后，判该年元旦若是星期天，据算法分析（2），再判该年是润年吗，若是润年，则验算该年每月31号都应不是星期天。上述二者，都要验算除31号外，该年总有某月某号为星期天。程序运行时每找到一年便打印出。结论请见打印结果。

2）框图 见图45

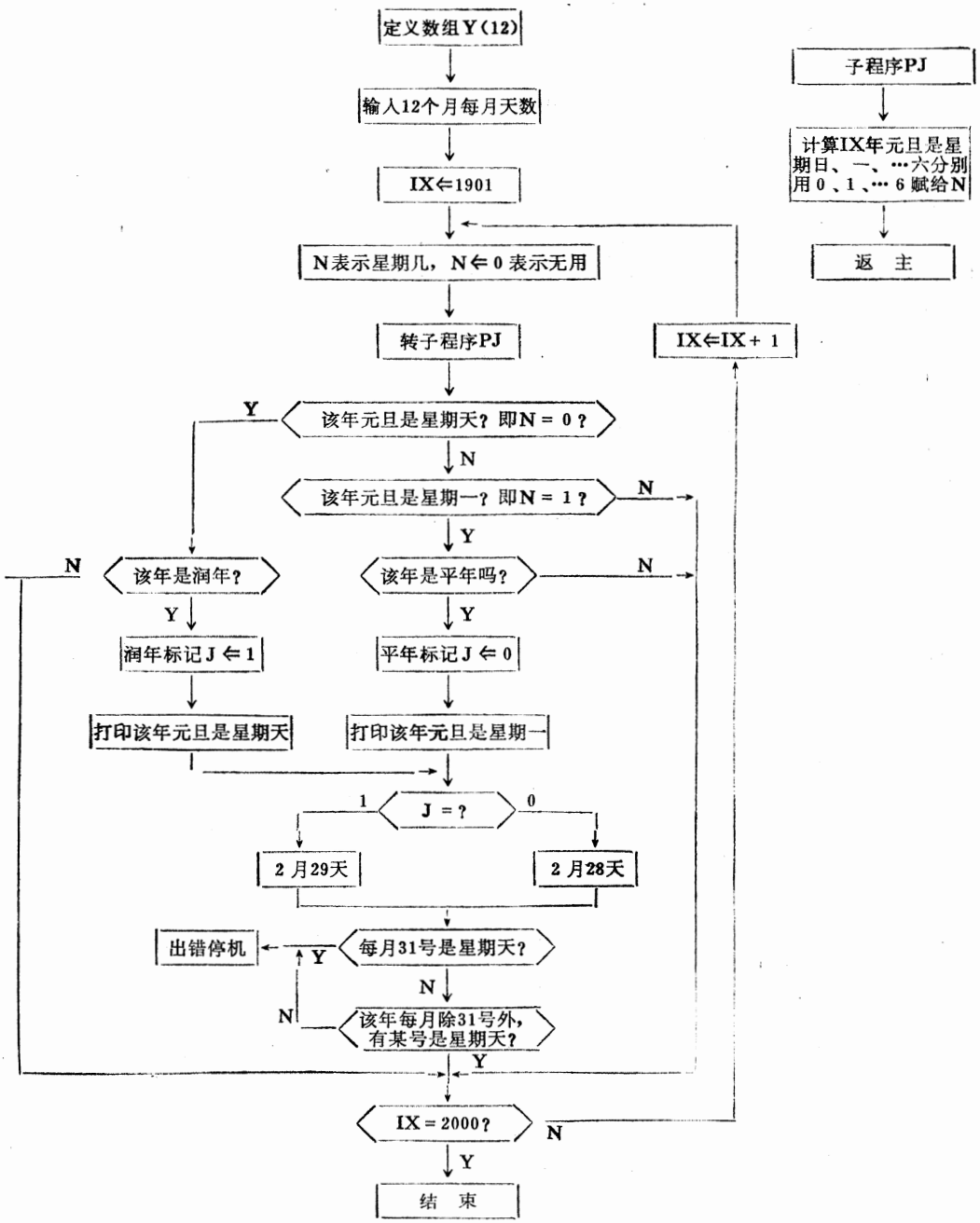


图 45

3）源程序及运行结果

```
DIMENSION  Y(12)
COMMON  /YJI/Y
DATA  Y/31. 0, 28. 0, 31. 0, 30. 0, 31. 0, 30. 0, 31. 0, 31. 0, 30. 0,
    31. 0, 30. 0, 31. 0/
DO  30  IX = 1901, 2000
```

```
      N = 10
      CALL  PJ (IX, 1, N)
      IF (N. EQ. 0) GOTO  40                    据算法分析（2）
      IF (N. EQ. 1) GOTO  50                    据算法分析（1）
      GOTO  30                                  元旦不是星期日也不是星期一
40    IF (IX. NE. IX/4*4) GOTO  30
      IF (IX. NE. IX/100*100) GOTO  60          判润年
      IF (IX. NE. IX/400*400) GOTO  30
60    J = 1                                     润年标记 J = 1
      WRITE (10, 70) IX
70    FORMAT (/5X, I4, 1H:, 24HNEW  YEAR'S  DAY  IS  SUNDAY)
      GOTO  100
50    IF (IX. NE. IX/4*4) GOTO  80
      IF (IX. NE. IX/100*100) GOTO  30          判润年
      IF (IX. NE. IX/400*400) GOTO  80
      GOTO  30
80    J = 0                                     平年标记 J = 0
      WRITE (10, 90) IX
90    FORMAT (/5X, I4, 1H:, 24HNEW  YEAR'S  DAY  IS  MONDAY)
100   IC = 0
      DO  120  K = 1, 12
      IC = IC + Y(K)
      IF (Y(K) . NE. 28. 0) GOTO  125           判该月Y是28天吗
      IF (J. NE. 1) GOTO  120                   判该年是润年吗
      IC = IC + 1
      GOTO  120
125   IF (Y(K) . NE. 31. 0) GOTO  120           只判该月31号是否是星期天
      N = 10
      CALL  PJ (IX, IC, N)
      IF (N. EQ. 0) STOP  'ERR1'                要判的年份某月31号是星期天，便 错。
      N = 10
      DO  200  I = 1, 30                        找到某年的每月31号都不是星期 天后，
      IC = I                                    再验证该年每月的1～30号总有某号
      CALL  PJ (IX, IC, N)                      是星期天。
      IF (N. EQ. 0) GOTO  120
200   CONTINUE
      STOP  'ERR2'
120   CONTINUE
      WRITE (10, 130)                           循环完了说明算法分析
130   FORMAT(5X,21HNOT  FIND 31 IS SUNDAY)      （1）和（2）正确
30    CONTINUE
      STOP
      END
```

```
SUBROUTINE  PJ (IX, IC, N)
S = FLOAT (IX-1)
S = S+(IX-1)/4-(IX-1)/100+(IX-1)/400+IC
P = AMOD(S, 7. 0)
N = IFIX(P)
RETURN
END
```

该子程序是计算IX年的元旦是星期几

```
1906 : NEW  YEAR'S  DAY IS MONDAY
NOT  FIND  31  IS  SUNDAY
```

1906年元旦是星期一，该年每月31号没有星期天

```
1917 : NEW  YEAR'S  DAY IS MONDAY
NOT  FIND  31  IS  SUNDAY
```

1917年元旦是星期一，该年每月31号没有星期天

```
1923 : NEW  YEAR'S  DAY IS MONDAY
NOT  FIND  31  IS  SUNDAY
```

1923年元旦是星期一，该年每月31号没有星期天

```
1928 : NEW  YEAR'S  DAY IS SUNDAY
NOT  FINO  31  IS  SUNDAY
```

1928年元旦是星期日，该年每月31号没有星期天

```
1934 : NEW  YEAR'S  DAY IS MONDAY
NOT  FIND  31  IS  SUNDAY
```

1934年元旦是星期一，该年每月31号没有星期天

```
1945 : NEW  YEAR'S  DAY IS MONDAY
NOT  FIND  31  IS  SUNDAY
```

1945年元旦是星期一，该年每月31号没有星期天

以下类同标注，省略

```
1951 : NEW  YEAR'S  DAY IS MONDAY
NOT  FIND  31  IS  SUNDAY
1956 : NEW  YEAR'S  DAY IS SUNDAY
NOT  FIND  31  IS  SUNDAY
1963 : NEW  YEAR'S  DAY IS MONDAY
NOT  FIND  31  IS  SUNDAY
1973 : NEW  YEAR'S DAY IS MONDAY
NOT  FIND  31  IS  SUNDAY
1979 : NEW  YEAR'S  DAY IS MONDAY
NOT  FIND  31  IS  SUNDAY
1984 : NEW  YEAR'S  DAY IS SUNDAY
NOT  FIND  31  IS  SUNDAY
1990 : NEW  YEAR'S  DAY IS MONDAY
NOT  FIND  31  IS  SUNDAY
STOP
```

（三）思 考 题

1）用其它算法来验证算法分析（1）和（2）是否正确。

2）对照主题程序运行后打印结果，你能找出20世纪这一百年里，还有哪一年的每月此日都不是星期天被丢掉了。

46 元旦遇到星期六还是星期天的机会多

试问阳历新年（元旦）较常碰到星期六还是星期天？

（一）算法分析和结论

如何计算请见44题，按此法计算1900～2300年这四百年，元旦时遇星期一为56天，星期二为58天，星期三为57天，星期四为57天，星期五为58天，星期六为56天，星期日为58天。可见，元旦遇到星期天比星期六的机会多。

（二）程 序 设 计

1）设计思路

定义数组 I S(7)，置放星期一至星期日。利用循环语句，控制变量1901（年）至2300（年），在循环体里转子程序计算每年的元旦是星期几，并累加之。出循环体便将这四百年元旦时遇到的星期一至星期天全打印出，最后比较星期六的天数多还是星期天的天数多，结论便出。

2）框图 见图46

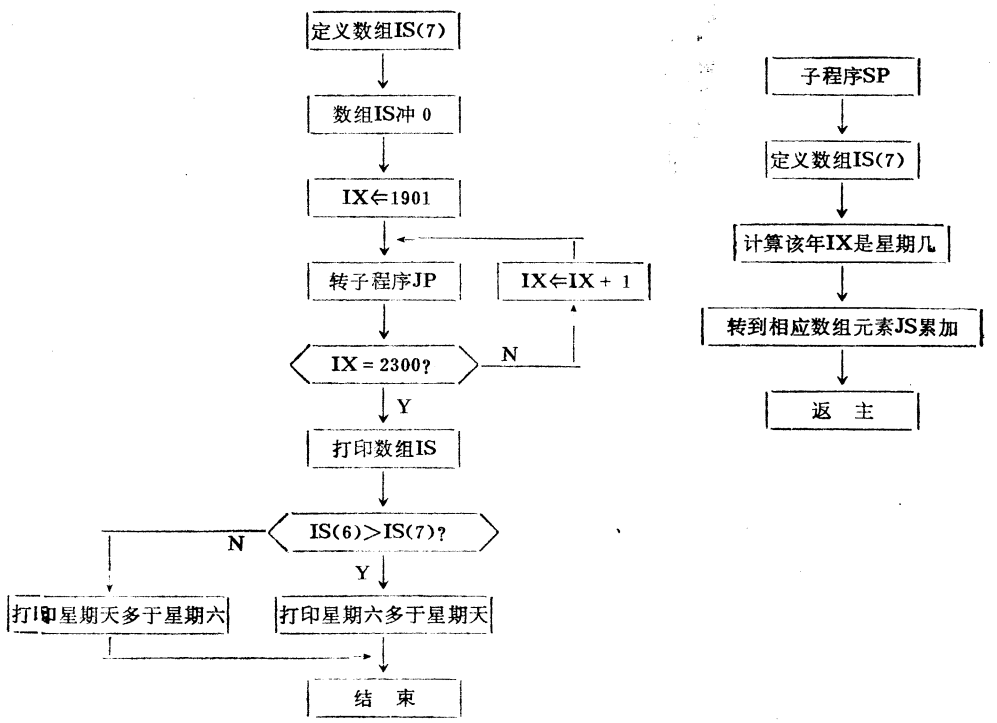


图 46

3）源程序及运行结果

```
      DIMENSION  IS(7)
      DO  10  J=1, 7
10    IS(J) =0
      DO  30  IX=1901, 2300
```

```
      CALL  JP (1, IX, IS)
30    CONTINUE
      WRITE (10, 150)  (J, IS (J) , J=1, 7)
150   FORMAT ((5X, 3HIS (, I1, 2H) =, I4))
      IF (IS(7). GT. IS(6)) GOTO  50
      WRITE (10, 40)
40    FORMAT (5X, 43 HNEW  YEAR'S  DAY  IS  MORE  SATURDAY  THAN
        SUNDAY)
      STOP
50    WRITE (10, 60)
60    FORMAT (5X, 43HNEW  YEAR'S  DAY  IS  MORE  SUNDAY  THAN
        SATURDAY)
      STOP
      END
      SUBROUTINE  JP (IC, IX, IS)
      DIMENSION  IS ( 7 )
      X = FLOAT (IX - 1)
      S = X + (IX - 1) /4 - (IX - 1) /100 + (IX - 1) /400 + IC
      P = AMOD (S, 7. 0)
      M = IFIX (P)
      IF (M. EQ. 0) GOTO  7
      IF (M. EQ. 1) GOTO  1
      IF (M. EQ. 2) GOTO  2
      IF (M. EQ. 3) GOTO  3
      IF (M. EQ. 4) GOTO  4
      IF (M. EQ. 5) GOTO  5
      IF (M. EQ. 6) GOTO  6
      GOTO  8
1     IS( 1 ) = IS( 1 ) + 1
      GOTO  8
2     IS( 2 ) = IS( 2 ) + 1
      GOTO  8
3     IS( 3 ) = IS( 3 ) + 1
      GOTO  8
4     IS( 4 ) = IS( 4 ) + 1
      GOTO  8
5     IS( 5 ) = IS( 5 ) + 1
      GOTO  8
6     IS( 6 ) = IS( 6 ) + 1
      GOTO  8
7     IS( 7 ) = IS( 7 ) + 1
8     RETURN
      END
```

IS（1）=56
IS（2）=58
IS（3）=57
IS（4）=57
IS（5）=58
IS（6）=56
IS（7）=58
NEW YEAR'S DAY IS MORE SUNDAY THAN SATURDAY

（元旦遇星期日比星期六机会多）

（三）思 考 题

请推论公元1年的元旦是星期几？

47 老人今年几何

吃西瓜时小明看见爷爷没有牙吃的还挺快，便问爷爷今年多大年岁了，爷爷风趣地说，你算算吧，我出生第二年过了个生日，如今过了20个生日，算完了告诉我，我今年多大年岁了，是哪年哪月哪日生的？

（一）笔算步骤和结果

老爷爷牙都掉了，肯定不会是20岁，他出生的年份一定是润年的2月29日，即出生在“润日”，则月、日已知，剩下的问题就要计算从今年（1983）往回推算第20个润年是哪一年。

怎样快速推算润年，一般采用方法是：凡是公元年数能被4除尽，或公元年数末二位数逢百且能被400除尽的，就是润年。你能否想出一种简便、快速的方法，不用计算，只要从年份的个位数和十位数或者百位数和千位数来推算出是否是润年吗？

阳历有润日（即2月为29天）的年份为润年，我们根据历法置润日知道：每400年共加97个润日，如果每4年置一个润日，400年中就有润日100天，与实际情况多了3天，因此，又在逢百之年不置润日，但最后一个逢百之年（400年，800年，1200年……）置润日，这样，就和实际情况一致了。

简便的计算方法是：凡十位数是单数，而个位数是2或6时，即为润年；十位数是双数，而个位数是0、4、8时也是润年，这是因为公元年数被4除，除到十位数时，如果十位数是单数时，它的余数也是单数，比4小的单数只有1和3，个位数为2或6（即12、16、32、36，…）时，正好能被4除尽；如果十位数是双数，它的余数也是双数，比4小的双数只有2，个位数为0、4、8（即20、24、28…）时，也正好能被4除尽，例如1980年，十位数是8是双数，个位数是0，所以是润年。

逢百之年，可将末两位的两个零去掉，同样可以采用上面方法来推算润年，例如2000年，去掉末尾两个零剩下20，十位数2是双数，个位数是零，正好是润年，这样，只要记住“0、4、8、2、6”就可以推算润年，这五个数字中前三个数字是十位数为双数时推算润年用；后两个数字是十位数为单数时推算润年用。

据上述推算润年方法，从1983年倒退至20个润年为1905年。老爷爷是1905年2月29日出生的，一般吃西瓜时是在夏、秋季节，是过了2月份，因此老爷爷今年已经78岁了。

（二）程 序 设 计

1）设计思路

从今年（1983年）往回倒推20个润年，再加1便是出生年份。前已介绍推算润年的方法，在程序中判断如下。（1）年份能被4除尽，且不能被100除尽的，为润年。（2）能被4又能被100除尽者，但不能被400除尽，也不是润年。（3）既能被4和100除尽，又能被400除尽的才为润年。（4）当然，若首先就不能被4除尽的，肯定不是润年，也不必再判。

设J是生日数，IX是年。利用循环语句倒推至循环终止变量110年，利用上述条件判断，一旦满足条件，便退出循环，打印出生年份。

2）框图　见图47

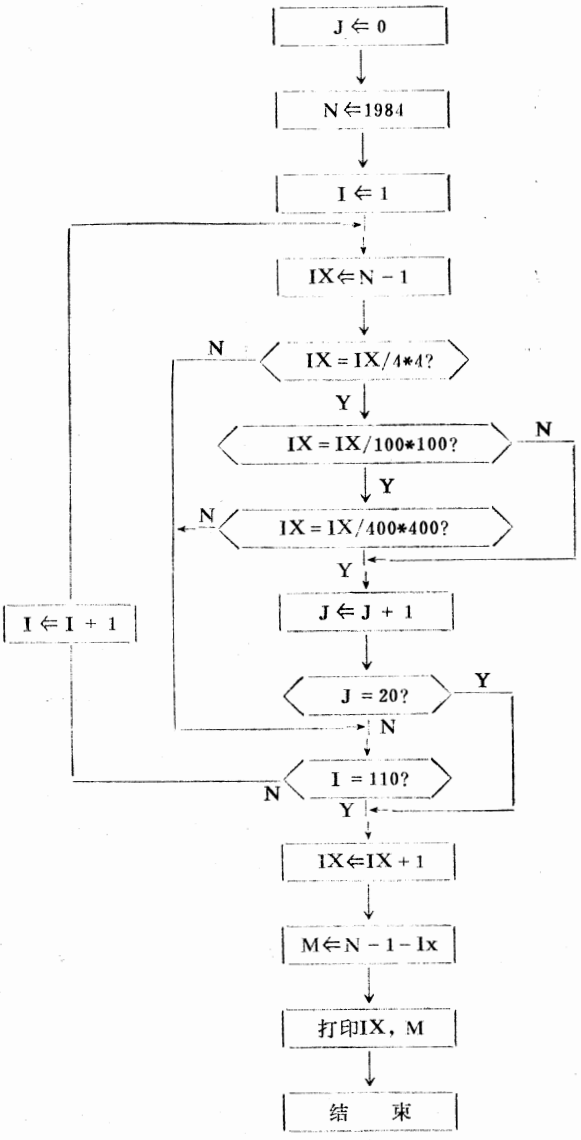


图 47

3）源程序及运行结果

```
      J = 0
      N = 1984
      DO  10  I = 1,110
      IX = N - I
      IF (IX, NE, IX/4*4) GOTO  10
      IF (IX, NE, IX/100*100) GOTO  15
      IF (IX, NE, IX/400*400) GOTO  10
15    J = J + 1
      IF (J. EQ. 20) GOTO  20
10    CONTINUE
      STOP  'ERR'                    20语句是出生的第二年才能过生日，故往回
                                     再倒推一年，才是出生年。
20    IX = IX + 1
      M = N - 1 - IX
      WRITE (10, 30) IX, M
30    FORMAT (10X, I4, 8H- - -1983 :, I2)
      STOP
      END
                                     程序运行打印结果
      1905- - -1983 : 78             1905年生，到1983年为78岁。
```

（三）思 考 题

猜年龄和出生月份——用2乘你出生的月份，再加上5，再乘50，再加上你的年龄，再减去365，然后把最后的得数告诉我，我就知道你今年是几岁，是在哪个月生的。例如，你告诉我是199，我就知道你今年是14岁，是在3月份生的。这是什么道理，你知道吗？请你编写程序验证是否对。

48 计算机会告诉你怎样度过年华

人们总是对未来有着美好的计划和愿望，很少去仔细算算往日的时刻是怎样度过的。本程序将要告诉你，从你诞生后至今度过了多少岁月，在这些宝贵的岁月里，你睡眠共用去了多少年、月、日；你吃饭共用去了多少年、月、日；你学习、工作、玩耍共占用了多少年、月、日；你正常的休息用去了多少年、月、日。当你看到这些数字时，你会感到惊讶和感叹。珍惜你未来的年华多做贡献吧！计算机最后告诉你将在哪一年退休。

（一）算 法 分 析

本程序采用的日历算法介绍省略。程序中各种算法可以计算出题目中要求统计的各项数字。

（二）程 序 设 计

1）设计思路

思路可由程序中旁注窥察出。以例子来说明操作。首先从键盘上输入今天的日期和你出生的日期，如5，10，1943，表示1943年5月10日，计算机马上会告诉你这一天是星期

儿。如果你正好是若干年前的今天诞生的，计算机还会打印出“HAPPY BIRTHDAY”（生日快乐），表示今天恰好是你的生日，向你表示祝贺。同时会告诉你从你诞生到今天，已经过了多少年、月、天（统一按一年365天、一月30天计算）。例如，1943年5月10日出生的那一天是星期四，到程序运行的那天（1983年5月13日）已过了40年零3天。其中睡了整整13年4个月零2天(每天按8小时计算)，用于吃饭时间4年（以每天有1/10时间即2.4小时用于吃三餐饭和饭前、后的准备和收拾整理），用于学习、工作和学前玩的时间共为13年4个月零2天（以每天8小时计算），用于休息的时间（即以上各项剩余下的时间）为9年3个月零2天。最后计算出你将在2003年退休（以60岁为退休年龄）。此例请看程序运行后打印结果。

如果是未入学的儿童，则不会打印出“工作和学习”的时间。假设5岁多以前的儿童是没上学的，白天是玩耍的，则打印出“YOU HAVE PLAYED”，指出“你用于玩的时间”为多少。

16岁以下的少年一般是未参加工作的学生，所以在你的结果信息中打印出“YOU HAVE PLAYED AND STUDIED”（你用来玩耍和学习的时间）。

程序中，920语句“K 5 > 3？”的含意是：判断出生以来的总日数减去睡眠和吃饭时间，如果小于或等于3（年），则此人必是不到5岁半的儿童，它不应有“学习和工作”的内容，（吃饭加睡眠共占43.33%，如果余下的56.67%是3年，则总年数应为 $\frac{3}{0.5666}=5.3$岁）。950语句的“K 5 > 9”的含意是：判断此人是否在16岁以下（因 $\frac{9}{0.5666}=16$），如果K 5 < 9，则表示未超过16岁，不应有“工作”的内容。

程序中有些语句是专门用来处理年、月、日的数字的，使月数不应小于零、大于12，日数大于0。

程序举例四个，40、26、17、6岁各一个例子，他们如何度过年华，请见程序运行后打印结果。

2）框图　见图48

3）源程序及运行结果

```
      DIMENSION  T(12)
      REAL  F, A8
      INTEGER  I5, I6, I7, K5, K6, K7 (分别表示年、月、日)
      FNA(A) =INT (A/4, 0)
      FNB(A) =INT (A/7, 0)
      WRITE (6, 10)
10    FORMAT (1X, 26X, 7HWEEKDAY)
      WRITE (6, 20)
20    FORMAT (1X, 20X, 18HTHE  COMPUTER  WORLD)
      WRITE (6, 30)
30    FORMAT (1X, 41HENTER  TODAY'S DATE  IN  THE  FORM  4, 18,
        1983? )          (你从这个日、月、年计算吗)
      READ (5, 35) M1, D1, Y1          (从键盘上输入今天的日、月、年)
```

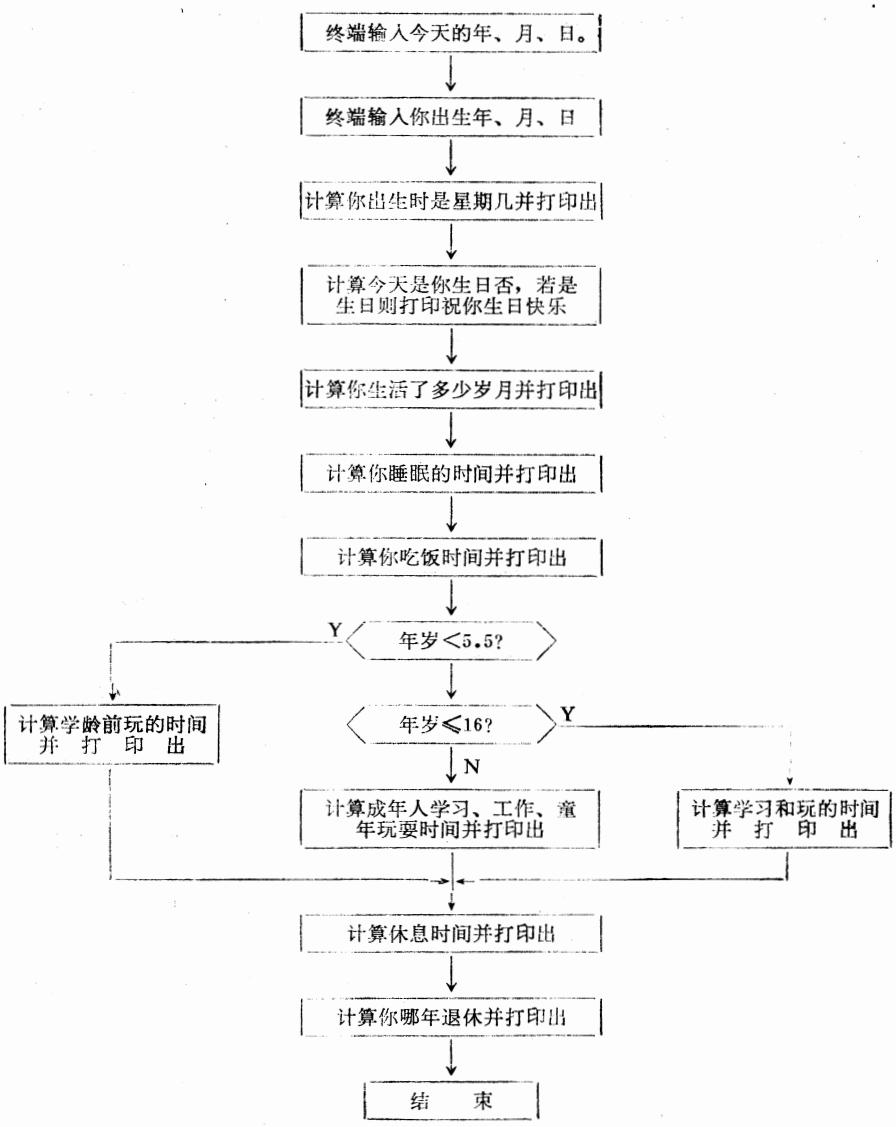


图 48

```
35      FORMAT (3F4.0)
        T(1) = 0.0
        T(2) = 3.0
        T(3) = 3.0
        T(4) = 6.0
        T(5) = 1.0
        T(6) = 4.0
        T(7) = 6.0
        T(8) = 2.0
        T(9) = 5.0
        T(10) = 0.0
```

```
      T (11) = 3.0
      T (12) = 5.0
      WRITE (6, 40)
40    FORMAT (1X, 46HENTER DAY OF BIRTH (OR OTHER DAY OF
        INTEREST) ? )
      READ (5, 35) M, D, Y          (从键盘上输入你的生日或你感兴趣的日、月、年)
      I1 = INT ((Y - 1500.0) /100.0)
      IF (Y - 1582. 0. LT.0.0) G0 T0 1100
      A = I1*5 + (I1 + 3) /4
      I2 = INT (A - FNB (A) *7.0)
      Y2 = INT (Y/100.0)
      Y3 = INT (Y - Y2*100,0)
      A = Y3/4.0 + Y3 + D + T (M) + FLOAT (I2)
      B = INT (A - FNB (A) *7.0) +1
      IF (M.GT.2) GO TO 370
      IF (Y3.EQ.0.0) GO TO 340
      TI = INT (Y - FNA (Y) *4.0)
      IF (T1.NE.0.0) GO TO 370
300   IF (B.NE.0.0) GO TO 320
      B = 6
320   B = B - 1.0
      GO TO 370
340   A = I1 - 1.0
      T1 = INT (A - FNA (A) *4.0)
      IF (TI.EQ.0.0) GO TO 300
370   IF (B.NE.0.0) GO TO 390
      B = 7
390   IF((Y1*12.0 + FLOAT(M1))*31.0 + DI.LT. (Y*12.0 + M)*31 + D)GO TO 450
      IF((YI*12.0 + FLOAT (MI)) *31 + DI.EQ. ( Y*12 + M)*31 + D) GO TO 430
C     ********************************** **
      N = B
      ID = D
      IY = Y
      IF (N, LT, 1, OR, N, GT, 7) GO TO 610
      GO TO (1, 2, 3, 4, 5, 6, 7) , N      N是1～7分别表示生日那天是星期日～星
1     WRITE (6, 80) M, ID, IY              期六，转去处理。

80    FORMAT (1X, I2, H/, 12, H/, I4, 14H/ WAS A SUNDAY)
        GO TO 610
2     WRITE (6, 82) M, ID, IY
82    FORMAT (1X, I2, H/, I2, H/, I4, 14H/ WAS A MONDAY)
      GO TO 610
3     WRITE (6, 84) M, ID, IY
```

```
84      FORMAT (1X, I2, H/, I2, H/, I4, 15H/  WAS  A  TUESDAY)
        GO  TO  610
4       WRITE (6, 86) M, ID, IY
86      FORMAT (1X, I2, H/, I2, H/, I4, 17H/  WAS  A  WEDNESDAY)
        GO  TO  610
5       WRITE (6, 88) M, ID, IY
88      FORMAT (1X, I2, H/, I2, H/, I4, 16H/  WAS  A  THURSDAY)
        GO  TO  610
6       WRITE (6, 90) M, ID, IY
90      FORMAT (1X, I2, H/, I2, H/, I4, 14H/  WAS  A  FRIDAY)
        GO TO  610
7       WRITE (6, 92) M, ID, IY
92      FORMAT (1X, I2, H/, I2, H/, I4, 16H/  WAS  A  SATURDAY)
        GO  TO  610
430     N = B
        ID = D
        IY = Y
        IF (N.LT.1.OR.N.GT.7) GO  TO  610
        GO  TO  (11, 22, 33, 44, 55, 66, 77) , N   N是1～7分别表示生日那天 是星期日
                                                   ～星期六，转去处理。
11      WRITE (6, 94) M, ID, IY
94      FORMAT (1X, I2, H/, I2, H/, I4, 13H/  IS  A  SUNDAY)
        GO  TO  610
22      WRITE (6, 96) M, ID, IY
96      FORMAT (1X, I2, H/, I2, H/, I4, 13H/  IS  A  MONDAY)
        GO  TO  610
33      WRITE (6, 98) M, ID, IY
98      FORMAT (1X, I2, H/, I2, H/, I4, 14H/  IS  A  TUESDAY)
        GO  TO  610
44      WRITE (6. 100) M, ID, IY
100     FORMAT (1X. I2. H/, I2, H/, I4, 16H/  IS  A  WEDNESDAY)
        GO  TO  610
55      WRITE (6, 102) M, ID, IY
102     FORMAT (1X. I2. H/, I2, H/, I4, 15H/  IS  A  THURSDAY)
        GO  TO  610
66      WRITE (6, 104) M, ID, IY
104     FORMAT (1X, I2, H/, I2, H/, I4, 13H/  IS  A  FRIDAY)
        GO  TO  610
77      WRITE (6, 106) M, ID, IY
106     FORMAT (1X, I2, H/, I2, H/, I4, 15H/  IS  A  SATURDAY)
        GO  TO  610
450     N = B
        ID = D
```

```
      IY=Y
      IF (N.LT.1.OR.N.GT.7) GO TO 610
      GO TO (111, 222, 333, 444, 555, 666, 777) , N
111   WRITE (6, 108) M, ID, IY
108   FORMAT (1X, I2, H/, I2, H/, I4, 16H/ WILL BE SUNDAY)
      GO TO 610
222   WRITE (6, 110) M, ID, IY
110   FORMAT (1X, I2, H/, I2, H/, I4, 16H/ WILL BE MONDAY)
      GO TO 610
333   WRITE (6, 112) M, ID, IY
112   FORMAT (1X, I2, H/, I2, H/, I4, 17H/ WILL BE TUESDAY)
      GO TO 610
444   WRITE(6, 114) M, ID, IY
114   FORMAT (1X, I2, H/, I2, H/, I4, 19H/ WILL BE WEDNESDAY)
      GO TO 610
555   WRITE (6, 116) M, ID, IY
116   FORMAT (1X, I2, H/, I2, H/, I4, 18H/ WILL BE THURSDAY)
      GO TO 610
666   WRITE(6, 118) M, ID, IY
118   FROMAT (1X, I2, H/, I2, H/, I4, 16H/ WILL BE FRIDAY)
      GO TO 610
777   WRITE (6, 120) M, ID, IY
120   FORMAT (1X, I2, H/, I2, H/, I4, 18H/WILL BE SATURDAY)
610   IF ( (Y1*12.0+FLOAT (M1) ) *31.0+D1.EQ.(Y+12+M) *31+D) GO
        TO 1100
      I5=Y1-Y
      WRITE (6, 1500)
1500  FORMAT (1X, /)
      I6=M1-M
      I7=D1-D
      IF (I7.GE.0) GO TO 690
      I6=I6-1
      I7=I7+30
690   IF (I6.GE.0) GO TO 720
      I5=I5-1
      I6=I6+12
720   IF (I5.LT.0) GO TO 1100
      IF (I7.NE.0) GO TO 760
      IF (I6.NE.0) GO TO 760
      WRITE (6, 1510)
1510  FORMAT (1X, 20H***HAPPY BIRTHDAY***)
760   WRITE (6, 1520)
1520  FORMAT (35X, 5HYEARS, 5X, 6HMONTHS, 4X,
```

判断年I5、月I6、日I7

计算生活了多长时间

```
      4HDAYS)
      WRITE (6, 1530) I5, I6, I7
1530  FORMAT (1X, 22HYOUR  AGE  IF
      BIRTHDATE, 13X, I2, 8X, I2, 8X, I2)
780   A8 = (I5*365) + (16*30) + I7 + I6/2
      K5 = I5
      K6 = I6
      K7 = I7
      E = Y + 60. 0
      F = 0.333333
      CALL  SUB (F, A8, I5, I6, I7, K5, K6, K7)
      WRITE (6, 1540) I5, I6, I7
1540  FORMAT (1X, 14HYOU  HAVE  SLEPT,
      21X, I2, 8X, I2, 8X, I2)
      F = 0. 1
      CALL  SUB (F, A8, I5, I6, I7, K5, K6, K7)
      WRITE (6, 1550) I5, I6, I7
1550  FORMAT (1X, 14HYOU  HAVE  EATEN,
        19X, I4, 6X, I4, 6X, I4)
      F = 0.333333
      IF (K5, GT, 3) GO  TO  950
      CALL  SUB (F, AB, I5, I6, I7, K5, K6, K7)
      WRITE (6, 1590) I5, I6, I7
1590  FORMAT (1X, 15HYOU  HAUE  PLAYED,
        18X, I4, 6X, I4, 6X, I4)
      GO  TO  1000
950   IF  (K5.GT.9) GO  TO  980
      CALL  SUB (F, A8, I5, I6, I7, K5,
        K6, K7)
      WRITE (6, 1600) I5, I6, I7
1600  FORMAT (1X, 27HYOU  HAVE  PLAYED
      ANDSTUDIED, 6X, I4, 6X,I4,6X, I4)
      GO  TO  1000
980   CALL  SUB (F, A8, I5, I6, I7, K5,
        K6, K7)
      WRITE (6, 1560) I5, I6, I7
1560  FORMAT (1X, 28HYOU  HAVE  STUDIED,
      WORKED, PLAY, 6X, I4, 6X, I4, 6X, I4)
1000  IF (K6.EQ.12) GO  TO  1020
      GO  TO  1040
1020  K5 = K5 + 1
      K6 = 0
1040  WRITE (6, 1570) K5, K6, K7
1570  FORMAT (1X, 17HYOU  HAVE  RELAXED,
        16X, I4, 6X, I4, 6X, I4, /)
```

将年、月、日暂存于K5、K6、K7

计算睡眠用的时间

计算吃饭用的时间

计算不到5岁半的儿童玩耍时间

计算5.5～16岁孩子玩耍和学习时间

计算成年人工作、学习、玩的时间

计算休息的时间

```
      N = E
      WRITE (6, 1580) N
1580  FORMAT (1X, 20X, 20H**YOU  MAY  RETIRE
      IN, I4, 2H**, //)
1100  STOP
      END
      SUBROUTINE  SUB (F, A8, I5, I6, I7,
      K5, K6, K7)          子程序SUB计算共用去的年、月、日。
      K1 = INT (F*A8)
      I5 = K1/365
      K1 = K1 - (I5*365)
      I6 = K1/30
      I7 = K1 - (I6*30)
      K5 = K5 - I5
      K6 = K6 - I6
      K7 = K7 - I7
      IF (K7, GE, 0) GO  TO  1270
      K7 = K7 + 30
      K6 = K6 - 1
1270  IF (K6. GE. 0) GO  TO  1300
      K6 = K6 + 12
      K5 = K5 - 1
1300  RETURN
      END
```

计算退休的年份

下面是程序举例

```
          WEEKDAY
     THE  COMPUTER  WORLD
ENTER  TODAY'S  DATE  IN  THE
  FORM  4, 18, 1983? 5, 13, 1983
ENTER  DAY  OF  BIRTH (OR
  OTHER DAY OF INTEREST)? 5,10,1943
5/10/1943/  WAS  A  MONDAY
```

举例：今天是1983年5月13日，输入你出生日期是1943年5月10日，计算机告诉你生日是星期天。

	YEARS	MONTHS	DAYS	
YOUR AGE IF BIRTHDATE	40	0	3	你已生活了40年零3天
YOU HAVE SLEPT	13	4	2	你睡眠用去13年4个月零2天
YOU HAVE EATEN	4	0	0	你吃饭用去4年
YOU HAVE STUDIED, WORKED, PLAY	13	4	2	你学习、工作、童年玩耍用去13年4个月零2天
YOU HAVE RELAXED	9	3	29	你休息用去9年3个月零29天

```
**  YOU  MAY  RETIRE  IN  2003**                    你将在2003年退休
                WEEKDAY
        THE COMPUTER  WURLD
ENTER  TODAY'S  DATE  IN  THE  FORM
  4, 18, 1983? 9  07, 1983                          举例：今天是1983年9月7日
ENTER  DAY  OF  BIRTH (OR  OTHER
  DAY  OF  INTEREST)? 9, 07, 1957                   输入你出生日期是1957年9月7日
9/  7/1957/  WAS  A  SATURDAY                       计算机告诉你的生日是星期六

**HAPPY  BIRTHDAY***                                祝贺你，今天正好是你的生日

                                  YEARS  MONTHS  DAYS
YOUR  AGE  IF  BIRTHDATE            26      0      0      你已生活了26年
YOU  HAVE  SLEPT                     8      8      3      你睡眠用去8年8
                                                          个月零3天
YOU  HAVE  EATEN                     2      7      9      你吃饭用去2年7
                                                          个月零9天
YOU  HAVE  STUDIED,                  8      8      3      你学习、工作、童
  WORKED, PLAY                                            年玩耍用去8年8
                                                          个月零3天
YOU  HAVE  RELAXED                   6      0     15      你休息用去6年零
                                                          15天

**YOU  MAY  RETIRE  IN  2017**                      你将在2017年退休
          WEEKDAY
  THE COMPUTER  WORLD
ENTER  TODAY'S  DATE  IN  THE
  FORM  4, 18, 1983? 5, 04, 1983                    举例：今天是1983年5月4日，
ENTER  DAY  OF  BIRTH (OR
  OTHER  DAY  OF  INTEREST) ?
  5, 04, 1966                                       输入你的出生日期是1966年5月4日，
5/  4/1966/  WAS  A  WEDNESDAY                      计算机告诉你的生日是星期三。

***HAPPY  BIRTHDAY***                               祝贺你，今天正好是
                                                    你的生日
                                  YEARS  MONTHS  DAYS
YOUR  AGE  IF  BIRTH DATE           17      0      0      你已生活了17年
YOU  HAVE  SLEPT                     5      8      3      你睡眠用去5年8个
                                                          月零3天
YOU  HAVE  EATEN                     1      8     15      你吃饭用去1年8个
                                                          月零15天
YOU  HAVE  PLAYED  AND               5      8      3      你学习和童年玩耍用
  STUDIED                                                 去5年8个月零3天
YOU  HAVE  RELAXED                   3     11      9      你休息用去3年11个
                                                          月零9天
        **YOU  MAY  RETIRE  IN  2026**                  你将在2026年退休
```

```
        WEEKDAY
  THE COMPUTER WORLD
ENTER TODAY'S DATE IN THE
  FORM 4, 18, 1983? 6, 01, 1983
ENTER DAY OF BIRTH (OR
  OTHER DAY OF INTEREST) ?
  6, 01, 1977
6/ 1/1977/ WAS A WEDNESDAY
```

举例：今天是1983年6月1日，

输入你的出生日期是1977年6月1日，

计算机告诉你的生日是星期三。

```
***HAPPY BIRTHDAY***
```

祝贺你，今天正好是你的生日。

	YEARS	MONTHS	DAYS	
YOUR AGE IF BIRTHDATE	6	0	0	你已生活了6年
YOU HAVE SLEPT	2	0	4	你睡眠用去2年零2天
YOU HAVE EATEN	0	7	9	你吃饭用去7个月零9天
YOU HAVE PLAYED	2	0	4	你玩耍用去2年零4天
YOU HAVE RELAXED	1	4	13	你休息用去1年4个月零13天

```
**YOU MAY RETIRE IN 2037**
```

你将在2037年退休

（三）思 考 题

请你在周围人中多找几个例子练习，把结果告诉他们。他们将会感到惊讶！更加珍惜自己的年华。

八 二 人 博 弈

49 巧 夺 奇 数

共15颗棋子，二人轮流取，每次只许取 1 、 2 或 3 颗，直到取尽为止，谁手中的棋子总数是奇数者为赢。

（一）算法分析和结论

15本身是个奇数，把它拆开成为两堆，总是一个奇数加上一个偶数，所以本游戏必是一胜一负，不可能平局。取胜者的秘诀是：第一次必须先取 2 颗，以后的取法则按下列规则行事：

第一条：刚取的棋子数如果和已取得的棋子数加起来是个奇数，那么剩下的棋子数必须是 1 或 8 或 9；

第二条：刚取的棋子数如果和已取得的棋子数加起来是个偶数，那么剩下的棋子数必须是 4 或 5 。

这两条规则是互相补充的，就是说，如果第一条做不到的话，那么第二条一定能做得到，依据规则，百战百胜。

如第一次甲取 2 颗，乙取 2 颗，余11颗。甲若再取 1 ， 1 + 2 = 3 是奇数，余10，与第一条不符；若取 2 ，则 2 + 2 = 4 是偶数，余9，与第二条不符；甲若取 3 ， 3 + 2 = 5 是奇数，余 8 ，与第一条相符，所以第二次甲取 3 。

若乙第二次取 1 颗，据上述的条件转移指令，易看出甲第三次取 1 ，不符合第二条；取 2 不符合第一条；所以仍取 3 ，符合第二条，这时余 4 ，不论乙第三次取 1 或 3 ，甲都可以全数取尽，使甲手中总数为奇数而胜。

若乙第三次取 2 ，甲取 1 ，余 1 ，乙不能不取，甲手中自然为奇数还是胜。

（二）程 序 设 计

1）设计思路

由键盘输入总棋数N为15颗，计算机CPU为一方，取K颗，你为另一方，取M颗，据题意K和M不大于 3 。先让你取，然后CPU据算法分析第一、二条判断它如何取，如此轮流下去，你若掌握了窍门，你胜。否则，CPU获胜。请见程序中举的两局例子，各有胜负。每取一次，形象地打印出过程。

2）框图 见图49

3）源程序及运行结果

```
5       WRITE (10)  (30H INPUT N(N=15：PLAY N=-1：STOP))
        READ (11, 8) N
8       FORMAT (I2)
        IF (N. EQ. -1) STOP
        IF (N. NE. 15) GOTO 5
        WRITE (10, 9) N
9       FORMAT (/5X, 4HNO. =, I2)
```

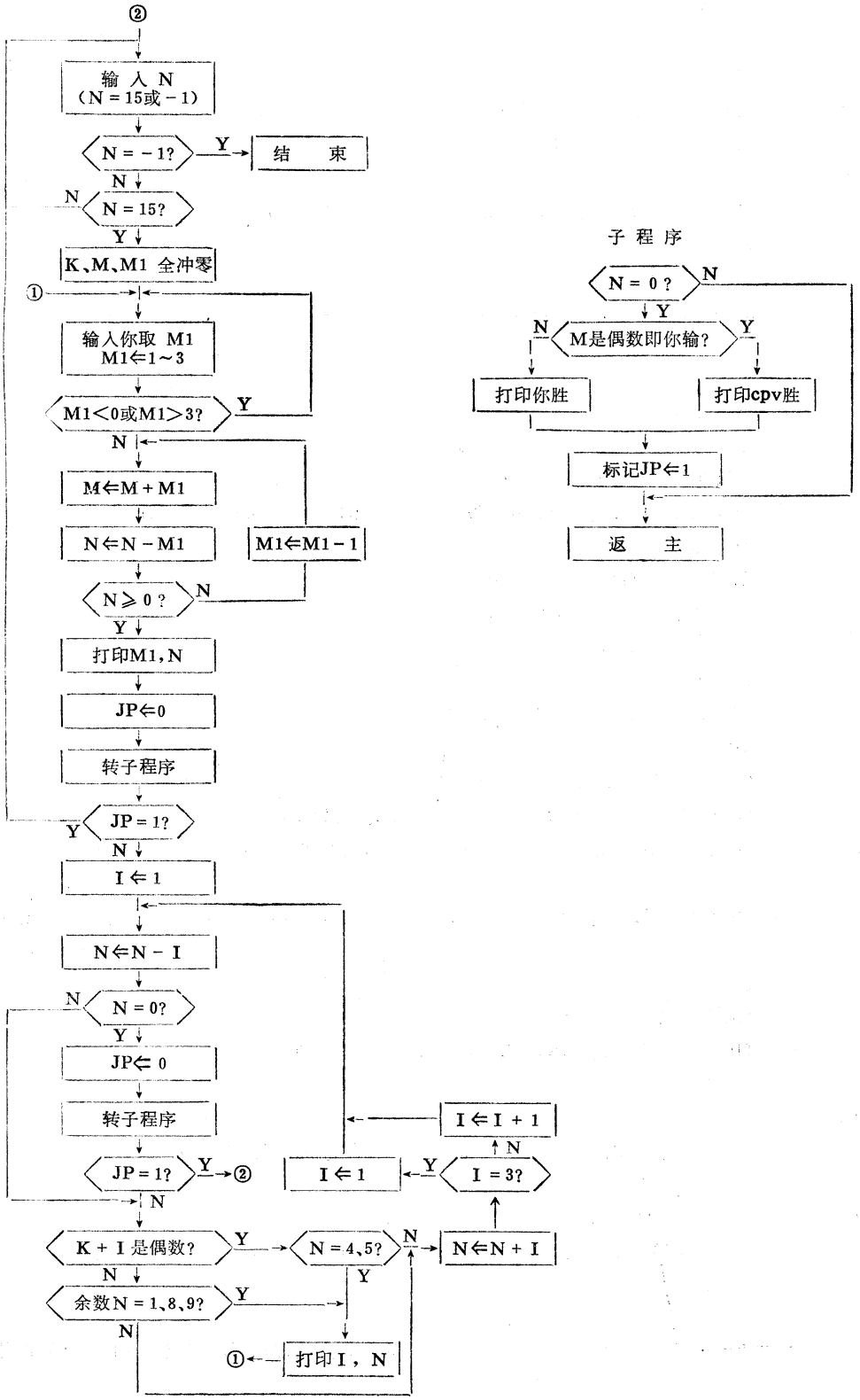

②
输 入 N
（N = 15或 - 1）
N = - 1?
Y
结 束
N
N = 15?
N
Y
K、M、M1 全冲零
①
输入你取 M1
M1⇐1～3
M1＜0或M1＞3?
Y
N
M⇐M + M1
N⇐N - M1
M1⇐M1 - 1
N ⩾ 0 ?
N
Y
打印M1，N
JP⇐0
转子程序
JP = 1?
Y
N
I ⇐ 1
N⇐N - I
N = 0?
N
Y
JP⇐ 0
转子程序
JP = 1?
Y
②
N
I⇐I + 1
N
I = 3?
Y
I ⇐ 1
K + I 是偶数?
Y
N = 4、5?
N
N⇐N + I
N
Y
余数N = 1、8、9?
Y
N
Y
①
打印 I ， N
子 程 序
N = 0 ?
N
Y
N
M是偶数即你输?
Y
打印你胜
打印cpv胜
标记JP⇐1
返 主

图 49

```
      K = 0
      M = 0
      M1 = 0
10    WRITE (10)  (19H READ M1 (1<=M1<=3))
      READ (11, 12) M1
12    FORMAT (I1)
      IF (M1. LE. 0. OR. M1. GT. 3) GOTO 10
15    M=M+M1
      N=N-M1
      IF (N. GE. 0) GOTO 30
      M=M-M1
      N=N+M1
      M1=M1-1
      GOTO 15
30    WRITE (10, 40) M1, N
40    FORMAT (10X, 9HYOU TAKE, I1, 8H       LEFT:, I2)
      JP = 0
      CALL PR (N, K, M, JP)
      IF (JP. EQ. 1) GOTO 5
      DO 50 I=1, 3
      N=N-I
      IF (N. NE. 0) GOTO 45
      JP = 0
      CALL PR (N, K, M, JP)
      IF (JP. EQ. 1) GOTO 5 (已取完，转语句5)
45    IF ((K+I)/2*2. EQ. K+I) GOTO 60 (没取完，判CPU取总数 (K+I) 符合算
                                       法二吗)
      IF (N. EQ. 1. OR. N. EQ. 8. OR. N. EQ.9)GOTO 70 (余数N，据算法一取
                                                      胜条件来判)
      N=N+I
      GOTO 50
60    IF (N. EQ. 4. OR. N. EQ. 5) GOTO 70
      N=N+I
50    CONTINUE
      I=1 (出循环说明CPU取1～3都不符合算法一、二条人为令CPU取1。)
      N=N-I
70    WRITE (10, 80) I, N
80    FORMAT (10X, 9HCPU TAKE, I1, 8H       LEFT:, I2)
      GOTO 10
      END
      SUBROUTINE PR (N, K, M, JP) 子程序判取尽否，若取尽 (N=0) 谁胜。
      IF (N. NE. 0) GOTO 40
      IF (M/2*2. EQ. M) GOTO 10
```

```
      WRITE (10, 20) M  你取总数M为奇数
20    FORMAT (10X, 26HYOU WIN·····YOU HAVE TAKE:, I2)
      JP=1
      GOTO 40
10    I=15-M  CPU取总数为奇数
      WRITE (10, 30) I
30    FORMAT (10X, 26HCPU WIN·····CPU HAVE TAKE:, I2
      JP=1
40    RETURN
      END
```

```
 INPUT N (N=15:PLAY  N=-1:STOP)
15
    NO. =15                                    (第一局键盘输入15)
 READ M1 (1<=M1<=3)
3
         YOU TAKE 3  LEFT:12                   你取3，余12。
         CPU TAKE 3  LEFT: 9                   CPU取3，余9。
 READ M1 (1<=M1<=3)
3
         YOU TAKE 3  LEFT: 6                   你取3，余6。
         CPU TAKE 2  LEFT: 4                   CPU取2，余4。
 READ M1 (1<=M1<=3)
3
         YOU TAKE 3  LEFT: 1                   你取3，余1。
         YOU WIN·····YOU HAVE TAKE: 9          你有奇数9，你胜。

 INPUT N (N=15:PLAY N=-1:STOP)
15
    NO. =15                                    (第二局键盘输入15)
 READ M1 (1<=M1<=3)
2
           YOU TAKE 2  LEFT:13                 你取2，余13。
           CPU TAKE 1  LEFT:12                 CPU取1，余12。
 READ M1 (1<=M1<=3)
1
           YOU TAKE 1  LEFT:11                 你取1，余11。
           CPU TAKE 3  LEFT: 8                 CPU取3，余8。
 READ M1 (1<=M1<=3)
2
           YOU TAKE 2  LEFT: 6                 你取2，余6。
           CPU TAKE 2  LEFT: 4                 CPU取2，余4。
 READ M1 (1<=M1<=3)
```

```
3
        YOU TAKE 3   LEFT：1                    你取3，余1。
        CPU WIN……CPU HAVE TAKE：7    CPU取1后为奇数7，CPU胜。

 INPUT N (N=15：PLAY N=-1：STOP)
-1
STOP
```

（三）思 考 题

巧夺偶数——与主题巧夺奇数恰恰相反。在桌上放25个小棋子，由双方轮流取棋子，每人每次至少要取一颗棋子，最多可取三颗。两个局中人照这种方式取下去，直到桌面上所有的棋子全部被取完，于是双方各自把手中的棋子加以清点，这时必定有一个人拿到的棋子总数是偶数，他就是获胜者。

这种游戏局中人既要考虑到自己手里的棋子数，又要考虑到对方手里的棋子数，变化比较多。请你设计如何才能取胜的程序。

50 取 尽 者 胜

共20张牌，二人轮流取，谁最后取完便胜，但每次只许取1～2张牌。

（一）算法分析和结论

要想最后拿到20，必先拿到17，要想拿到17，必先拿到14，…，依次类推，只要拿到3的倍数再加2的数字，就一定可取胜。如先拿到2，对方不论拿到3或4，你都能拿到5，这样便可拿到所有3的倍数加2的数字，最后，必然拿到20，取尽了则胜。

（二）程 序 设 计

1）设计思路

计算机CPU为一方，你为另一方，据算法分析先取者必胜。让CPU先取N1=2，你取N2（N2≦2），以后CPU取时先判，取余数N的3的倍数（即剩下3的倍数），这是CPU必胜的算法。首先用READ语句由键盘输入20张牌，以后你从键盘敲入数字做为取牌的张数，依题意每次不超过2张，掌握窍门者是先取，并且先取2。

我们在程序中举了两局例子，都被计算机赢了。双方每取一次牌，都形象地打印出来，请见程序运行时打印结果。

2）框图 见图50

3）源程序及运行结果

```
5       WRITE (10)  (30H INPUT N (PLAY：N=20 STOP：N=-1))
        READ (11, 10) N
10      FORMAT (I2)
        IF (N. EQ. -1) STOP
        IF (N. NE. 20) GOTO 5
        WRITE (10, 11) N
11      FORMAT (/5X, I2, 6H CARDS)
        N1=2
17      WRITE (10, 20) N1
```

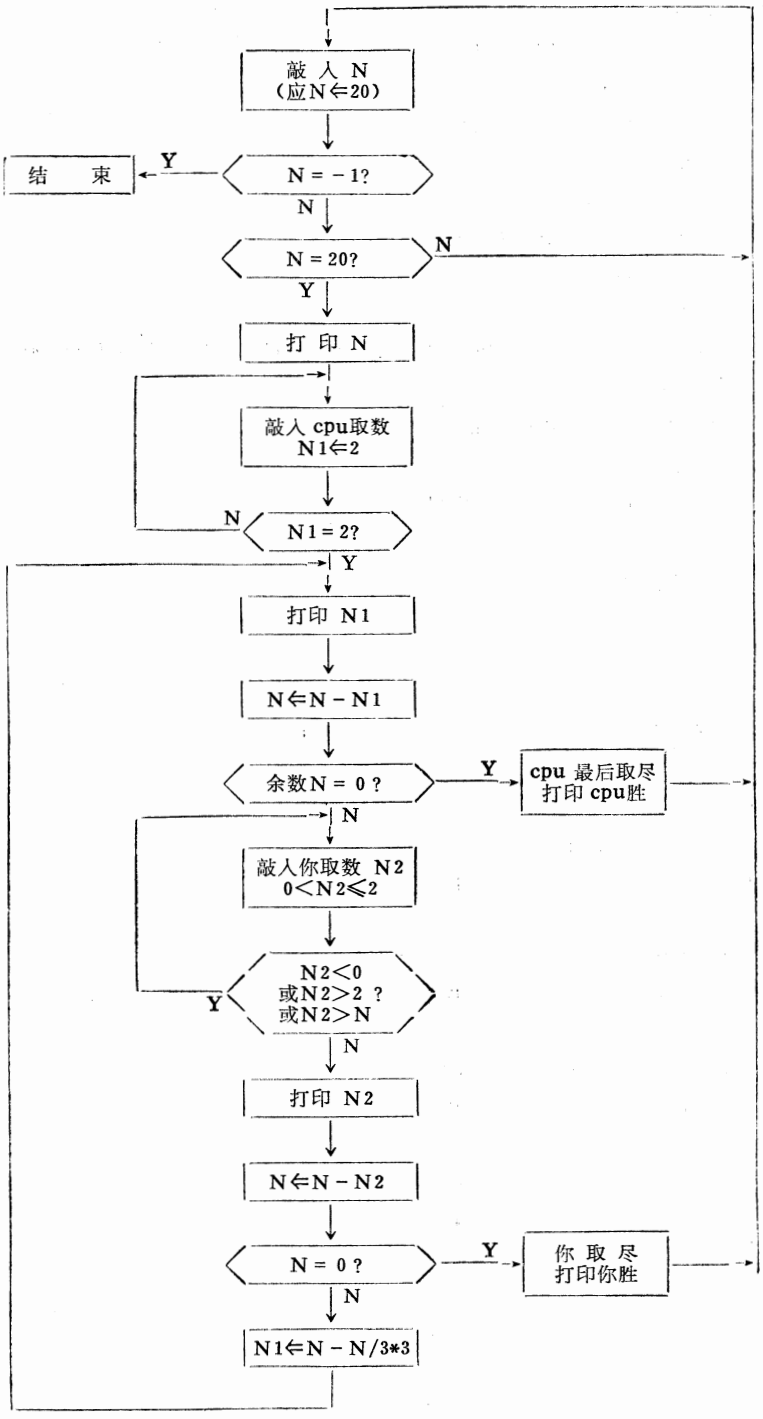

敲 入 N
(应N⇐20)
结 束
Y
N = -1?
N
N = 20?
N
Y
打 印 N
敲入 cpu取数
N1⇐2
N
N1 = 2?
Y
打印 N1
N⇐N - N1
余数N = 0 ?
Y
cpu 最后取尽
打印 cpu胜
N
敲入你取数 N2
0<N2≤2
N2<0
或N2>2 ?
或N2>N
Y
N
打印 N2
N⇐N - N2
N = 0 ?
Y
你 取 尽
打印你胜
N
N1⇐N - N/3*3

图 50

```
20      FORMAT (10X, 9HCPU TAKE, I1)
        N=N-N1
        IF (N. NE. 0) GOTO 40
        WRITE (10, 30)
30      FORMAT (10X, 7HCPU WIN)
        GOTO 5
40      WRITE (10, 41)
41      FORMAT (18H YOU TAKE 1 OR 2 :)
        READ (11, 45) N2
45      FORMAT (I1)
        IF (N2. LE. 0. OR. N2. GT. 2. OR. N2. GT. N) GOTO 40
        N=N-N2
        IF (N. NE. 0) GOTO 70
        WRITE (10, 60)
60      FORMAT (10X, 7HYOU WIN)
        GOTO 5
70      N1=N-N/3*3
        GOTO 17
        END
```

```
 INPUT N (PLAY : N=20 STOP : N=-1)
20
    20 CARDS                    第一局输入20
          CPU TAKE 2            CPU取 2
 YOU TAKE 1 OR 2 :              你取 2
2
          CPU TAKE 1            CPU取 1
 YOU TAKE 1 OR 2 :              你取 2
2
          CPU TAKE 1            CPU取 1
 YOU TAKE 1 OR 2 :              你取 1
1
          CPU TAKE 2            CPU取 2
 YOU TAKE 1 OR 2 :              你取 1
1
          CPU TAKE 2            CPU取 2
 YOU TAKE 1 OR 2 :              你取 1
1
          CPU TAKE 2            CPU取 2
 YOU TAKE 1 OR 2 :              你取 1
1
          CPU TAKE 2            CPU取 2
          CPU WIN               CPU胜
```

```
 INPUT N (PLAY : N=20 STOP : N=-1)
20
    20 CARDS                    第二局输入20
          CPU TAKE 2            CPU取2
 YOU TAKE 1 OR 2 :              你取1
1
          CPU TAKE 2            CPU取2
 YOU TAKE 1 OR 2 :              你取2
2         CPU TAKE 1            CPU取1
 YOU TAKE 1 OR 2 :              你取2
2
          CPU TAKE 1            CPU取1
 YOU TAKE 1 OR 2 :              你取1
1
          CPU TAKE 2            CPU取2
 YOU TAKE 1 OR 2 :              你取1
1
          CPU TAKE 2            CPU取2
 YOU TAKE 1 OR 2 :              你取2
2
          CPU TAKE 1            CPU取1
          CPU WIN               CPU胜
 INPUT N (PLAY : N=20 STOP : N=-1)
-1
STOP
```

（三）思 考 题

如果主题目中牌的总数不是20而是24张牌，每次只许取1～2张，取尽者胜。请分析获胜者诀窍：先取还是后取，应先拿到几最后必胜。编写程序，玩几局，看看你的结论对否？（可在本书找到答案）。

51 有 奖 储 蓄

父亲买一只储蓄盒，对莉莉和文文说，“咱们来搞有奖储蓄游戏”，办法是这样：“你们俩每天轮流把一枚硬币投进盒里，而我每天后投进一枚1分的硬币，今天你投，明天他投，投1分的、2分的、5分的都可以，谁先把盒里的钱凑成三元或超过三元，谁就中奖…。”

请分析如何投才能中奖呢？

（一）算法分析和结论

每天父亲后投入1分。这样每天投入数可能2分（1＋1）、3分（1＋2）或6分（1＋5）。在投储过程中，谁先使盒内储金数达到292分或293分，谁就能保证自己先储满或超过三元。

为防止293分这个数被对方控制，自己就要避免占据291分(293－2)、290分(293－3)以及287分（293－6）；为了不让对方占据，自己就要避开286、285、282各数，而去占领284和283两数…。如此不断往下推算，可得下列结果

300	299	298	297	296	295		
294	293	292	291	290	289		
288	287	286	285	284	283		
282	281	280	279	278	277		
276	275	274	273	272	271		
270	269	268	267	266	265		
264	263	262	261	…	…		
20	19	18	17	16	15		
14	13	12	11	10	9		
8	7	6	5	4	3	2	1

表中数字下画“—”者，可取胜，其他数字要避开。

从上表可看出规律，凡是9的整数倍以及被9除余4或5的数（297除外）都能取胜！

（二）程 序 设 计

1）设计思路

由题意是三人游戏斗智，不过第三方即父亲一方，每次在你或我投后，他固定投1分钱，规则是你、我双方轮流投（不在同一天投）。这样，今天我投一个数目，明天你投K数目，投后总是加父亲的1分钱，盒里累积为Y分钱。

程序第60语句及其下两个语句，第110语句及其下五个语句，都是据算法分析如何占领有利位置。当你暂时占据有利位置、自己只好随便投入1、2、5任一枚，如投入1，便是70语句意。

当你最后投满300，打印出你赢。否则，打印出你失败。谁先投都可以，自己若不先投，可由键盘输入J为零。程序中举了一例，结果你赢，得奖。

2）框图　见图51

3）源程序及运行结果

```
1       K = 0
        WRITE ( 6 , 10)
10      FORMAT (1X, 47HLET J TO BE NOT EQAUL 0
           IF I WANT START FIRST/)
        WRITE ( 6 , 11)
11      FORMAT (1X, 3HJ = ? )
        READ ( 5 , 15) J
15      FORMAT (I1)
        Y = 0
        IF (J. NE. 0 ) GO TO 60
40      WRITE ( 6 , 16)
16      FORMAT (1X, 3HK = ? )
        READ ( 5 , 15) K
        IF (K. NE. 1 . AND. K. NE. 2 . AND. K. NE.
           5 ) GO TO 40
```

开始若自己先投硬币，则投J数。

K为对方投放钱数

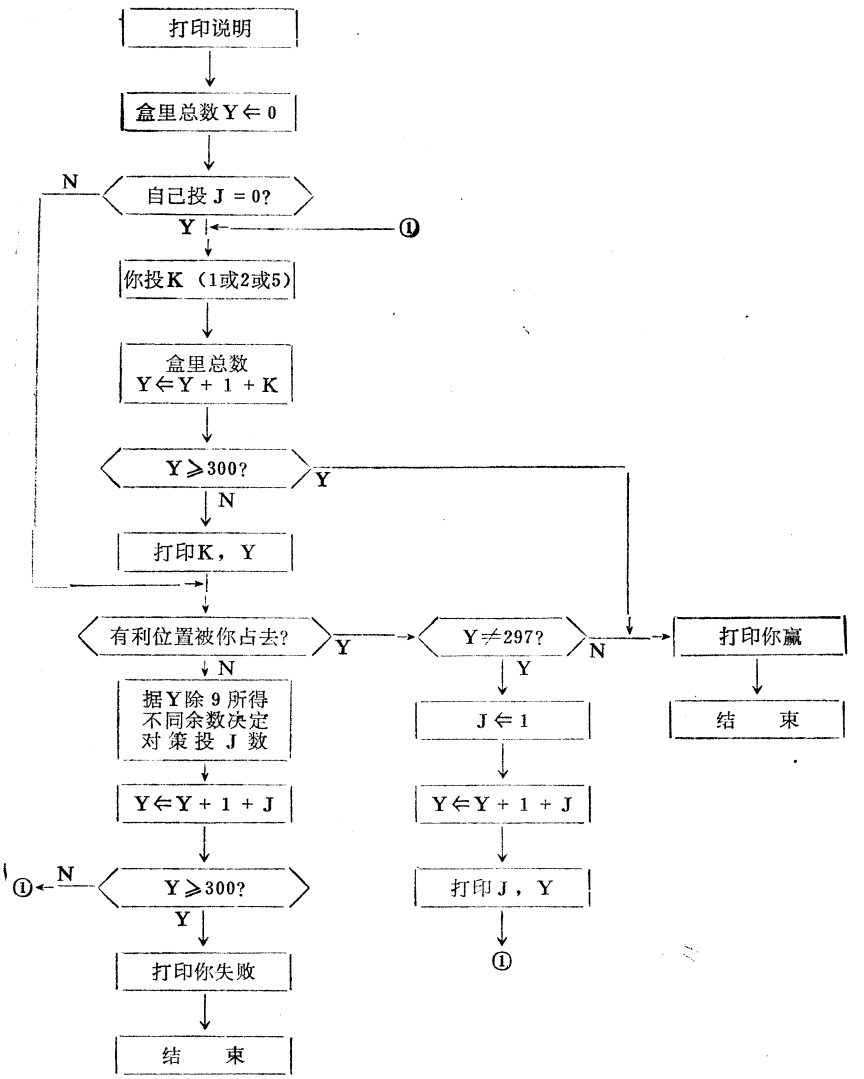


图 51

```
      Y = Y + 1 + K                          } Y是储蓄盒里总钱数，因为
      IF (Y. GE. 300) GO TO 500              } 父亲每次总投1分，故加1。
      WRITE (6, 20) K, Y
20    FORMAT (1X, 2HK=, I1, 3H 1, 2HY=, F4.0)
      K = 0
60    IF (Y/9.0. NE. INT (Y/9.0) . AND. (Y-4.
          0) /9.0. NE. INT ((Y-4.0)/9.0) . AND.
       (Y-5.0) /9.0. NE. INT ((Y-5.0) /9.0) )   } 判断有利位置被对方占去否
          GO TO 110
      IF (Y. NE. 297) GO TO 70
      J = 5                                  } J是自己投放钱数
      GO TO 510
70    J = 1
```

```
      IF (J. NE. 1. AND. J. NE. 2. AND. J. NE. 5) GO TO 70
      Y=Y+1+J
      WRITE (6, 22) J, Y
      GO TO 40
110   IF ((Y-1.0)/9.0. EQ. INT ((Y-1.0)/9.0))
     1   GO TO 200
      IF ((Y-2.0)/9.0. EQ. INT ((Y-2.0)/9.0))
     1   GO TO 210
      IF ((Y-3.0)/9.0. EQ. INT ((Y-3.0)/9.0))
     1   GO TO 220
      IF ((Y-6.0)/9.0. EQ. INT ((Y-6.0)/9.0))
     1   GO TO 200
      IF ((Y-7.0)/9.0. EQ. INT ((Y-7.0)/9.0))
     1   GO TO 210
      IF ((Y-8.0)/9.0. EQ. INT ((Y-8.0)/9.0))
     1   GO TO 220
200   J = 2
      GO TO 230
210   J= 1
      GO TO 230
220   J = 5
230   Y=Y+1+J
      WRITE (6, 22) J, Y
 22   FORMAT (1X, 2HJ=, I1, 3H 1, 2HY=, F4.0)
      IF (Y. GE. 300) GO TO 510
      GO TO 40
500   WRITE (6, 30)
 30   FORMAT (1X, 12HYOU HAVE WON)
      STOP
510   WRITE (6, 35)
 35   FORMAT (1X, 15HYOU HAVE FAILED)
      STOP
      END
```

对方没占有利位置时，Y除9的余数分别是1、2、3、6、7、8，根据每一余数选择自己不同对策。

打印你赢

打印你失败

```
LET J TO BENOT EQAUL 0 IF I WANT START FIRST
J =? 1
J = 1    1   Y = 2.
K =? 2
K = 2    1   Y = 5.
J = 1    1   Y = 7.
K =? 1
K = 1    1   Y = 9.
J = 1    1   Y =11.
```

若开始第一次自己先投，则投钱J数目。

自己投1加父亲1共有2

你投2加父亲1共有5

自己投1加父亲1共有7

你投1加父亲1共有9

自己投1加父亲1共有11

```
K = ? 5
K = 5   1   Y = 17.          你投 5 加父亲 1 共有17
J = 5   1   Y = 23.          自己投 5 加父亲 1 共有23
K = ? 5
K = 5   1   Y = 29.          你投 5 加父亲 1 共有29
J = 1   1   Y = 31.          自己投 1 加父亲 1 共有31
K = ? 1
……                           (因较长省略中间部分)
K = 5   1   Y = 285.         你投 5 加父亲 1 共有285.
J = 2   1   Y = 288.         自己投 2 加父亲 1 共有288
K = ? 1
K = 1   1   Y = 290.         你投 1 加父亲 1 共有290
J = 1   1   Y = 292.         自己投 1 加父亲 1 共有292
K = ? 1
K = 1   1   Y = 294.         你投 1 加父亲 1 共有294
J = 2   1   Y = 297.         你投 2 加父亲 1 共有297
K = ? 5                      你再投 5 (共有钱数≥300)
YOU HAVE WON                 你赢
```

(三) 思 考 题

若每天三个人都投(先后投),所投数目都不固定,由你安排,如何进行程序设计,才能取胜。

52 请试试你几次猜中

计算机随机地选择一个英文字母,从A到Z(依次称为由小到大),让你猜。如果你猜得过大,计算机会告诉你"太大",让你再猜一个小一些的字母。并统计出你一共猜了几次。若五次之内猜中了,计算机会打印或显示出"你猜得好极了。"否则,打印出猜中了,但比五次多,并让你跟它再玩一次。

(一) 算法分析和结论(省略)

(二) 程 序 设 计

1) 设计思路

计算机随机选择一个英文字母,让你猜。我们选用计算机型号为HP/3000,其FORTRAN语言具有字符类型的有关语句READ CA,相当BASIC语言的ASC函数,产生某字母的ASCII码。该机A字母的ASCII码为16640 + 256 * 1,B的ASCII码为16640 + 256 * 2,……,Z的ASCII码为16640 + 256 * 26。注意,程序中语句100,是计算机随机选择一个用ASCII码表示的一个字母,即该语句产生一个随机数L。然后你猜,你用READ CA输入一个字母,将CA与M放在同一单元(用等价语句说明),计算机产生的L和你猜的M,二者比较哪个大或小,打出信息,你再猜,当猜到M与L相等时,便是猜中了。每猜一次,计数器K加1。猜中时,据K大于或小于5,打印出不同信息。你可跟计算机多玩几次。

程序中举了一例:计算机说我有一字母,你猜是什么字母?猜是G。打出信息,猜的

大了，应小些。再猜是D。打出信息，仍大，应小些。再猜是B。打出信息，猜的小了，应大些。再猜是C。打出信息“你4次猜中了，好极了！”再玩一局吧！每玩一局，计数器I加1。

2）框图　见图52

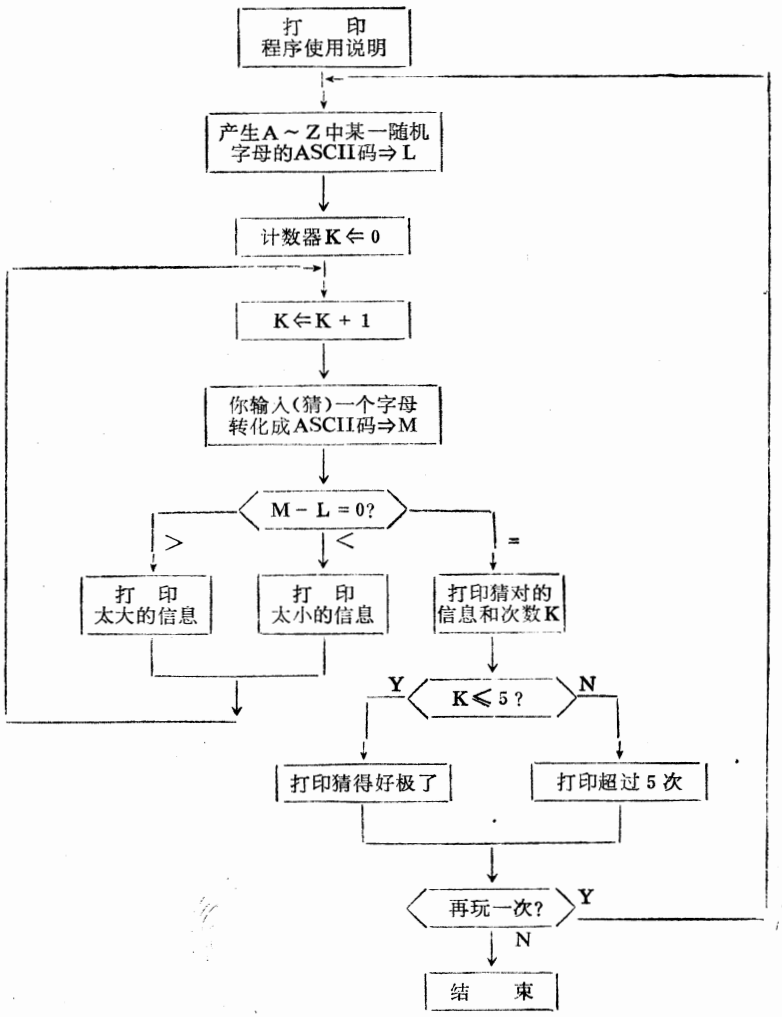


图 52

3）源程序及运行结果

```
      CHARACTER*1 CA
      EQUIVALENCE (M, CA)
      WRITE (6, 10)
10    FORMAT (13X, 38H…… LETTER GUESSING GAME ……/)
      WRITE (6, 15)
15    FORMAT (8X, 48HI WILL THINK OF A LETTER OF THE APHABET, A
     TO Z.)
      WRITE (6, 20)
20    FORMAT (8X, 48HTRY TO GUESS MY LETTER AND I WILL GIVE YOU
```

```
      CLUES)
      WRITE (6, 25)
25    FORMAT (8X, 45HAS TO HOW CLOSE YOU ARE GETTING TO MY
     LETTER. /)
100   L=16640+INT (ABS (SIN (L/256+0. 1)) *26) *256          产生随机数，转
                                                    化成A～Z某个字母的ASCII码。
      K = 0
      WRITE (6, 30)
30    FORMAT (8X, 35HOK, I HAVE A LETTER. START GUESSING/)
110   WRITE (6, 32)
32    FORMAT (8X, 19HWHAT IS YOUR GUESS? )
      K=K+1
      READ (5, 35) CA          你输入一个字母的ASCII码
35    FORMAT (A1)
      IF (CA. EQ. "0") STOP
      IF (M-L) 180, 220, 200     计算机产生的L和你猜(输入)的M，二者比较合适否。
180   WRITE (6, 40)
40    FORMAT (8X, 29HTOO LOW, TRY A HIGHER LETTER. )
      GO TO 110
200   WRITE (6, 45)
45    FORMAT (8X, 29HTOO HIGH, TRY A LOWER LETTER. )
      GO TO 110
220   WRITE (6, 50) G
      I=I+1
50    FORMAT (8X, 14HYOU GET IT IN. , I2, 9H GUESSES. )
      IF (K. LE. 5) GO TO 270
      WRITE (6, 55)
55    FORMAT (8X, 43HBUT IT SHOULD NOT TAKE MORE THAN 5
     GUFSSES! /)
      GO TO 280
270   WRITE (6, 60)
60    FORMAT (8X, 12HGOOD JOB ! ! ! / /)
280   WRITE (6. 65)
65    FORMAT (8X, 19HLET'S PLAY AGAIN……/)
      I=I+1
      GO TO 100
      END
```

```
    …… LETTER GUESSING GAME ……
I WILL THINK OF A LETTER OF THE APHABET, A TO Z.
TRY TO GUESS MY LETTER AND I WILL GIVE YOU CLUES
AS TO HOW CLOSE YOU ARE GETTING TO MY LETTER.
OK, I HAVE A LETTER. START GUESSING          计算机产生一个字母。
```

```
WHAT IS YOUR GUESS? G                    你猜是G。
TOO HIGH, TRY A LOWER LETTER.            猜大了，再小些。
WHAT IS YOUR GUESS? D                    猜是D。
TOO HIGH, TRY A LOWER LETTER.            猜大了，再小些。
WHAT IS YOUR GUESS? B                    猜是B。
TOO LOW, TRY A HIGHER LETTER.            猜小了，再大些。
WHAT IS YOUR GUESS? C                    猜是C。
YOU GET IT IN. 4 GUESSES.                你猜对了，猜4次。
GOOD JOB! ! !                            好极了!
LET'S PLAY AGAIN--                       再玩一局…
OK, I HAVE A LETTER. START GUESSING
WHAT IS YOUR GUESS?
```

（三）思 考 题

适当改动本程序，使之两个字母也能进行比较。

53 先 取 者 输

若有24张牌，二人轮流每次拿一张或两张，谁拿到最后一张牌便胜，如果知道窍门，先取牌的人必定输，后拿牌的人，必定胜，应怎样取法才能胜呢?

（一）算法分析和结论

将24张纸牌依次编上1～24号，因为24是3的倍数，故应先拿到3，这样就保证依次可以拿到3的倍数。若将24张牌，依次叫做1，2，3，…，24号。若想取到最后一张牌（24），必须在这以前先取到21，若想取得21，需先取得18，15，12，…，3。若首先取到3，不能先取牌，让对方先取1或2，你总能在后取时拿到3。因为24是3的倍数，只要能取到3的倍数，必胜无疑。

（二）程 序 设 计

1）设计思路

先由键盘敲入总牌数N = 24。假若你为一方，计算机为另一方。计算机由算法分析，让你先取N 1张，余下N⇐N − N 1，这时计算机判它如何取N 2张（3的倍数）。尚余N⇐N − N 2，再由你取N 1张，如此反复按规定取，取到最后一张时，必是计算机，打印出CPU WIN。相信否，请试试。程序中举一例，由键盘敲入的取牌张数，已输给了计算机。

2）框图 见图53

3）源程序及运行结果

```
1       WRITE (10, 2)
2       FORMAT (31H  INPUT  N (PLAY : N = 24  STOP : N = -1) :)
        READ (11, 6) N
6       FORMAT (I2)
        IF (N. EQ. -1) STOP
        IF (N. NE. 24) GOTO 1
        WRITE (10, 10) N
10      FORMAT (/5X, I2, 6H  CARDS)
```

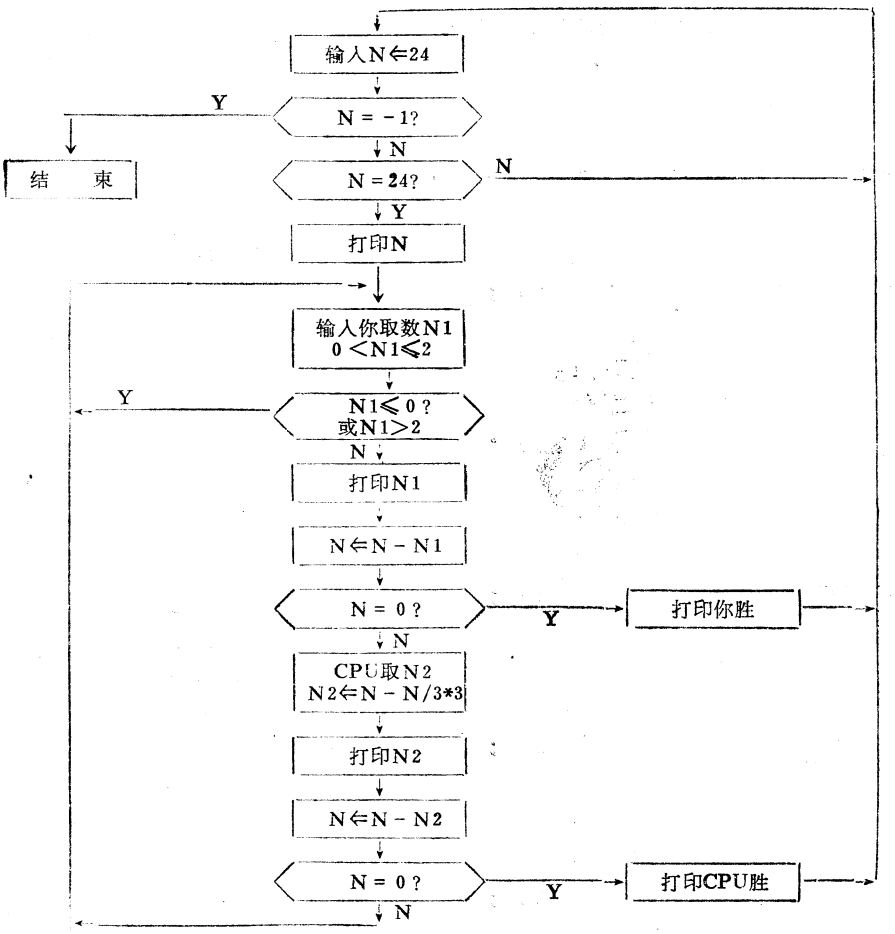


图 53

```
15      WRITE (10)  (23HYOU  WILL  TAKE (1  OR  2) :)
        READ (11, 16) N1
16      FORMAT (I1)
        IF (N1. LE. 0. OR. N1. GT. 2) GOTO 15
        WRITE (10, 20) N1
20      FORMAT (10X, 9HYOU  TAKE, I1)
        N = N - N1
        IF (N. NE. 0) GOTO 40
        WRITE (10, 30)
30      FORMAT (10X, 7HYOU  WIN)
        GOTO 1
40      N2 = N - N/3*3
        WRITE (10, 50) N2
50      FORMAT (10X, 9HCPU  TAKE, I1)
        N = N - N2
        IF (N. NE. 0) GOTO 15
```

```
        WRITE (10, 60)
60      FORMAT (10X, 7HCPU  WIN)
        GOTO  1
        END

INPUT  N (PLAY : N=24  STOP : N=-1) :
24
   24  CARDS                                    输入24张牌
YOU  WILL  TAKE (1  OR  2) :
1
           YOU  TAKE  1
           CPU  TAKE  2
YOU  WILL  TAKE (1  OR  2) :
2
           YOU  TAKE  2
           CPU  TAKE  1
YOU  WILL  TAKE (1  OR  2) :
2
           YOU  TAKE  2
           CPU  TAKE  1
YOU  WILL  TAKE (1  OR  2) :
1
           YOU  TAKE  1
           CPU  TAKE  2
YOU  WILL  TAKE (1  OR  2) :
1
           YOU  TAKE  1
           CPU  TAKE  2
YOU  WILL  TAKE (1  OR  2) :
1
           YOU  TAKE  1
           CPU  TAKE  2
YOU  WILL  TAKE (1  OR  2) :
2
           YOU  TAKE  2
           CPU  TAKE  1
YOU  WILL  TAKE (1  OR  2) :
2
           YOU  TAKE  2
           CPU  TAKE  1
           CPU  WIN

INPUT  N (PLAY : N=24  STOP : N=-1) :
```

```
- 1
STOP
```

（三）思 考 题

甲、乙二人各有一副牌(11张)，每副牌的牌面上分别写着0到10，二人轮流将手中的牌每次取出一张计算得分，得分办法如下规定：设甲的牌面数值为X，乙的为Y，按照$2x^2-y^2$的式子记分，如得正数，是甲的分数；得负数，是乙的分数。例如：甲取X = 5，乙取Y = 2，代入式中得46，是甲得46分。

又如，甲取X = 1，乙取Y = 3，代入式中得 - 7，是乙得7分。

请你想一想，甲、乙各采用什么策略，最后得总分最多？

54 由你指定总数量和取法

假设有一堆小石子，二人轮流去取，谁拿走最后一颗石子便输。还让你先规定石子总数N以及每次最多取多少颗数K（当然N≥K≥1），智谋者才能赢你。

请想想如何才能取胜。

（一）笔算步骤和结果

若指定总数N以及每次取走最大数K（N≥K≥1）。先取的人按此规则去取，后取者便要计算还剩下多少，应该取多少颗石子才能取胜，智谋者应取N - 1除以K - 1的余数才能取胜，每次保持这样取法，才能迫使对方（先取者）取走最后一颗石子——输了。

（二）程 序 设 计

1）设计思路

你和计算机各为一方，和计算机比智慧。由你指定一大堆石子数量N，并由你指定每次最多取K颗。然后由你先取（从键盘敲入取走的数量K 1），N⇐N - K 1，而计算机（CPU）取数先判N＞0时，取K 2 = （N - 1）-（N - 1）/（K + 1）×（K + 1），才能迫使你取最后一颗石子，当判K 1 = 0时，计算机便赢了，打印出CPU WIN。否则，打印你赢（YOU WIN）。

程序中暂定N为三位正整数。玩了三局，都是计算机赢了。第一局，N = 1，K = 1；第二局，N = 999，K = 678；第三局，N = 43，K = 28。请看程序，形象地打印出计算机在每局中是如何取胜的。

2）框图 见图54-1

3）源程序及运行结果

```
5       WRITE (10) (36H TOTAL NUMBER(1<=N<=999 N=-1:STOP):)
        READ (11, 10) N
        WRITE (10) (28H NUMBER OF TAKING (1<=K<=N) :)
        READ (11, 10) K
10      FORMAT (I3)
        IF (N. EQ. - 1) STOP
        IF (K. GT. N) GOTO 5
        WRITE (10, 30) N K
30      FORMAT (/5X, 10HALL CARDS:, I3, 5X, 19HEVERY TAKE
           MAXIMUM:, I3)
```

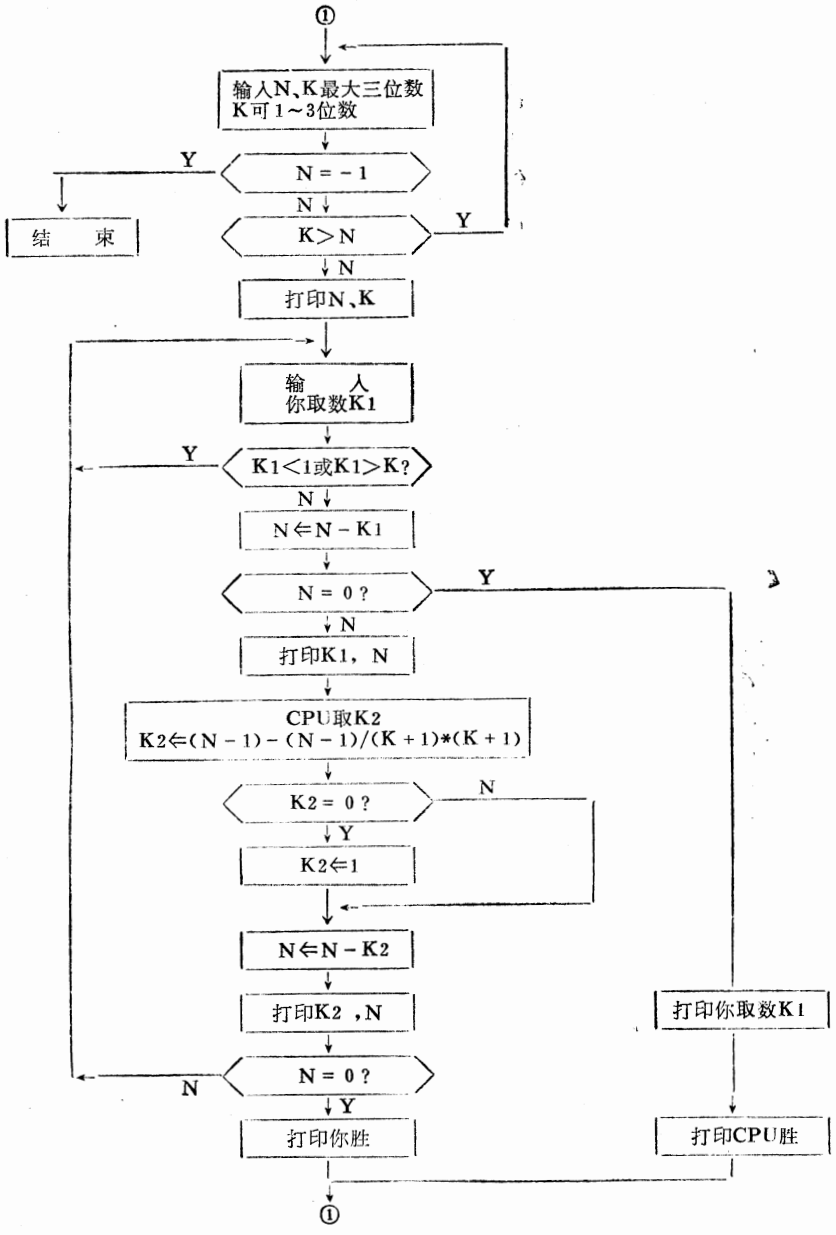


图 54-1

```
35      WRITE (10)  (11H  YOU  TAKE:)
        READ (11, 40) K1
40      FORMAT (I3)
        IF (K1. LT. 1. OR. K1. GT. K) GOTO  35
        N=N-K1
        IF (N. EQ. 0) GOTO  80
        WRITE (10, 50) K1, N
50      FORMAT (5X, 9HYOU  TAKE:, I3, 5X, 5HLEFT:, I3)
```

```
      K2 = (N-1) - (N-1) / (K+1) * (K+1)
      IF (K2. NE. 0) GOTO 60
      K2=1                                  当算出K2=0时，令K2=1，以免违例。
60    N = N - K2
      WRITE (10, 70) K2, N
70    FORMAT (5X, 9HCPU  TAKE:, I3, 5X, 5HLEFT:, I3)
      IF (N. EQ. 0) GOTO 100
      GOTO 35
80    WRITE (10, 90) K1
90    FORMAT (5X, 9HYOU  TAKE:, I3)
      WRITE (10, 95)
95    FORMAT (5X, 7HCPU  WIN)
      GOTO 5
100   WRITE (10, 110)
110   FORMAT (5X, 7HYOU  WIN)
      GOTO 5
      END
```

```
TOTAL  NUMBER (1<=N<=999  N=-1:STOP) :
  1
NUMBER  OF  TAKING (1<=K<=N) :
  1
   ALL  CARDS:  1       EVERY  TAKE  MAXIMUM:  1
YOU  TAKE
  1
   YOU  TAKE:  1
   CPU  WIN

TOTAL  NUMBER (1<=N<=999  N=-1  STOP) :
 999
NUMBER  OF  TAKING (1<=K<=N) :
 678
   ALL  CARDS: 999       EVERY  TAKE  MAXIMUM: 678
YOU  TAKE
 385
   YOU  TAKE:  38       LEFT: 961
   CPU  TAKE: 281       LEFT: 680
YOU  TAKE:
 197
   YOU  TAKE: 197       LEFT: 183
   CPU  TAKE: 483       LEFT:  1
YOU  TAKE:
  1
```

```
    YOU  TAKE:  1
    CPU  WIN
TOTAL  NUMBER (1<=N<=999  N=-1:STOP) :
 43
NUMBER  OF  TAKING (1<=K<=N) :
 28
   ALL  CARDS: 43        EVERY  TAKE  MAXIMUM:28
YOU  TAKE:
 14
   YOU  TAKE: 14        LEFT: 29
   CPU  TAKE: 28        LEFT:  1
YOU  TAKE
 1
   YOU  TAKE:  1
   CPU  WIN

TOTAL  NUMBER (1<=N<=999  N=-1:STOP) :
-1
NUMBER  OF  TAKING (1<=K<=N) :
STOP
```

（三）思　考　题

围棋盘上下国际象棋——下过国际象棋的人都知道，在这一棋种中，“皇后”的威力最大。它不仅纵横自如，不限步数，好象中国象棋里的“车”，而且在对角线上斜飞攻杀也不限步数，真可以说是“八面威风”了。但是，在下面将要说到的一种博弈问题里，“皇后”的威力却要差一些。虽然它也仍然不限步数，可是却永远只能沿着三个方向移动（见图54-2中的箭头，它只能向左、向下，以及向左下方移动），而且只能前进，不准倒退。图中用=符号表示皇后初始位置。

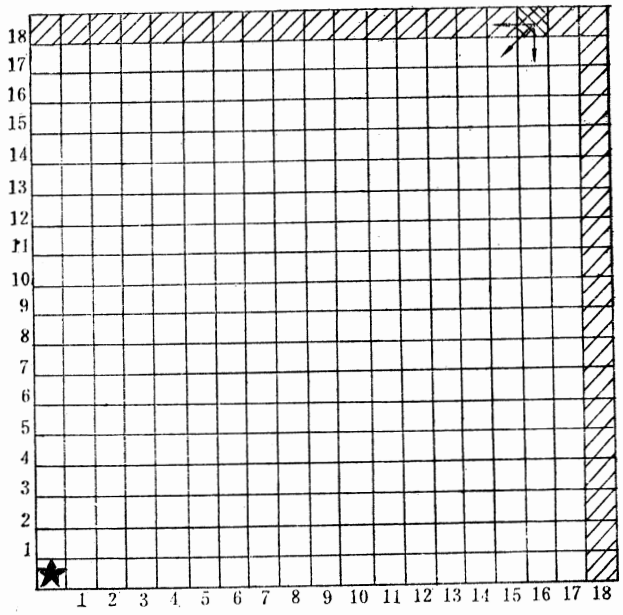


图 54-2

图54-2就是一个19×19格的棋盘，它比通常的围棋盘略大一些。可能大家都了解，围棋子是放在线的交叉点上的，但是，国际象棋子却是放在格子里头的。

棋盘的左下角方格打印一个五角星，它就是“皇后”要进入的目标。本游戏非常特殊，只用一只棋子，就是“皇后”，有两个局中人，双方轮流搬动棋子。开始的时候，甲方可以把一只

“皇后”棋子放在顶上一行或右边一列的任意一个格子里（就是图中打着荫影线的那一部分区域）。甲方把棋子放好以后，就轮到乙方走棋，他可以按照“皇后”的走法，向左、向下或向左下方任意走几步。乙方走好后，再轮到甲方来走，他也必须按照上面的规则走棋。就这样双方轮流地进行下去。谁先把“皇后”走进标有五星的左下角，谁就算赢了。

这种博弈游戏是不可能下和棋的。现在我们先来看一个 8 ×8格的棋盘，也就是标准的国际象棋棋盘（见图54-3）。非常明显，如果乙方把“皇后”走入图中的荫影线部分，那么甲方就可以把“皇后”一步移到左下角的终点，他就赢了。可是，乙方也不是傻瓜，不能设想他会心甘情愿地把“皇后”放在图中的荫影线区域里。因此，甲方的目标就是怎样才能迫使乙方非把皇后放在图上的荫影区域去不可。

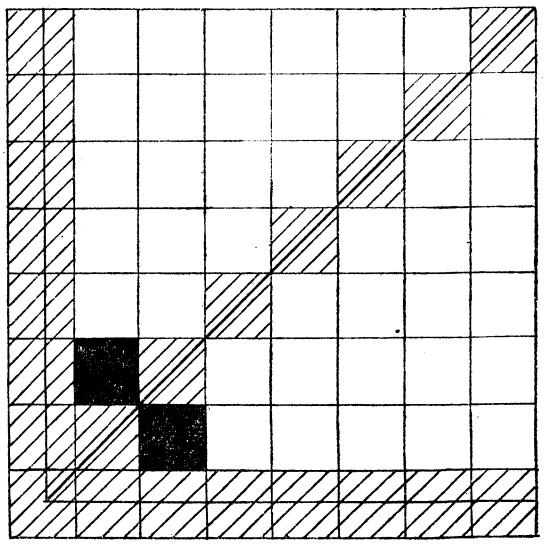


图 54-3

由图54-3可见，如果甲方能把皇后放进最靠近终点的两个黑方格中去，那末，无论是两者中的哪一个，下一步轮到乙方走棋的时候，乙方已没有选择的余地，只好把皇后走入荫影线区域里。由此可见，这两个黑方格就好比是军事上的“制高点”，谁抢先占领了它，只要下面的步子得当，谁就能稳稳取胜。

那末，怎样才能占领这对“制高点”呢，请你依次倒推。在图54-2上可以采用坐标的记法。它们是：

（1，2），（3，5），（4，7），（6，10），（8，13），（9，15），（11，18）。

上述数组，将纵横坐标的位置交换一下就得出了全部有利位置，最后取胜。请你进行二人博弈的程序设计。

55 由双方共同指定扑克牌总数

有一迭纸牌，两个选手参加比赛。第一个选手可取任意张，但不得取全部。以后两个选手交替取牌，每个人均可取一张或一张以上，但取的牌数不得多于对方刚取的数目的一倍。谁取得最后一张谁胜。

（一）笔算步骤——省略，可见设计思路

题目并未限定纸牌总数目，也不由一方给定总数目，所以，和别的题不同，而是由玩牌双方各指定一个数目，总数目是二者之和的平均。只限制轮流取牌数量不得超过一方刚取走数量的一倍，因此玩时，随时留心观察取法，结局可能各有胜负，编写程序试试二人的智力。

（二）程 序 设 计

1）设计思路

（1）该迭牌数目由甲、乙二人共同指定，若甲、乙分别指定的值为A、B，则取N=(A+B)/2。

（2）按游戏规则进行检查，对符合规则的输入，则从N中减去。若N=0，则说明刚才取牌的一方胜。

（3）若其输入不符合规则，则输出相应的信息，并请求重新输入。

（4）程序中玩了两局，都是后取者胜，（取牌过程和结局，详见输出打印）。请你再来一局，试试谁胜。

2）框图　见图55

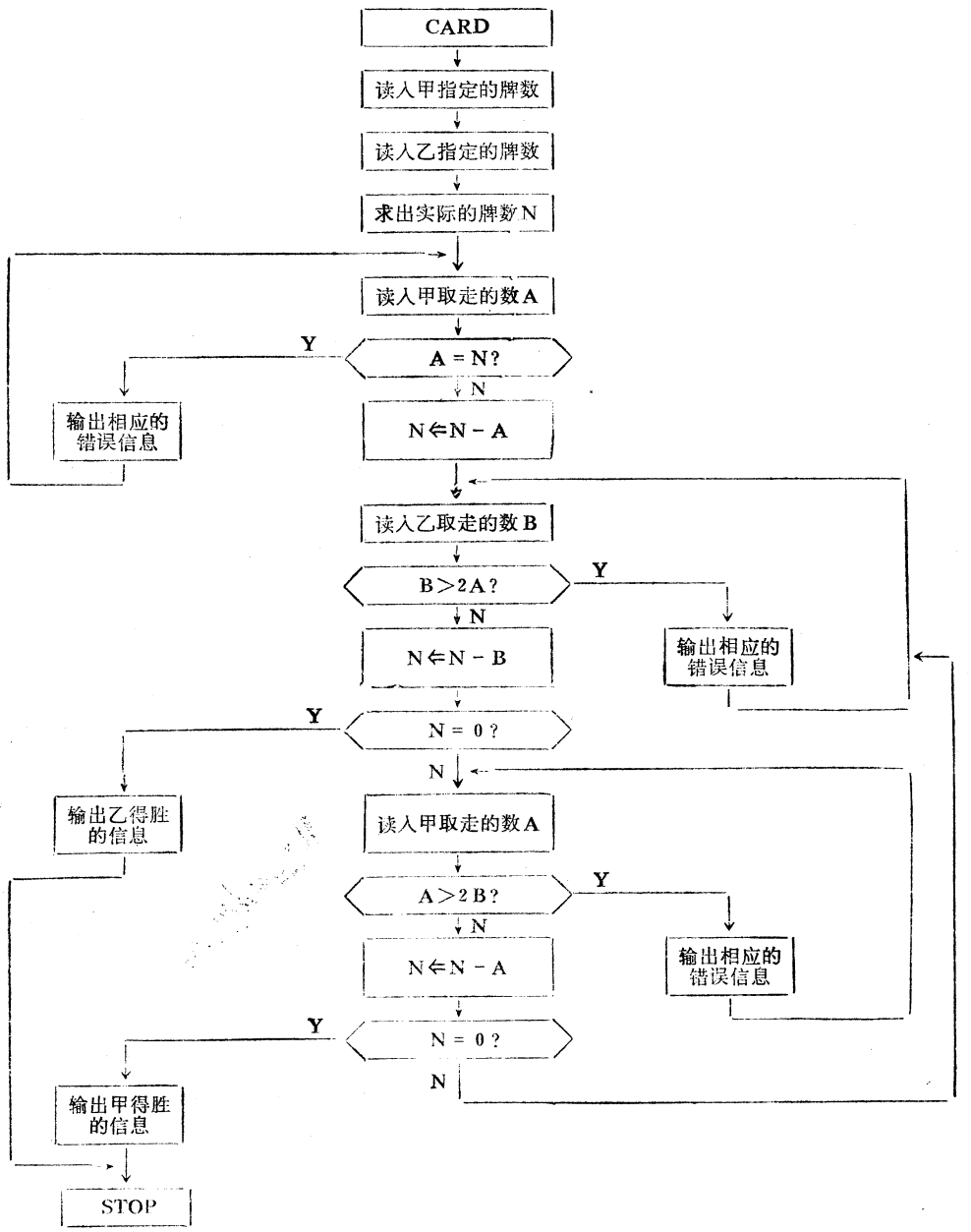


图 55

3）源程序及运行结果

```
C  ***  N CARDS FOR TWO PLAYING
        INTEGER A, B, N
        WRITE (1, 10)
 10     FORMAT (5X, 'TYPEING THE NUMBER OF THE DECK SPECIFIED BY
           FIRST PLAYE 1R', /)
        READ (1, 20) A
 20     FORMAT (I4)
        WRITE (1, 30)
 30     FORMAT (5X, 'TYPEING THE NUMBER OF THE DECK SPECIFIED
           BY SECOND PLAY 1ER', /)
        READ (1, 20) B
        N=INT ( (A+B) /2)
        WRITE(1,40)N
 40     FORMAT (5X, 25HTHE NUMBER OF THE DECK IS, 2X, I4, /)
 45     WRITE (1, 50)
 50     FORMAT (5X, 42HHOW MANY CARDS TAKEN AWAY BY FIRST
           PLAYER? , /)
        READ (1, 20) A
        IF (A. LT. N) GO TO 70
C  ***  WRONG, ONCE AGAIN
        WRITE (1, 60)
 60     FORMAT (5X, 41HTOO MUCH YOU TAKEN AWAY, ONCE AGAIN
           PLEAS, /)
        GO TO 45
 70     N = N - A
C  ***  TURN TO SECOND
 75     WRITE (1, 80)
 80     FORMAT (5X, 43HHOW MANY CARDS TAKEN AWAY BY SECOND
           PLAYER? , /)
        READ (1, 20) B
        IF (B. LE. 2*A) GO TO 90
 85     WRITE (1, 60)
        GO TO 75
 90     IF (N-B. LT. 0) GO TO 85
        N = N - B
        IF (N. EQ. 0) GO TO 130
C  ***  TURN TO FIRST AGAIN
 95     WRITE (1, 50)
        READ (1, 20) A
        IF (A. LE. 2*B) GO TO 100
 98     WRITE (1, 60)
        GO TO 95
```

```
100     IF (N-A. LT. 0) GO TO 98
        N = N - A
        IF (N. EQ. 0) GO TO 110
        GO TO 75
110     WRITE (1, 120)
120     FORMAT (5X, 37HTHE LAST CARD WAS TAKEN AWAY BY
           FIRST, /)
        GO TO 150
130     WRITE (1, 140)
140     FORMAT (5X, 38HTHE LAST CARD WAS TAKEN AWAY BY
           SECOND, /)
150     STOP
```

（第一局）

```
TYPEING THE NUMBER OF THE DECK SPECIFIED BY FIRST
   PLAYER
```

（送入甲指定的牌数）

```
15
TYPEING THE NUMBER OF THE DECK SPECIFIED BY SECOND
   PLAYER
```

（送入乙指定的牌数）

```
25
THE NUMBER OF THE DECK IS 20
```

（该选牌数为20张）

```
HOW MANY CARDS TAKEN AWAY BY FIRST PLAYER?
```

（甲取走多少张？）

```
4
HOW MANY CARDS TAKEN AWAY BY SECOND PLAYER?
```

（乙取走多少张？）

```
3
HOW MANY CARDS TAKEN AWAY BY FIRST PLAYER?
```

（甲取走多少张？）

```
4
HOW MANY CARDS TAKEN AWAY BY SECOND PLAYER?
```

（乙取走多少张？）

```
1
HOW MANY CARDS TAKEN AWAY BY FIRST PLAYER?
```

（甲取走多少张？）

```
2
HOW MANY CARDS TAKEN AWAY BY SECOND PLAYER?
```

（乙取走多少张？）

```
1
HOW MANY CARDS TAKEN AWAY BY FIRST PLAYER?
```

（甲取走多少张？）

```
2
HOW MANY CARDS TAKEN AWAY BY SECOND PLAYER?
(乙取走多少张？)
3
THE LAST CARD WAS TAKEN AWAY BY SECOND
(最后一张牌是乙取走的，乙胜)
          (第二局)
TYPEING THE NUMBER OF THE DECK SPECIFIED BY FIRST PLAYER
(输入甲指定的牌数)
20
TYPEING THE NUMBER OF THE DECK SPECIFIED BY SECOND
   PLAYER
(输入乙指定的牌数)
21
THE NUMBER OF THE DECK IS 20
(该迭牌数为20张)
HOW MANY CARDS TAKEN AWAY BY FIRST PLAYER?
(甲取走多少张？)
4
HOW MANY CARDS TAKEN AWAY BY SECOND PLAYER?
(乙取走多少张？)
3
HOW MANY CARDS TAKEN AWAY BY FIRST PLAYER?
(甲取走多少张？)
4
HOW MANY CARDS TAKEN AWAY BY SECOND PLAYER?
(乙取走多少张？)
2
HOW MANY CARDS TAKEN AWAY BY FIRST PLAYER?
(甲取走多少张？)
3
HOW MANY CARDS TAKEN AWAY BY SECOND PLAYER?
(乙取走多少张？)
7
TOO MUCH YOU TAKEN AWAY, ONCE AGAIN PLEAS
HOW MANY CARDS TAKEN AWAY BY SECOND PLAYER?
(乙要取的牌数太多，超过甲刚才取的二倍，重新输入)
5
TOO MUCH YOU TAKEN AWAY, ONCE AGAIN PLEAS
HOW MANY CARDS TAKEN AWAY BY SECOND PLAYER?
(乙要取的牌数太多，超过了该迭牌余下的总数，重新输入)
1
HOW MANY CARDS TAKEN AWAY BY FIRST PLAYER?
```

(甲取走多少张？)

3

TOO MUCH YOU TAKEN AWAY, ONCE AGAIN PLEAS
HOW MANY CARDS TAKEN AWAY BY FIRST PLAYER?
(甲取走的牌数太多，重新输入)

1

HOW MANY CARDS TAKEN AWAY BY SECOND PLAYER?
(乙取走多少张？)

2

THE LAST CARD WAS TAKEN AWAY BY SECOND
(乙取走最后一张，乙胜)

注：左边一列数字均为甲、乙二人分别从键盘输入的数字

(三) 思 考 题

本题虽未论述取牌方法，但却是个有趣的数学问题。试问，当程序告诉你该迭牌的实际数目N，且由你作为甲首先取牌，你知道第一次应取多少张才能获胜吗？

56 二 人 博 弈

一个方形棋盘分成9个方格，甲乙两人交替地在空白的方格里分别放自己的棋子。设甲有若干红色棋子，乙有若干白色棋子。如果某个人先占有一行或一列或一条对角线的全部三个棋子，则为胜。试给出摸拟这种下棋的算法和程序设计。

(一) 算法分析和结论

据得胜的条件先看经过各格的直线总数：中心一格是4条，故中心写上“4”（见图56-1），角上一格是3条，故写上“3”，边上一格是2条，故写上“2”。（若先填一方以“1”表之，后填一方以“－1”表示）。第一个“1”应画在当中（见图56-2），第一个“－1”应画在角上。这样看来，先填者似乎可以取胜。其实不然，因为后填者还可设法阻止先填者联成一线，如图56-2中局势，后填者便在左右上角方格填“－1”，使先填者不能沿对角线联成一线）。一般情况是：如果双方都掌握了其中的奥妙，此竞赛便成和局，如果先填一方掌握规律，后填者稍有不慎，先填者便有取胜的可能。反之，后填一方掌握规律。先填者随机乱填，后填一方便可乘机取胜，至少也不会输掉，变为和局。

3	2	3
2	4	2
3	2	3

图 56-1

－1	1	－1
	1	
1		

图 56-2

(二) 程 序 设 计

1) 设计思路

甲乙双方，计算机为甲方，它的棋子为1，你是乙方，你的棋子为－1，空白棋格为0。棋盘为一个3×3的数组。

总的策略是，整体优先值判断法。把各行、列、对角线依次检查，每个点赋给一个优

先值，赋值的原则是，每点都有自己的基值，即是过其所在位置的可能的直线数目（见图56-1），棋盘上各点基值构成一个3×3的数组，即B（3，3）。每行（或列、对角线）若有一个你的棋子或一个计算机棋子，则此行（或列、对角线）上的空格的优先值加4；若有两个你的棋子，则加15；若有两个计算机棋子则加40；（上述加4、15、40都是在原棋盘格上优先数基值基础上加的）。若某直线上有你三个子，VOCT置1，你胜；若计算机先占有直线上三个位置，VOCT置－1，计算机胜。每个点的优先值存于数组A（3，3，2）中，其中没有棋子的空白点的优先值，据上述不同着子位置，加不同数值，一旦该格着落棋子，便将该格（点）的优先值置0，因为不再使用该格（点）了。

计算出各空白点的优先值之后，找出一个最大优先值的点，即得胜可能性最大的着子点，计算机把该点赋值为1（着子）。在计算优先值的过程中同时检查有无三个同类棋子在同一行（列、对角线）上的，若有，要判出是哪一方占领，并打印出来。若没有，则由你走下一步。若一直走到没有空格位子时，还没分胜负，这时VOCT单元里为0，则打印出0，即为和局标志。若打印出VOCT值为－1，是计算机胜，若打印出1，是你胜。程序中举一例，请见程序运行时打印结果。

2）框图　见图56-3

3）源程序及运行结果

```
C       PROGRAM  XQ
        INTEGER  A, B, Z, P, V, E, F, VOCT
        DIMENSION  A(3, 3, 2) , B(3, 3) ,                B存放优先数基数
           E (3, 2)
        COMMON  /XQ1/B
        DATA  B/3, 2, 3, 2, 4, 2, 3, 2, 3/
3       DO  4  I2=1, 3                                   A（I，J，1）表示棋盘九
                                                         格，先清0。机器着子为数1.
                                                         你着子为－1。
        DO  4  I1=1, 3
4       A(I1, I2, 1) =0                                  A（I，J，2）表示棋盘九
                                                         格优先数。
        DO  6  I1=1, 3
        DO  6  I2=1, 3
6       A (I1, I2, 2) =B(I1, I2)
        F=0
        VOCT=0                                           棋盘纵横各三条直线，双对角
                                                         直线，共8条，填满时，F=
                                                         8。
        LCON=1                                           指示器LCON判谁走，1时机
                                                         器走，－1时你走。
5       DO  200  K=1, 4                                  K为：  1  2  3  4
        DO  190  I=1, 3                                  I表示：行  列  正对  逆对
                                                                  角线  角线
        IF (I. GT. 1. AND. K. GT. 2) GOTO  190
```

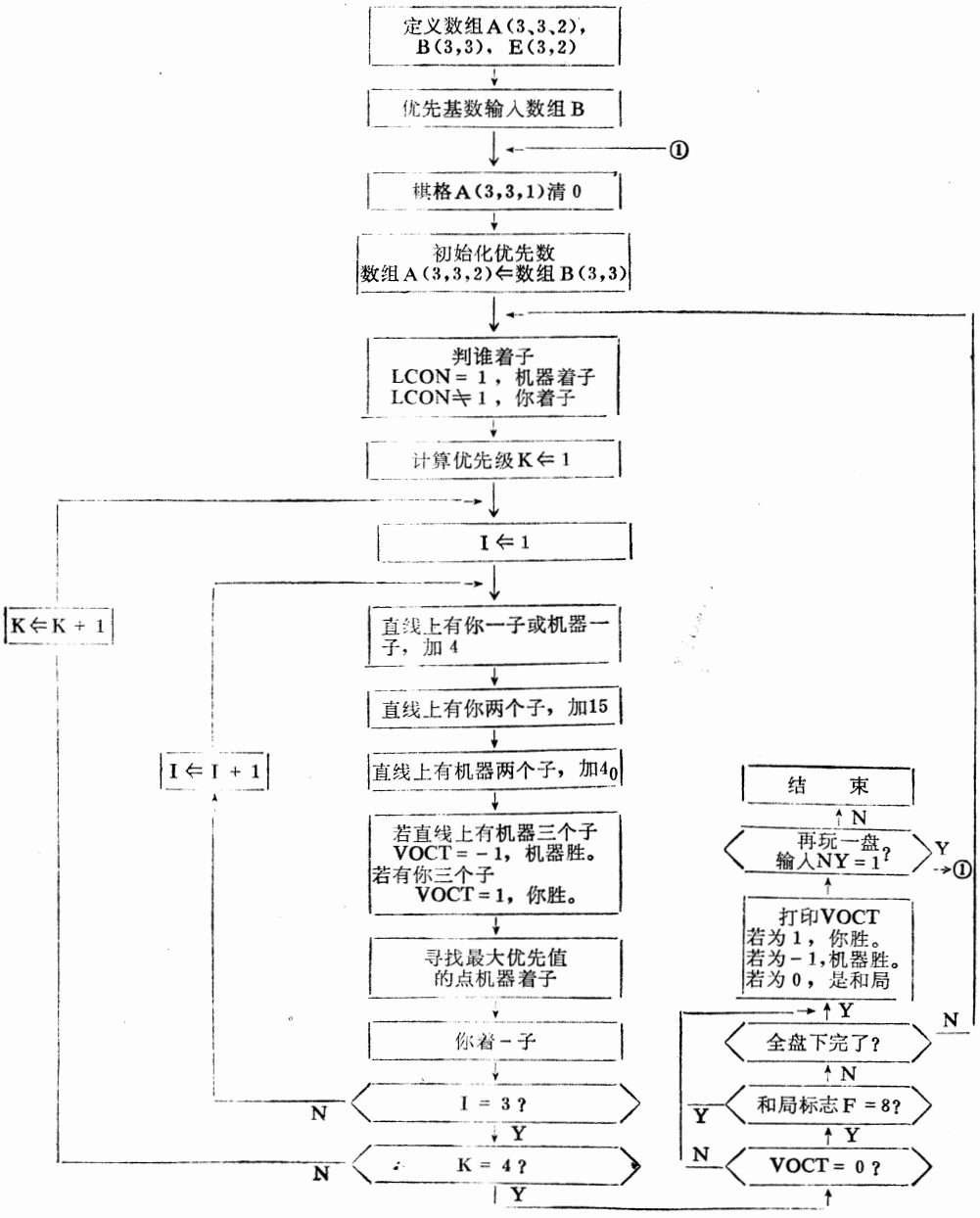


图 56-3

```
      N = 0                                  同一直线上着子计数器，N为你，
      P = 0                                  P为机器，Z为空格。
      Z = 0
      DO   8   J1 = 1, 3                     工作单元E清0。
      DO   8   J2 = 1, 2
8     E(J1, J2) = 0
      DO   80  J = 1, 3                      一直线上三个格着子情况，分
```

```
                                                    别给N、P、Z赋值
      GOTO (10, 12, 14, 16) K                       据K值不同，行列在变化。
10    II = I
      JJ = J
      GOTO  18
12    II = J
      JJ = I
      GOTO  18
14    II = J
      JJ = J
      GOTO  18
16    II = J
      JJ = 4 - J
18    M = A(II, JJ, 1)                              ]判该格着子情况
      IF (M) 20, 25, 30                             ]
20    N = N + 1
      GOTO  80
25    Z = Z + 1
      E(J, 1) = II                                  ]保留空格位置
      E(J, 2) = JJ                                  ]
      GOTO  80
30    P = P + 1
80    CONTINUE
      V = 0                                         优先数累加值V
      IF ( (N+P. EQ. 3) . AND. (N. NE. 3) .         该直线已填满，且该线和局。
         AND. (P. NE. 3) ) GOTO  180
      IF (N. EQ. 0) GOTO  87
      GOTO (82, 84, 86) N                           据你在该直线上着子几个，加
                                                    不同的优先数。
82    V = V + 4                                     你着子一个，加 4 。
      GOTO  87
84    V = V + 15                                    你着子两个，加15。
      GOTO  87
86    VOCT = 1                                      你着子三个，VOCT = 1，你
                                                    胜。
87    IF (P. EQ. 0) GOTO  95
      GOTO (88, 89, 90) P                           据机器在该直线上着子几个，
                                                    加不同的优先数。
88    IF (V. NE. 4) V = V + 4                       机器着一子，加 4 。
      GOTO  95
89    V = V + 40                                    机器着两子，加40。
      GOTO  95
90    VOCT = -1                                     机器着三子，VOCT = - 1，
```

```
                                                    机器胜。
 95     DO  100  L1=1, 3                            ┐
        IF (E (L1, 1) . EQ. 0) GOTO  100            │
        II=E (L1, 1)                                │
        JJ=E (L1, 2)                                │ 改变优先数
        IF (A(II, JJ, 2) . EQ. 0 ) GOTO  100        │
        A(II, JJ, 2) =V+A (II, JJ, 2)               │
100     CONTINUE                                    ┘
        GOTO  190
180     F = F + 1                                   已填满的直线F加1
190     CONTINUE
200     CONTINUE
        IF (VOCT. NE. 0) GOTO  400                  有胜者
        IF (F. EQ. 8) GOTO  400                     已布满子，和局。
        IF (LCON. NE. 1) GOTO  300                  你走（着子）
        LCON= -1                                    机器走，下次该你走。
        M1=0
        DO  250  I1=1, 3                            ┐
        DO  250  I2=1, 3                            │
        IF (M1. GE. A(I1, I2, 2) ) GOTO  250        │ 找最大优先数M1及其位置
        M1=A(I1, I2, 2)                             │ (II, JJ)
        II=I1                                       │
        JJ=I2                                       │
250     CONTINUE                                    ┘
        A(II, JJ, 1) =1                             机器着子，数1表示
        A(II, JJ, 2) =0                             着子后，该格优先数M1清0，
                                                    不再用。
        WRITE (10, 450)  ( (A (I1, J1, 1) , J1=1,   打印你和机器着子布局
           3) , I1=1, 3)
        DO  260  I1=1, 3                            ┐
        DO  260  J1=1, 3                            │
        IF (A (I1, J1, 1) . EQ. 0) GOTO  5          │ 判全盘下完否
260     CONTINUE                                    ┘
        GOTO  400                                   未完，转400句
300     LCON=1                                      未完，下次机器走。
        READ (11) II, JJ                            你着子（走），从终端输着子
                                                    的格子位置（行、列号）。
        A(II, JJ, 1) = -1                           你着子用数－1
        A(II, JJ, 2) =0                             着子后该格优先数清0，不再
                                                    用。
        WRITE (10, 450) ((A (I1, J1, 1), J1=1, 3,)  打印你和机器着子布局
        I1=1, 3)
        GOTO  5
```

```
400       WRITE (10, 420) VOCT                          全盘结束打印，1为你胜，
                                                        -1为机器胜，0为和局。
420       FORMAT (1X, 5HVOCT=, I3)
450       FORMAT (1X, 3I5/)
          WRITE (10, 460)
460       FORMAT (1X, 14HLET  US  AGAIN! )              若输入NY=1，再继续玩。
          READ (11, 470) NY                             NY≠1，则停。
470       FORMAT (I2)
          IF (NY. EQ. 1) GOTO  3
          STOP
          END
```

下面是举一例子

```
    0     0     0
    0     1     0      首先机器选优先数基数最大的中心格填一子“1”
    0     0     0
1                      你选择在1行1列填子
1
   -1     0     0
    0     1     0      第二次你选择在（1，1）位置填一子“-1”
    0     0     0
   -1     0     1
    0     1     0      第三次机器选在（1，3）位置填一子“1”
    0     0     0
3                      你选择在3行1列填子
1
   -1     0     1
    0     1     0      第四次你选择在（3，1）位置填一子“-1”
   -1     0     0
   -1     0     1
    1     1     0      第五次机器选在堵住你能成直线
   -1     0     0      的位置（2，1）填一子“1”
2                      你选择在2行3列填子
3
   -1     0     1
    1     1    -1      第六次你选在（2，3）位置填一子“-1”
   -1     0     0      以便堵住机器能成直线的空格
   -1     0     1
    1     1    -1      第七次机器选在（3，3）位置填一子“1”
   -1     0     1
```

```
1                                    你选在1行2列填子
2
   -1    -1     1
    1     1    -1                    第八次由上观查和局已定
   -1     0     1                    你任选位置（1，2）填一子“-1”
   -1    -1     1
    1     1    -1                    第九次机器在只有一位置（3，2）填一子“1”
   -1     1     1
VOCT=0                               0表示和局
LET US AGAIN!                        再来一局!
0                                    输入0，不玩了。
STOP                                 停
```

（三）思 考 题

15点游戏，又称“井”字游戏。正方形由九个小方格组成，1～9九个数字，有规律地填在小方格里，如图56-4所示。甲、乙双方轮流在写有不同数字的小方格里着红、白棋子，每次放一颗棋子，谁先占据行、列或对角线中三个数字加起来为15，便是赢者。请给出模拟这种游戏的其它算法和程序设计。

8	1	6
3	5	7
4	9	2

图 56-4

（还有些巧妙的文字游戏及网络游戏都是与上述游戏同构的。然而，所有这些游戏都建立在3×3魔方的基础上，这个方阵是最古老的组合方面的奇珍之一。）

九 计 算 机 算 式

57 BC乘BC等于几

两位数B C取何值时，有

$$\begin{array}{r} B\ C \\ \times\ B\ C \\ \hline A\,B\,C \end{array}$$

式中A≠0，可取1～9，B、C可取0、1……9，但A、B、C代表数字不得相同。

（一）笔算步骤和结果

既然两位数B C和它的平方数具有相同的个位数字，因其C只可能取0、1、5和6。因B≠0，故C≠0。又因C＝1时，B C平方的十位数字应等于2 B的个位数字，所以C≠1，C只可能等于5和6。

因为 $35^2=1225$，故B＜3，用15、16、25、26代入上式通过验证知B C将唯一地等于25，且算式为：

$$\begin{array}{r} 2\ 5 \\ \times\ 2\ 5 \\ \hline 6\ 2\ 5 \end{array}$$

（二）程序设计

1）设计思路

利用二重循环语句确定B、C。据分析B、C不可能有0数字，故使循环控制变量由1至9变化，据算式知B C＊B C是三位数字，如果大于三位数字则不符合题意，重取二数，若是三位数字，其末两位应与乘数或被乘数相同，这又是一个判断条件。满足条件后，打印出算式形式及结果。

2）框图　见图57

3）源程序及运行结果

```
      DO  10  IB=1, 9
      DO  10  IC=1, 9
      N=10*IB+IC
      M=N*N
      IF (M. GT. 1000) GOTO  10
      IF (MOD (M, 100) . NE. N) GOTO  10
      WRITE (5, 20) N, N, M
      WRITE (2, 20) N, N, M
20    FORMAT (5X, 2HBC, 5X, I2/, 2X, 'X  ', 2HBC, 2X, 'X  ', I2/
     1  1X, '……', 1X, '……'/, 4X, 3HABC, 4X, I3/)
10    CONTINUE
      STOP
      END
```

```
   BC      25
×  BC   ×  25
.....   .....
  ABC     625
```

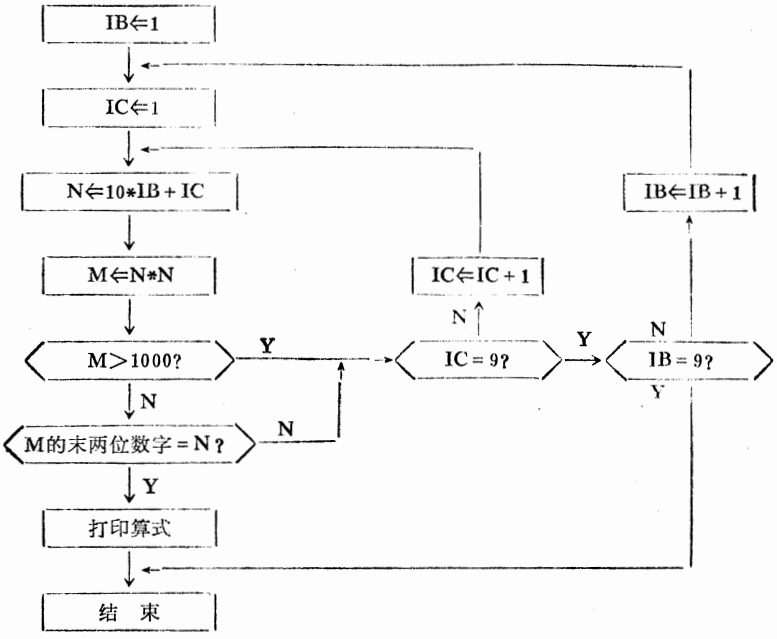


图 57

（三）思 考 题

1）如下算式，找出乘数、被乘数和乘积的十进制的数字。

```
   BC
×  BC
------
  ABBC
```

笔算分析结果为

```
   76
×  76
------
 5776
```

2）如下乘法算式，找出不同字母所代表的十进制不同的数字。

```
   EF
×  EF
------
 GHGE
```

笔算分析结果为

```
   93
×  93
------
 8649
```

58 计算机算术字谜

求出下面算式不同字母代表十进制中不同的一个数字。

```
  ONE
+ ONE
-----
  TWO
```

（一）笔算步骤和结果

这是一道较简单的算术字谜。式中的每一个字母都表示 0 ～ 9 中一个数字，相同的字母代表相同数字，不同字母代表不同的数字，并规定 0 不许出现在最前面，例 0 1 2、0 3 5 等是不允许出现的。通过分析可以知道，当 E = 1，N = 3 时，上式即成立。即当 E = 1，N = 3 时，O = 2，W = 6，T = 4，上式的意思就是 231 + 231 = 462。当然要使上式成立的解不止一个，例如 E = 2，N = 3，也能使上式成立，即 432 + 432 = 864。

（二）程序设计

1）设计思路

该算式有九个字符，其中有五个是不同的字符，用二重循环语句可确定这五个不同的字符所代表的数字，先由个位 E 确定，每确定一个字符时，要判与其它不同字符所确定的数字相同否。要特别注意进位的判别。例如，IE + IE = IO，IN + IN = IW，要判 IO 有否进位，有进位要加在 IW 上。同理判 IW 有否进位，有则加在 IT 上。全确定后，打印出算式和结果。找出一种答案便令其结束。

2）框图　见图58

3）源程序及运行结果

```
      WRITE (5, 70)
      DO  10  IE=1, 9
      IO=IE+IE
      DO  20  IN=0, 9
      IF (IN. EQ. IO. OR. IN. EQ. IE) GOTO  20
      IW=IN+IN
      IF (IO/10) 30, 25, 30
30    IW=IW+1
25    IF (IW. EQ. IO. OR. IW. EQ. IN. OR. IW. EQ. IE) GOTO  20
      IT=IO+IO
      IF (IW/10) 40, 50, 40
40    IT=IT+1
50    IF (IT. EQ. IN. OR. IT. EQ. IW. OR. IT. EQ. IE) GOTO  20
      N=100*IO+10*IN+IE
      M=100*IT+10*IW+IO
      IF (2*N. NE. M. OR. M. GT. 999) GOTO  20
      WRITE (5, 60) N, N, M
70    FORMAT (8X, '  ONE'/, 8X, '+ONE'/, 7X, '.....'/, 8X, '  TWO')
60    FORMAT (8X, I5/, 8X, '+', I4/, 7X, '.....'/, 8X, I5/)
20    CONTINUE
```

```
10      CONTINUE
        STOP
        END
```

```
      O N E
+     O N E
.....................
      T W O
      2 3 1
+     2 3 1
.....................
      4 6 2
```

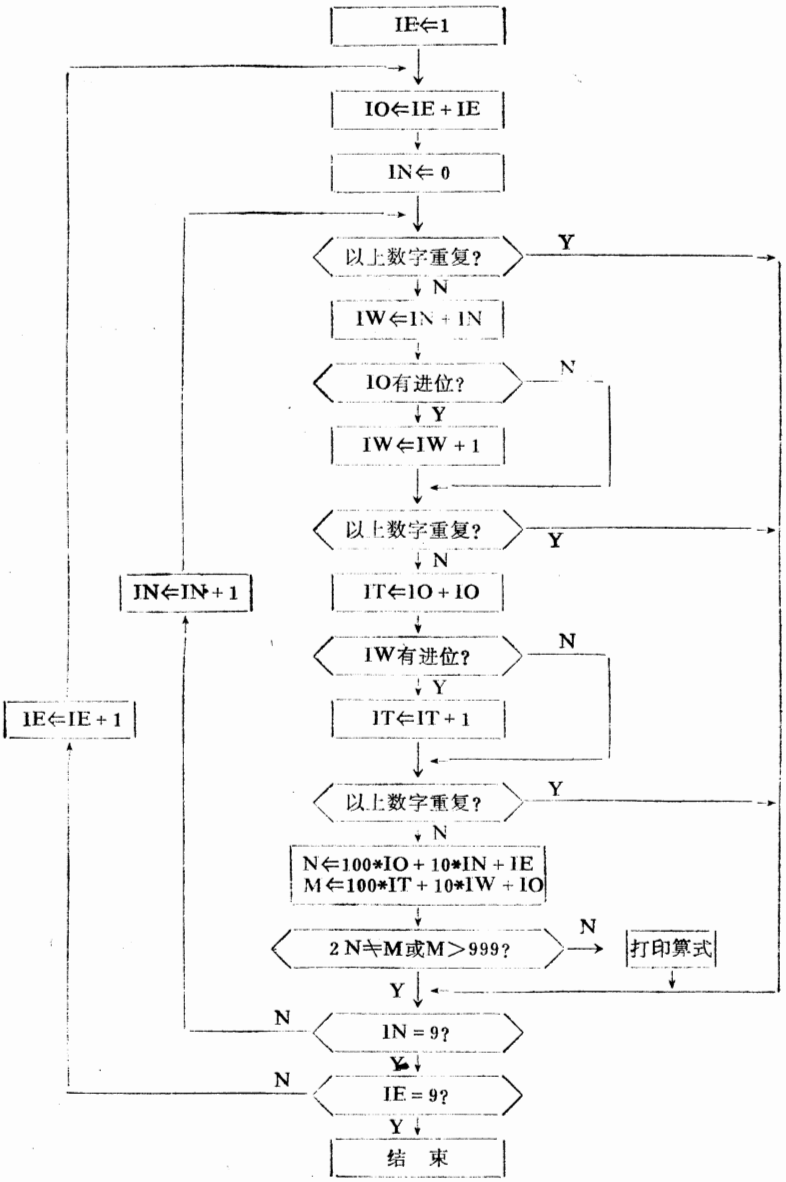


图 58

（三）思 考 题

1）有一个年份数，它与由它的数字倒排所成的数的和为8657，且已知它的最后两位数字相等，求证这个年份必为1966。

笔算分析提示

$$\begin{array}{r} A\,B\,C\,C \\ +\ C\,C\,B\,A \\ \hline 8\,6\,5\,7 \end{array}$$

2）计算加法算式

$$\begin{array}{r} \quad\quad A \\ M\,A\,T\,H \\ M\,A\,G\,S \\ +\ M\,A\,T\,H \\ \hline G\,A\,M\,E \end{array}$$

单词G A M E（游戏）表示一个素数。试求不同字母所表示的不同数字。

笔算分析结果——字母M应表示集合｛1，2，3｝中的一个元素，字母A应表示集合｛4，5，9｝中一个元素，由此推出字母G等于集合｛4，7，8｝中的某个元素。看一看素数表，发现只有8923才是唯一合适的素数，然后容易把整个密码译出：

$$\begin{array}{r} \quad 9 \\ 2\,9\,6\,7 \\ 2\,9\,8\,0 \\ +\ 2\,9\,6\,7 \\ \hline 8\,9\,2\,3 \end{array} \qquad \text{或} \qquad \begin{array}{r} \quad 9 \\ 2\,9\,6\,5 \\ 2\,9\,8\,4 \\ +\ 2\,9\,6\,5 \\ \hline 8\,9\,2\,3 \end{array}$$

59 除 法 游 戏

在下述除法游戏中，每个字母唯一地代表一个十进制数字

$$\frac{A\,H\,H\,A\,A\,H}{J\,O\,K\,E} = H\,A$$

试求出所有字母代表的十进制数字。

（一）笔算步骤和结果

改写原式为

$$J\,O\,K\,E = \frac{A\,H\,H\,A\,A\,H}{H\,A} = 100 + \frac{A\,H\,(10001)}{H\,A}$$

进而将10001唯一地分解为素因数之乘积，得

$$10001 = 73 \times 137$$

现在考察$\frac{A\,H\,(10001)}{H\,A}$。由题意，它必须是整数，注意到10001是五位数，所以借10001的分解式知，两位数H A = 73，最后，上述除法应该是$\frac{377337}{73} = 5169$。

（二）程 序 设 计

1）设计思路

算式左端分子有六个字符，分母有两个字符，共八个字符，但却由两个字符A和H排列组合成除法算式。用二重循环，先确定A和H，循环参数的初值为1，终值为9，增量为1。A和H必须代表不同数字，同时算式左端相除后所得的商，应是四位整数，因为算式右端是由四个字符组成。定义数组K（6），将四位数字的商由低位至高位分别置入K（1）～K（4），将算式左端的A和H置入K（5）和K（6）。然后判数组K（6）中六个数字应该互不相同，找对时，停在STOP OK。否则，停在STOP ERROR。

2）框图 见图59

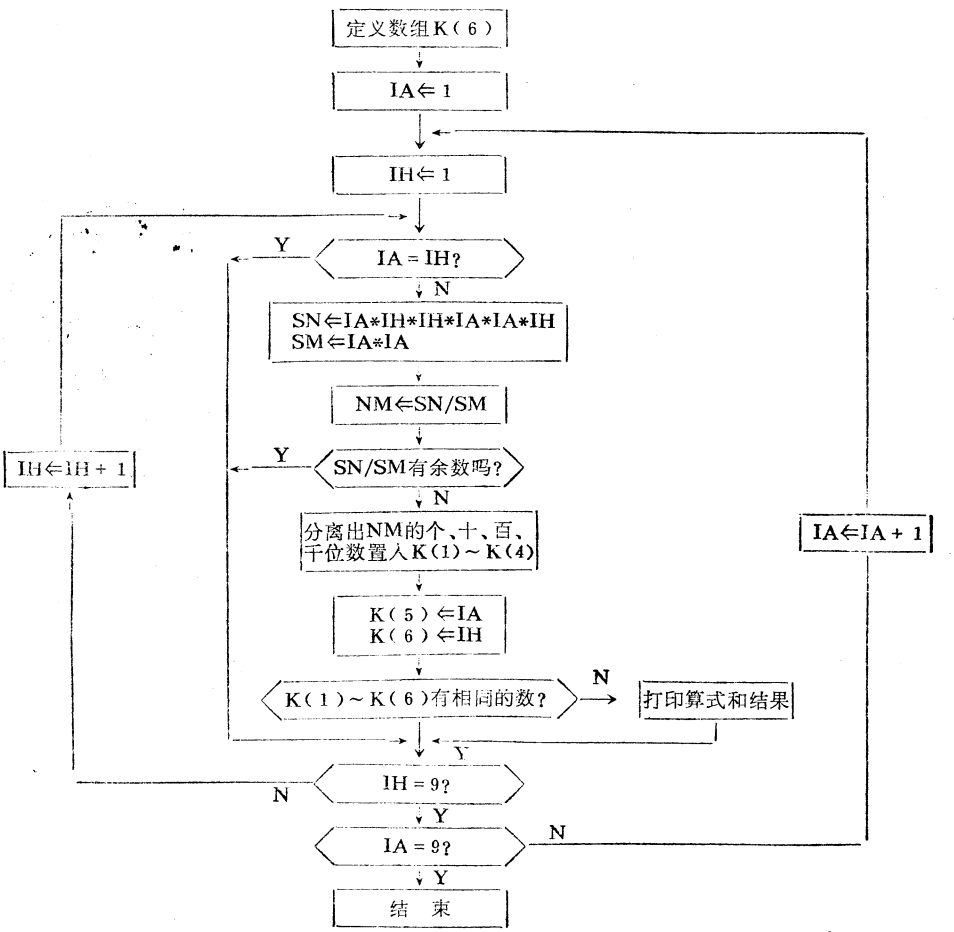


图 59

3）源程序及运行结果

```
DIMENSION  K(6)
DO  10  IA = 1, 9
DO  20  IH = 1, 9
IF (IA. EQ. IH) GOTO  20
SN = 100000. *IA + 10000. *IH + 1000. *IH + 100. *IA + 10. *IA + IH
SM = 10. *IH + IA
```

```
      NM=INT (SN/SM)
      IF ( (SN/SM-FLOAT (NM) ) . GT. 0.00001) GOTO  20
      DO  30  I=1, 4
30    K(I) = (NM-NM/10**I*10**I) /10** (I-1)
      K(5) =IA
      K(6) =IH
      DO  40  I=1, 5
      I1=I+1
      DO  40  J=I1, 6
      IF  (K(I) . EQ. K(J) )  GOTO  20
40    CONTINUE
      WRITE (5, 50)  IA, IH, IH, IA, IA, IH, IH, IA, NM
50    FORMAT (8X, 20HAHHAAH/HA=JOKE- - ->, 6I1, 1H/, 2I1, 1H=,
       I4///)
      STOP  OK
20    CONTINUE
10    CONTINUE
      STOP  ERROR
      END
```

程序运行后打印结果

```
AHHAAH/HA=JOKE- - ->377337/73=5169
```

（三）思 考 题

烤火腿——在英语“烤火腿”：“F R Y H A M”一词中，让每个字母代表十进制数，其中H=F+1，请从关系式7（F R Y H A M）=6（H A M F R Y）中，确定这些数字。

笔算分析和程序设计等，详见《趣味程序设计100例》第43题。

60 古诗词算式

写在一张纸上的算式被蛀虫咬得支离破碎，字迹看不清楚，但利用四则运算的道理，通过逻辑推理，可以把原来的算式剖析出来，这就是经常为人称道的“虫食算”。它从创立到现在，已有相当一段历史，有人把中国古代诗词中的名言佳句与虫食算结合起来，制作了一些风格优异的小品，下面就是其中之一：

年年×岁岁=花相似　　（1）

岁岁÷年年=人÷不同　　（2）

上面两个算式中的各个汉字分别代表0～9十个数码。相同的汉字表示相同的数码，不同的汉字表示不同的数码。请你求出上面两个算式？

（一）笔算步骤和结果

为了叙述方便，可将这两个算式编上号。依次编为（1）式和（2）式。

由（2）式的右边可知，这一除式的商数是真分数，即分子比分母小。可见岁岁＜年年，

两个重迭文字所表示的二位数只能是11，22，33…99。据（1）可知，“岁岁”不能是11，因为任何数乘上1均是其本身，若“岁”是1，那末乘积的末位数应是年，而绝对不会是“似”。

如果“岁岁”是33，则因岁岁＜年年；所以“年年”至少应是44，然而$33\times44=1452$，是四位数，与（1）式矛盾。由此可知，岁岁只能是22，而年年只能是33或44（∵年年若是55，它与22的乘积已达四位数，就与（1）式相矛盾）。

又由于$22\times33=726$，这个三位数乘积中的一个数字是2，与被乘数中的一个数字重复，按条件它们用同一数字表示，但（1）式中的被乘数和乘积中没有相同的汉字。因此年年只能是44，于是这两道算式为：

$$44\times22=968$$

$$22\div44=5\div10$$

（二）程序设计

1）设计思路

我们将这富有诗意的算式，除年用Y代替外，其余用汉语拼音第一个字符替代该汉字。即每个字符代表一个汉字，形成两个互有联系的算式：

$$YY*SS=H\times Z \qquad (1)$$

$$YY/SS=BT/R \qquad (2)$$

利用循环语句确定这些字符。首先确定（1）式S和Y，除S和Y代表不相同数字外，还要使YY＊SS等于三位数，将这三位数的个位、百位、千位数字分别赋给（1）式右端的Z、X、H，使得Y、S、H、X、Z，所代表的数字都不相等。从（1）式暂定了Y和S，便暂算出（2）式左端，利用循环语句确定（2）式右端R、B、T，除代表数字互不相同外，还要与（2）式左端的商数相等。最后打印出算式和结果。

2）框图　见图60

3）源程序及运行结果

```
WRITE (5, 70)
DO  10  IS=2, 9
DO  20  IY=2, 9
IF (IS. EQ. IY)   GO  TO  20
IYY=10*IY+IY
ISS=10*IS+IS
N=IYY*ISS
IF (N/1000. NE. 0. OR. N/100. LT. 1) GOTO  20
IZ=MOD (N, 10)
IX=MOD (N, 100) /10
IH=N/100
IF (IZ. EQ. IX. OR. IZ. EQ. IH. OR. IX. EQ. IH) GOTO  20
IF (IZ. EQ. IY. OR. IZ. EQ. IS)   GOTO  20
IF (IX. EQ. IY. OR. IX. EQ. IS)   GOTO  20
IF (IH. EQ. IY. OR. IH. EQ. IS)   GOTO  20
M=IYY/ISS
DO  30  IR=1, 9
```

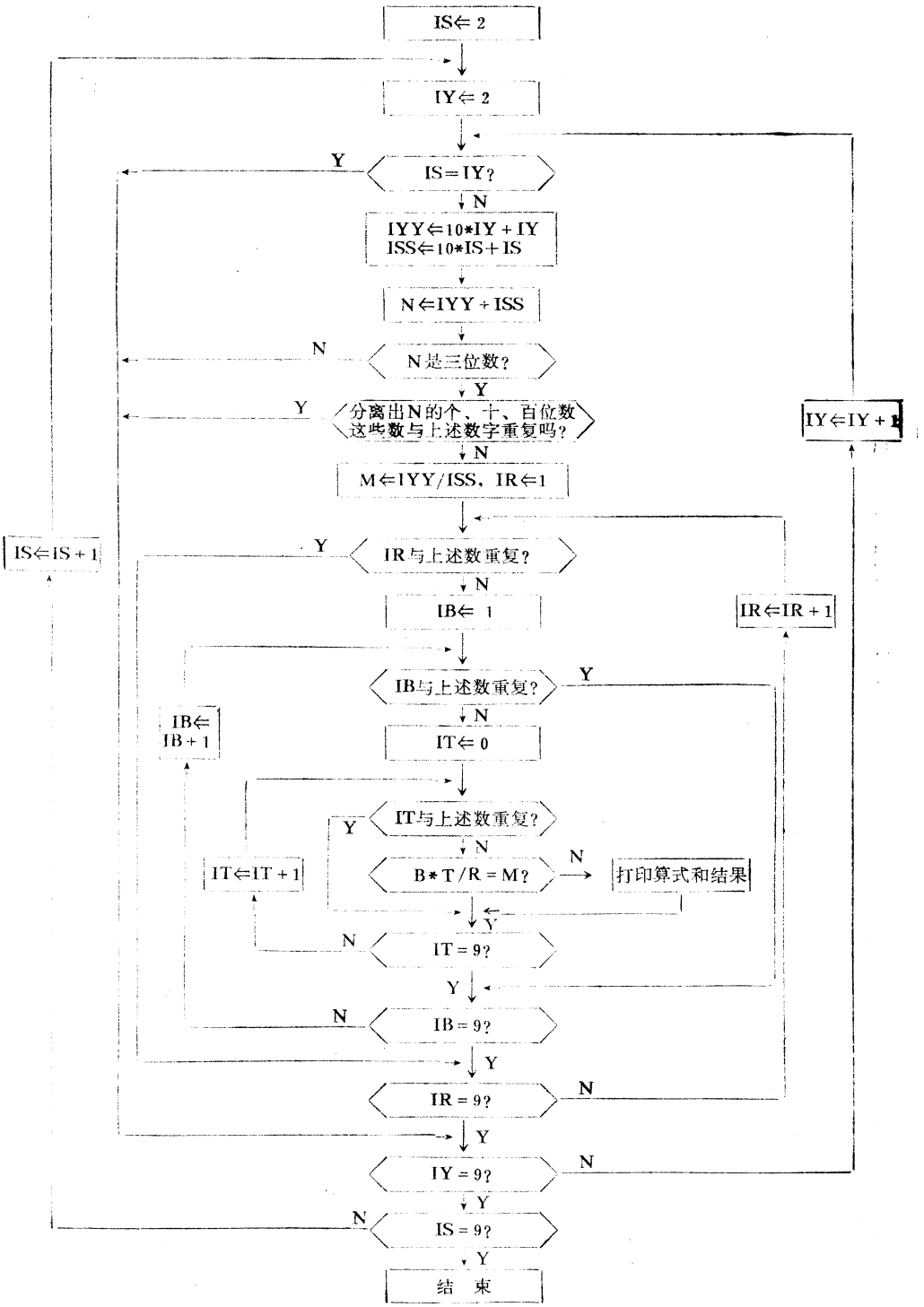

IS⇐2
IY⇐2
IS=IY?
Y
N
IYY⇐10*IY+IY
ISS⇐10*IS+IS
N⇐IYY+ISS
N是三位数?
分离出N的个、十、百位数
这些数与上述数字重复吗?
M⇐IYY/ISS, IR⇐1
IR与上述数重复?
IB⇐1
IB与上述数重复?
IT⇐0
IT与上述数重复?
B*T/R=M?
打印算式和结果
IT⇐IT+1
IT=9?
IB⇐IB+1
IB=9?
IR=9?
IR⇐IR+1
IY=9?
IY⇐IY+1
IS=9?
IS⇐IS+1
结 束

图 60

```
      IF (IR. EQ. IY. OR. IR. EQ. IS) GOTO  30
      IF (IR. EQ. IZ. OR. IR. EQ. IX. OR. IR. EQ. IH) GOTO  30
      DO  40  IB=1, 9
      IF (IB. EQ. IY. OR. IB. EQ. IS. OR. IB. EQ. IR) GOTO  40
      IF (IB. EQ. IZ. OR. IB. EQ. IX. OR. IB. EQ. IH) GOTO  40
      DO  50  IT=0, 9
      IF (IT. EQ. IB. OR. IT. EQ. IR. OR. IT. EQ. IY. OR. IT. EQ. IS)
        GOTO  50
      IF (IT. EQ. IZ. OR. IT. EQ. IX. OR. IT. EQ. IH) GOTO  50
      IBT=10*IB+IT
      BT=FLOAT (IBT)
      R=FLOAT (IR)
      IF (ABS (BT/R-FLOAT (M)) . GT. 0.00001) GOTO  50
      WRITE (5, 60) IYY, ISS, N, IYY, ISS, IBT, IR
60    FORMAT (8X, I2, '*', I3, '=', I4/, 8X, I2, '/', I3, '=', I3,
      '/', I2)
70    FORMAT (8X, 'YY  *  SS=HXZ'/, 8X, 'YY/SS=BT/R')
50    CONTINUE
40    CONTINUE
30    CONTINUE
20    CONTINUE
10    CONTINUE
      END

      YY  *  SS=HXZ
      YY  /  SS=BT/R
      44  *  22=968
      44/22=10/5
```

（三）思 考 题

辩认淋湿的数字——雨淋湿了算术本上一道题，8个数字只能看清3个（第1个数字虽然模糊不清楚，但可看出不是1）。

$$(□\times(□3+□))^2=8□□9$$

其余的数字请帮助找找。

笔算步骤、结果和程序设计等，详见《趣味程序设计100例》第45题

61 计 算 机 算 式

确定下式等号两边的英文字母唯一地代表十进制中的一个数字

$$SIX+SIX+SIX=NINE+NINE$$

（一）笔算步骤和结果

不难推算出，上式结果为

$$942+942+942=1413+1413$$

（二）程 序 设 计

1）设计思路

算式等号左端是三个英文数字 6 相加，右端是两个英文数字 9 相加，左右英文意思都是等于18。从数学角度看也巧妙成立。任务是要推断不同字母所代表的不同数字是什么，虽有五个不同字母，分析后仅用三重循环便可找出。先从等号左端个位数X着手，I X的循环变量由 1 ～ 9，使得 3X = 2E。再从等号左端十位的 I 着手，循环变量由 1 ～ 9 寻找，使得 3I = 2N。注意，个位的 3X 可能有进位JW，进到十位的 3 I 上，所以需要 3I + JW = 2N，依次确定等号左端百位 S。同理，若 3I有进位要加在 3S上，而且3I的进位还可能受到3X进位的影响。故3I要加上前位（个位）的进位 JW后，再往百位进位，形成 3S，要使 3S的个位等于 2I，3S的进位等于 2N。每生成一个数字，要与不同字母所代表的数字不相同。当满足要求时形象地打印出算式与结果。

2）框图　见图61

3）源程序及运行结果

```
      DO  10  IX = 1, 9
      IE = MOD (3*IX, 10)
      IF (IE/2*2. NE. IE)  GOTO  10
      IE = IE/2
      DO  20  II = 1, 9
      IF (II. EQ. IX. OR. II. EQ. IE)  GOTO  20
      JW = 3*IX/10
      IN = MOD (3*II + JW, 10)
      IF (IN/2*2. NE. IN)  GOTO  20
      IN = IN/2
      IF (IN. EQ. IX. OR. IN. EQ. II. OR. IN. EQ. IE) GOTO  20
      DO  30  IS = 1, 9
      IF (IS. EQ. IX. OR. IS. EQ. II. OR. IS. EQ. IE. OR. IS. EQ. IN)
        GOTO  30
      JW =  (3*II + JW) /10
      IF (2*II. NE.  (MOD (3*IS + JW, 10) ) . OR. 2 * IN. NE.  ( (3*IS +
        JW) /10) ) GOTO  30
      ISIX = 100*IS + 10*II + IX
      NINE = 1000*IN + 100*II + 10*IN + IE
      ISIX3 = 3*ISIX
      NINE2 = 2*NINE
      IF (ISIX3. NE. NINE2)  GOTO  30
      WRITE (5, 40) ISIX, ISIX, NINE, ISIX, NINE, ISIX3, NINE2
40    FORMAT (15X, 3HSIX, I6/, 15X, 3HSIX, I6, 5X, 4HNINE, I7/, 12X,
     1  8H      SIX +, I4, 2X, 7H +     NINE, I7/, 10X,
     1  32H  ------------------------------  /, 20X, I4, 10X, I6/)
30    CONTINUE
20    CONTINUE
10    CONTINUE
```

END

```
      S I X     9 4 2
      S I X     9 4 2          N I N E     1 4 1 3
+     S I X     9 4 2      +   N I N E     1 4 1 3
-------------------------------------------------------
          2 8 2 6                        2 8 2 6
```

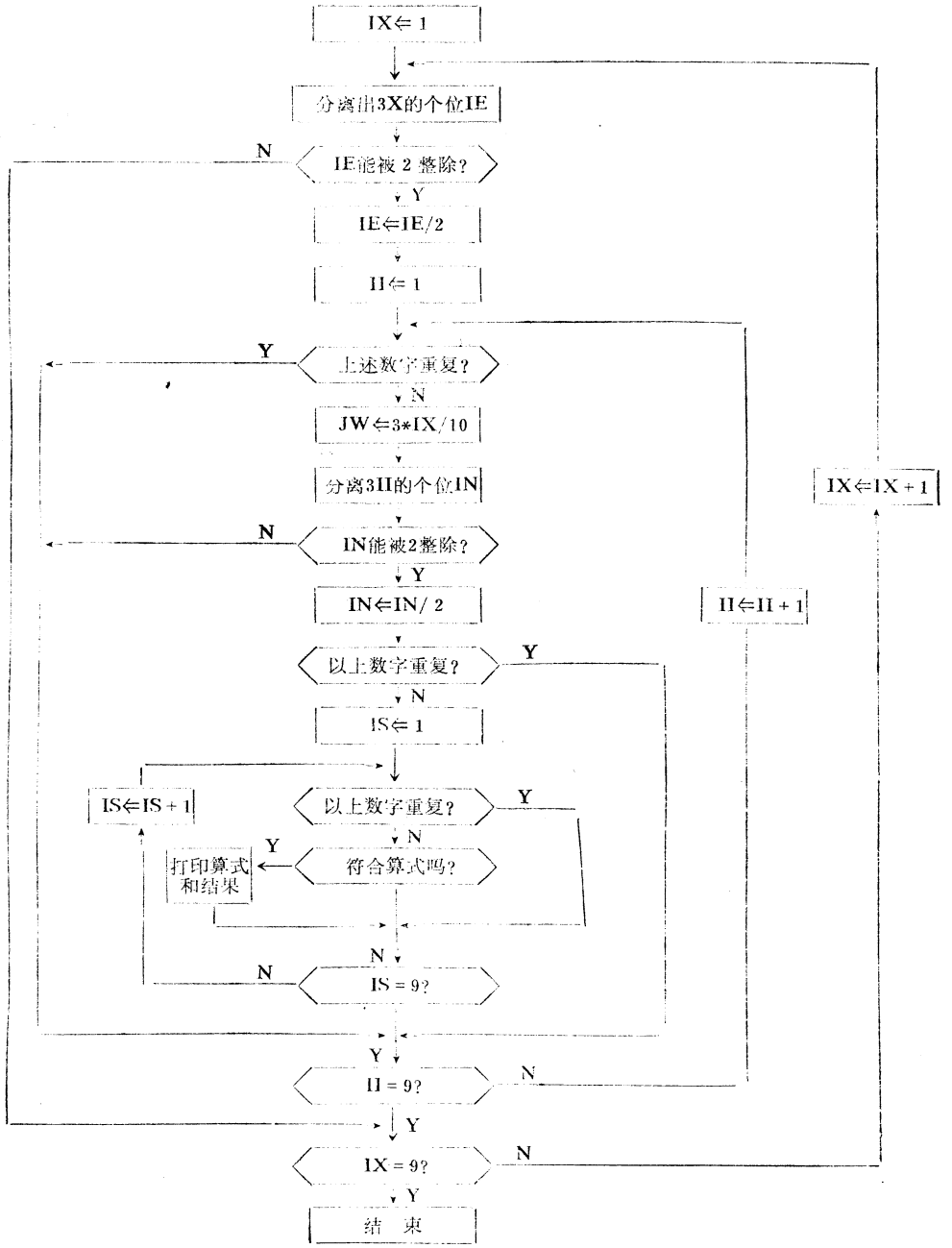


图 61

（三）思　考　题

请计算SEND加上MOKE等于多少钱（MONEY）。

```
  S E N D
+ M O K E
---------
M O N E Y
```

笔算分析结果为

```
  9 5 6 7
+ 1 0 8 5
---------
1 0 6 5 2
```

62 古　纸　残　篇

在一位数学家的藏书中夹有一张古旧的纸片。纸片上的字迹早已模糊不清了，只留下曾经写过字的痕迹，依稀还可以看出它是一个乘法算式，如图62-1所示。这个算式上原来的数字是什么呢？夹着这张纸片的书页上，“素数”两字被醒目地勾划了出来。难道说，这个算式与素数有什么关系吗？有人对此作了深入的研究，果然发现这个算式中的每一个数字都是素数，而且这样的算式是唯一的。请你也研究一番，并把这个算式写出来。

```
    * * *
×     * *
---------
  * * * *
* * * *
---------
* * * * *
```

图 62-1

（一）笔算步骤和结果

据题意，在每一个“*”号的地方只能填素数，即2、3、5或7。由于式中第三、四行都是四位数，因此首先要求一个三位数和一个一位数，使其乘积是一个四位数，并且在被乘数、乘数及乘积中只能出现上面的四个数码。通过试填，只有以下四种可能：$775\times3=2325$，$555\times5=2775$，$755\times5=3775$及$325\times7=2275$。

在上面这四种情形中，被乘数都不相同，因此，要满足题中的条件，乘数只能是两个数码相同的二位数，即只能是以下四种情况：775×33，555×55，755×55，325×77。

在这四种情形中，只有第一种才能使所得的数的数字的每一位都是素数，因此古纸残篇上的算式只能是：

```
    7 7 5
×     3 3
---------
  2 3 2 5
2 3 2 5
---------
2 5 5 7 5
```

（二）程序设计

1）设计思路

本题的算式连一个确切数字也没给，仅给每位数字都是素数，据此条件推算。定义数

组L(4)、置放素数2、3、5、7。算式从上至下共有五行，算式第一行的三位数和第二行的两位数相乘，利用五重循环确定乘数与被乘数。每变动一个数字时，要判是否满足题意条件。第三行和第四行各四位数，若每位都是素数，数值最小为2222，这是判断条件之一，第五行是五位数，若每位都是素数，数值最大为77777，这又是判断一个条件。此外，每生成一行都要判断该行的每位数字是否为素数。程序将巧妙安排这些判断和循环语句，得出正确推论后，形象地打印出完整算式。

2）框图　见图62-2

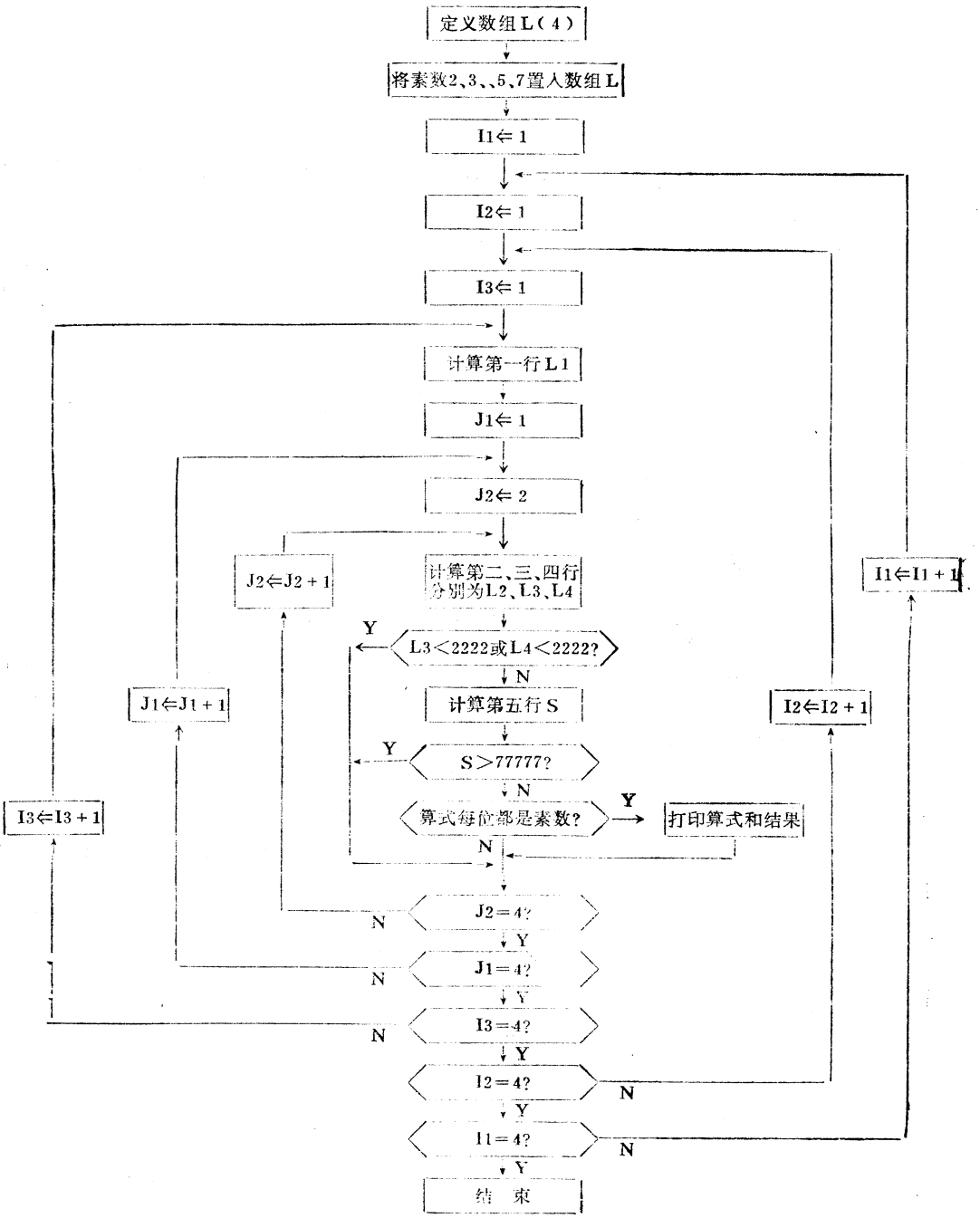


图 62-2

3）源程序及运行结果

```
      INTEGER  L(4)
      L(1) = 2
      L(2) = 3
      L(3) = 5
      L(4) = 7
      DO  10  I1 = 1, 4
      DO  10  I2 = 1, 4
      DO  10  I3 = 1, 4
      L1 = 100*L(I1) + 10*L(I2) + L(I3)
      DO  20  J1 = 1, 4
      DO  20  J2 = 1, 4
      L2 = 10*L(J1) + L(J2)
      L3 = L1*L(J1)
      L4 = J1*L(J2)
      IF ( (L3. LT. 2222) . OR. (L4. LT.
     2  2222) ) GOTO  20
      S = 10. *L4 + L3
      IF (S. GT. 77777. ) GOTO  20
      K3 = 4
      DO  30  K1 = 1, 3
      GOTO  (70, 60, 50) , K1
70    Q = FLOAT (L3)
      GOTO  40
60    Q = FLOAT (L4)
      GOTO  40
50    Q = S
      K3 = 5
40    DO  30  K2 = 1, K3
      N = IFIX (AMOD (Q, 10. ) )
      Q = (Q - N) /10.
      IF (N. NE. 2. AND. N. NE. 3. AND. N. NE. 5. AND. N. NE. 7)
     2  GOTO  20
30    CONTINUE
      WRITE (5, 5) L1, L2, L3, L4, S
20    CONTINUE
10    CONTINUE
5     FORMAT (//15X, I3/, 13X, 1HX, I4/, 12X, 10H·····/,
     1  14X, I4/, 13X, I4/10X, 12H·····/, 12X, F7. 0/)
      STOP  OK!
      END
```

L1、L2、L3、L4、S分别表示算式第1、2、3、4、5行

```
      7 7 5
  ×     3 3
-----------
    2 3 2 5
  2 3 2 5
-----------
  2 5 5 7 5
```

（三）思　考　题

英文SIX是6，TWO是2，这两个数相乘。巧好等于TWELVE，即12。请确定演算过程中不同字母代表十进制中不同的数字。

```
      S I X
  ×   T W O
-------------
    — — — —
  — — — —
— — — —
-------------
T W E L V E
```

笔算分析的结果

```
      9 8 6
  ×   3 4 5
-------------
    4 9 3 0
  3 9 4 4
2 9 5 8
-------------
3 4 0 1 7 0
```

63 祝圣诞节快乐

“A MERRY XMAS TO ALL”是英语中一句祝愿的话，它本身也是个谜：其中每个字母代表十进制中的一个数字，而每一个单词是一个平方数。此外，每个单词中数码字之和也是某个平方数。请解开这个字谜。

（一）笔算步骤和结果

借助于平方数表，可以很快确定：ALL是100、144、400、900中的一个数，而TO则是36或81。

如果要找一个四位的平方数，它的数字之和是个平方数，它的十位数字是1、4或9，那么这个唯一的四位数是7396 = XMAS；所以ALL = 900。因每个字母只代表一个数码，故TO只能是81。于是，MERRY要么是35224，要么是34225，而这两数中只有后者是个平方数。所以本题的答案是：

9　34225　7396　81　900

（二）程序设计

1）设计思路

本题共15个字符，除去重复字符尚有10个不同的字符，将用遍0，1，2，…，9这10个数字；字符串长短不一，且有限制条件作为判断的依据，耐人寻味。

我们先从长字符串MERRY开始。虽有5个字符，但有一个重复字符R，利用四重

循环暂定这5个字符。而后再试定4个字符 XMAS，其中M和前一个字符串中M相同，故用三重循环便可暂定。注意，其中字符A与本算式第一个单独字符A相同，故在此应顺便判A所代表的一位数字，应是平方根数，从而确定了算式第一个字符A。再确定字符TO，二重循环便可。最后确定字符ALL，虽然是三个字符，但A已在前暂定，L又是重复字符，故仅用一重循环便可试定出三个字符。

上述每串字符暂定后，立即判断每个字符相加是否是平方数，该串字符本身是否是平方数，且要判定每个不同字符是否唯一代表一个不同的数字。将算式结果形象地打印出。

2）框图　见图63

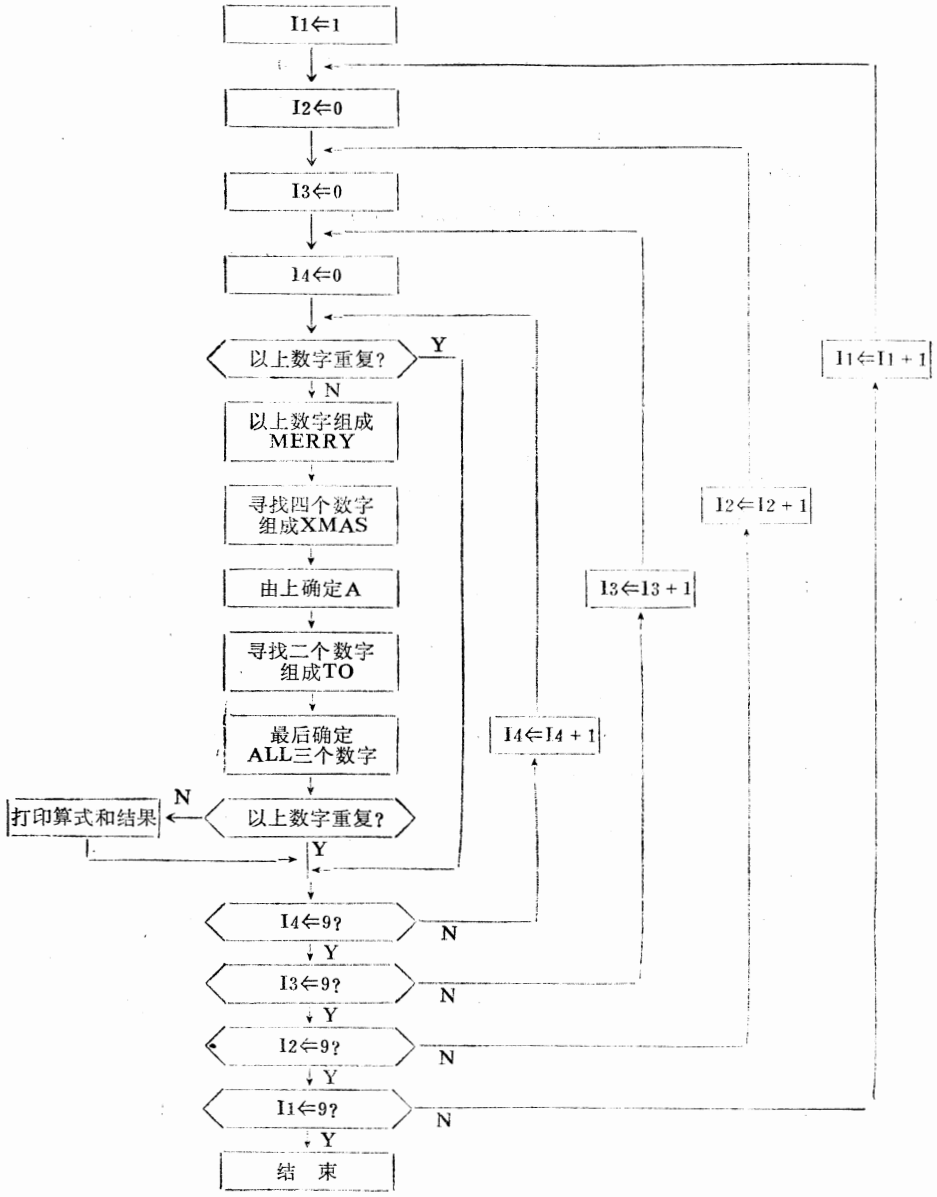


图 63

3）源程序及运行结果

```
DO  10  I1=1, 9
DO  10  I2=0, 9
DO  10  I3=0, 9
DO  10  I4=0, 9
IF (I1. EQ. I2. OR. I1. EQ. I3. OR. I1. EQ. I4)  GOTO  10
IF (I2. EQ. I3. OR. I2. EQ. I4. OR. I3. EQ. I4)  GOTO  10
S=SQRT (FLOAT (I1+I2+I3+I3+I4) )
IF ( (S-INT (S) ) . GT. 0.0001) GOTO  10
S=SQRT (10000. *I1+1000. *I2+100. *I3+10. *I3+I4)
IF ( (S-INT (S) ) GT. 0.0001) GOTO  10
DO  20  J1=1, 9
DO  20  J2=1, 9
DO  20  J3=0, 9
Q=SQRT (FLOAT (J1+I1+J2+J3) )
IF ( (Q-INT (Q) ) . GT. 0.00001)  GOTO  20
Q=SQRT (FLOAT (J2) )
IF ( (Q-INT (Q) ) . GT. 0.0001) GOTO  20
IF (J1. EQ. I1. OR. J1. EQ. I2. OR. J1. EQ. I3. OR. J1. EQ. I4)  GOTO
  20
IF (J2. EQ. I1. OR. J2. EQ. I2. OR. J2. EQ. I3. OR. J2. EQ. I4)  GOTO
  20
IF (J3. EQ. I1. OR. J3. EQ. I2. OR. J3. EQ. I3. OR. J3. EQ. I4)  GOTO
  20
IF (J1. EQ. J2. OR. J1. EQ. J3. OR. J2. EQ. J3)  GOTO  20
Q=SQRT (1000. *  J1+100. *  I1+10. *  J2+J3)
IF ( (Q-INT (Q) ) . GT. 0.0001)  GOTO  20
DO  30  K1=1, 9
DO  30  K2=0, 9
R=SQRT (FLOAT (K1+K2) )
IF ( (R-INT (R) ) . GT. 0.00001)  GOTO  30
IF (K1. EQ. I1. OR. K1. EQ. I2. OR. K1. EQ. I3. OR. K1. EQ. I4)
  GOTO  30
IF (K2. EQ. I1. OR. K2. EQ. I2. OR. K2. EQ. I3. OR. K2. EQ. I4)
  GOTO  30
IF (K1. EQ. J1. OR. K1. EQ. J2. OR. K1. EQ. J3)  GOTO  30
IF (K2. EQ. J1. OR. K2. EQ. J2. OR. K2. EQ. J3. OR. K2. EQ.  K1)
  GOTO  30
R=SQRT (10. *  K1+K2)
IF ( (R-INT (R) ) . GT. 0.0001)  GOTO 30
DO  40  L=0, 9
T=SQRT (FLOAT (J2+L+L) )
```

```
      IF ( (T-INT (T) ) . GT. 0.00001) GOTO  40
      IF (L. EQ. I1. OR. L. EQ. I2. OR. L. EQ. I3. OR. L. EQ. I4)  GOTO
        40
      IF (L. EQ. J1. OR. L. EQ. J2. OR. L. EQ. J3)  GOTO  40
      IF (L. EQ. K1. OR. L. EQ. K2)  GOTO  40
      T=SQRT (100. *  J2+10. *  L+L)
      IF ( (T-INT (T) ) . GT. 0.0001)  GOTO  40
      WRITE (5, 50) J2, I1, I2, I3, I3, I4, J1, I1, J2, J3, K1, K2, J2, L, L
50    FORMAT (/10X, 19HA  MERRY  XMAS  TO  ALL/, 10X, I1, 1H, 5I1,
     1  1H, 4I1, 1H, 2I1, 1H, 3I1/)
40    CONTINUE
30    CONTINUE
20    CONTINUE
10    CONTINUE
      STOP  OK  !
      END
```

程序运行后打印结果

```
A  MERRY  XMAS  TO  ALL
9   34225   7396   81  900
```

（三）思 考 题

○×○○=○○○=○○×○

该乘法共九个空格，用遍了1～9这九个数字，并且不重复。请你求出该等式，进行程序设计。

64 巧妙的算式

如此算式

```
    F O R T Y
        T E N
+)      T E N
-------------
    S I X T Y
```

其中每一个字母表示一个数字，且不同的数字对应着不同的字母，请将这些字母求出来。

（一）笔算步骤和结果

在上述算式中，我们约定从右向左分别称为第一位、第二位…第五位。因为第一位的数字和的末尾等于Y，故字母N或者等于0，或者等于5。但当N = 5时，这位便向第二位进位1，此时第二位上的数字和就不可能以T结尾，所以N = 0；完全类似地可以指出，E只能取0或5，但N = 0，所以E = 5；因为第三位向前进位，不会大于2，故第四位的数字和（即O与第三位向前的进位和）不大于11，所以第四位向第五位进位为1，但N = 0，所以I = 1，且O = 9，这同时说明了由第三位向第四位的进位是2；因为X

不可能取数字0和1，且从第二位向第三位的进位不大于1，故第三位的数字和不大于24，所以字母T = 7或T = 8。若T = 7，则R = 8，而X = 3。

注意到F和S是相邻的数字，它们仅可能是2和3，或3和4，所以X = 3时，F和S就不再是相邻的数字，从而T = 8，R = 7，且X = 4；此时只剩下数字2、3、6，由上说明可知F = 2，S = 3，且Y = 6。

这样，该算式仅有唯一解：

```
    2 9 7 8 6
      8 5 0
+)    8 5 0
-------------
  3 1 4 8 6
```

（二）程序设计

1）设计思路

有十个英文字母，自然用到0～9十个数字，显然E⇐5，N⇐0，其余八个字母用多重循环来寻找所代表的数字。每找到一个字母所代表的数字，要与暂确定的字母表示的数字不得重复。全确定后，打印出英文字母算式和其对应的数字结果。

2）框图　见图64

3）源程序及运行结果

```
KE=5
KN=0
DO 50 KY=1, 9
IF (KY. EQ. KE)  GO TO 50
DO 40 KT=1, 9
IF (KT. EQ. KY. OR. KT. EQ. KE)  GO TO 40
DO 30 KR=1, 9
IF (KR. EQ. KT. OR. KR. EQ. KY. OR. KR. EQ. KE)  GO TO 30
IF ( (1+KR+2*KT) /10. LT. 1)  GO TO 30
KX=1+KR+2*KT- ( (1+KR+2*KT) /10) *10
IF (KX. LT. 1) GO TO 30
IF (KX. EQ. KR. OR. KX. EQ. KT. OR. KX. EQ. KY. OR. KX. EQ.
  KE) GO TO 30
DO 20 KO=1, 9
IF (KO. EQ. KX. OR. KO. EQ. KR. OR. KO. EQ. KT) GO TO 20
IF (KO. EQ. KY. OR. KO. EQ. KE) GO TO 20
K1=KO+ (1+KR+2*KT) /10-10
IF (K1. LT. 1. OR. K1. GT. 9)  GO TO 20
IF (K1. EQ. KO. OR. K1. EQ. KX. OR. K1. EQ. KT) GO TO 20
IF (K1. EQ. KY. OR. K1. EQ. KE. OR. K1. EQ. KR) GO TO 20
DO 10 KF=1, 8
IF (KF. EQ. K1. OR. KF. EQ. KO. OR. KF. EQ. KX. OR. KF. EQ.
  KT)  GO TO 10
IF (KF. EQ. KY. OR. KF. EQ. KE. OR. KF. EQ. KR)  GO TO 10
```

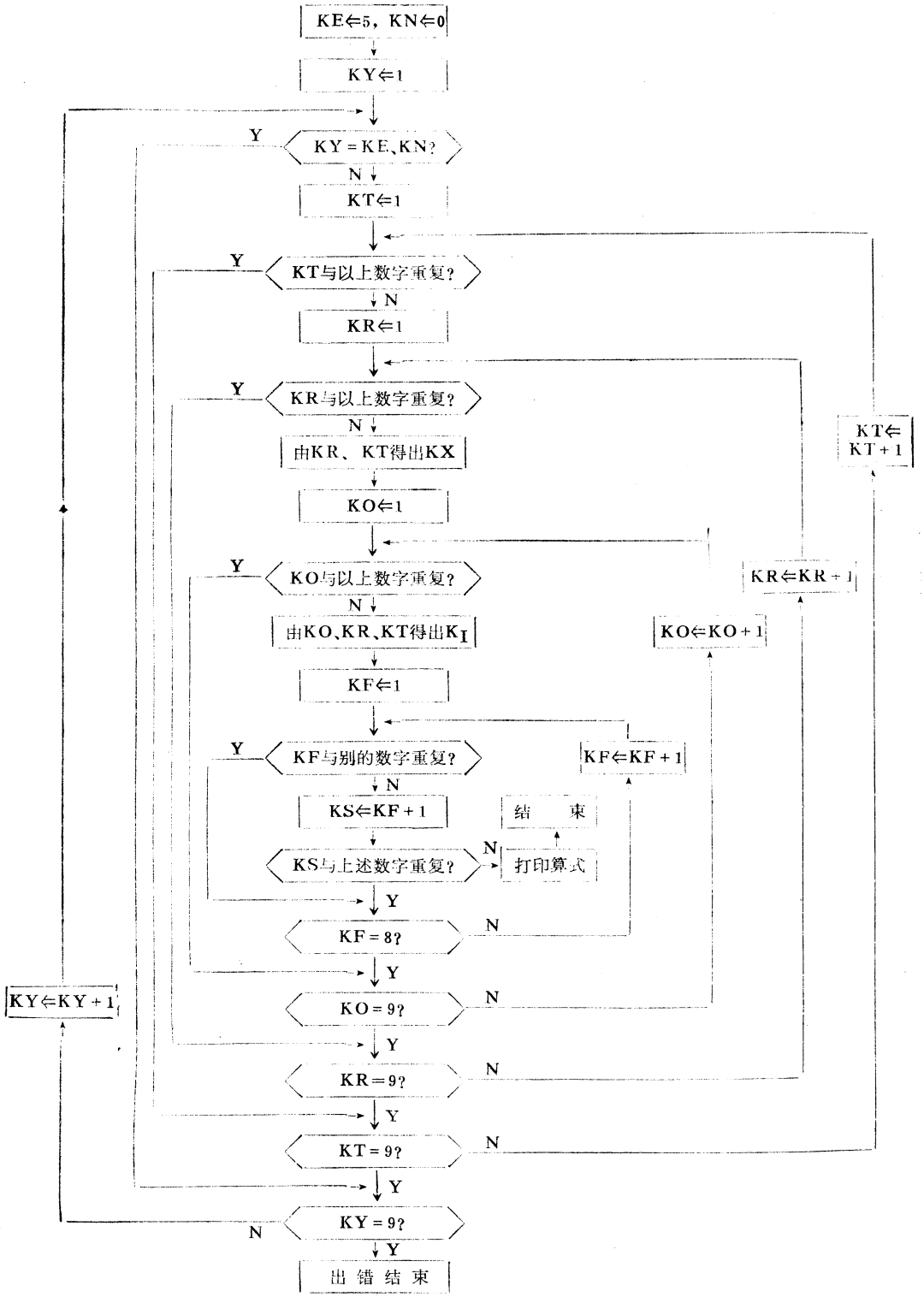

KE⇐5，KN⇐0
KY⇐1
Y
KY=KE、KN?
N
KT⇐1
Y
KT与以上数字重复?
N
KR⇐1
Y
KR与以上数字重复?
N
由KR、KT得出KX
KO⇐1
Y
KO与以上数字重复?
N
由KO、KR、KT得出K_I
KF⇐1
Y
KF与别的数字重复?
N
KS⇐KF+1
KS与上述数字重复?
N
打印算式
结 束
Y
KF=8?
N
Y
KO=9?
N
Y
KR=9?
N
Y
KT=9?
N
Y
KY=9?
N
Y
出错结束
KY⇐KY+1
KF⇐KF+1
KO⇐KO+1
KR⇐KR+1
KT⇐KT+1

图 64

```
      KS=KF+1
      IF (KS. EQ. KI. OR. KS. EQ. KO. OR. KS. EQ. KX)  GO TO 10
      IF (KS. EQ. KT. OR. KS. EQ. KY. OR. KS. EQ. KE. OR. KS. EQ. KR)
        GO TO 10
      WRITE (6, 11) KF, KO, KR, KT, KY
11    FORMAT (8X, 7H FORTY, 8X, 5I1)
      WRITE (6, 22) KT, KE, KN
      WRITE (6, 33) KT, KE, KN
22    FORMAT (8X, 7H       TEN, 10X, 3I1)
33    FORMAT (8X, 7H +     TEN, 7X, 3H+    , 3I1)
      WRITE (6, 44)
44    FORMAT (8X, 8H·····, 5X, 8H·····)
      WRITE (6, 55) KS, KI, KX, KT, KY
55    FORMAT (8X, 7H SIXTY, 8X, 5I1)
      STOP
10    CONTINUE
20    CONTINUE
30    CONTINUE
40    CONTINUE
50    CONTINUE
      STOP 'ERR'
      END
```

```
     FORTY          29786
       TEN            850
+      TEN     +      850
.....          .....
     SIXTY          31486
```

(三) 思 考 题

```
      FIVE                  9021
       TWO                   846
+       ONE     +             671
-------        -------
      EIGHT                 10538
```

提示结果：

十 里 程 碑

65 发展家庭养殖事业

王大娘要用100元钱买100头小牲畜，不多不少要求“双百”。若小牛每头10元，羊羔每只3元，小兔每只0.5元。请你替她算算应该怎样买法？

（一）笔算步骤和结果

这群牲畜平均每头值1元，小牛的价钱与平均值差正9元，小羊差正2元，小兔差负0.5元。

所以每买一头小牛就得买18只小兔，而每买一只小羊就得买4只小兔。设买了X头小牛，Y只小羊，Z只小兔，于是有

$$X(1+18)+Y(1+4)=100$$

或

$$5Y=100-19X$$

要想$100-19X$非负，并可被5整除，只有当$X=0$或$X=5$时，$Y=20$或$Y=1$。所以或者买20只小羊，80只小兔，或者买5头小牛，1只小羊和94只小兔。

（二）程 序 设 计

1）设计思路

令X、Y、Z表示牛、羊、兔的只数。王大娘要求“双百”，我们以此打印程序运行结果和说明。打印表头为：

$$100=10*X+3*Y+0.5*Z \quad AND \quad 100=X+Y+Z$$

利用循环语句，先求牛的数目，0～10头，当牛数定后，由算法分析算出羊、兔数，用算术条件语句判断，若成立，则打印出三种牲畜的英文名字和只数。

2）框图 见图65

3）源程序及运行结果

```
C       100=10*X+3*Y+0.5*Z  AND  100=X+Y+Z
        DO  11  I=1, 11
        NC=I-1
        NP= (19*NC) / 5
        IF (NP*5. NE. 19*NC)  GO  TO  11
        IF (NP. GT. 20) GO  TO  11
        NS=20-NP
        NR=80-NC+NP
        WRITE (6, 66) NC, NS, NR
11      CONTINUE
66      FORMAT (11X, 20HTHE  ANSWER  ARE: CART=, I2, 7H, SHEEP=,
        I2, 9H, RABBIT=, I2)
        STOP
        END
```

```
 THE  ANSWER  ARE : CART= 0 , SHEEP=20, RABBIT=80
 THE  ANSWER  ARE : CART= 5 , SHEEP= 1, RABBIT=94
```

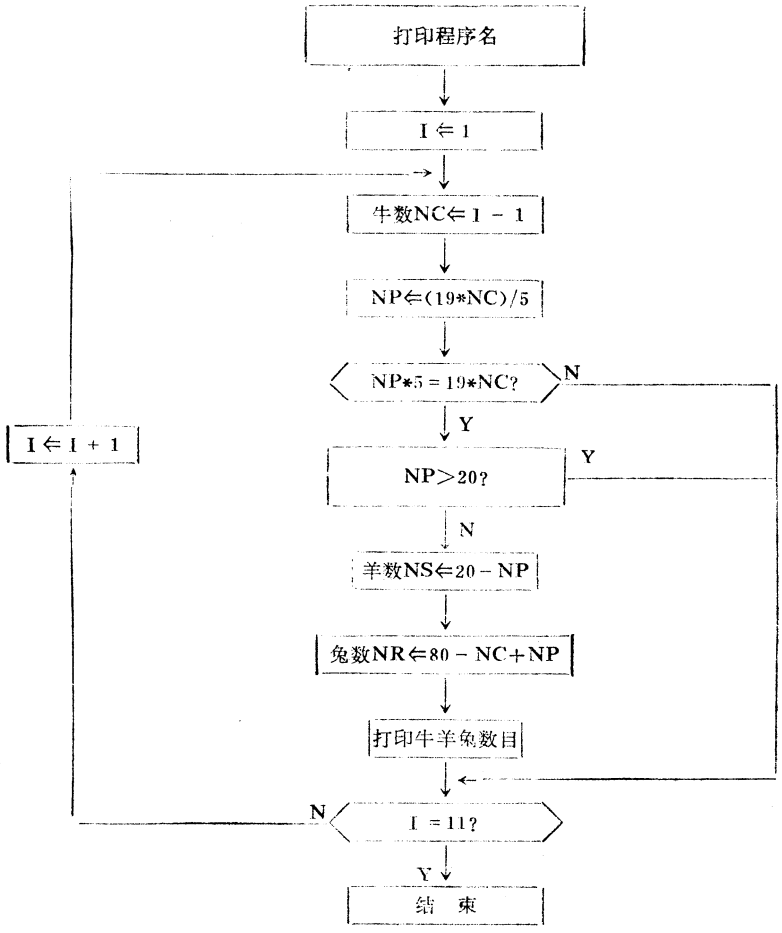


图 65

（三）思 考 题

100元要买100只鸡。若大母鸡每只5元，大公鸡每只3元，小鸡每三只1元。问怎样买法，编写程序打印出结果。

66 运动员和学习成绩

班级里有25个学生，其中17人会骑自行车，13人会游泳，8人会滑冰，这三个项目全都掌握的学生一个也没有，但是不论会骑自行车的或会游泳的或会滑冰的学生，他们的数学成绩都是良好或及格，这点是值得注意的，因为全班有6个人数学不及格。试问：全班数学成绩优秀的学生有几人？有几人既会游泳又会滑冰？（学生数学成绩分为优秀、良好、及格、不及格四种）。

（一）笔算步骤和结果

全班共有学生25人，因为有6个学生数学成绩不及格，所以成绩至少是及格的学生共有19人，全班“运动员”（如果认为一个学生只会一个运动项目）共有17＋13＋8＝38人，但是由于没有一个学生掌握三个运动项目，所以参加运动的学生共有19人（根据上面

所得），每人掌握两个运动项目，现在可以容易地回答题目所提的问题了：

（1）班里没有一个学生数学成绩为优秀；

（2）19个运动员中，17人会骑自行车，所以只有两个学生会游泳又会滑冰。

（二）程 序 设 计

1）设计思路

令全班学生人数为I，不及格人数为J，会骑车人数IRB，会游泳人数IS，会滑冰人数ISK，这些数字用数据赋值语句DATA输入。调试该题的机器要求在有名公用区来传递这些数据，令区名为J1。

再设运动员数IAL，及格人数IG，优秀生人数IEX。

由题意知，成绩优秀的学生一定出现在三者都不会的同学中，又三者都不会的同学又没有。故将不及格人数和优秀学生人数除去后，此数恰好为运动员数IAL（即包括重复项目在内运动员的数目）的1/2时，才有解，否则为不可解，（因为题中没有告诉各项目的重复运动员的数目），这个事实说明只要是运动员，每个运动员必定会两个运动项目，此外，优秀学生只可能在IAL＜2＊IG的情况下才有，否则没有。

$$IEX = 1/3(2IG - IAL)$$

∵IEX = 2(IG − IEX) − IAL（因每个运动员会两个项目，故乘2）

∴3IEX = 2IG − IAL

IEX = 1/3 (2IG − IAL)

这算法要在程序中考虑。

2）框图　见图66

3）源程序及运行结果

```
      COMMON/J1/ I , J , IRB, IS, ISK
      DATA  I , J , IRB, IS, ISK/25, 6 , 17, 13, 8 /
      IG = I - J
      IAL = IRB + IS + ISK
      IF (IAL. LT. 2*IG) GO TO 20
      IF (IAL. EQ. 2*IG) GO TO 30
      STOP'ERR1'
30    IEX = 0
      ISKS = IG - IRB
35    WRITE (10, 50) IEX, ISKS
50    FORMAT (3X, 4HIEX = , I2/3X, 21HICE - SKATING AND SWIM = ,
      I 2)
      STOP
20    IEX = ( 2 *IG - IAL) / 3
      IM = IG - IEX
      IF (IAL. NE. 2 *IM) STOP 'ERR 2'
      ISKS = IM - IRB
      GOTO 35
      END
```

```
IEX = 0                          没有优秀学生
ICE - SKATING AND SWIM = 2       会游泳与滑冰的有2人
```

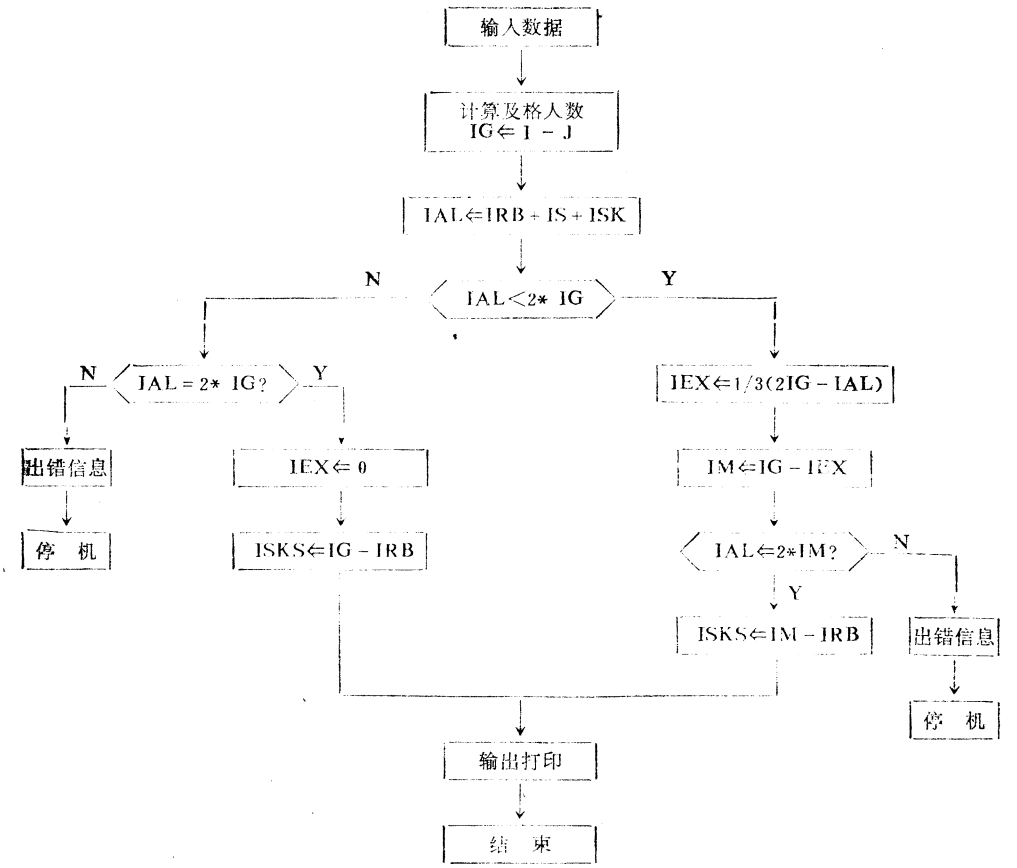


图 66

（三）思 考 题

某医院对一天的门诊病人作了统计，门诊病人总数1000人，其中公费医疗者525人，女病人312人，青年病人470人，公费青年病人147名，公费女病人42名，女青年病人86人，公费女青年病人25人，请判断一下，各数字之间有无矛盾。

67 人口非控制不可

1982年中国第三次人口普查结果是全国总人口为10亿3千多万，若不计划生育，发展下去，在可预见的若干年内，将带来何种程度的严重后果，请举例计算。

（一）算法分析和结论

若R、P1、Y（J）分别代表人口增长率、人口基数、年数，总人口计算公式为

$$P(J)=P1\times(1+R)^{Y(J)}$$

如果每年人口增长率为3%，2000年全国人口为17.5亿，到2050年时达77亿，而1000年以后则为7×10^{21}人，即七十万亿亿，这是个天文数字。它到底多大呢，不妨以包括高山、沙漠、江河、湖泊在内的960万平方公里为基数计算一下，每平方米上要居住7.3×10^{8}人，即7.3亿人。即使全国都盖满100层的摩天大楼，也无法容纳这么多人。如果真出现这样的情况，那真是人无立锥之地的可怕的情景。如果人口年增长率为2%，则2000年时

为14.7亿人，1000年以后为4.1×10^{17}人，每平方米居住43000人。如果年增长率为1%，1000年后为2.16×10^{18}人，即二万亿人，每平方米居住2人。如能将出生率降到千分之五，则1000年后我国人口为150亿人，每平方米只居住0.0001人。

每个有远见的中国人，必须宣传人口非控制不行。我们用FORTRAN语言，也协助再次呼吁！

（二）程序设计

1）设计思路

程序中安排六种人口增长率R，对应每种R，计算从1982年（人口第三次普查）后18、68、1000年，即公元2000、2050、3982年时分别总人口数，以及相应的每平方米人口数。由于数字大，定义双精度数组P（3）、A（3）分别置放总人口数、人口数/米^2。

对应每种增长率，程序运行后打印出两行，第一行是人口总数，第二行是每平方米人数，它是全国总人数除以960万平方公里（96×10^{12}米^2）得到的。程序中以数组Y代表年数，Y（1）、Y（2）、Y（3）分别代表距今多少年。P数组代表总人口数，A数组代表人口密度。读者可以由READ语句输入任何一年的年份数，来求该年的总人口数。

45语句的下一个语句是求R（为保证2000年时全国人口不超过12亿时，年人口增长率），可以看出，R应小于或等于0.852%。这是由算法公式得出来的。若是上述六种增长率R，问若干年后便超过12亿，见附注。

2）框图　见图67

3）源程序及运行结果

```
      DOUBLE PRECISION P(3), A(3), P1      (双精度数组和变量)
      INTEGER Y
      DIMENSION Y(3)
      P1=1030000000.0 (以我国第三次人口普查10亿3千万为基数)
      WRITE (6, 5)
5     FORMAT (1X, 1H?)
      READ (5, 10) Y(1), Y(2), Y(3)
10    FORMAT (I3, I3, I4)
      WRITE (6, 20)
20    FORMAT (1X, 68H RATE    2000    2050    AFTER 1000 YEARS/)
      R=0.03
      DO 30 I=1, 6
      DO 35 J=1, 3
      P(J)=P1**(1+R)**(Y(J))
35    A(J)=P(J)/9600000000000.0
      WRITE (6, 40) R, (P(J), J=1, 3)
40    FORMAT (1X, F5.3, 2(F18.1, 2X), F24.1)
      WRITE (6, 42) (A(J), J=1, 3)
42    FORMAT (1X, 5X, 2(F18.6, 2X), F24.6/)
30    R=R-0.005
      WRITE (6, 45)
```

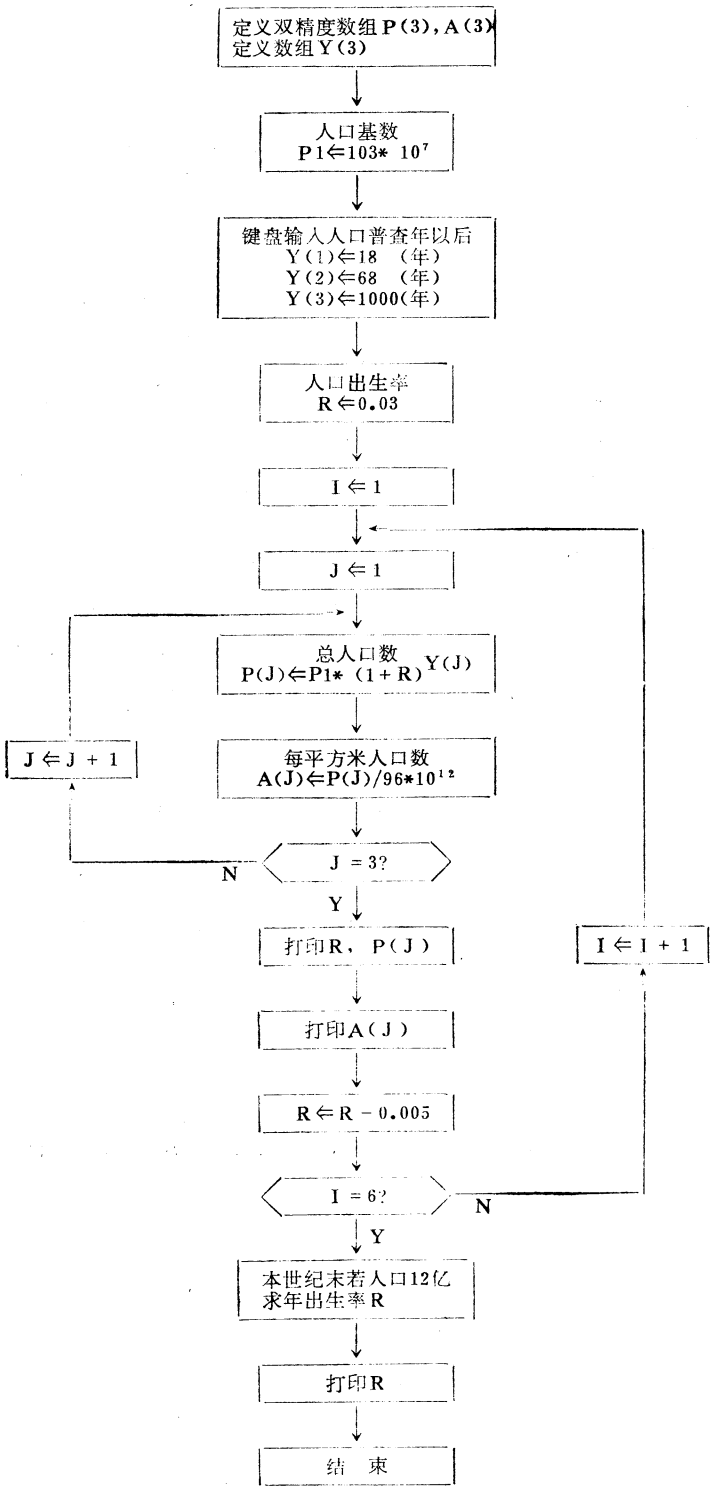

定义双精度数组P(3),A(3)
定义数组Y(3)
人口基数
P1⇐103* 10⁷
键盘输入人口普查年以后
Y(1)⇐18 (年)
Y(2)⇐68 (年)
Y(3)⇐1000(年)
人口出生率
R⇐0.03
I⇐1
J⇐1
总人口数
P(J)⇐P1*(1+R)^Y(J)
每平方米人口数
A(J)⇐P(J)/96*10¹²
J⇐J+1
J=3?
N
Y
打印R, P(J)
打印A(J)
R⇐R-0.005
I=6?
N
I⇐I+1
Y
本世纪末若人口12亿
求年出生率R
打印R
结 束

图 67

```
45      FORMAT (1X/)
        R = (12.0/10.3) **(1.0/18.0) - 1
        WRITE (6, 50) R
50      FORMAT (1X, 10HIF RATE IS, F5.3,
           42H, THE POPULATION OF 2000 IS 1, 200, 000, 000)
        STOP
        END
```

```
?18, 68, 1000
RATE (增长率)     2000 (年)              2050 (年)          AFTER 1000 YEAR (一千年后)
. 030          1753506102.6(总人口)   7687208566.7      708051781812224004000.0
                 . 000183(人/米²)       . 000801              737553894.072422
. 025          1606450493.3           5521564846.0      545424062873600003000.0
                 . 000167               . 000575              5681500.305862
. 020          1471094629.8           3959588513.4      410228861440000002.0
                 . 000153               . 000412              42732.170441
. 015          1346564524.2           2834865832.3      3012619190000000.0
                 . 000140               . 000295              313.814480
. 010          1232031128.4           2026235196.6      21587335468750.0
                 . 000128               . 000211              2.248681
. 005          1126749870.8           1445385770.3      150996277465.8
                 . 000117               . 000151              . 015729
IF RATE IS. 009, THE POPULATION OF 2000 IS 1,200,000,000
(若增长率为0.009，到本世纪末人口为12亿)
```

4) 附注

若我国人口年增长率R分别为0.03、0.025、0.02、0.015、0.010、0.005，问若干年我国人口便超过12亿？

程序设计思路同主题。令N为若干年，Y是公元年数，P1是N年时人口总数。仍以全国第三次人口普查10亿3千多万为基数。

源程序及运行结果如下。

```
        DOUBLE PRECISION P1
        R = 0.03
        DO 10 I = 1, 6
        N = 0
        P1 = 1030000000.0
50      P1 = P1* (1 + R)
        N = N + 1
        IF (P1, GT, 1200000000.0) GO TO 40
        GO TO 50
40      WRITE (6, 20) R, N, 1982 + N, P1
20      FORMAT (1X, 2HR =, F5, 3, 4X, 2HN =, I2, 4X, 2HY =, I4, 4X,
        3HP1 =, F12, 1)
```

```
10      R = R - 0.005
        STOP
        END
```

```
R = .030    N =  6    Y = 1988    P1 = 1229873660.5
R = .025    N =  7    Y = 1989    P1 = 1224347123.7
R = .020    N =  8    Y = 1990    P1 = 1206808981.9
R = .015    N = 11    Y = 1993    P1 = 1213288784.9
R = .010    N = 16    Y = 1998    P1 = 1207755821.8
R = .005    N = 31    Y = 2013    P1 = 1202227539.7
```

（三）思 考 题

假设我国工业产值增长速度每年为5%、7%、9%、11%、13%时，请你编写程序计算出需要多少年工业产值可以增长一倍？

68 如何找这个数

对任意指定的整数N，找出一个最小的数S，使其满足$S=P^N+Q^N=R^N+T^N$，且P、Q、R、T不全相等。

（一）算 法 分 析

（1）首先确定出可计算的最大整数MAX。对任意指定的N，分别求出0^N，1^N，2^N，…，X_{max}^N，且$X^N \leqslant MAX$。

（2）在求出的序列中作二、二之和的比较，即$X_i^N+X_j^N$与$X_l^N+X_K^N$，$1 \leqslant i, j, l, K \leqslant max$，进行比较，若不等则改变i，j，$l$，K的值，若相等则做（3）。

（3）X_i、X_j分别与X_l，X_K进行比较，若全等，则说明是同一个和，即$X_i^N+X_j^N=X_l^N+X_l^N$，再返回（2），重新查找。若不全等，则说明已找到，其中$P=X_i$，$Q=X_j$，$R=X_l$，$T=X_K$即为所求。

又因为i、j、l、K均从最小值开始查找，因此满足题中所求S要求最小的条件。

（二）程 序 设 计

1）设计思路

设置一个二维数组A（M，2），用于存放候选序列，其第一列存放X_i的值，第二列存放相应的X_i^N的值。

进行比较时，从M=1起，分别取第二列的两个数相加，然后与第二列中的另外两个数进行比较。输出结果是：

$$S_1=X_i**N+X_j**N=S \qquad S_2=X_l**N+X_K**N=S$$

其中X_i，X_j，X_l，X_K，S均为实际数值。

注：此程序运行的结果取决于机器可表达的最大正整数。例如，若N=04，则S=635318657，则要32位字长的机器才能求出。若使用16位机器，可能给出没找到S或其它错误信息（作双倍字长运算时例外）。因此，可根据欲求的N，估计S的最大值，以便确定MAX的位数。此外，对于运算速度为百万次以下的机器，亦需要相当长的时间。

当从键盘上输出的N值为04时，运算结果为：

$S_1 = 59 ** 4 + 158 ** 4 = 635318657$

$S_2 = 133 ** 4 + 134 ** 4 = 635318657$

2）框图　见图68

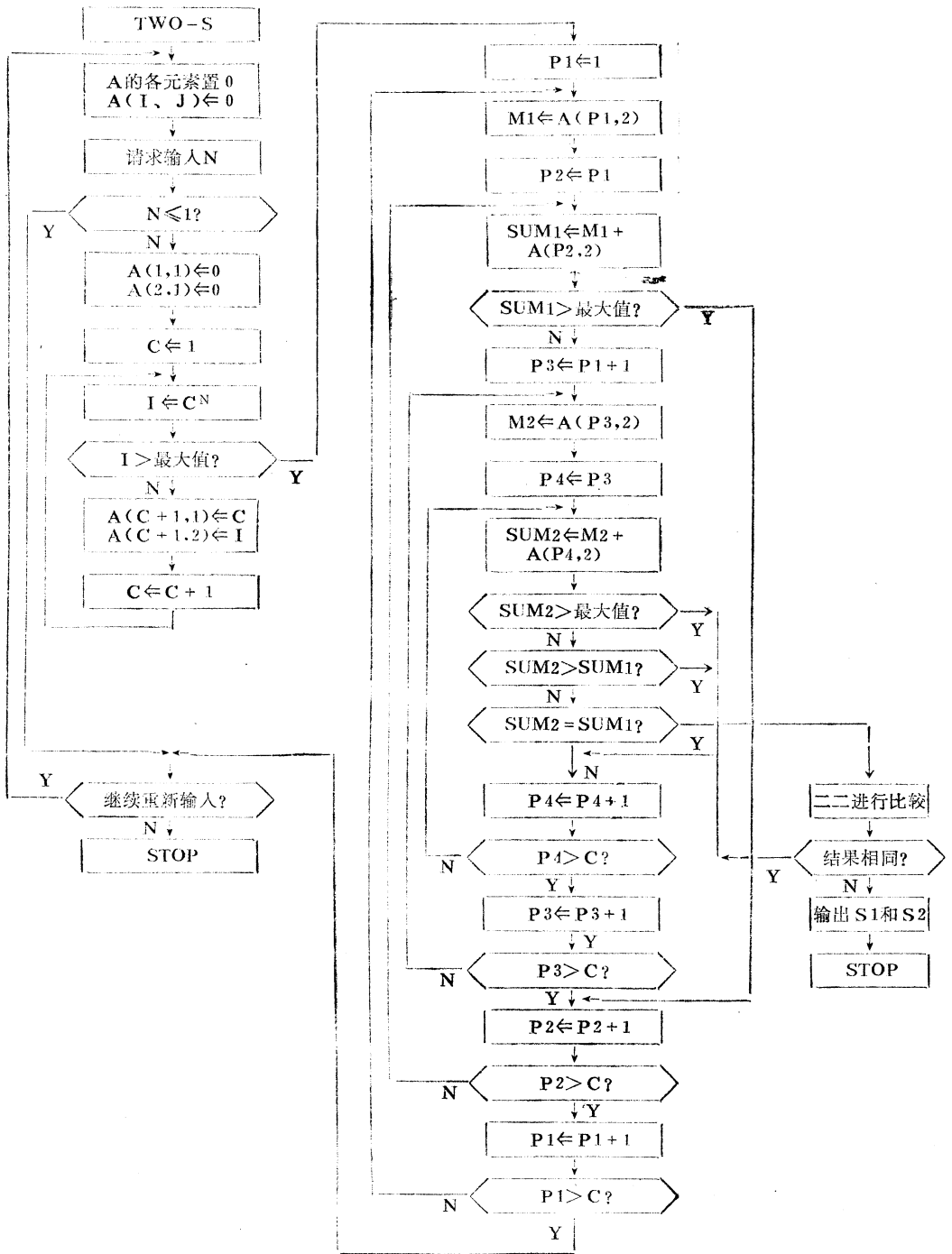


图 68

3）源程序及运行结果

```
C***TWO-S
      INTEGER  N, C, I, P1, P2, P3, P4, M1, M2, SUM1, SUM2
      CHARACTER  PLAY
      INTEGER  A(1000, 2)
5     DO  10  J=1, 2      } A的各元素置0
      DO  10  I=1, 1000   }
10    A(I, J) = 0
      WRITE (1, 15)
15    FORMAT (5X, 45HTYPE A NUMBER YOU SPECIFIED
      AS  POWER  EXPONENT, /)                          读入N
      READ (1, 20) N
20    FORMAT (I2)
      IF (N. EQ. 1) GO TO 85
C  ***  COMPUTING THE X**N
      A(1, 1) = 0
      A(1, 2) = 0
      C = 1
30    I = C**N
      IF (I. GT. 6000000000) GO TO 40
      A(C+1, 1) = C                          计算≤最大值的C^N，
      A(C+1, 2) = I                          并放入A中。
      C = C + 1
      GO TO 30
C  ***  SERACH FOR THE S1 AND S2
40    DO  80  P1=1, C
      M1 = A(P1, 2)
      DO  70  P2=P1, C
      SUM1 = M1 + A(P2, 2)
      IF (SUM1. GT. 10000000000)GO TO 80
      DO  60  P3=P1+1, C                     查找S1=P^N+Q^N
      M2 = A(P3, 2)                          及S2=R^N+T^N
      DO  50  P4=P3, C
      SUM2 = M2 + A(P4, 2)
      IF (SUM2. GT. 10000000000) GO TO 60
      IF (SUM2. GT. SUM1) GO TO 60
      IF (SUM2. NE. SUM1) GO TO 50
C  ***  COMPARE THEM
      IF ((A(P1, 1). EQ. A(P3, 1)). AND. (A(P2,
      1). EQ. A(P4, 1)). OR. (A(P1, 1).
     1 EQ. A(P4, 1)). AND. (A(P2, 1). EQ. A(P3,    比较
      1))) GO TO 50
      GO TO 105
50    CONTINUE
```

```
 60        CONTINUE
 70        CONTINUE
 80        CONTINUE
 85        WRITE (1, 90)
 90        FORMAT (5X, 26HTHE SOLUTION IS NOT EXIST;, /)
           WRITE (1, 93)
 93        FORMAT (5X, 17HTO CONTINUE? (Y/N), /)
           READ (1, 95) PLAY
 95        FORMAT (1A)
           IF (PLAY. EQ.'Y') GO TO 5
           GO TO 150
C  ***  OUTPUT THE RESULTS
105        WRITE (1, 110)
110        FORMAT (5X, 17HTHE S1 AND S2 ARE)
           WRITE (1, 120) A (P1, 1), N, A (P2, 1), N,
           SUM1
120        FORMAT (5X, 3HS1 =, I15, 2H**, I3, 1H+, I15, 2H
           **, I3, 1H=, I15)
           WRITE (1, 130) A (P3, 1), N, A (P4, 1), N,
           SUM2
130        FORMAT (5X, 3HS1 =, I15, 2H**, I3, 1H+, I15,
           2H**, I3, 1H=, I15)
150        STOP
           END
```

找印结果

程序运行打印结果如下：

$$S1 = 59^{**}4 + 158^{**}4 = 635318657$$

$$S2 = 133^{**}4 + 134^{**}4 = 635318657$$

（三）思 考 题

请验证：$X^4 + Y^4 = Z^2$，$X^3 + Y^3 = Z^3$，$X^4 + Y^4 = Z^4$都没有正整数解。

69 苹 果 装 箱

请你将1000个苹果分装在10筐中，有人向你要苹果时，不论他要几个(在1000以内)，你只能拿几筐给他，且不许你打开苹果筐一个个去数，而这几筐的苹果的总数，正好与他向你要的数目一样，问你应该怎样地分装？

（一）笔算步骤和结果

因为每一个自然数都可用1、2、4、8、16、32、…等数中的若干个相加而得，所以在10筐内分别装入1、2、4、8、16、32、64、128、256、512个苹果，并顺次记住这些筐中苹果个数，如果要的苹果数少于4，在第1、2筐之间拿。若少于8个，在1～3#筐拿，…依次类推，如果少于512个，要在1～9#筐之间拿。若所要的苹果个数大于512，便给10#，并在1～9#筐间补余数。

（二）程 序 设 计

1）设计思路

定义数组K（10）和M（10）用来装10筐苹果。用循环语句据算法分析，将2^0、2^1、……、2^9依次输入相应数组M（1）～M（10）。由键盘敲入你所要的苹果数N，来找应付的哪几筐。先用最大筐M（10）即512和N相比，若N大于M（10），则N减去M(10)，即N⇐N－512，余下的N再与M（10）比较，若N＜M（10），再依次比较N＜M(9)吗，若成立，则比较N＜M（8）吗，是，则N⇐N－M（8），如此反复比较，直至N＝0为止。凡是减去的筐数赋给数组K，最后输出数组K便是所付的筐数。

2）框图　见图69

3）源程序及运行结果

```
      DIMENSION  M(10) , K(10)
2     WRITE (10)  (39H  REQUIRED  NUMBER (1<=N<=1000  N=
      -1 : STOP) )
      READ (11, 5) N
5     FORMAT (I 4)
      IF (N. EQ. -1) STOP
      N1 = N
      WRITE (10, 10) N
10    FORMAT (/2X, 2HN=, I 4)
      DO  20  I = 1, 10
      L = I - 1
      M(I) = 2**L
20    K(I) = 0
      IF (N. GE. M (10))GOTO  70
35    DO  60  I = 1, 9
      J = 10 - I
      IF (N-M (J))  60, 61, 62
60    CONTINUE
      STOP  'ERR'
61    K(J) =M(J)
      GOTO  40
62    N = N-M(J)
      K(J) =M(J)
      GOTO  60
70    N = N-M(10)
      K(10) =M(10)
      IF(N. EQ. O) GOTO  40
      GOTO  35
40    WRITE (10, 50) N1,  (K(JJ) , JJ= 1, 10)
50    FORMAT (5X, I4, 1H=, 9 (I3, 1H+) , I3)
      GOTO  2
      END
```

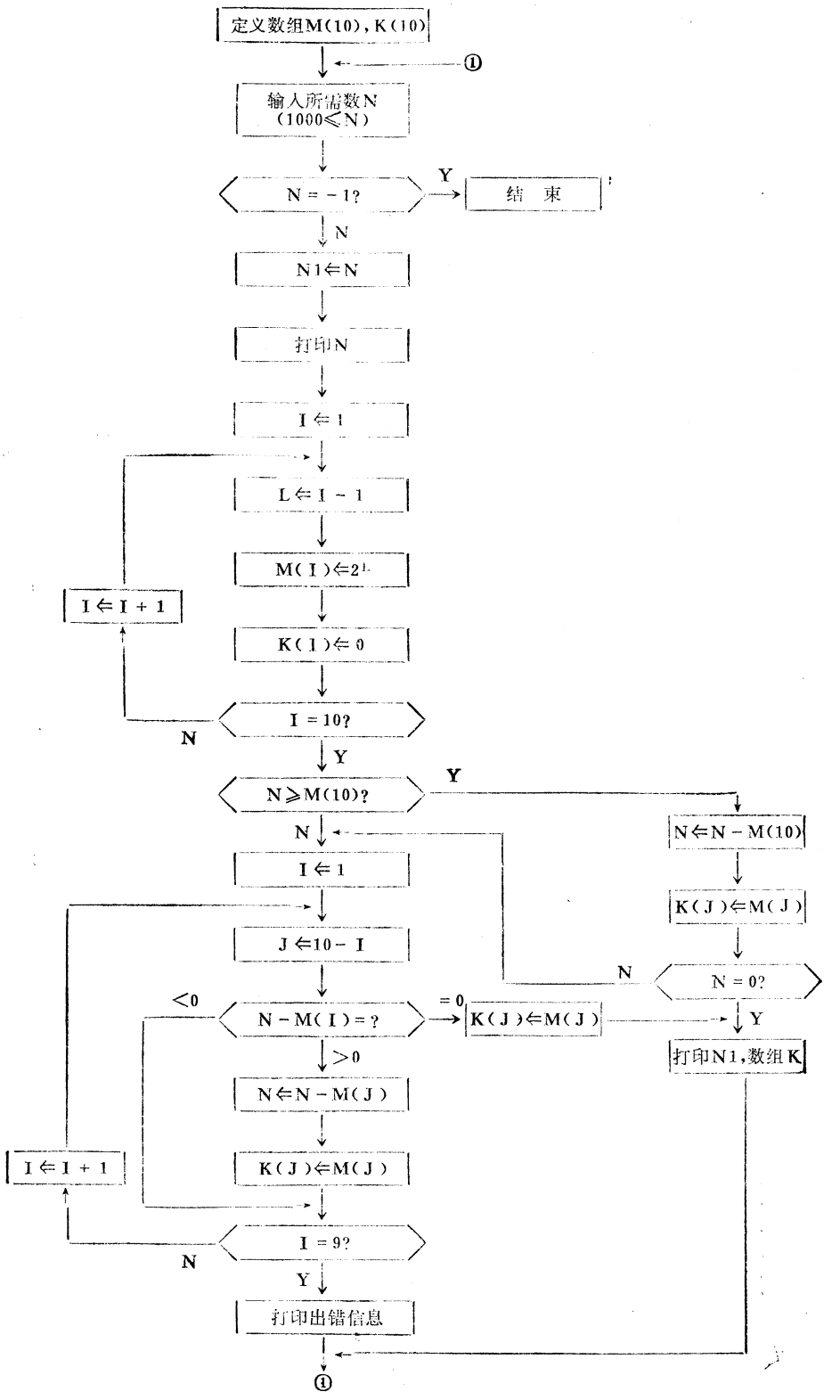

定义数组M(10),K(10)
①
输入所需数N
(1000≤N)
N = -1?
Y
结 束
N
N1⇐N
打印N
I ⇐ 1
L⇐I - 1
M(I)⇐2^L
K(I)⇐0
I⇐I + 1
I = 10?
N
Y
N≥M(10)?
Y
N
N⇐N - M(10)
I ⇐ 1
K(J)⇐M(J)
J ⇐10 - I
N
N = 0?
<0
N - M(I) = ?
= 0
K(J)⇐M(J)
Y
打印N1,数组K
>0
N⇐N - M(J)
I⇐I + 1
K(J)⇐M(J)
I = 9?
N
Y
打印出错信息
①

图 69

```
REQUIRED  NUMBER (1<=N<=1000  N=-1 : STOP)程序运行后打印结果
1000
N=  1000
     1000=   0 +   0 +   0 +   8 +   0 +32 +  64 +128 +256 +512
                                                    等号左端是所要的数量
REQUIRED  NUMBER (1<=N<=1000  N=-1      等号右端是所付的数量
: STOP)                                          (每加一项是一个筐)
100
N=100
    100=   0 +   0 +   4 +   0 +   0 +  32 +  64 +   0 +   0 +   0
REQUIRED  NUMBER (1<=N<=1000  N=-1 : STOP)
10
N=10
     10=   0 +   2 +   0 +   8 +   0 +   0 +   0 +   0 +   0 +   0
REQUIRED  NUMBER  (1<=N<=1000  N=-1 : STOP)
  1
N=   1
    1=   1 +   0 +   0 +   0 +   0 +   0 +   0 +   0 +   0 +   0
REQUIRED  NUMBER (1<=N<=1000  N=-1 : STOP)
658
N=658
    658=   0 +   2 +   0 +   0 +  16 +   0 +   0 +128 +   0 +512
REQUIRED  NUMBER (1<=N<=1000  N=-1 : STOP)
640
N=  640
    640=   0 +   0 +   0 +   0 +   0 +   0 +   0 +128 +   0 +512
REQUIRED  NUMBER (1<=N<=1000  N=-1 : STOP)
-1
STOP
```

（三）思　考　题

甲记住一个位于1和1024之间的正整数，让乙来猜，如果对于乙的猜测，甲只能回答“是”或“不是”，问怎样以最少的次数猜测后，就一定能猜中？

笔算分析：在具体猜测时，可采用逐步对半缩小范围的方法：首先将1024分为二相等部分（1到512和513到1024），据第一次猜测判断此数在那一部分，把所在范围缩至512个范围数内；然后将所得部分再对半分来进行猜测，把所在范围缩至256个数范围内；逐次对半猜测至第九次，范围被缩到仅剩两个数，于是再猜一次，便一定可以判断出甲决定的数。

举例说明如下：

假定甲所选的数为2，则十次猜测过程如下：

（1）“选好的数大于512？”“不是”

（2）“大于256”？“不是”

（3）“大于128”？“不是”

（4）“大于64”？“否”

（5）“≥32”？“否”

（6）“≥10”？“否”

（7）“＞8”？“否”

（8）“＞4”？“否”

（9）“＞2”？“否”

（10）“＞1”？“是”

∴甲选之数等于2

又如甲选之数为924，则

（1）“所选数＜512？”“否”

（2）“＜768？”“否”

（3）“＜896？”“否”

（4）“＜960？”“是”

此时乙必须记住甲所选之数在896与960之间

（5）“＜928？”“是”则要猜的数在896～928之间（相隔32个数）

（6）“＜912？”“否”范围缩小到16个数

（7）“＜920？”“否”范围缩小到8个数

（8）“＜924？”“否”范围缩小到924～928

（9）“＜926？”“是”范围缩小到924～925

（10）“是924？”“是”

当猜测次数少于10次时，虽然因猜测的选择，有时也可猜中，但少于10次猜测将是不充分的。

请你编写程序做这种游戏并进行验证对否。

70 找出这样的四位数

在2000以内的四位数中，有这样一个四位数：其个位数与千位数之差等于其百位数与十位数之差；其个位数之和等于其百位数与千位数之和，且个位数与十位数（或百位数与千位数）之和是个位数与千位数（或百位数与十位数）之差的10倍。

问这个四位数是多少？

（一）笔算步骤和结果

设此四位数为a b c d，依题意可知d－a＝b－c

c＋d＝a＋b，c＋d＝（d－a）×10

因为c＋d≤9＋9＝18＜20，又因为c＋d≠0，所以d－a＝1，c＋d＝10，因为d、c不全为0，又因为a b c d在2000以内，所以a＝1，d＝1＋a＝2，c＝10－d＝8。

因而 b＝d－a＋c＝2－1＋8＝9。

所以至少有一个四位数为1982。有否大于2000的其它四位数，请编写程序，让计算机搜寻所有满足条件的这样四位数。

（二）程序设计

1）设计思路

设这四位数为A B C D，给出四个条件：D－A＝B－C，D＋C＝B＋A，D＋C＝10(D－A)，B＋C＝10（B－C），但其中有重复条件．前两个条件实际由一个条件变换即可，因而后两个条件也如此（可变为一个条件），即共有两个条件。利用四重循环确定A B C D，第一个字不会是0，循环控制变量由1开始至9，其余皆由0～9。当满足上述两个判断条件时，便打印出一组解，当循环完毕，便找出所有的解。

2）框图　见图70

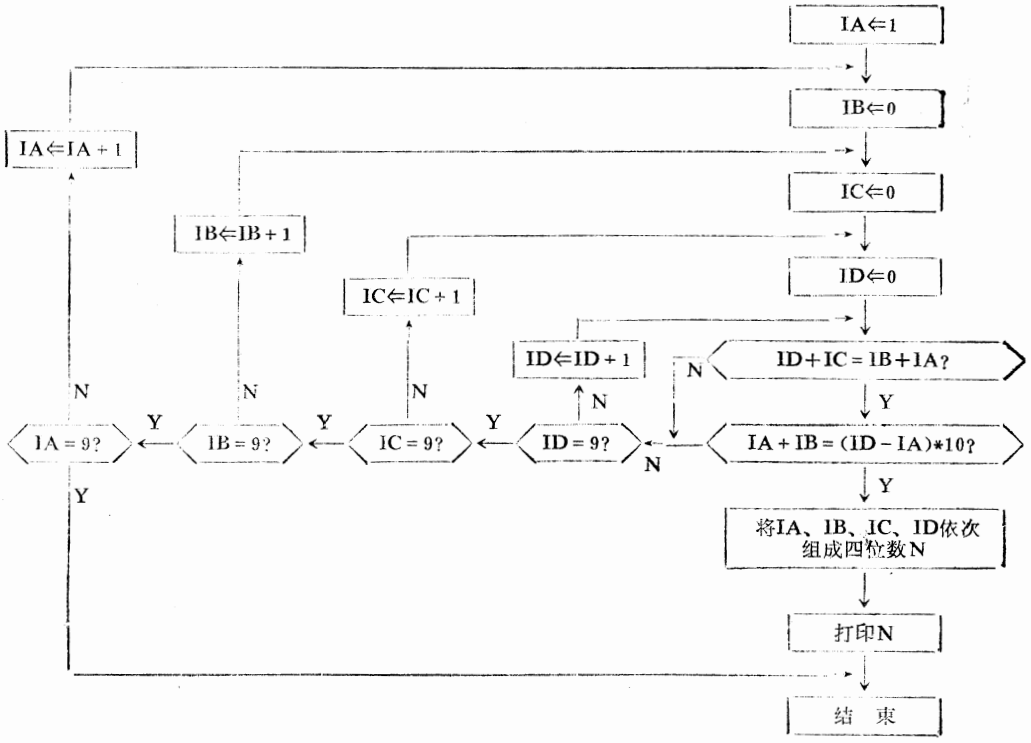


图 70

3）源程序及运行结果

```
      DO  10  IA = 1,  9
      DO  10  IB = 0,  9
      DO  10  IC = 0,  9
      DO  20  ID = 0,  9
      IF ( (IA+IB) . NE.  (IC+ID) ) GOTO  20
      IF ( (IA+IB) . NE.  (ID-IA) *10) GOTO  10
      N = 1000*IA + 100*IB + 10*IC + ID
      WRITE ( 5, 50) N
50    FORMAT (15X, I 4 )
20    CONTINUE
10    CONTINUE
      STOP
      END
```

1982
2873
3764
4655
5546
6437
7328
8219

程序运行时打印结果

（三）思 考 题

一个四位数，其千位、百位、十位、个位数字依次组成等差数列，百位上的数字是个位、千位数字的等比中项，把该四位数的数字反着次序排列所得的数与原四位数字之和为11110，求原四位数。

71 求出特定的四位数

求一个四位数 a b c d，它是11的倍数，且 b + c = a，同时 b c 是完全平方数。

（一）笔算步骤和结果

∵ a b c d 可被11整除，∴ a + c − (b + d) = 11K，∵ a + c ≤ 9 + 9 = 18，∴ K = 0，或K = 1，下面分别讨论 K为 0 或 1 时的两种情况。

（1）当K = 0时，使得 a + c − b − d = 0

∵ b + c = a，∴ b = a − c，将 a、b 代入式中得 b + c + c − b − d = 0，从而 2c = d。

∵ b c 为完全平方数，又∵一位或二位的完全平方数有 0，1，4，9，16，25，36，49，64，81。即 c ∈{6，5，9，0，1，4}，因 2c = d，0 ≤ d ≤ 9，∴ c ∈{0，1，4}

当 c = 0 时，b = 0，d = 0，但这时无论 a 为何数（除 0 外），而明显 a ≠ 0，a b c d 均不成为11的倍数，∴ c ≠ 0，c ∈{1，4}。当 c 取 4 时，∵个位为 4 的一位或二位平方数只有 0 4，64。

又因 d = 2 c = 2 × 4 = 8，∴此时 b c d 后三位数为048、648。而648显然不成立，因为 a = b + c = 6 + 4 = 10 > 9，（10不是一位数字）。

当 b = 0，c = 4，d = 8 时，a = b + c = 0 + 4 = 4 时即四位数4048满足条件。

当 c = 1，d = 2，此时 b = 0 或 b = 8。

当 b = 0 时，a = b + c = 0 + 1 = 1，∴四位数1012也为一解。

当 b = 8 时，a = b + c = 8 + 1 = 9，∴四位数9812也为一解。

（2）讨论当K = 1 时，a + c − b − d = 11。

在上述（1）可导出 2c = d + 11，∵ 0 ≤ d ≤ 9，d = 2c − 11，∴ c > 5，即 c ∈{6，9}。当 c = 6 时，d = 1，b 取 3，a = b + c = 3 + 6 = 9，∴其四位数9361为一组解。

当 c = 9，d = 2c − 11 = 18 − 11 = 7，故 b = 0 或 4。当 b = 4 时，a = b + c = 4 + 9 = 13 > 9（显然不成立），由此 b 只有取 0，a = b + c = 0 + 9 = 9。

则四位数9097又为一组解。

如此继续计算，尚有1012，4048，7161，9812等组解。

（二）程 序 设 计

1）设计思路

虽然是求四位数A B C D，据已知条件，用二重循环，求B和C便可。判B C是完全平方数时，将B + C⇒A。因A是四位数的首位，它不应大于9或等于0，满足这些条件时便打印出一组A B C D。继续循环再找，直至出外层循环便将所有可能的四位数全找出。

2）框图　见图71

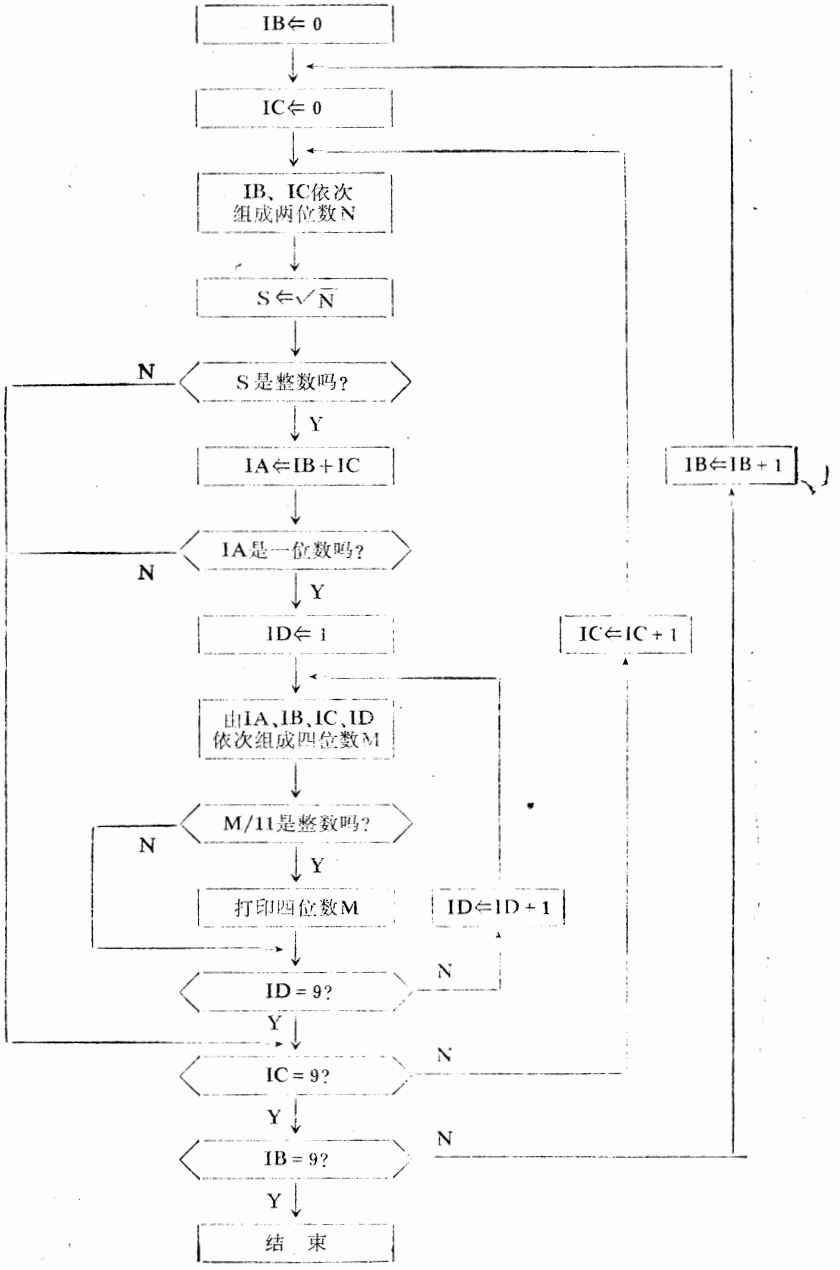


图 71

3）源程序及运行结果

```
      DO  10  IB = 0 , 9
      DO  20  IC = 0 , 9
      N = 10*IB + IC
      S = SQRT (FLOAT (N))
      IF (ABS (IFIX ( S) - S) . GT. 0.00001)  GOTO  20
      IA = IB + IC
      IF((IA. GT. 9 ) . OR.  (IA. LE. 0 ))  GOTO  20
      DO  30  ID = 1 , 9
      M = 1000*IA + 100*IB + 10*IC + ID
      IF (M. NE. M/11*11) GOTO  30
      WRITE ( 5 , 40) M

40    FORMAT(15x, I4)                    1012
30    CONTINUE                           4048
20    CONTINUE                           9097
10    CONTINUE                           7161   程序运行时打印结果
      STOP                               9361
      END                                9812
```

（三）思　考　题

已知四位数满足下列条件。

（1）若同时将其个位数字与百位数字，十位数字与千位数字的位置互换，则其数值增加5940；

（2）除以 9 余 8 。

求：这些四位数中的最小奇数。

72　五个人的年龄

王、张两位老师一起外出，年龄刚到半百的王老师，遇到他的三个邻居（一个大人、两个孩子）。王对张说：“他们三个人的年龄加起来恰好是你的年龄的两倍，三个岁数相乘，乘积是2450，他们个人的年龄虽然都比我小，但大人的年龄与我相差无几”。请问这五个人的年龄各多少？

（提示：先把2450分解成三个数乘积，要把各种可能都列出来）。

（一）笔算步骤和结果

把2450分解成三数的乘积，有下列各种可能（括号数字是三个数之和）：

1 ×49×50 （100）； 2 ×25×49 （76）； 5 × 7 ×70 （82）； 5 × 10 ×49 （64）；

5 ×14×35 （54）； 7 × 7 ×50 （64）； 7 ×10×35 （52）； 7 ×14×25 （46）。

还有一些分解，比如 1 ×25×98， 2 × 5 ×245等不适合的年龄未列出。由题意分析，王老师是50岁（半百），只有 5 ×10×49 （64）这一组与题意相符。

答案是：王老师50岁，张老师是64/2 = 32 （岁），三个邻居中大人为49岁（比王老师年岁小，但最接近），两个孩子年龄分别为 5 岁和10岁。

（二）程 序 设 计

1）设计思路

由题分析，王老师年龄半百即50，令IW＝50，三个邻居用I1、I2、I3表示循环控制变量。大人的年龄与王老师接近，设I1在45～50岁之间，两个孩子I1与I2的年龄，设在1～15岁范围，张老师年龄IZ＝(I1＋I2＋I3)/2，将几种可能分别打印出来。

2）框图　见图72

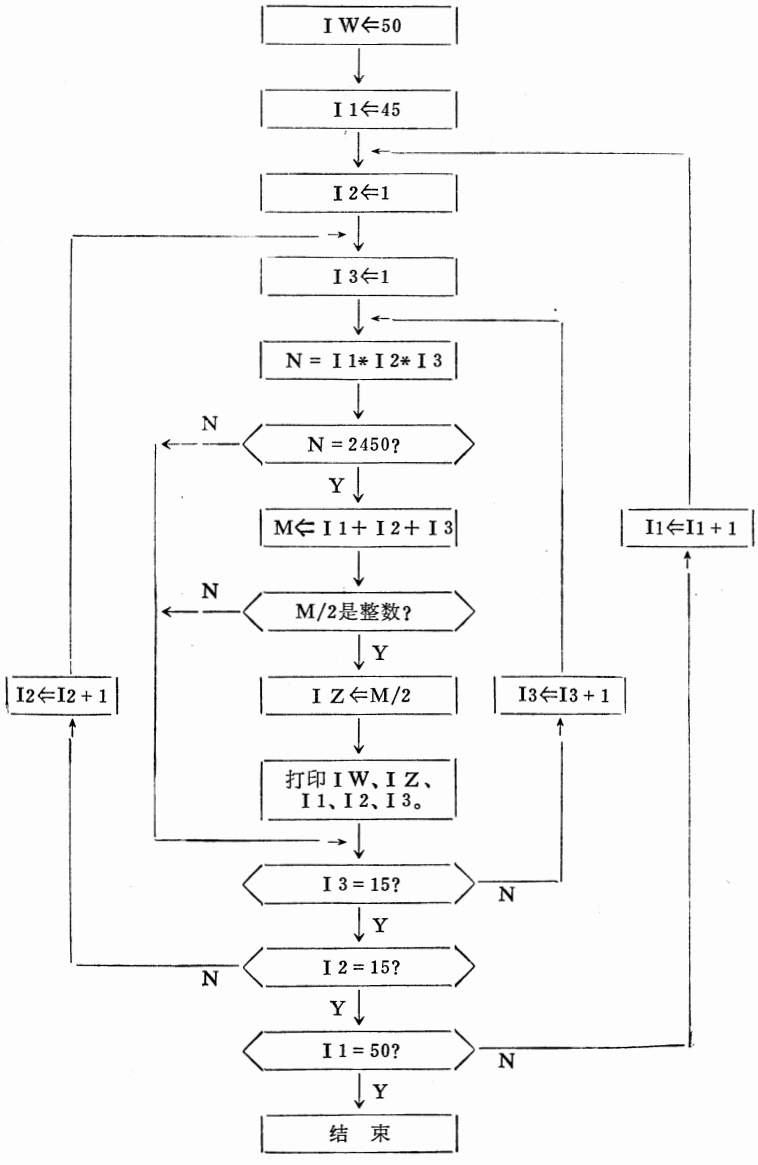


图 72

3）源程序及运行结果

```
IW = 50
DO  20  I 1 = 45, 50
```

```
      DO  20  I2 = 1 , 15
      DO  30  I3 = 1 , 15
      N = I1 * I2 * I3
      IF (N - 2450)  30, 50, 20
50    M = I1 + I2 + I3
      IZ = M/2
      IF (M. NE. IZ * 2) GOTO  20
      WRITE (5, 40) IW, IZ, I1, I2, I3
      GOTO  20
40    FORMAT (10X, 5 I 4)
30    CONTINUE
20    CONTINUE
      STOP
      END
```

```
50  32  49   5  10
50  32  49  10   5
50  32  50   7   7
```

三种情况，数字依次为王老师、张老师、邻居大人、邻居两个孩子的年龄。

（三）思 考 题

李老师多少岁——张老师问李老师："你今年多少岁？你爱人今年多少岁？"李老师笑笑说："我年岁的平方与我爱人年岁之和恰好等于1053。而我爱人年岁的平方与我的年岁之和却等于873。你计算一下吧！"

算法分析，框图和程序设计等，详见《趣味程序设计100例》第73题。

73 数学谜的年龄

少林是班里有名的数学谜，有人问他今年几岁，他说"我的岁数的三次方是个四位数，四次方是个六位数。要组成我岁数的三次方和四次方，需要用遍从0～9十个数字。"请问少林今年几岁？

（一）笔算步骤和结果

三次方是四位数，四次方是六位数；可见不会大于22岁（$\because 22^3 = 10648$，已经超过四位数），也不会小于17岁（$17^4 = 83521$，还不到六位数），只能从18，19，20，21四个数字中去找。

组成少林岁数的三次方和四次方的十个数码字，必须是从0到9这十个数全都用遍，即不能漏掉任何一个，也不能有任何重复。而$20^3 = 8000$，$19^4 = 130321$，$21^4 = 194481$，这三种数字都出现数码重复。余下的只有18，$18^3 = 5832$，$18^4 = 104976$，完全符合上述两个条件，可见少林今年是18岁。

（二）程 序 设 计

1）设计思路

该数字应是两位数，利用循环语句寻找该两位数。循环控制变量初值为10，21为终值，步长为1。

定义数组K（10），循环变量I。将其立方赋给整型变量K1，将其四次方赋给实型变量S2。判K1、S2符合四位数和六位数后，将四位数K1分离出千、百、十、个位数字分别置入K（1）～K（4）。将六位数S2分离出十万、万、千、百、十、个位数字，分别置入K（5）～K（10）。利用二重循环比较数组K（10）中10个数是否有相同的，有则不合题意要求，重新更换I值，按上述判定。若数组K（10）中10个数全不相同，则已求出。打印第一行排列为I^3和I^4，其后为不同的10个数字，打印第2行是进一步指出I^3的四位数字，I^4的六位数字。

2）框图　见图73

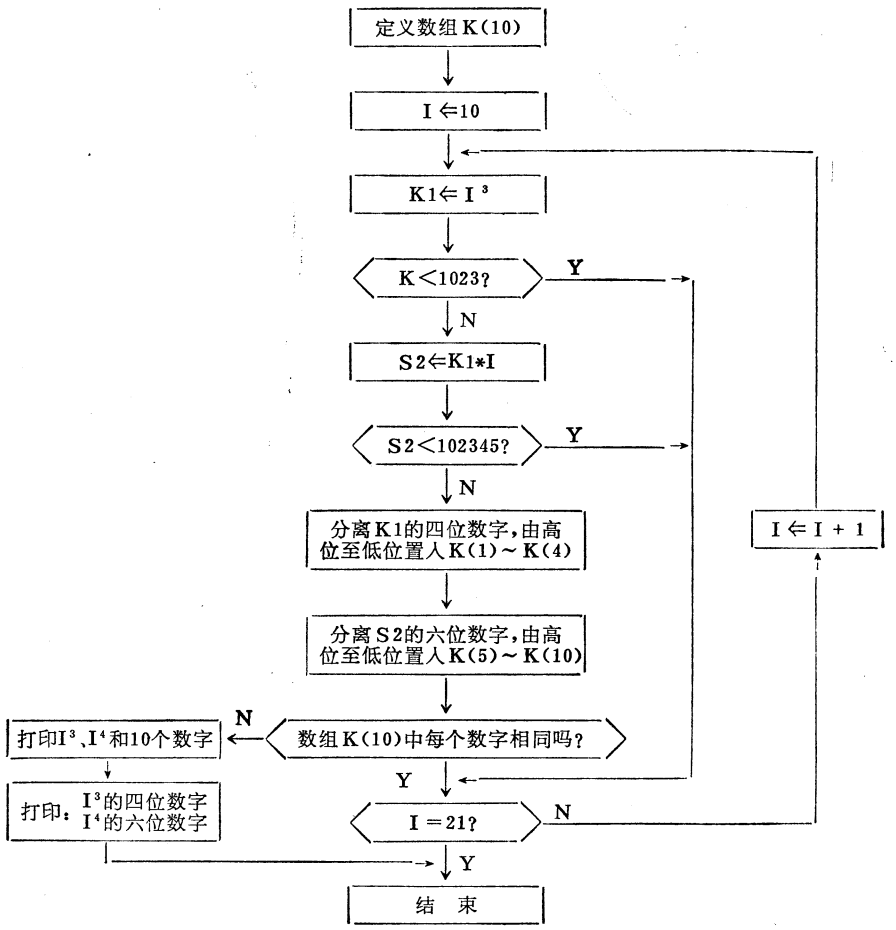


图 73

3）源程序及运行结果

```
INTEGER  K(10)
DO  10  I=10, 21
K1=I**3
IF (K1. LT. 1023) GOTO  10
S2=FLOAT (K1) *FLOAT (I)
IF (S2. LT. 102345. 0) GOTO  10
```

```
      K (1) =K1/1000
      K (2) =MOD (K1, 1000) /100
      K (3) =MOD (K1, 100) /10
      K (4) =MOD (K1, 10)
      K (5) =INT (S2/100000. 0)
      K (6) =INT ( (S2-INT (S2/100000. ) *100000. ) /10000. )
      K (7) =INT ( (S2-INT (S2/10000. ) *10000. ) /1000. )
      K (8) =INT ( (S2-INT (S2/1000. ) *1000. ) /100. )
      K (9) =INT ( (S2-INT (S2/100. ) *100. ) /10. )
      K (10) =INT (S2-INT (S2/10. ) *10. )
      DO  20  J=1, 9
      JJ=J+1
      DO  20  L=JJ, 10
      IF (K(J) . EQ. K(L) ) GOTO  10
20    CONTINUE
      WRITE (5, 30) I, I, K
      WRITE (5, 40) I,  (K(J) , J=1, 4) , I,  (K(J) , J=5, 10)
      STOP  OK!
30    FORMAT (5X, I2, '**3&', I3, '**4=', 10I3)
40    FORMAT (5X, I2, 4H**3=, 4I1, 5X, I2, 4H**4=, 6I1)
10    CONTINUE
      STOP  ERROR
      END

      18**3  &  18**4=   5  8  3  2  1  0  4  9  7  6
      18**3=5832          18**4=104976
```

（三）思 考 题

一位中学生说他的年龄加上10，开方以后得到的一个平方根，恰好是他的年龄减去10，问他的年龄多大？

74 里 程 碑

甲、乙两个城市之间，有一条999公里长的公路。公路旁每隔一公里竖立着一个里程碑。里程碑的半边写着从甲城到乙城的距离，另半边写着从乙城到甲城的距离。

有位汽车司机开车从甲城去乙城，他注意到有的里程碑上所写的数仅用了两个不同数字，例如：|0|999|仅用了0和9。他想数一数具有这种特征的里程碑共有多少个，可是他没有数清。从乙城返回甲城时，他又想再数一数，由于让车、躲人，分散了他数里程碑的注意力，还是没有数成。为了让他安心开车，以免出事故，我们替他算一算，这样的里程碑共有多少个。

（一）笔算步骤和结果

我们先考虑从甲城到乙城的里程数。任一里程数可用一个三位数表示，记为

(abc) =100a+10b+c，其中a、b、c是0，1，2，…，9中的某一个数字。若选定数字

a和数字b来组成三位数，那么只可能有下面八种：

（a a a），　（a a b），　（a b a），　（a b b），

（b b b），　（b b a），　（b a b），　（b a a）。

在同一块里程碑上，还标志着从乙城到甲城的里程数。由于同一里程碑上的左右两个里程数之和等于999，所以与上面八种里程碑数标在同一里程碑上的里程数应对应为：

（9－a，9－a，9－a），（9－a，9－a，9－b），（9－a，9－b，9－a），（9－a，9－b，9－b），

（9－b，9－b，9－b），（9－b，9－b，9－a），（9－b，9－a，9－b），（9－b，9－a，9－a）。

要使里程碑上只用到两个数字，那么必有9－a＝b，9－b＝a。故上面八个从乙城到甲城的里程数为：

（b b b），　（b b a），　（b a b），　（b a a），

（a a a），　（a a b），　（a b a），　（a b b）。

在0，1，2，…，9十个数字中，适合a＋b＝9的数字只有五个，它们是（0，9），（1，8），（2，7），（3，6），（4，5）。所以在这条公路上仅用了两个数字表示里程碑共有5×8＝40个。

把它们列出来就是：

9，990	118，881	227，772	…	990，9
90，909	181，818	272，727		909，90
99，900	811，188	722，277		900，99
0，999	111，888	222，777		999，0

（二）程序设计

1）设计思路

利用循环语句，其循环控制变量I为0～999，做为左半边数字，则右半边数字N⇐999－I，然后据算法分析转子程序判合乎条件的三位数字，置放在数组L（4）、M（4）、返主后，每组打印在一行上，即打印数组M（4），L（4）。程序搜寻后，与结论完全同，请见程序运行后打印结果。

2）框图　见图74

3）源程序及运行结果

```
      DIMENSION  L(4), M(4)
      K = 0
      DO  10  I=0, 999
      N=999-I
      IF (I. GE. 900) GOTO  30
      CALL  YY (N, I, K, L, M)
      IF (K. LT. 4) GOTO  10
      WRITE (10, 20)  (M(KK), L(KK), KK = 1, 4)
20    FORMAT (1X, 4 (5X, I3, 2H, I3) )
      DO  25  J=1, 4
      L (J) = 0
```

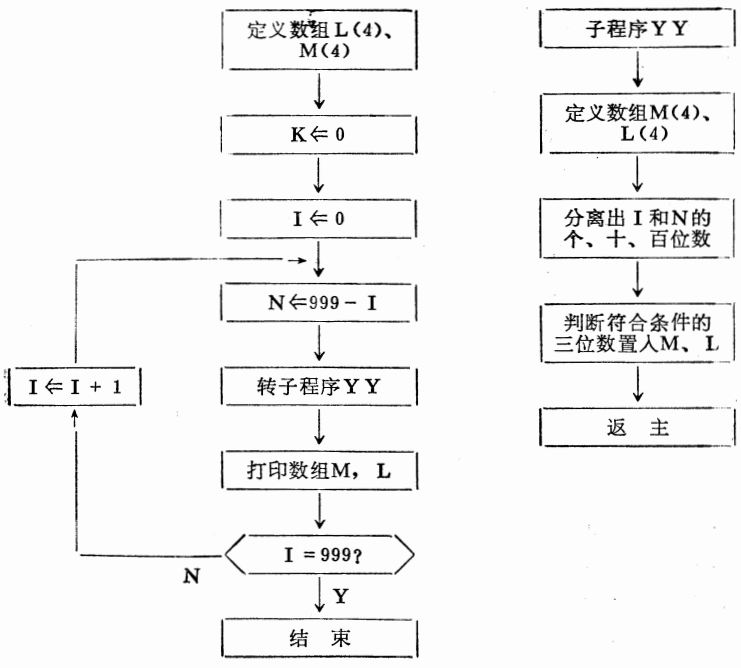


图 74

```
25      M(J) = 0
        K = 0
10      CONTINUE
        STOP
30      CALL  YY (I, N, K, L, M)
        IF (K. LT. 4) GOTO 10
        WRITE (10, 20) (L(KK) , M(KK) , KK = 1, 4 )
        DO 35 J = 1 , 4
        L(J) = 0
35      M(J) = 0
        K = 0
        GOTO 10
        END
        SUBROUTINE YY (N1, I, K, L, M)
        DIMENSION L( 4 ) , M( 4 )
        IC = N1 - N1/10*10
        IB = (N1 - N1/100*100) /10
        IA = N1/100
        IF (IA. EQ. IB. AND. IA. EQ. IC) GOTO 106
        IF (IA. NE. IB. AND. (IB. EQ. IC. OR. IA. EQ. IC) ) GOTO 100
        IF (IA. EQ. IB. AND. IB. NE. IC) GOTO 100
        GOTO 200
100     DO 105 J = 1 , 3
```

```
      GOTO (101, 102, 103) J
101   I1 = I/100
      GOTO 104
102   I1 = (I - I/100*100) /10
      GOTO 104
103   I1 = I - I/10*10
104   IF (I1. NE. IA. AND. I1. NE. IB. AND. I1. NE. IC) GOTO 200
105   CONTINUE
106   K = K + 1
      L (K) = N1
      M (K) = I
200   RETURN
      END
```

0,	999	9,	990	90,	909	99,	900
111,	888	118,	881	181,	818	188,	811
222,	777	227,	772	272,	727	277,	722
333,	666	336,	663	363,	636	366,	633
444,	555	445,	554	454,	545	455,	544
544,	455	545,	454	554,	445	555,	444
633,	366	636,	363	663,	336	666,	333
722,	277	727,	272	772,	227	777,	222
811,	188	818,	181	981,	118	888,	111
900,	99	909,	90	990,	9	999,	0

（三）思 考 题

请你适当改变程序，将里程碑上三个不同数字，也一起找出来。

75 第十八届国际数学竞赛题

若干个正整数之和为1976，试求它们乘积的最大值。

（一）笔算步骤和结果

因为和为1976的正整数组总是存在的，并且只有有限多组，故所求的最大乘积也必然存在，若通过逐一试算，以求最大积，则计算量大，我们限定这些和为1976的若干个正整数中有一个不小于1969，求它们积的最大值。

先由下面试算中找规律，比如：

$1 + 1975 = 1976$，　$1 \times 1975 = 1975$；

$2 + 1974 = 1976$，　$2 \times 1974 = 3948$；

$1 + 2 + 1973 = 1976$，　$1 \times 2 \times 1973 = 3946$；

$2 + 3 + 1971 = 1976$，　$2 \times 3 \times 1971 = 11826$；

$3 + 4 + 1969 = 1976$，　$3 \times 4 \times 1969 = 23628$；

$5 + 1971 = 1976$，　$5 \times 1971 = 9855$；

$1 + 5 + 1970 = 1976$，　$1 \times 5 \times 1970 = 9850$；

…… ……

易知在所列出的积中，第五个最大。初步观察不难发现，加数的个数相对地多一些，各个加数要相对地小一点，并且：

1）凡加数中有1的，不能达到最大值（因为1与任何数之积等于该数本身）；

2）非最大的加数若大于4，则积相对地要小些。

下面论证如不限定若干个正整数中有一个不小于1969，求若干个正整数的值在什么范围内选取，使之相加仍为1976，而乘积最大。

方法一

设X_i（$i=1, 2, \cdots, n \leqslant 1976$）为正整数，且和

$$\sum_{i=1}^{n} X_i = 1976, \qquad (1)$$

则最大的乘积$\prod_{i=1}^{n} X_i$中：

1）任意的非最大的$X_i=1$，否则，不妨设$x_1=1$，并将1与另一任意加数，比如x_i合并，以$1+X_2$取代1与X_2这两个数，易知（1）不变，和仍为1976，而积中因子“$1 \cdot X_2$”代之以“$1+X_2$”，显然

$$1 \cdot X_2 \cdot X_3 \cdots X_n < (1+X_2) \cdot X_3 \cdots X_n。$$

即积增大。

2）任意的非最大的$X_i \leqslant 4$。否则，不妨设$X_1 > 4$，此时可令

$$X_1 = 2+a \qquad (a>2),$$

易知（1）不变，和仍为1976；而积中因子X_1，以

$$2 \cdot a = 2(X_1-2) = 2X_1 - 4 = X_1 + (X_1-4) > X_1$$

代替，显然

$$(2+a) X_2 \cdot X_3 \cdots X_n < 2 \cdot a \cdot X_2 \cdot X_3 \cdots X_n$$

即积增大。

由1）、2）可知，最大积中的因子X_2只能在2、3、4中选取。然而$4=2+2=2\times 2$，故4可看作两个2，因而X_i只能在2与3中选取。这就是说，最大积为$2^m \times 3^n$的形式，其中自然数m、n满足条件

$$2m + 3n = 1976 \qquad (2)$$

因为三个2可用二个3代替，和不变，但乘积增大，所以2的个数$m=1$或2。（和不变、积增大举例：$8=2\times 2\times 2 < 3\times 3 = 9$。）

但由（2）知$m \neq 2$，否则$n = 1972/3 = 657.3$不是整数，因而只能是$m=1$，此时$n=(1976-2)/3=658$。

故所求最大乘积为2×3^{658}

方法二

在确定非最大的任意的$X_i \leqslant 4$时，也可以采用下述的反证，其余部分同方法一。

假定$X_i \leqslant 4$，则至少存在一个自然数i，使$X_i > 4$，对此不等式作如下变形，依次得到：

$X_i \geqslant 5$，$X_i > 4.5$，$2X_i > 9$，$3X_i - 9 > X_i$，

$$X_i < 3(X_i - 3)。$$

这表明，当把X_i换成3与X_i-3乘积时，和不变，但乘积增大，与

$\prod_{i=1}^{n} X_i$为最大的乘积矛盾，因而必有非最大的$X_i \leqslant 4$。

（二）程 序 设 计

1）设计思路

据上述限定若干个正整数中有一个不小于1969来编写程序，由前述分析其余的正整数不妨设1、2、3、4、5为加数中的某几个数，加数个数不同，看乘积是哪个大，大者便保存于N单元，最后将所有加数的各种可能全试算完，则打印出暂存单元N中的数（便是其中最大的乘积数），并将哪几个数相加也一起打印出，检验是否上述笔算分析第五个最大（指乘积）。

定义数组K(5)，置放1～5(年)。主程序依次安排1～5不同组合数字（最少一个数，最多五个数），分别转P1～P5等五个子程序去处理：一个年数、两个年数、…五个年数的乘积，最大的乘积存于N。

2）框图　见图75

3）源程序及运行结果

```
      DIMENSION  K(5)
      DO  5  I=1, 5
5     K(I) = 0
      N=0
      JP=1
10    K1=1
      CALL  P1 (K1, N, K)          转子程序处理1～5中一个数
20    K2=2
      GO  TO (21, 22) JP
21    CALL  P2 (K1, K2, N, K)
      GO  TO  30
22    CALL  P1 (K2, N, K)
30    K3=3
      GO  TO (31, 32, 33) JP
31    CALL  P2 (K1, K2, K3, N, K)          转子程序处理1～5中两个数
      GO  TO  40
32    CALL  P2 (K2, K3, N, K)
      GO  TO  40
33    CALL  P1 (K3, N, K)
40    K4=4
      GO  TO (41, 42, 43, 44) JP
41    CALL  P4 (K1, K2, K3, K4, N, K)
      GO  TO  50
42    CALL  P3 (K2, K3, K4, N, K)          转子程序处理1～5中三个数
      GO  TO  50
43    CALL  P2 (K3, K4, N, K)
      GO  TO  50
```

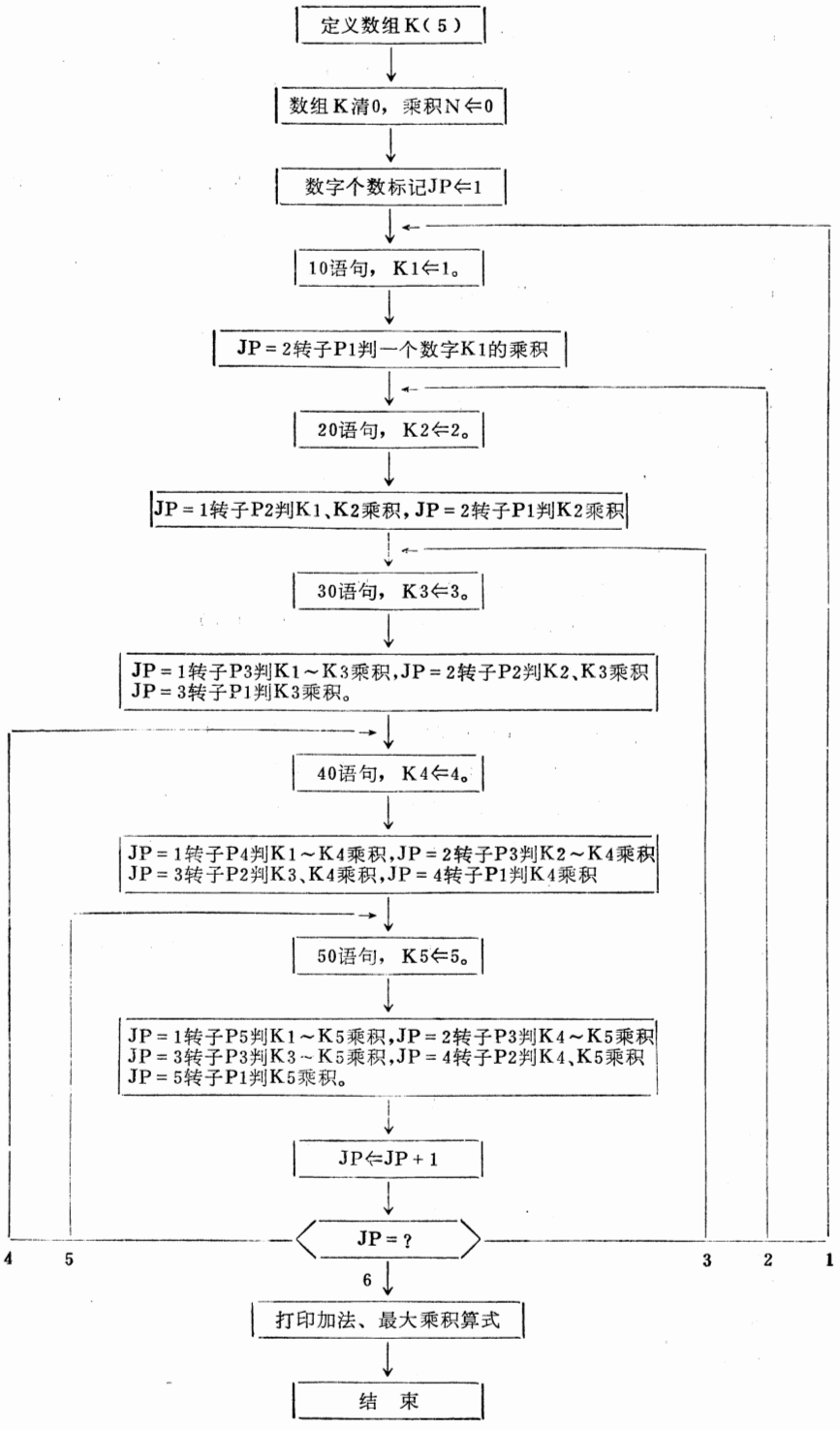

定义数组K(5)
数组K清0，乘积N⇐0
数字个数标记JP⇐1
10语句，K1⇐1。
JP=2转子P1判一个数字K1的乘积
20语句，K2⇐2。
JP=1转子P2判K1、K2乘积，JP=2转子P1判K2乘积
30语句，K3⇐3。
JP=1转子P3判K1～K3乘积，JP=2转子P2判K2、K3乘积
JP=3转子P1判K3乘积。
40语句，K4⇐4。
JP=1转子P4判K1～K4乘积，JP=2转子P3判K2～K4乘积
JP=3转子P2判K3、K4乘积，JP=4转子P1判K4乘积
50语句，K5⇐5。
JP=1转子P5判K1～K5乘积，JP=2转子P3判K4～K5乘积
JP=3转子P3判K3～K5乘积，JP=4转子P2判K4、K5乘积
JP=5转子P1判K5乘积。
JP⇐JP+1
JP=?
4
5
3
2
1
6
打印加法、最大乘积算式
结 束

图 75

```
44      CALL  P1 (K5, N, K)
50      K5 = 5
        GO  TO (51, 52, 53, 54, 55) JP
51      CALL  P5 (K1, K2, K3, K4, K5, NK)          转子程序处理 1 ～ 5 中五个数
        JP = JP + 1
        GO  TO  59
52      CALL  P4 (K2, K3, K4, K5, N, K)            转子程序处理 1 ～ 5 中四个数
        JP = JP + 1
        GO  TO  59
53      CALL  P3 (K3, K4, K5, N, K)
         JP = JP + 1
        GO  TO  59
54      CALL  P2 (K4, K5, N, K)
        JP = JP + 1
        GO  TO  59
55      CALL  P1 (K5, N, K)
        JP = JP + 1
59      GO  TO (10, 20, 30, 40, 50, 60) JP
60      KJ=K ( 1 ) +K ( 2 ) +K ( 3 ) +K ( 4 ) +K ( 5 )
        KM = 1976 - KJ
        M = 1976
        WRITE ( 5 , 70) M,  (K (JJ) , JJ= 1 ,  5 ) KM
        WRITE ( 5 , 71) N,  (K (II) , II= 1 ,  5 ) KM
70      FORMAT (5X, I4, 1H=, 5 (I1, 1H+) , I4)
71      FORMAT (5X, I5, 1H=, 5 (I1, 1H*) , I4)
        STOP
        END
        SUBROUTINE  P1 (M1, N, K)                  子程序一个数M1的乘积
        DIMENSION  K( 5 )
        J1 = 1976 - M1
        N1 = M1 * J1
        IF (N1. LE. N) GO  TO  30
        N = N1
        DO  20  I= 1 ,  5
        IF (I. NE. M1) GO  TO  10
        K ( I ) =I
        GO  TO  20
10      K ( I ) = 0
20      CONTINUE
30      RETURN
        END
        SUBROUTINE  P2 (M1, M2, N, K)              子程序两个数M1、M2的乘积
        DIMENSION  K( 5 )
```

```
      J2 = 1976 - M1 - M2
      N2 = M1*M2*J2
      IF (N2. LE. N) GO TO 30
      N = N2
      DO 20 I = 1, 5
      IF (M1. NE. I) GO TO 10
      K(I) = I
      GO TO 20
10    IF (M2. NE. I) GO TO 15
      K(I) = I
      GO TO 20
15    K(I) = 0
20    CONTINUE
30    RETURN
      END
      SUBROUTINE P3 (M1, M2, M3, N, K)      子程序三个数M1、M2、M3的乘积
      DIMENSION K (5)
      J3 = 1976-M1-M2-M3
      N3 = M1*M2*M3*J3
      IF (N3. LE. N) GO TO 30
      N = N3
      DO 20 I = 1, 5
      IF (M1. NE. I) GO TO 10
      K(I) = I
      GO TO 20
10    IF (M2. NE. I) GO TO 15
      K(I) = I
      GO TO 20
15    IF(M3. NE. I) GO TO 16
      K(I) = I
      GO TO 20
16    K(I) = 0
20    CONTINUE
30    RETURN
      END
      SUBROUTINE P4 (M1, M2, M3, M4, N, K)   子程序四个数M1～M4的乘积
      DIMENSION K(5)
      J4 = 1976-M1-M2-M3-M4
      N4 = M1 *M2*M3*M4*J4
      IF (N4. NE. I) GO TO 30
      N = N4
      DO 20 I = 1, 5
      IF (M1. NE. I) GO TO 10
```

```
      K(I) = I
      GO TO 20
10    IF (M2. NE. I) GO TO 15
      K(I) = I
      GO TO 20
15    IF (M3. NE. I) GO TO 16
      K(I) =I
      GO TO 20
16    IF (M4. NE. I) GO TO 17
      K(I) =I
      GO TO 20
17    K(I) = 0
20    CONTINUE
30    RETURN
      END
      SUBROUTINE P5 (M1, M2, M3,M4,M5,N, K)    子程序五个数M1～M5的乘积
      DIMENSION K(5)
      J5=1976-M1-M2-M3-M4-M5
      N5=M1*M2*M3*M4*M5*J5
      IF (N5. LE. N) GO TO 30
      N=N5
      DO 20 I=1, 5
20    K(I) =1
30    RETURN
      END
```

程序运行后打印结果

1976= 0 +0+ 3 +4+ 0 +1969	年数相加仍为1976
23628= 0 * 0 * 3 * 4 * 0 *1969	年数乘积最大者（其中0表示空位，并非乘0）。

（三）思 考 题

如果若干个正整数之和为1976，而又不限制若干个正整数中每个正整数的值，请编写程序，搜寻出哪几个正整数的乘积最大？

76 找出三个不同的数字

有三个不同的数字(其中没有零)，用它们可能组合的所有各个三位数的和都是2886。如果把三位从大到小和从小到大依次排列成两个三位数，其差为495。求这三位数？

（一）笔算步骤和结果

设这三位数为A、B、C，那么用它们可能搭配出来的所有三位数是：ABC、ACB、BAC、BCA、CAB、CBA。其和为100A + 10B + C = 222 (A + B + C) = 2286，所以，A + B + C = 13。

再设A的权是最大的一个数，B是中间的一个数，C的权是最小的一个数。据题意，那么（100A＋10B＋C）－（100C＋10B＋A）＝495，由此式可得出A＝C＋5。把A＝C＋5代入A＋B＋C＝13，则B＝8－2C。

从A＝C＋5和B＝8－2C可知，C不可能等于1。因为若C等于1，则A＝6，B＝6，不合题意；同时C也不等于3，因为C等于3，B就等于2，这样中间的数就小于最小的数了。因此C只可能等于2。代入上述关系式，即可知A＝7，B＝4。

所以这三个数分别为7，4，2。

（二）程 序 设 计

1）设计思路

利用三重循环，控制变量由1～9，分别寻找这三个数字，先判三个数字全不相同，再判三个数字之和等于13，以及三个数字顺排和逆排二者之差是否等于495。当符合条件时，打印出这三个数字。

2）框图　见图76

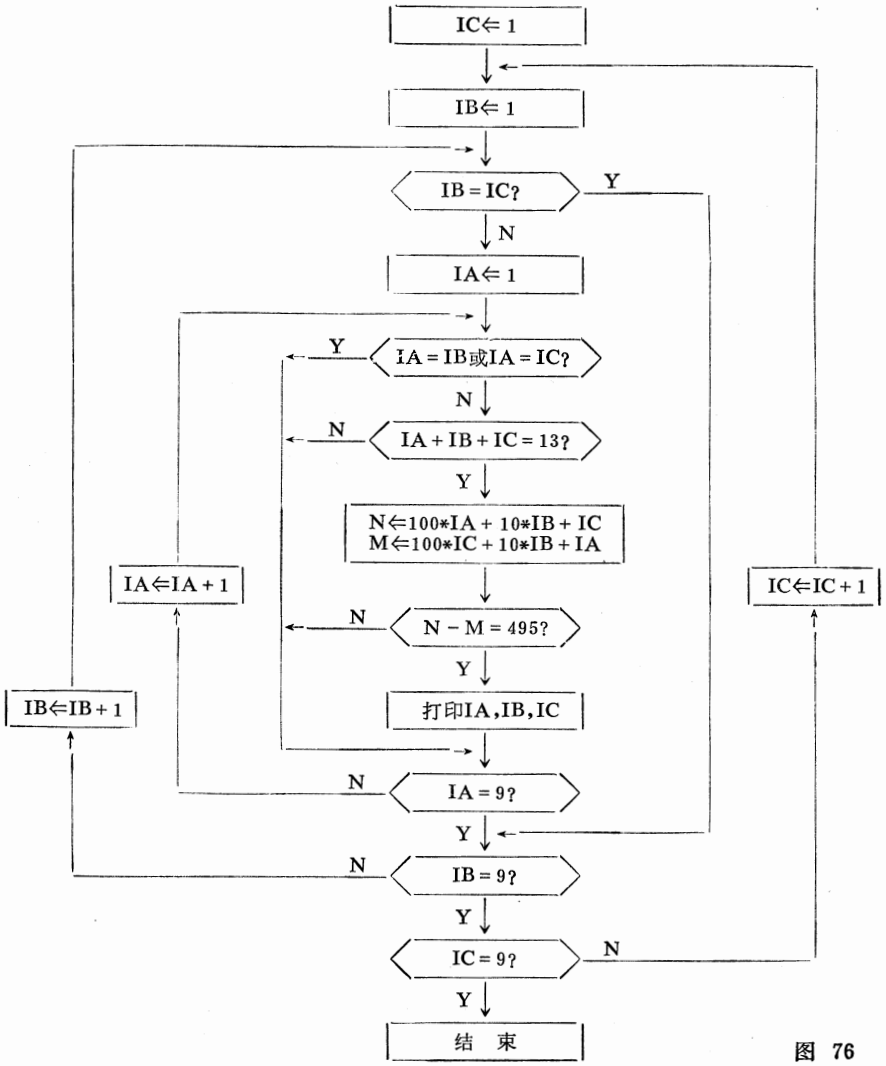


图 76

3）源程序及运行结果

```
      DO  10  IC=1, 9
      DO  20  IB=1, 9
      IF (IB. EQ. IC) GO  TO  20
      DO  30  IA=1, 9
      IF (IA. EQ. IB. OR. IA. EQ. IC) GO  TO  30
      IF (IA+IB+IC. NE. 13) GO  TO  30
70    N=100*IA+10*IB+IC
      M=100*IC+10*IB+IA
      IF (ABS (N-M) . NE. 495) GO  TO  30
      WRITE (6, 40) IA, IB, IC
      STOP'FIND'
40    FORMAT (10X, 2HA=, I1, 3X, 2HB=, I1, 3X, 2HC=, I1/)
30    CONTINUE
20    CONTINUE
10    CONTINUE
      STOP
      END
```

A=7　　B=4　　C=2　　（打印结果）

（三）思 考 题

把316这个数表示为两个加数的和，使其中一个加数能被13整除，而另一个加数能被11整除，并推出这两个加数是多少？

77　找出四个数之和的最大及最小

在下列所示杂乱分布的1～20个数字，首尾相连，试找出四个相邻的数之和最大，再找出四个相邻的数之和最小。20个数排序如下：

1，18，4，13，6，10，15，2，17，3，19，7，16，8，11，14，9，12，5，20。

（一）算法分析和结论

可先固定某一个起点，如从1开始按顺时针连续四个数相加求和为第一组，再从第五个数的6开始连加四个数为第二组，再从第九项的17连加四个数为第三组，…，当加到第五组时其终点项落在20，为第一次循环；第二次循环从18开始，加到第五组时落在1上；第三次从4开始…；第四次循环即最后一次从13开始…。共得20组数字，从中找出四个连续项相加：

最小者为　　4＋13＋6＋10＝33。

最大者为　　19＋7＋16＋8＝50。

（二）程 序 设 计

1）设 计 思 路

定义数组IA（20）置放20个数，数组IC（4）存放四个相邻数。利用三重循环可以取到任意相邻四个数之和，N和M分别存放大的和及小的和。最后打印输出N和M。

2）框图　见图77

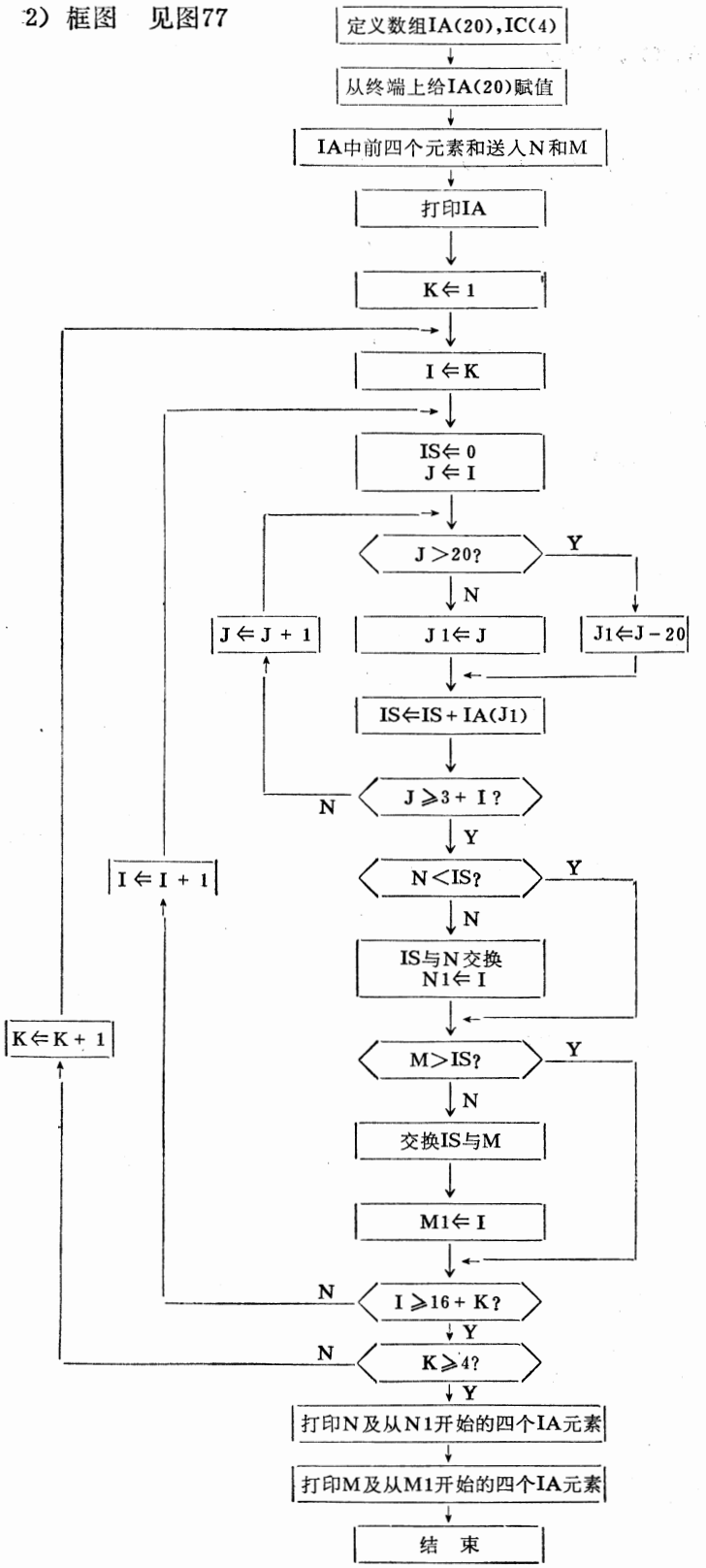


图 77

IA存放20个任意整数。
IC为打印时，存放四个所求连续整数。

N和M分别存放最小和数及最大和数，开始时暂存放前四个数之和

K、I、J为三重循环的控制变量，这三重循环保证可以取到任意相连的四个数之和。

IS存放这样的和

处理J超过20的情况

把比N小的IS放到N中

把比M大的IS放到M中

3）源程序及运行结果

```
      DIMENSION  IA(20) , IC(4)
      DO  20  I=1, 20
      WRITE (5, 10) I
10    FORMAT (2X, 3HIA (, I2, 2H) =)
20    READ (5, 15) IA(I)
15    FORMAT (I2)
      N=0
      DO  30  I=1, 4
30    N=N+IA(I)
      M=N
      WRITE (5, 35) IA
35    FORMAT (20 (1X, I2) )
      DO  200  K=1, 4
      II=16+K
      DO  190  I=K, II, 4
      IS=0
      DO  45  J=I, 3+I
      IF (J. GT. 20) GO  TO  40
      J1=J
      GO  TO  45
40    J1=J-20
45    IS=IS+IA(J1)
      IF (N. LT. IS) GO  TO  105
      IB=IS
      IS=N
      N=IB
      N1=I
105   IF (M. GT. IS) GO  TO  190
      IB=IS
      IS=M
      M=IB
      M1=I
190   CONTINUE
200   CONTINUE
      N4=N1+3
      JJ=0
      DO  300  ND=N1, N4
      NP=ND
      JJ=JJ+1
      IF (ND. LE. 20) GO  TO  300
      NP=20-NP
300   IC (JJ) =IA(NP)
```

```
      PRINT*, 'MIN:', N
      WRITE (5, 400) IC
400   FORMAT (4 (3X, I3) )
      M4 = M1 + 3
      JJ = 0
      DO  350  MD = M1, M4
      JJ = JJ + 1
      NP = MD
      IF (MD. LE. 20) GO  TO  350
      NP = 20 - MD
350   IC(JJ) = IA(NP)
      PRINT*, 'MAX:', M
      WRITE (5, 400) IC
      STOP
      END
```

```
 1  18  4  13  6  10  15  2  17  3  19  7  16  8  11  14  9  12  5  20
      MIN:              33
              4         13        6         10
      MAX:              50
             19         7         16        8
```

（三）思 考 题

已知由20项所组成的等差数列的偶数项之和等于250，奇数项之和等于220，试确定此数列的中间两项。

78 排 队 买 票

小强跟父亲排队买球票，时间一久，便不耐烦了，他父亲便给他出了道题让他去算。父亲说：“小强，你去从排头按一、二、三、四…的顺序数下来，第一个人为第1号，第二个人为第2号，依次类推，算出我们前面所有人的号数总和是多少？”

小强高兴了，带着铅笔和纸忙碌起来，数了一阵回到父亲身边报告说：“前面的孩子数为成年人数的三分之一，有趣的是成年人的号数总和也刚好是孩子号数之和的三倍。”

小强以为给父亲出了个难题，而父亲却胸有成竹地问：“那么，前面所有人的号数总和是多少呢？”

小强故意为难父亲说：“在800至1000之间”。他父亲却高兴地笑了笑，立即说出了前面的人数，成人与孩子各有多少，以及所有号数的总和数。

请你也想一想，得出什么结果？

（一）笔算步骤和结果

设孩子数为X，则成年人数为3X，总人数为4X，他们的号数分别为1、2、3、4、5…4X这些号数的总和为

$$1+2+3+\cdots+4X=\frac{4X(4X+1)}{2}=2X(4X+1)$$

根据小强的话知　　$799<2X(4X+1)<1001$

化简，解不等式得　　$8<X<12$

因X为整数，且成年人号数总和等于孩子号数总和的三倍，可知所有人号数的总和必为4的倍数。又因，不论X为何值时，$4X+1$ 总为奇数。因此，若要求$2X(4X+1)$为4的倍数，X必为偶数，从上不等式得　　$X=10$

于是　　$2X(4X+1)=20(40+1)=820$

由此得出他们前面有40人，其中孩子10人，成年人30人，他们的号数总和为820。

（二）程 序 设 计

1）设计思路

设总人数N上限为250，用循环控制变量N由1～250来找正确人数，每循环一次，累加总人数排队号码，判人数与号码符合否，若符合，打印出总人数和号码（不计算大人与孩子的明显比例数）。若不符合，再继续更换循环变量N。直到符合条件为止。否则，打印出错信息。

2）框图　见图78

3）源程序及运行结果

```
      JS=0
      DO  10  N=1, 250                    总人数N
      JS=JS+N                             计数JS
      IF (JS. LT. 800) GO  TO  10
      N1=N                                总人数下限N1
      JS1=JS                              总人数下限号码
      GO  TO  15
10    CONTINUE
      STOP · ERR1 ·
15    JS2=JS1                  15句从下限人数的号码继续计数
      M1=N1+1
      DO  20  M=M1, 250
      JS2=JS2+M                           JS2为上限号码
      GO  TO  25
20    CONTINUE
25    IF (N1. EQ. N1/4*4. AND. JS1. EQ. JS1/4*4) GO  TO  40
                                          判下限人数与号码符合否
      N1=N1+1
      DO  30  NX=N1, N2                   在N1、N2上下限间找
      JS1=JS1+NX
      IF (NX. EQ. NX/4*4. AND. JS1. EQ. JS1/4*4) GO  TO  60
                                          判找到总人数否
30    CONTINUE
      STOP · ERR2 ·
```

```
40        IF (JS1. GT. 1000) STOP ·ERR3·
          WRITE (6, 50) N1, JS1
50        FORMAT (5X, 4HMEN=, I2, 5X, 3HNO=, I3/)
          STOP
60        IF (JS1. GT. 1000) STOP ·ERR3·
          WRITE (6, 50) NX, JS1
          STOP
          END
```

打印结果

MEN = 40　　NO = 820　　总人数40，　　总号码820。

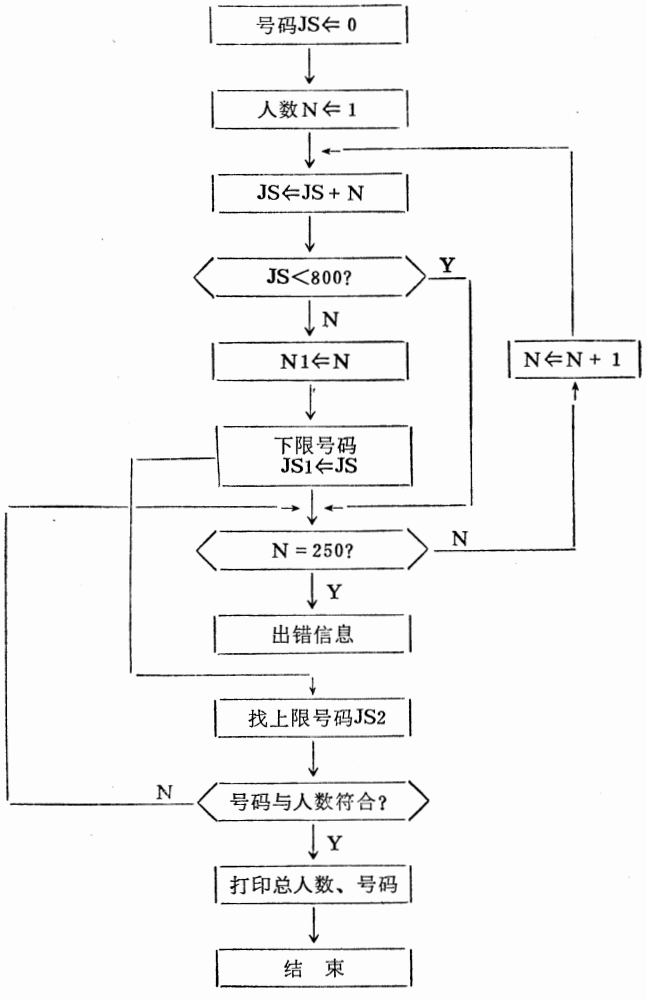


图 78

（三）思　考　题

五个男生从单号入座，彼此挨着，座号加起来是85；五个女生从双号入座，也彼此挨着，座号加起来是80，请推算这十人是坐在哪些座位号上？

79 乐 队 人 数

美国编写趣味难题的萨姆·洛伊德想出了下面题：在爱尔兰守神节那天，举行每年一度的庆祝游戏，指挥者若将乐队排成10人、9人、8人、7人、6人、5人、4人、3人和2人时，最后的一排总是缺少一个人，那些人想这个位置大概是给数月前死去的乐队成员凯西还留着位置。指挥者见到总缺1人恼火了，叫大家排成一列纵队前进。

假定人数不超过7000人，那么乐队究竟有多少人？

（一）笔算步骤和结果

该题是求最小公倍数M，$M = 2^3 \times 3^2 \times 5 \times 7 = 2520$(人)，这是假若按乐队原指挥的人数排队不缺1人时的总人数。那样实际排法总缺1人，故乐队实际总人数为2520－1＝2519（人）。

（二）程 序 设 计

1）设计思路

本趣味题所给排队人数是有规律的，可不用求最小公倍数的方法，用一个循环语句便可。控制变量I由2到10，是假设每行排2～10人时，都能排成整齐的队形（不缺人），若判断N不能被I整除时，则N⇐N＋1，出了循环体时，N是能被I整除的人数，再减1，便是乐队实际人数，令输出打印N值。

2）框图　见图79

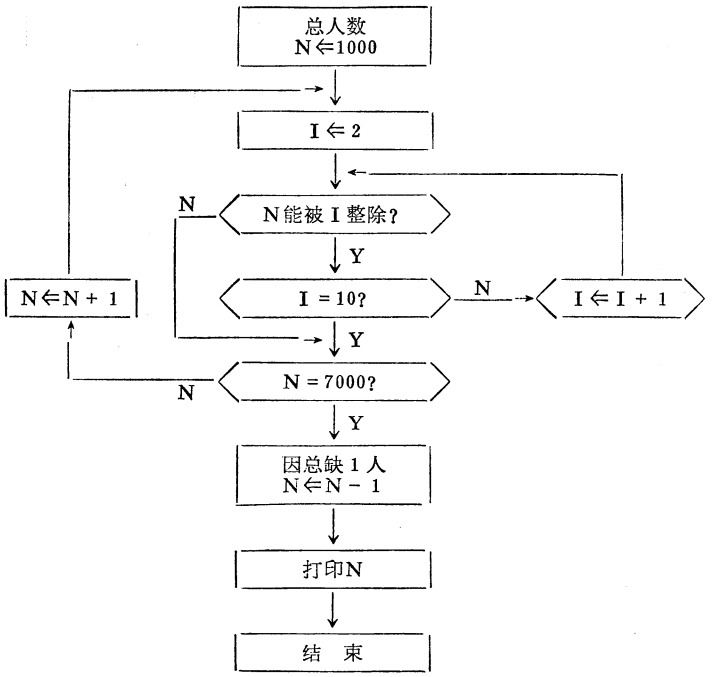


图 79

3）源程序及运行结果

```
DO  22  N = 1000, 7000
DO  20  I = 2, 10
```

```
      IF (N. NE. N/I*I) GOTO  22
20    CONTINUE
      GOTO  25
22    CONTINUE
25    N=N-1
      WRITE (10, 30) N
30    FORMAT (10X, 2HN=, I4)
      STOP
      END
```

```
N=2519
```

（三）思　考　题

公元七世纪印度算术书有道经典趣味难题：某女士手里拎了一篮鸡蛋，从她身边奔跑而过一匹惊马，吓了她一跳，结果把篮里的鸡蛋全打碎了。有人问她篮里原有多少个鸡蛋，她说两个一数，三个一数，四个一数，五个一数时，余数分别为1、2、3和4。问篮里原有多少个鸡蛋？

十一　高斯八皇后92种解

80　行列各种四棵树

学校订购了三十六棵树苗，学生们按预订计划先挖掘36个坑，行列整齐的构成正方形。但运输时不慎，损坏了十二棵。学生问老师：这怎么种？老师烦燥地说：每行每列保持四棵就可以，你们试着种吧！

（一）算 法 分 析

已挖掘36个坑，构成了整齐的正方形，那必是每行每列各 6 个坑，如果全种上树苗如图80-1所示。图中用 1 表示树苗，共36个 1 。但损坏的12棵不能用，若按老师意见去种，则每行或者每列要去掉两棵，去掉后 1 变为 0 ，如图80-2所示，便保证了每行、每列各有四个 1 。此题有多种解法，我们仅举其中之一作为例子，如图80-2中画＊号的 0 指定要删去 1 变为 0 ，一般地说，对任意指定位于 a(i, j) 的某个 1 ，还要删除另外 11个 1 ，其取法为：

a(i, j),	a(i+1, j)
a(i+1, j+1),	a(i+2, j+1)
a(i+2, j+2),	a(i+3, j+2)
a(i+3, j+3),	a(i+4, j+3)
a(i+4, j+4),	a(i+5, j+4)
a(i+5, j+5),	a(i+6, j+5)

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

图 80-1

1	0	0	1	1	1
1	1	0	0*	1	1
1	1	1	0	0	1
1	1	1	1	0	0
0	1	1	1	1	0
0	0	1	1	1	1

图 80-2

其中的加法为模 6 加。

（二）程 序 设 计

1）设计思路

36个 1 放在6×6的二维数组 A 中。按算法分析决定应删除的 1 的位置，相应地添入0，最后输出数组 A 即为结果。

2）框图　见图80-3

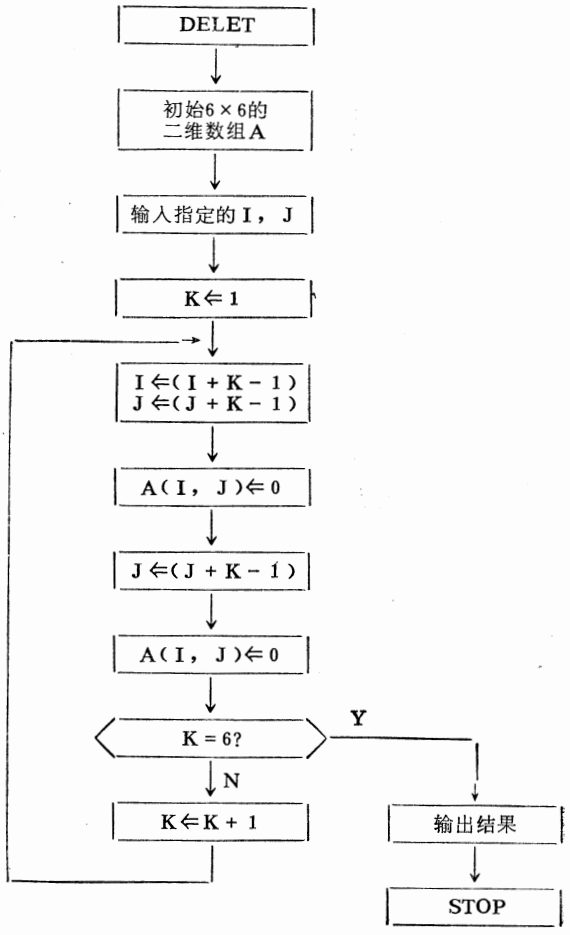


图 80-3

3）源程序及运行结果

```
C
C       THIS  PROGRAM  DELETS  TWELVE  1'S
C
        INTEGER  I, J, K, II, JJ
        INTEGER  A(6, 6)
        DO  5  I=1, 6                          } 把6×6的二维数组A
        DO  5  J=1, 6                          } 的各元素置1
5       A(I, J)=1
        OPEN  (3, FILE='D5', ACCESS='WRITE')
C
C       READ  INPUTS
        WRITE (1, 10)
```

```
10      FORMAT (3X, 33HTYPE THE POSITION
        OF 1 YOU WANTED/)                          请求输入指定删除的1的位置
        READ (1, 20) I
        READ (1, 20) J
20      FORMAT (I1)
        WRITE (3, 22) I
22      FORMAT (36H THE LINE OF ELEMENT YOU DELETED IS, I2)
        WRITE (3, 24) J
24      FORMAT (37H THE COLU OF ELEMENT YOU DELETED IS, I2)
C
C       II = (I+K-1)MOD6
C
        DO 60 K=1, 6
        IF ((I+K-1). GT. 6) GO TO 30
        II=I+K-1                                   (I+K-1) 模6
        GO TO 40
30      II=I+K-7
C
C       JJ = (J+K-1)MOD6
C
40      IF ((J+K-1). GT. 6) GO TO 45
        JJ=J+K-1                                   (J+K-1) 模6
        GO TO 50
45      JJ=J+K-7
50      A (II, JJ) = 0                             在得出的位置上添0
C
C       JJ = (J+K)MOD6
C
        IF ((J+K). GT. 6) GO TO 55
        JJ=J+K                                     (J+K) 模6
        GOTO 60
55      JJ=J+K-6
60      A (II, JJ) = 0                             在得出的位置上添0
        WRITE (3, 80)
80      FORMAT (20H THIS IS THE RESULT)            输出结果
        WRITE (3, 70) ((A(I, J), J=1, 6), I=1, 6)
        CLOSE(-1)
        STOP
        END
```

```
THE LINE OF ELEMENT YOU DELETED IS 3 指定删除元素的行为3
THE COLU OF ELEMENT YOU DELETED IS 5 指定删除元素的列为5
THIS IS THE RESULT
```

```
1  1  0  0  1  1
1  1  1  0  0  1
1  1  1  1  0  0                    结果
0  1  1  1  1  0
0  0  1  1  1  1
1  0  0  1  1  1
```

（三）思 考 题

本题有多种解法，我们仅举其中之一作为例子，已打印输出正确结果。请你找出共有几种解法，并将每种解法打印出。

81 高斯八皇后92种解

八皇后是个古老而有趣的游戏，是由高斯于1850年首先提出的。要求在国际象棋的棋盘上放置八个皇后，使其不能相互攻击，即任意二个皇后不能处于棋盘上的同一行、同一列、同一条斜线上。如图81-1所示。试问共有多少种摆法？你能找出多少种摆法？

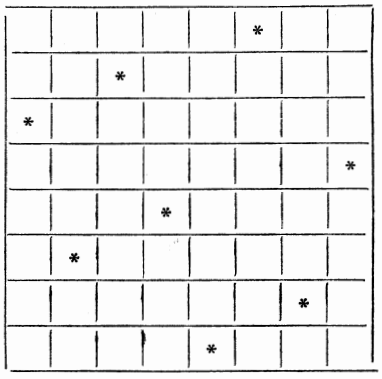


图 81-1

（一）算 法 分 析

本题虽由高斯提出，但他本人并未能完全解决，他仅求出约全部解的三分之二。这个问题的求解需要有大量的试验和计算，因此用手工计算是难以胜任的。

为了简单，我们取国际象棋的棋盘为8×8共64个格子。并把棋盘的格子按行、列划分好，用坐标区别。八个皇后顺序编号为1，2，…，8。由于每一行只能放一个皇后，为了叙述方便，把放在第一行上的称为一号皇后，放在第二行上的叫做二号皇后，顺序称以下诸皇后。

算法的基本思路是：首先把一号皇后放在（1，1）的位置上，然后把二号皇后放在（2，j_2）的位置上，并使其不能相互攻击。接下放三号皇后，找到一个满足条件的（3，j_3），把其放在该位置。依此顺序放下去。遇到某个皇后，假设六号皇后，无论把她放在该放的任何一个位置上均不满足条件，则重新布置其前面的一个皇后，即五号皇后，比如放在（5，j_5+1）的位置上，接下继续放六号皇后，如此试探下去，可找到满足条件的布局——全部92种解（详见打印结果）。

下面详细叙述算法：

（1）把1号皇后放在（1，j_1）的位置上。

（2）把二号皇后放在（2，j_2）的位置上。若与一号皇后之间不满足条件，则$j_2 \Leftarrow j_2+1$，返回（2）。若在$1 \leqslant j_2 \leqslant 8$的范围内不满足条件，则$j_2 \Leftarrow 1$，$j_1 \Leftarrow j_1+1$，返回算法(1)。

（3）设置L(3≤L≤8）号皇后于（L，j_L）位置上，并应使其满足条件。若与已放置的皇后冲突，则$j_L \Leftarrow j_L+1$，并返回（3）。若在$1 \leqslant j_L \leqslant 8$的范围内都不能满足条件，则$j_L \Leftarrow 1$，$L \Leftarrow L-1$，$j_L \Leftarrow j_L+1$，并返回（3）。若（L，$j_L$）位置满足条件，则把L号皇后放在该位置上，$L \Leftarrow L+1$，若$L \neq 9$则返回（3），若L＝9，则说明已找到一种摆法，继续做（4）。

若进行L⇐L－1后有L＝2，则返回算法（2）。

（4）此时已找到一种摆法，记下该种摆法后，试找下一种摆法。于是L⇐L－1，返回算法（3）。

本算法的关键是（3），是递归查找过程，下面举例说明。

当L号皇后放在（L，j_L）位置后，则要找L＋1号皇后的位置，设找到的位置为（L＋1，j_L＋1），接下找L＋2号皇后的位置，若找到满足条件的位置为（L＋2，j_L＋2），则放下L＋2号皇后。若L＋2号皇后在L＋2行没有满足条件的位置，则退回重新摆L＋1号皇后，接下继续摆L＋2号皇后。若无论L＋1号皇后在L＋1行的什么位置上，L＋2号皇后均没有满足条件的位置，则要退回重新摆L号皇后。一直可退到L＝1。

（二）程序设计

1）设计思路

由上述算法可见，这是一个递归查找过程。由于FORTRAN语言不允许递归，所以用循环语句解决，虽然增加了程序的长度，却锻练了技能。

程序设置一个8×8的数组DISC，用于表示棋盘，放置皇后的位置置1，其余为0，主要用于检查。当要把皇后放置在某个位置时，首先要检查是否与棋盘已放入的皇后冲突。RE为100×8的数组，每一行记录一种可能的摆法。COUNT为摆法的计数变量，每找到一种摆法，该变量加1。L1，L2，…，L8分别表示八个皇后在该行上的位置（即列号），程序中做为循环控制变量。一维数组LADY有八个元素，用于存放一次查找过程中所找到的位置，当八个位置都找到后，则把LADY送入DISC。

子程序CHECK用于检查给定的位置（I，J）是否与棋盘已放置的I－1个皇后冲突。要根据I、J的值，分别检查RE的I行、J列，以及（I，J）所在的二条斜线。

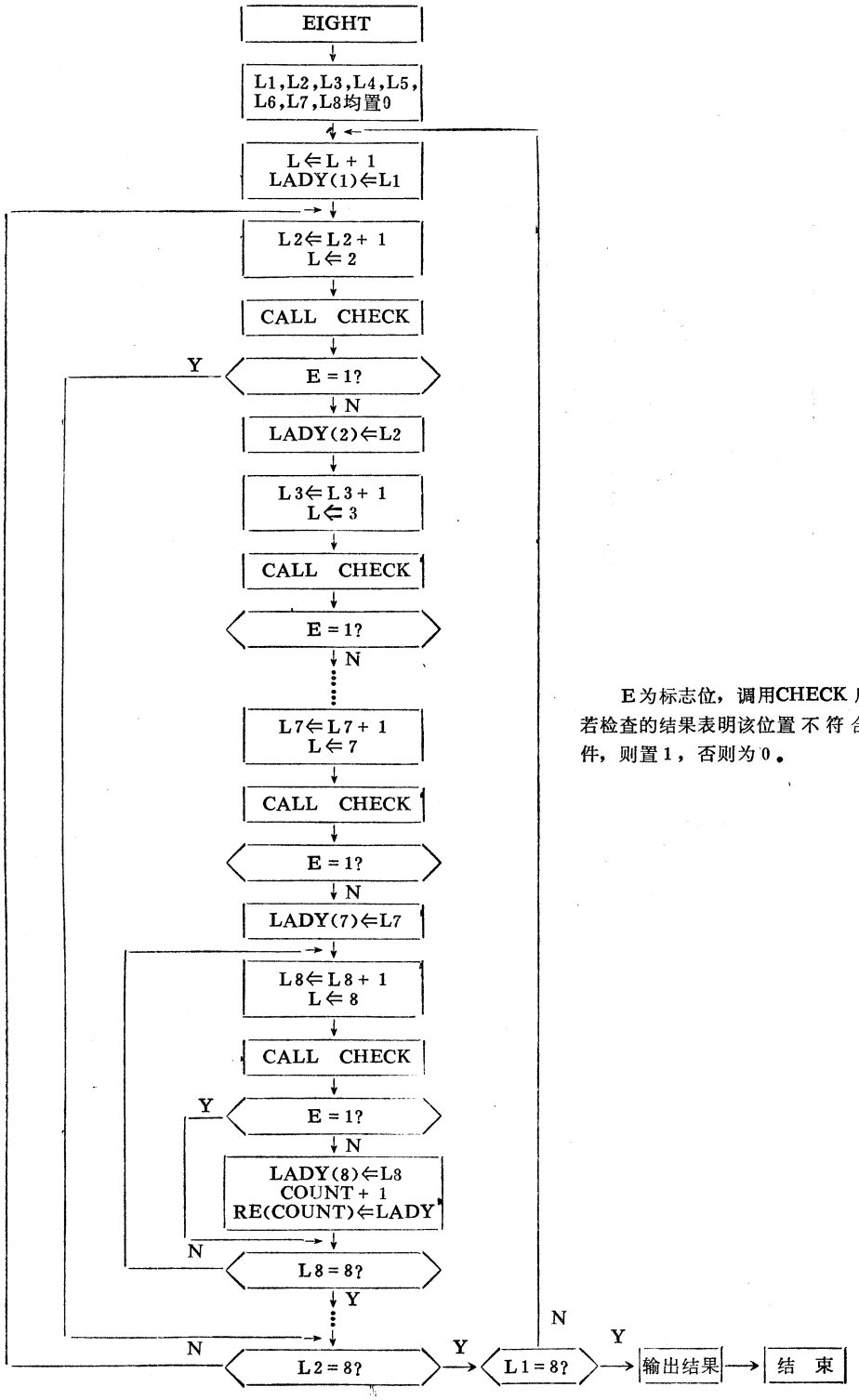
每一个皇后用了一个DO循环，一号皇后最先放置，因此是最外层的循环。八号皇后最后放置，因此是最内层循环。若八号皇后找到合适的位置，则把1～8个皇后的位置记下来。在某一层循环中无合适的位置，则退出一层循环，在其外层循环中重新查找。

当八层循环都做完后，则说明所有的可能布局都已找遍，于是将COUNT的计数值，输出全部可能的布局。

2）框图　见图81-2、图81-3。

3）源程序及运行结果

```
C***  PROGRAM  NAME: EIGHT          八皇后
C***  EIGHT  QUEENES
      INTEGER  L1, L2, L3, L4, L5, L6, L7, L8, L, COUNT, E
      INTEGER  DISC (8, 8), RE (100, 8), LADY(8)
      OPEN (4, FILE='EMP', ACCESS='W')
      COUNT=0
C***  CLEAR  THE  STACK  USED  BY  QUEENES
C***
      DO  4  L=1, 8
 4    LADY(L)=0
C***  CLEAR  ELEMENTS  OF  THE  DISC, I. E. CHESSBOARD
C***
```



E为标志位，调用CHECK 后，若检查的结果表明该位置不符合条件，则置1，否则为0。

图 81-2

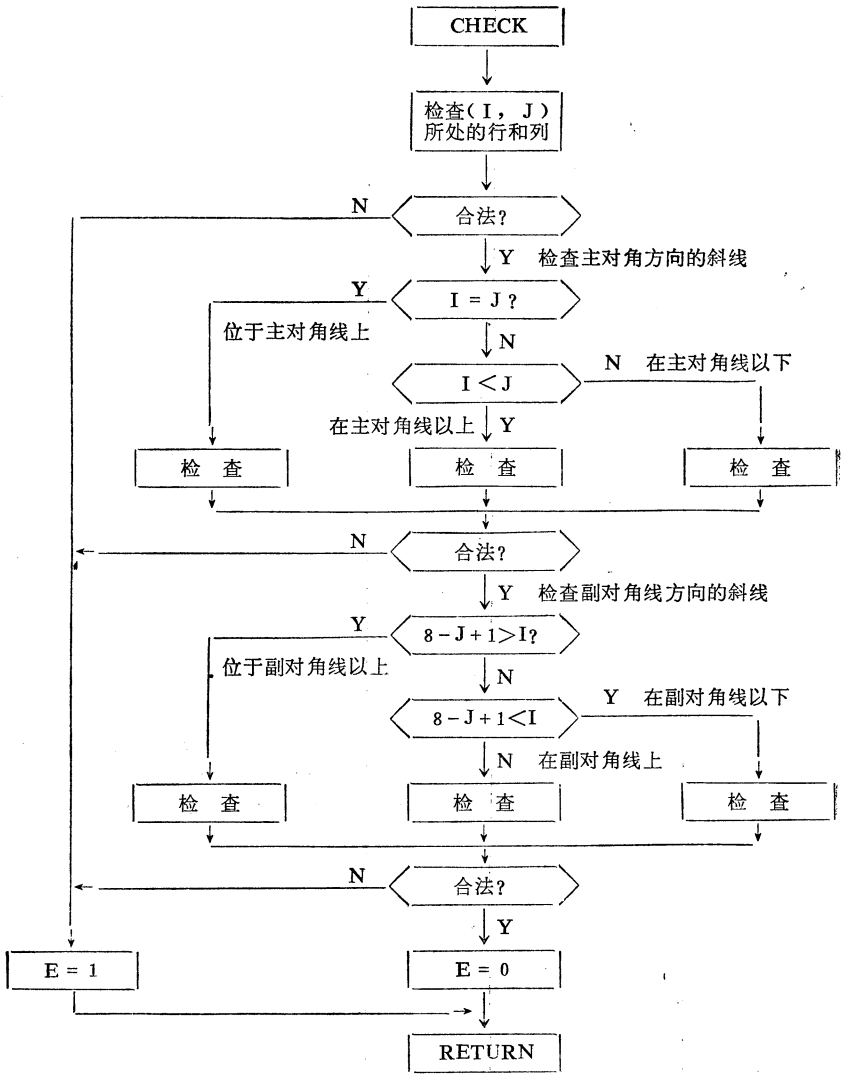


图 81-3 示意性框图

```
      DO  5  L=1, 8
      DO  5  L1=1, 8
5     DISC  (L, L1) = 0
C***  CLEAR  ELEMENTS  OF  THE  RE, I. E. RESULTS
C***
      DO  8  L=1, 100
      DO  8  L1=1, 8
8     RE(L, L1) = 0
C***  PLACE  THE  QUEEN  NO. 1  ON  THE  CHESSBOARD  放一号皇后
C***
      DO  10  L1=1, 8
      IF (L1. NE. 1)  DISC (1, L1-1) = 0
```

```
      L=1
      LADY(1)=L1
      DISC (1, L1)=1
C***  PLACE THE QUEEN NO. 2 ON THE CHESSBOARD 放二号皇后
C***
      DO 20 L2=1, 8
      IF (L2. NE. 1) DISC(2, L2-1)=0
      L=2
      CALL CHECK(DISC, L, L2, E)
      IF (E. EQ. 1) GO TO 20
      LADY(2)=L2
      DISC(2, L2)=1
C***  PLACE THE QUEEN NO. 3 ON THE CHESSBOARD 放三号皇后
C***
      DO 30 L3=1, 8
      IF (L3. NE. 1) DISC(3, L3-1)=0
      L=3
      CALL CHECK(DISC, L, L3, E)
      IF (E. EQ. 1) GO TO 30
      LADY(3)=L3
      DISC(3, L3)=1
C***  PLACE THE QUEEN NO. 4 ON THE CHESSBOARD 放四号皇后
C***

      DO 40 L4=1, 8
      IF (L4. NE. 1) DISC(4, L4-1)=0
      L=4
      CALL CHECK (DISC, L, L4, E)
      IF (E. EQ. 1) GO TO 40
      LADY(4)=L4
      DISC (4, L4)=1
C***  PLACE THE QUEEN NO. 5 ON THE CHESSBOARD 放五号皇后
C***
      DO 50 L5=1, 8
      IF (L5. NE. 1) DISC(5, L5-1)=0
      L=5
      CALL CHECK(DISC, L, L5, E)
      IF (E. EQ. 1) GO TO 50
      LADY(5)=L5
      DISC(5, L5)=1
C***  PLACE THE QUEEN NO. 6 ON THE CHESSBOARD 放六号皇后
C***
      DO 60 L6=1, 8
```

```
      IF (L6. NE. 1) DISC(6, L6-1) = 0
      L = 6
      CALL CHECK(DISC, L, L6, E)
      IF (E. EQ. 1) GO TO 60
      LADY(6) = L6
      DISC(6, L6) = 1
C***  PLACE THE QUEEN NO. 7 ON THE CHESSBOARD 放七号皇后
C***
      DO 70 L7 = 1, 8
      IF (L7. NE. 1) DISC(7, L7-1) = 0
      L = 7
      CALL CHECK(DISC, L, L7, E)
      IF (E. EQ. 1) GO TO 70
      LADY(7) = L7
      DISC(7, L7) = 1
C***  PLACE THE QUEEN NO. 8 ON THE CHESSBOARD 放八号皇后
C***
      DO 80 L8 = 1, 8
      IF (L8. NE. 1) DISC(8, L8-1) = 0
      L = 8
      CALL CHECK(DISC, L, L8, E)
      IF (E. EQ. 1) GO TO 80
      LADY(8) = L8
      DISC(8, L8) = 1
      COUNT = COUNT + 1                    摆法计数器加 1
      DO 90 I = 1, 8                       把本次找到的摆法
90    RE(COUNT, I) = LADY(I)               保存在RE中
      DISC(8, L8) = 0
80    CONTINUE
      DISC(8, LADY(8)) = 0
70    CONTINUE
      DISC(7, LADY(7)) = 0
60    CONTINUE
      DISC(6, LADY(6)) = 0
50    CONTINUE
      DISC(5, LADY(5)) = 0
40    CONTINUE
      DISC(4, LADY(4)) = 0
30    CONTINUE
      DISC(3, LADY(3)) = 0
20    CONTINUE
      DISC(2, LADY(2)) = 0
10    CONTINUE                             输出
```

```
      WRITE(1, 110)
      WRITE(4, 110)
110   FORMAT (5X, 'FOLLOWING ARE THE RESULTS THIS PROGRAM
     FOUND', /)
      WRITE(1, 120)
      WRITE(4, 120)
120   FORMAT (5X, 'NOTE : 1 : EACH LINE REPRESETS A WAY THE
     QUEEN WERE PLACED ON THE CHESSBOARD AND')

      WRITE(1, 130)
      WRITE(4, 130)
130   FORMAT (13X, 'WAS NUMBERED FROM 1 TO 92 SEQUENCELY', ')
      WRITE(1, 140)
      WRITE(4, 140)
140   FORMAT (11X, '2 : EACH COLUMU REPRESENTS A QUEEN, THE
     NUMBERS IN THE (··) REPRESENT THE POSITION')

      WRITE(1, 150)
      WRITE(4, 150)
150   FORMAT(13X,'THE QUEEN WAS PLACED ON THE CHESSBOARD'
     ////)
      WRITE(1, 160)
      WRITE(4, 160)
160   FORMAT (5X, 'NUMBER', 23X, 'THE POSITIOS THE QUEENES
     ON', //)
      DO 200 L=1, COUNT
      WRITE (4, 220) L, RE(L, 1), RE(L, 2), RE(L, 3), RE(L, 4), RE(L, 5),
     RE(L,  6), RE(L, 7), RE(L, 8)

200   WRITE (1, 220) L, RE(L, 1), RE(L, 2), RE(L, 3), RE(L, 4), RE (L,5),
     RE(L,  6), RE(L, 7), RE(L, 8)

220   FORMAT ( 5X, ' NO. ', I3, 5X, ' (1, ', I1, ')', 5X, ' (2, ', I1, ') ',
     5X, ' (3, ', I1,  ')',5X, '(4, 'I1, ') ', 5X, ' (5', ', I1, ')',
     5X, ' (6, ', I1, ')', 5X, '(7, ', I1, ')  ', 5X, '(8, ', I1, ')', /)

      CLOSE(-1)
      STOP
      END
C***      SUBROUTINE                    子程序CHECK
C***
      SUBROUTINE CHECK(CD, CL, CJ, CE)
      INTEGER T1, T2, T3, CL, CJ, CE
```

```
      INTEGER  CD(8, 8)
C***  CHECK  THE  LINE  CL  AND  CJ  COLUMU
      CE = 0
      DO  1110  T1 = 1, 8                          检查该位置所在的行、列
      IF  (CD(CL, T1). EQ. 0. AND. CD(T1, CJ). EQ. 0) GO  TO  1110
      CE = 1
      GO  TO  1280
1110  CONTINUE
      IF (CL. EQ. CJ) GO  TO  1140
      IF (CL. LT. CJ) GO  TO  1160

C***  BELOW  THE  MAIN  DIAGONAL                   该位置在主对角线以下
C***  T2 = BEGINNING  LINE, T3 = LAST  COLUMU
C***
      T2 = CL - CJ + 1
      T3 = 8 - T2 + 1
      DO  1130  T1 = 1, T3
      IF (CD(T2 + T1 - 1, T1). EQ. 0) GO  TO  1130
      CE = 1
      GO  TO  1280
1130  CONTINUE
      GO  TO  1200

C***  ON  THE  MAIN  DIAGONAL                      该位置在主对角线上
C***

1140  DO  1150  I = 1, 8
      IF (CD(I, I). EQ. 0) GO  TO  1150
      CE = 1
      GO  TO  1280
1150  CONTINUE
      GO  TO  1200

C***  OVER  THE  MAIN  DIAGONAL                    该位置在主对角线以上
C***  T2 = BEGINNING  COLUME, T3 = LAST  LINE
C***
1160  T2 = CJ - CL + 1
      T3 = 8 - T2 + 1
      DO  1170  T1 = 1, T3
      IF (CD(T1, T2 + T1 - 1). EQ. 0) GO  TO  1170
      CE = 1
      GO  TO  1280
1170  CONTINUE
```

```
C***  CHECK IF ON OR BELOW OR OVER THE SECOND DIAGONAL
                检查副对角线方向
  C***
1200      IF (8-CJ+1. GT. CL) GO TO 1230
          IF (8-CJ+1. LT. CL) GO TO 1260
  C***  ON THE SECOND DIAGONAL
          DO 1210 T1=1, 8                          该位置在副对角线上
          IF (CD(T1, 8-T1+1). EQ. 0) GO TO 1210
          CE=1
          GO TO 1280
1210      CONTINUE
          GO TO 1280
  C***  OVER THE SECOND DIAGONAL          该位置在副对角线以上
  C***  T2=LAST LINE AND COLUMU
  C***
1230      T2=CL+CJ-1
          DO 1250 T1=1, T2
          IF (CD(T1, T2-T1+1). EQ. 0) GO TO 1250
          CE=1
          GO TO 1280
1250      CONTINUE
          GO TO 1280
  C***  BELOW THE SECOND DIAGONAL
  C***  T2=BEGINNING LINE AND COLUMU          该位置在副对角线以下
  C***
1260      T2=CL+CJ-8
          DO 1270 T1=1, 8-T2+1
          IF (CD(T2+T1-1, 8-T1+1). EQ. 0) GO TO 1270
          CE=1
          GO TO 1280
1270      CONTINUE
1280      RETURN
          END
```

FOLLOWING ARE THE RESULTS THIS PROGRAM FOUND （本程序所找到的各种摆法如下）

NOTE： 1：EACH LINE REPRESENTS A WAY THE QUEEN WERE PLACED ON THE CHESSBOARD AND WAS NUMBERED FROM 1 TO 92 SEQUENCELY（每一行为一种摆法，各行顺序从1—92编号）

2：EACH COLUMU REPRESENTS A QUEEN, THE NUMBERS IN THE(••) REPRESENT THE POSITION THE QUEEN WAS PLACED ON THE CHESSBOARD（每一列表示一个皇后，括号中的数字表示该皇后在棋盘上的位置）

NUMBER THE POSITIOS THE QUEENES ON

NO.	1	(1, 1)	(2, 5)	(3, 8)	(4, 6)	(5, 3)	(6, 7)	(7, 2)	(8, 4)
NO.	2	(1, 1)	(2, 6)	(3, 8)	(4, 3)	(5, 7)	(6, 4)	(7, 2)	(8, 5)
NO.	3	(1, 1)	(2, 7)	(3, 4)	(4, 6)	(5, 8)	(6, 2)	(7, 5)	(8, 3)
NO.	4	(1, 1)	(2, 7)	(3, 5)	(4, 8)	(5, 2)	(6, 4)	(7, 6)	(8, 3)
NO.	5	(1, 2)	(2, 4)	(3, 6)	(4, 8)	(5, 3)	(6, 1)	(7, 7)	(8, 5)
NO.	6	(1, 2)	(2, 5)	(3, 7)	(4, 1)	(5, 3)	(6, 8)	(7, 6)	(8, 4)
NO.	7	(1, 2)	(2, 5)	(3, 7)	(4, 4)	(5, 1)	(6, 8)	(7, 6)	(8, 3)
NO.	8	(1, 2)	(2, 6)	(3, 1)	(4, 7)	(5, 4)	(6, 8)	(7, 3)	(8, 5)
NO.	9	(1, 2)	(2, 6)	(3, 8)	(4, 3)	(5, 1)	(6, 4)	(7, 7)	(8, 5)
NO.	10	(1, 2)	(2, 7)	(3, 3)	(4, 6)	(5, 8)	(6, 5)	(7, 1)	(8, 4)
NO.	11	(1, 2)	(2, 7)	(3, 5)	(4, 8)	(5, 1)	(6, 4)	(7, 6)	(8, 3)
NO.	12	(1, 2)	(2, 8)	(3, 6)	(4, 1)	(5, 3)	(6, 5)	(7, 7)	(8, 4)
NO.	13	(1, 3)	(2, 1)	(3, 7)	(4, 5)	(5, 8)	(6, 2)	(7, 4)	(8, 6)
NO.	14	(1, 3)	(2, 5)	(3, 2)	(4, 8)	(5, 1)	(6, 7)	(7, 4)	(8, 6)
NO.	15	(1, 3)	(2, 5)	(3, 2)	(4, 8)	(5, 6)	(6, 4)	(7, 7)	(8, 1)
NO.	16	(1, 3)	(2, 5)	(3, 7)	(4, 1)	(5, 4)	(6, 2)	(7, 8)	(8, 6)
NO.	17	(1, 3)	(2, 5)	(3, 8)	(4, 4)	(5, 1)	(6, 7)	(7, 2)	(8, 6)
NO.	18	(1, 3)	(2, 6)	(3, 2)	(4, 5)	(5, 8)	(6, 1)	(7, 7)	(8, 4)
NO.	19	(1, 3)	(2, 6)	(3, 2)	(4, 7)	(5, 1)	(6, 4)	(7, 8)	(8, 5)
NO.	20	(1, 3)	(2, 6)	(3, 2)	(4, 7)	(5, 5)	(6, 1)	(7, 8)	(8, 4)
NO.	21	(1, 3)	(2, 6)	(3, 4)	(4, 1)	(5, 8)	(6, 5)	(7, 7)	(8, 2)
NO.	22	(1, 3)	(2, 6)	(3, 4)	(4, 2)	(5, 8)	(6, 5)	(7, 7)	(8, 1)
NO.	23	(1, 3)	(2, 6)	(3, 8)	(4, 1)	(5, 4)	(6, 7)	(7, 5)	(8, 2)
NO.	24	(1, 3)	(2, 6)	(3, 8)	(4, 1)	(5, 5)	(6, 7)	(7, 2)	(8, 4)
NO.	25	(1, 3)	(2, 6)	(3, 8)	(4, 2)	(5, 4)	(6, 1)	(7, 7)	(8, 5)
NO.	26	(1, 3)	(2, 7)	(3, 2)	(4, 8)	(5, 5)	(6, 1)	(7, 4)	(8, 6)
NO.	27	(1, 3)	(2, 7)	(3, 2)	(4, 8)	(5, 6)	(6, 4)	(7, 1)	(8, 5)
NO.	28	(1, 3)	(2, 8)	(3, 4)	(4, 7)	(5, 1)	(6, 6)	(7, 2)	(8, 5)
NO.	29	(1, 4)	(2, 1)	(3, 5)	(4, 8)	(5, 2)	(6, 7)	(7, 3)	(8, 6)
NO.	30	(1, 4)	(2, 1)	(3, 5)	(4, 8)	(5, 6)	(6, 3)	(7, 7)	(8, 2)
NO.	31	(1, 4)	(2, 2)	(3, 5)	(4, 8)	(5, 6)	(6, 1)	(7, 3)	(8, 7)
NO.	32	(1, 4)	(2, 2)	(3, 7)	(4, 3)	(5, 6)	(6, 8)	(7, 1)	(8, 5)
NO.	33	(1, 4)	(2, 2)	(3, 7)	(4, 3)	(5, 6)	(6, 8)	(7, 5)	(8, 1)
NO.	34	(1, 4)	(2, 2)	(3, 7)	(4, 5)	(5, 1)	(6, 8)	(7, 6)	(8, 3)
NO.	35	(1, 4)	(2, 2)	(3, 8)	(4, 5)	(5, 7)	(6, 1)	(7, 3)	(8, 6)
NO.	36	(1, 4)	(2, 2)	(3, 8)	(4, 6)	(5, 1)	(6, 3)	(7, 5)	(8, 7)
NO.	37	(1, 4)	(2, 6)	(3, 1)	(4, 5)	(5, 2)	(6, 8)	(7, 3)	(8, 7)
NO.	38	(1, 4)	(2, 6)	(3, 8)	(4, 2)	(5, 7)	(6, 1)	(7, 3)	(8, 5)
NO.	39	(1, 4)	(2, 6)	(3, 8)	(4, 3)	(5, 1)	(6, 7)	(7, 5)	(8, 2)
NO.	40	(1, 4)	(2, 7)	(3, 1)	(4, 8)	(5, 5)	(6, 2)	(7, 6)	(8, 3)
NO.	41	(1, 4)	(2, 7)	(3, 3)	(4, 8)	(5, 2)	(6, 5)	(7, 1)	(8, 6)
NO.	42	(1, 4)	(2, 7)	(3, 5)	(4, 2)	(5, 6)	(6, 1)	(7, 3)	(8, 8)
NO.	43	(1, 4)	(2, 7)	(3, 5)	(4, 3)	(5, 1)	(6, 6)	(7, 8)	(8, 2)

NO.	44	(1, 4)	(2, 8)	(3, 1)	(4, 3)	(5, 6)	(6, 2)	(7, 7)	(8, 5)
NO.	45	(1, 4)	(2, 8)	(3, 1)	(4, 5)	(5, 7)	(6, 2)	(7, 6)	(8, 3)
NO.	46	(1, 4)	(2, 8)	(3, 5)	(4, 3)	(5, 1)	(6, 7)	(7, 2)	(8, 6)
NO.	47	(1, 5)	(2, 1)	(3, 4)	(4, 6)	(5, 8)	(6, 2)	(7, 7)	(8, 3)
NO.	48	(1, 5)	(2, 1)	(3, 8)	(4, 4)	(5, 2)	(6, 7)	(7, 3)	(8, 6)
NO.	49	(1, 5)	(2, 1)	(3, 8)	(4, 6)	(5, 3)	(6, 7)	(7, 2)	(8, 4)
NO.	50	(1, 5)	(2, 2)	(3, 4)	(4, 6)	(5, 8)	(6, 3)	(7, 1)	(8, 7)
NO.	51	(1, 5)	(2, 2)	(3, 4)	(4, 7)	(5, 3)	(6, 8)	(7, 6)	(8, 1)
NO.	52	(1, 5)	(2, 2)	(3, 6)	(4, 1)	(5, 7)	(6, 4)	(7, 8)	(8, 3)
NO.	53	(1, 5)	(2, 2)	(3, 8)	(4, 1)	(5, 4)	(6, 7)	(7, 3)	(8, 6)
NO.	54	(1, 5)	(2, 3)	(3, 1)	(4, 6)	(5, 8)	(6, 2)	(7, 4)	(8, 7)
NO.	55	(1, 5)	(2, 3)	(3, 1)	(4, 7)	(5, 2)	(6, 8)	(7, 6)	(8, 4)
NO.	56	(1, 5)	(2, 3)	(3, 8)	(4, 4)	(5, 7)	(6, 1)	(7, 6)	(8, 2)
NO.	57	(1, 5)	(2, 7)	(3, 1)	(4, 3)	(5, 8)	(6, 6)	(7, 4)	(8, 2)
NO.	58	(1, 5)	(2, 7)	(3, 1)	(4, 4)	(5, 2)	(6, 8)	(7, 6)	(8, 3)
NO.	59	(1, 5)	(2, 7)	(3, 2)	(4, 4)	(5, 8)	(6, 1)	(7, 3)	(8, 6)
NO.	60	(1, 5)	(2, 7)	(3, 2)	(4, 6)	(5, 3)	(6, 1)	(7, 4)	(8, 8)
NO.	61	(1, 5)	(2, 7)	(3, 2)	(4, 6)	(5, 3)	(6, 1)	(7, 8)	(8, 4)
NO.	62	(1, 5)	(2, 7)	(3, 4)	(4, 1)	(5, 3)	(6, 8)	(7, 6)	(8, 2)
NO.	63	(1, 5)	(2, 8)	(3, 4)	(4, 1)	(5, 3)	(6, 6)	(7, 2)	(8, 7)
NO.	64	(1, 5)	(2, 8)	(3, 4)	(4, 1)	(5, 7)	(6, 2)	(7, 6)	(8, 3)
NO.	65	(1, 6)	(2, 1)	(3, 5)	(4, 2)	(5, 8)	(6, 3)	(7, 7)	(8, 4)
NO.	66	(1, 6)	(2, 2)	(3, 7)	(4, 1)	(5, 3)	(6, 5)	(7, 8)	(8, 4)
NO.	67	(1, 6)	(2, 2)	(3, 7)	(4, 1)	(5, 4)	(6, 8)	(7, 5)	(8, 3)
NO.	68	(1, 6)	(2, 3)	(3, 1)	(4, 7)	(5, 5)	(6, 8)	(7, 2)	(8, 4)
NO.	69	(1, 6)	(2, 3)	(3, 1)	(4, 8)	(5, 4)	(6, 2)	(7, 7)	(8, 5)
NO.	70	(1, 6)	(2, 3)	(3, 1)	(4, 8)	(5, 5)	(6, 2)	(7, 4)	(8, 7)
NO.	71	(1, 6)	(2, 3)	(3, 5)	(4, 7)	(5, 1)	(6, 4)	(7, 2)	(8, 8)
NO.	72	(1, 6)	(2, 3)	(3, 5)	(4, 8)	(5, 1)	(6, 4)	(7, 2)	(8, 7)
NO.	73	(1, 6)	(2, 3)	(3, 7)	(4, 2)	(5, 4)	(6, 8)	(7, 1)	(8, 5)
NO.	74	(1, 6)	(2, 3)	(3, 7)	(4, 2)	(5, 8)	(6, 5)	(7, 1)	(8, 4)
NO.	75	(1, 6)	(2, 3)	(3, 7)	(4, 4)	(5, 1)	(6, 8)	(7, 2)	(8, 5)
NO.	76	(1, 6)	(2, 4)	(3, 1)	(4, 5)	(5, 8)	(6, 2)	(7, 7)	(8, 3)
NO.	77	(1, 6)	(2, 4)	(3, 2)	(4, 8)	(5, 5)	(6, 7)	(7, 1)	(8, 3)
NO.	78	(1, 6)	(2, 4)	(3, 7)	(4, 1)	(5, 3)	(6, 5)	(7, 2)	(8, 8)
NO.	79	(1, 6)	(2, 4)	(3, 7)	(4, 1)	(5, 8)	(6, 2)	(7, 5)	(8, 3)
NO.	80	(1, 6)	(2, 8)	(3, 2)	(4, 4)	(5, 1)	(6, 7)	(7, 5)	(8, 3)
NO.	81	(1, 7)	(2, 1)	(3, 3)	(4, 8)	(5, 6)	(6, 4)	(7, 2)	(8, 5)
NO.	82	(1, 7)	(2, 2)	(3, 4)	(4, 1)	(5, 8)	(6, 5)	(7, 3)	(8, 6)
NO.	83	(1, 7)	(2, 2)	(3, 6)	(4, 3)	(5, 1)	(6, 4)	(7, 8)	(8, 5)
NO.	84	(1, 7)	(2, 3)	(3, 1)	(4, 6)	(5, 8)	(6, 5)	(7, 2)	(8, 4)
NO.	85	(1, 7)	(2, 3)	(3, 8)	(4, 2)	(5, 5)	(6, 1)	(7, 6)	(8, 4)
NO.	86	(1, 7)	(2, 4)	(3, 2)	(4, 5)	(5, 8)	(6, 1)	(7, 3)	(8, 6)

NO.	87	(1, 7)	(2, 4)	(3, 2)	(4, 8)	(5, 6)	(6, 1)	(7, 3)	(8, 5)
NO.	88	(1, 7)	(2, 5)	(3, 3)	(4, 1)	(5, 6)	(6, 8)	(7, 2)	(8, 4)
NO.	89	(1, 8)	(2, 2)	(3, 4)	(4, 1)	(5, 7)	(6, 5)	(7, 3)	(8, 6)
NO.	90	(1, 8)	(2, 2)	(3, 5)	(4, 3)	(5, 1)	(6, 7)	(7, 4)	(8, 6)
NO.	91	(1, 8)	(2, 3)	(3, 1)	(4, 6)	(5, 2)	(6, 5)	(7, 7)	(8, 4)
NO.	92	(1, 8)	(2, 4)	(3, 1)	(4, 3)	(5, 6)	(6, 2)	(7, 7)	(8, 5)

以上是本程序运行后搜寻出八皇后92种布局

（三）思 考 题

1）若读者熟悉某种可以递归调用的语言，如何把本程序用递归的方法写得更为简洁一些。

2）若有10个皇后用10×10的棋盘如何摆法？

82 小虫奇妙地移动

徐工程师将画有9 × 9的网格草图挂在墙上，吸着香烟构思某设计方案。有九只小虫落在图上，恰巧每行、每列都有一只小虫，而且对角线上也不超过一只，如图82-1所示。

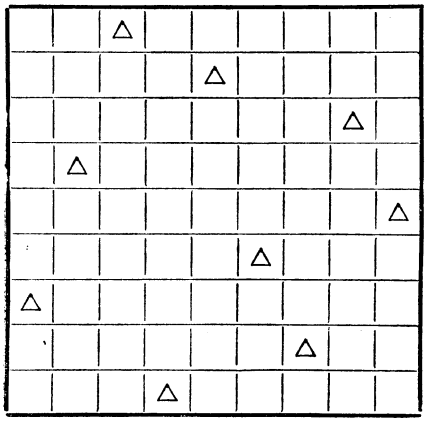


图 82-1

如此巧妙地分布，转移了他的思路，引起他观察的兴趣。突然有三只小虫同时向相邻的格子中移动，其余小虫虽然没动，但仍构成原状态那样的协调（每行每列以及对角线上仍有一只）。他在烟雾缭绕中，只看见了一只小虫移动了，但又没看清移动的方向。于是，他好奇地用笔画了画，微笑着自语说："噢，可能是这样移动的。"请读者考虑是怎样移动的？

（一）算 法 分 析

移动位置的小虫必定为相邻行和相邻列上的，否则会出现一行（列）上有多个小虫的情况，并出现空行（列）。

移动位置的三个小虫中，其中每只都有若干种可能的移动方向，若用 a_x、a_y、a_z 分别表示编号为x、y、z的三只小虫，则 $a_x(i, j) \rightarrow a_x(i+1, j+1)$ 表示处于i行、j列的小虫x移动到i + 1行、j + 1列。各种可能的移动方向为：

（1）若 $a_x(i, j) \longrightarrow a_x(i+1, j+1)$，其中 $i \neq 9$、$j \neq 9$，则有

$$a_y(i+1, k) \longrightarrow a_y(i, k)$$

$$a_z(l, j+1) \longrightarrow a_z(l, j)$$

（2）若 $a_x(i, j) \longrightarrow a_x(i+1, j-1)$，其中 $i \neq 9$，$j \neq 1$，则有

$$a_y(i+1, k) \longrightarrow a_y(i, k)$$

$$a_z(l, j-1) \longrightarrow a_z(l, j)$$

（3）若 $a_x(i, j) \longrightarrow a_x(i+1, j)$，其中 $i \neq 9$，则当 $k \neq 9$ 时有

$$a_y(i+1, k) \longrightarrow a_y(i, k+1)$$

$$a_z(l, k+1) \longrightarrow a_z(l, k)$$

而当 k ≠ 1 时有

$$a_y(i+1,\ k) \longrightarrow a_y(i,\ k-1)$$
$$a_z(l,\ k-1) \longrightarrow a_z(l,\ k)$$

可以看出，当 $1 < k < 9$ 时，上面二种移动方向都存在，以下同。

（4）若 $a_x(i,\ j) \longrightarrow a_x(i,\ j-1)$，其中 $j \neq 1$，则当 $l \neq 9$ 时有

$$a_y(l,\ j-1) \longrightarrow a_y(l+1,\ j)$$
$$a_z(l+1,\ k) \longrightarrow a_z(l,\ k)$$

而当 $l \neq 1$ 时有

$$a_y(l,\ j-1) \longrightarrow a_y(l-1,\ j)$$
$$a_z(l-1,\ k) \longrightarrow a_z(l,\ k)$$

（5）若 $a_x(i,\ j) \longrightarrow a_x(i,\ j+1)$，其中 $j \neq 9$，则当 $l \neq 1$ 时有

$$a_y(l,\ j+1) \longrightarrow a_y(l-1,\ j)$$
$$a_z(l-1,\ k) \longrightarrow a_z(l,\ k)$$

而当 $l \neq 9$ 时有

$$a_y(l,\ j+1) \longrightarrow a_y(l+1,\ j)$$
$$a_z(l+1,\ k) \longrightarrow a_z(l,\ k)$$

（6）若 $a_x(i,\ j) \longrightarrow a_x(i-1,\ j)$，其中 $i \neq 1$，则当 $k \neq 1$ 时有

$$a_y(i-1,\ k) \longrightarrow a_y(i,\ k-1)$$
$$a_z(l,\ k-1) \longrightarrow a_z(l,\ k)$$

而当 $k \neq 9$ 时有

$$a_y(i-1,\ k) \longrightarrow a_y(i,\ k+1)$$
$$a_z(l,\ k+1) \longrightarrow a_z(l,\ k)$$

（7）若 $a_x(i,\ j) \longrightarrow a_x(i-1,\ j-1)$，其中 $i \neq 1$，$j \neq 1$，则有

$$a_y(i-1,\ k) \longrightarrow a_y(i,\ k)$$
$$a_z(l,\ j-1) \longrightarrow a_z(l,\ j)$$

（8）若 $a_x(i,\ j) \longrightarrow a_x(i-1,\ j+1)$，其中 $i \neq 1$，$j \neq 9$ 则有

$$a_y(i-1,\ k) \longrightarrow a_y(i,\ k)$$
$$a_z(l,\ j+1) \longrightarrow a_z(l,\ j)$$

（二）程 序 设 计

1）设计思路

（1）由观察者给出的初始状态和一只移动位置的小虫（设编号为 x），则可找出另外二只小虫移动的方向。由于每行只能有一只小虫，因此用其行号作为小虫的编号，这样我们可以用 2 × 9 的数组 A 存放九只小虫的初始状态，其第一行的元素依次存放小虫的编号，第二行分别存放小虫所在的列。用十个只有二个元素的数组存放各小虫移动前和移动后的位置，例如，FROM1存放小虫 x 移动前的行和列，TO1存放小虫 x 移动后所位于的行和列。

（2）对任一只小虫（$1 \leqslant i \leqslant 9$，$1 \leqslant j \leqslant 9$），根据其 i、j 的值可做上述八种算法中几种，例如，当 $i = 1$，$j = 1$ 时，只能做上述算法中的（1）、（3）、（5）。全部列出如下：

1＜i＜9，1＜j＜9	1、2、3、4、5、6、7、8
i＝1，j＝9	2、3、4
i＝9，j＝1	5、6、8
i＝9，j＝9	4、6、7
i＝1，j＝1	1、3、5
i＝1，1＜j＜9	1、2、3、4、5
i＝9，1＜j＜9	4、5、6、7、8
1＜i＜9，j＝1	1、3、5、6、8
1＜i＜9，j＝9	2、3、4、6、7

为了找出各种可能的移动方向，要把上面的情况编成一个表，程序中为一个 9 × 9 的常数数组，叫做CM。该表内容如下：

1	2	3	4	5	6	7	8	8
2	3	4	0	0	0	0	0	3
5	6	7	0	0	0	0	0	3
4	6	8	0	0	0	0	0	3
1	3	5	0	0	0	0	0	3
1	2	3	4	5	0	0	0	5
4	5	6	7	8	0	0	0	5
1	3	5	6	8	0	0	0	5
2	3	4	6	7	0	0	0	5

表中第 9 列表示可执行的算法个数，其它列不为 0 的数表示可执行的算法编号。

（3）根据输入的已知小虫的 i 、 j 值，取出CM的第 9 列中的相应元素 做 为循环控制变量，然后分别取出该行的元素作为转移控制变量，转入相应算法执行部分，找出可能的路径并输出。各种可能的算法执行完后，则会找出全部可能的路径。

2）框图　见图82-2

3）源程序及运行结果

```
C       THIS PROGRAM SEARCH THE MOVING PATHES OF THREE INSECT,
          FROM POINTS
C       TO POINTS
C
        INTEGER I, J, K, L, INS, CASE, COUNT, JMP
        INTEGER A(2, 9), FROM1(2), TO1(2), FROM2(2), TO2(2),
       1  FROM3(2), TO3(2), TO22(2), FROM33(2), TO33(2), CM(9, 9)

C
C       READ IN INITIATE STATE, THE INSECT' S NO.,AND ITS DIREC-
          TION MOVED
C
```

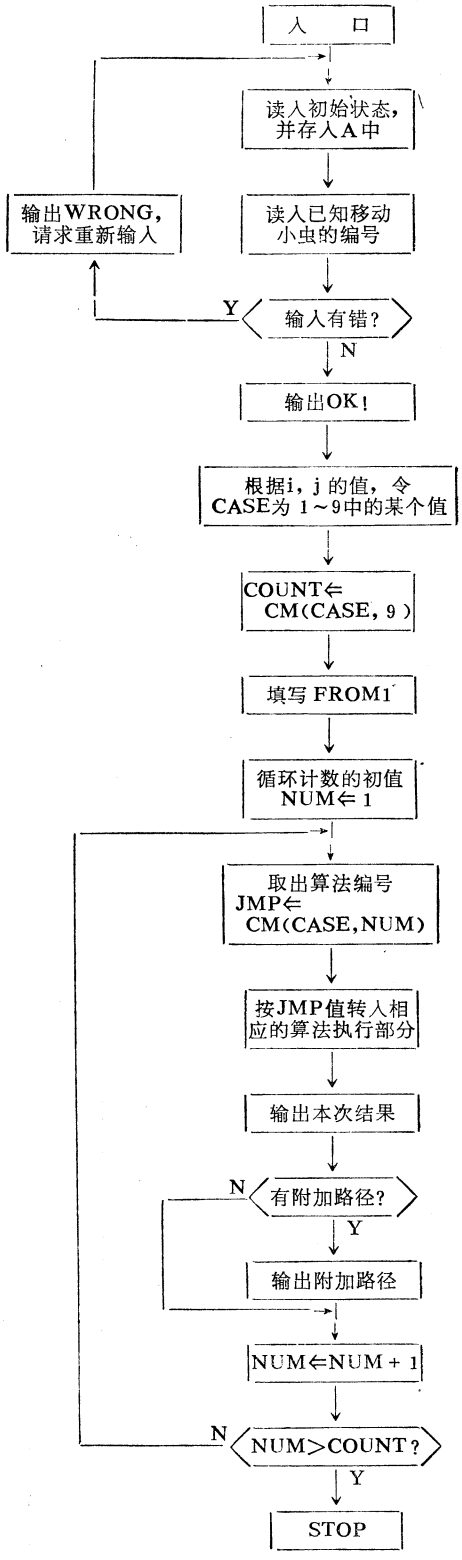

入　口
读入初始状态，并存入A中
读入已知移动小虫的编号
输入有错?
Y
N
输出WRONG，请求重新输入
输出OK!
根据i，j 的值，令CASE为 1～9中的某个值
COUNT⇐ CM(CASE, 9)
填写 FROM1
循环计数的初值 NUM⇐1
取出算法编号 JMP⇐ CM(CASE,NUM)
按JMP值转入相应的算法执行部分
输出本次结果
有附加路径?
N
Y
输出附加路径
NUM⇐NUM + 1
NUM>COUNT?
N
Y
STOP

图 82-2

```
      DATA CM (1, 1) , CM (2, 1) , CM (3, 1) ,  CM
        (4, 1) , CM (5, 1) , CM (6, 1) , CM (7, 1) ,
     1  CM (8, 1) , CM (9, 1) , CM (1, 2), CM (2, 2) ,
        CM (3, 2) , CM (4, 2) , CM (5, 2) , CM (6, 2),
     2  CM (7, 2) , CM (8, 2) , CM (9, 2) , CM (1, 3),
        CM (2, 3) , CM (3, 3) , CM (4, 3) , CM (5, 3),
     3  CM (6, 3) , CM (7, 3) , CM (8, 3) , CM (9, 3),
        CM (1, 4) , CM (2, 4) , CM (3, 4) , CM (4, 4),
     4  CM (5, 4) , CM (6, 4) , CM (7, 4) , CM (8, 4),
        CM (9, 4) , CM (1, 5) , CM (2, 5) , CM (3, 5),
     5  CM (4, 5) , CM (5, 5) , CM (6, 5) , CM (7, 5),
        CM (8, 5) , CM (9, 5) , CM (1, 6) , CM (2, 6),
     6  CM (3, 6) , CM (4, 6) , CM (5, 6) , CM (6, 6),
        CM (7, 6) , CM (8, 6) , CM (9, 6) , CM (1, 7),
     7  CM (2, 7) , CM (3, 7) , CM (4, 7) , CM (5, 7),
        CM (6, 7) , CM (7, 7) , CM (8, 7) , CM (9, 7),
     8  / 1, 2, 5, 4, 1, 1, 4, 1, 2, 2, 3, 6, 6, 3, 2, 5,
        3, 3, 3, 4, 7, 8, 5, 3,
     9  6, 5, 4, 4, 4*0, 4, 7, 6, 6, 5, 4*0, 5, 8,8,7, 6,
        8*0, 7, 8*0/
        DATA  CM (1, 8) , CM (2, 8) , CM (3, 8) , CM
        (4,8) , CM (5, 8) , CM (6, 8) , CM (7, 8) ,
     1  CM (8, 8) , CM (9, 8) , CM (1, 9) , CM (2, 9),
        CM (3, 9) , CM (4, 9) , CM (5, 9) , CM (6, 9),
     2  CM (7, 9) , CM (8, 9) , CM (9, 9) /8, 8*0, 8, 4
        *3, 4*5/
```

常数数组CM

```
     1  WRITE (1, 2)
     2  FORMAT (3X, 43HTYPE  THE  INITATE
          STATE  OF  INSECTS  POSITION, /)
        DO  4  J=1, 9
     4  READ (1, 5)  (A(I, J) , I=1, 2)
     5  FORMAT (I1, 1X, I1)
        WRITE (1, 7)
```

读入各只小虫的初始位置

```
     7  FORMAT (3X, 28HTYPE  THE  MOVING
          INSECT' S  NO., /)
        READ (1, 10) INS
    10  FORMAT (I1)
```

读入移动位置的某只小虫的编号

```
        DO  40  I=1, 2
        DO  40  J=1, 9
        IF(A(I, J) . LT. 1. OR. A(I, J) .
          GT. 9) GO  TO  80
    40  CONTINUE
        IF (INS. LT. 1. OR. INS. GT. 9) GO  TO  80
        WRITE (1, 70)
```

输入数据检查

```
   70  FORMAT (3X, 3HOK! )
       GO  TO  100
   80  WRITE (1, 85)
   85  FORMAT (23HWRONG! INPUT  ONCE  AGAIN)
       GO  TO  1
C
C        1 < I < 9 ,  1 < J < 9
C
         100  IF (A (1, INS) . GT. 1. AND. A (1, INS) . LT. 9. AND. A (2,
         INS) . GT. 1. AND. A (2, INS) . LT. 9) CASE = 1
C
C        I = 1,  J = 9
C
         IF (A (1, INS) . EQ. 1. AND. A (2, INS) . EQ. 9) CASE = 2
C
C        I = 9,  J = 1
C
         IF (A (1, INS) . EQ. 9. AND. A (2, INS) . EQ. 1) CASE = 3
C
C        I = 9,  J = 9
C
         IF (A (1, INS) . EQ. 9. AND. A (2, INS) . EQ. 9) CASE = 4
C
C        I = 1 J = 1
C
         IF (A (1, INS) . EQ. 1. AND. A (2, INS) . EQ. 1) CASE = 5
C
C        I = 1,  1< J <9
C
         IF (A(1, INS). EQ. 1. AND. A (2, INS). GT. 1. AND. A(2, INS).LT.9)
           CASE = 6
C
C        I = 9,  1< J <9
C
         IF (A (1, INS) . EQ. 9. AND. A (2, INS) . GT. 1. AND. A (2, INS) .
           LT. 9) CASE = 7
C
C        1 < I < 9 ,  J = 1
C
         IF (A (1, INS) . GT. 1. AND. A (1, INS) . LT. 9. AND. A (2, INS) .
           EQ. 1 )CASE = 8
C
C        1 < I < 9 ,  J = 9
```

```
      IF (A(1. INS) , GT. 1. AND. A(1, INS) . LT. 9. AND. A(2, INS) .
        EQ. 9) CASE = 9
      COUNT = CM (CASE, 9)
      I = INS
      J = A(2, INS)
      FROM1( 1 ) = I
      FROM1( 2 ) = J
      DO 300  NUM = 1, COUNT
      JMP = CM(CASE, NUM)
      GO  TO (110, 120, 130, 140, 150, 160, 170, 180) JMP
110   K = A(2, I + 1)
      CALL  SEARCH (L, J + 1, A)
      CALL  WRT (TO1, I + 1, J + 1, FROM2, I + 1,
        K, TO2, I, K,  FROM3 , L, J + 1, TO3, L, J)
      GO  TO  200
120   K = A (2,  J + 1)
      CALL  SEARCH (L, J - 1, A)
      CALL  WRT (TO1, I + 1, J - 1, FROM2, I + 1,
        K, TO2, I, K, FROM3, L, J - 1, TO3, L, J)
      GO  TO  200
130   K = A (2,  I + 1)
      IF (K. EQ. 9) GO  TO  132
      IF (K. EQ. 1) GO  TO  134
      CALL  SEARCH (L, K + 1, A)
      CALL  WRT (TO1, I + 1, J, FROM2, I + 1, K,
        TO2, I, K + 1, FROM3, L, K + 1, TO3, L, K)
      CALL  SEARCH (L, K - 1, A)
      CALL  WRT (TO1, I + 1, J, FROM2, I + 1, K,
        TO22, I, K - 1, FROM33, L, K - 1, TO33, L,K)
      GO  TO  200
132   CALL  SEARCH (L, K - 1, A)
      CALL  WRT (TO1, I + 1, J, FROM2, I + 1, K,
      TO2,I,K-1,FROM3,L,K-1,TO3, L,K)
      GO  TO  200
134   CALL  SEARCH (L, K + 1, A)
      CALL  WRT (TO1, I + 1, J, FROM2, I + 1, K,
        TO2, I, K + 1, FROM3, L, K + 1, TO3, L, K)
      GO  TO  200
140   CALL  SEARCH (L, J - 1, A)
      IF (L. EQ. 9) GO  TO  144
      IF (L. EQ. 1) GO  TO  142
      K = A (2, L + 1)
      CALL  WRT (TO1, I, J - 1, FROM2, L, J - 1,
        TO2, L + 1, J, FROM3, L + 1, K, TO3, L, K)
      K = A(2, L - 1)
```

算法（1）

算法（2）

算法（3）

```
      CALL WRT (TO1, I, J-1, FROM2, L, J-1,
        TO22, L-1, J, FROM33, L-1, K, TO33,
     1  L, K)
      GO TO 200
142   K = A (2, L+1)
      CALL WRT (TO1, I, J-1, FROM2, L, J-1,
        TO2, L+1, J, FROM3, L+1, K, TO3, L, K)
      GO TO 200
144   K = A (2, L-1)
      CALL WRT (TO1, I, J-1, FROM2, L, J-1,
        TO2, L-1, J, FROM3, L-1, K, TO3, L, K)
      GO TO 200

150   CALL SEARCH (L, J+1, A)
      IF (L. EQ. 9) GO TO 152
      IF (L. EQ. 1) GO TO 154
      K=A (2, L-1)
      CALL WRT (TO1, I, J+1, FROM2, L, J+1,
        TO2, L-1, J, FROM3, L-1, K, TO3, L, K)
      K=A (2, L+1)

      CALL WRT (TO1, I, J+1, FROM2, L, J+1,
        TO22, L+1, J, FROM33, L+1, K, TO33,
     1  L, K)
      GO TO 200

152   K = A (2, L-1)
      CALL WRT (TO1, I, J+1, FROM2, L, J+1,
        TO2, L-1, J, FROM3, L-1, K, TO3, L, K)
      GO TO 200
154   K=A (2, L+1)
      CALL WRT (TO1, I, J+1, FROM2, L, J+1,
        TO2, L+1, J, FROM3, L+1, K, TO3, L, K)
      GO TO 200
160   K=A (2, I-1)
      IF (K. EQ. 9) GO TO 162
      IF (K. EQ. 1) GO TO 164
      CALL SEARCH (L, K-1, A)
      CALL WRT (TO1, I-1, J, FROM2, I-1, K,
        TO2, I, K-1, FROM3, L, K-1, TO3, L, K)
      CALL SEARCH (L, K+1, A)
      CALL WRT (TO1, I-1, J, FROM2, I-1, K,
      TO22, I, K+1, FROM33, L, K+1, TO33,
     1  L, K)
      GO TO 200
```

算法（4）

算法（5）

算法（6）

```
162     CALL  SEARCH (L, K-1, A)
        CALL  WRT (TO1, I-1, J, FROM2, I-1, K,
          TO2, I, K-1, FROM3, L, K-1, TO3, L, K)
        GO  TO  200
164     CALL  SEARCH (L, K+1, A)
        CALL  WRT (TO1, I-1, J, FROM2, I-1. K,
          TO2, I, K+1, FROM3, L, K+1, TO3, L, K)
        GO  TO  200
170     K=A (2, I-1)
        CALL  SEARCH (L, J-1, A)
        CALL  WRT (TO1, I-1, J-1, FROM2, I-1, K,
          TO2, I, K, FROM3, L, J-1, TO3, L, J)
        GO  TO  200
180     K = A(2,  I -1)
        CALL  SEARCH (L, J+1, A)
        CALL  WRT (TO1, I-1, J+1, FROM2, I-1, K,
          TO2, I, K, FROM3, L, J+1, TO3, L, J)
200     WRITE (1, 270) NUM
270     FORMAT (///2X, 11HTHIS IS THE, I2, 29HTH
          MOVING  PATH  OF  THE  INSECTS)
        WRITE (1, 250) FROM1, TO1
        WRITE (1, 250) FROM2, TO2
        WRITE (1, 250) FROM3, TO3
        IF (JMP. LT. 3. OR. JMP. GT. 6) GO  TO  300
        IF ( (K. EQ. 1, OR. K. EQ. 9) . AND. (JMP.
          EQ. 3. OR. JMP. EQ. 6). OR. (L. EQ. 1.OR.
    1   L. EQ. 9). AND. (JMP. EQ. 4. OR. JMP. EQ.
          5) ) GO  TO  300
260     WRITE (1, 210) NUM
210     FORMAT (/2X, 35HTHIS IS THE OTHER ADDI-
          TIONAL  OF  THE, I2, 7HTH  PATH)
        WRITE (1, 250) FROM2, TO22
250     FORMAT (2X, 6HFROM (, I2, 1H., I2, 5H) TO
          (, I2, 1H. , I2, 1H) )
        WRITE (1, 250) FROM33, TO33
300     CONTINUE
        STOP
        END
C
C       THIS  SUBROUTINE  RESEARCH  THE  1 INE  NUMBER  OF  ARRAY
          BY  THE  1  TO. NUMBER
        SUBROUTINE  SEARCH (X, Y, A1)
        INTEGER  A1 (2, 9)
        INTEGER  X, Y, C
```

算法（7）

算法（8）

输出本次移动的结果

若存在可能的其它移动方向，则输出之。

子程序，由小虫的列号查出其所在的行号。

```
      DO 310 C=1, 9
      IF (A1 (2, C) . NE. Y) GO TO 310
      X=A1(1, C)
310   CONTINUE
      RETURN
      END
C
C     THIS SUBROUTINE FILL THE NUMBER INTO 子程序，分别添写五个
                                           数组。
C
      SUBROUTINE WRT (A1, I1, J1, A2, I2, J2, A3, I3, J3, A4, I4, J4, A5,
       I5, J5)
      INTEGER I1, I2, I3, I4, I5, J1, J2, J3, J4, J5
      INTEGER A1(2) , A2(2) , A3(2) , A4(2) , A5(2)
      A1( 1) = I 1
      A1( 2) = J 1
      A2( 1) = I 2
      A2( 2) = J 2
      A3( 1) = I 3
      A3( 2) = J 3
      A4( 1) = I 4
      A4( 2) = J 4
      A5( 1) = I 5
      A5( 2) = J 5
      RETURN
      END
```

下面是6号小虫可能有八种移动及其它二只小虫也相应移动：

```
THIS IS THE 1TH MOVING PATH OF
  THE INSECTS              第一种移动
FROM (6, 6) TO (7, 7)      6号虫由(6,6)至(7,7)
FROM (7, 8) TO (6, 8)      7号虫由(7,8)至(6,8)
FROM (9, 7) TO (9, 6)      9号虫由(9,7)至(9,6)
                           解释下同
THIS IS THE 2TH MOVING PATH OF THE
  INSECTS
FROM (6, 6) TO (7, 5)
FROM (7, 8) TO (6, 8)
FROM (1, 5) TO (1, 6)

THIS IS THE 3TH MOVING PATH OF
  THE INSECTS
FROM (6, 6) TO (7, 6)
```

注1：本输出结果的输入是：

```
八只小虫的初始位置为
 1, 5 (即位于第1行，第5列)
 2, 3
 3, 9
 4, 1
 5, 4
 6, 6
 7, 8
 8, 2
 9, 7
```

```
FROM (7, 8) TO (6, 9)
FROM (3, 9) TO (3, 8)
THIS IS THE OTHER ADDITIONAL OF THE
 3TH PATH
FROM (7, 8) TO (6, 7)
FROM (9, 7) TO (9, 8)

THIS IS THE 4TH MOVING PATH OF THE
 INSECTS
FROM (6, 6) TO (6, 5)
FROM (1, 5) TO (2, 6)
FROM (2, 3) TO (1, 3)

THIS IS THE 5TH MOVING PATH OF THE INSECTS
FROM (6, 6) TO (6, 7)
FROM (9, 7) TO (8, 6)
FROM (8, 2) TO (9, 2)

THIS IS THE 6TH MOVING PATH OF
 THE INSECTS
FROM (6, 6) TO (5, 6)
FROM (5, 4) TO (6, 3)
FROM (2, 3) TO (2, 4)
THIS IS THE OTHER ADDITIONAL
 OF THE 6TH PATH
FROM (5, 4) TD (6, 5)
FROM (1, 5) TO (1, 4)

THIS IS THE 7TH MOVING PATH OF
 THE INSECTS
FROM (6, 6) TO (5, 5)
FROM (5, 4) TO (6, 4)
FROM (1, 5) TO (1, 6)

THIS IS THE 8TH MOVING PATH OF THE INSECTS
FROM (6, 6) TO (5, 7)
FROM (5, 4) TO (6, 4)
FROM (9, 7) TO (9, 6)
```

移动位置的小虫编号为：6

注2：由于6号小虫有八种可能的移动方向，因此找出有这八种可能移动方向的其它二只小虫的移动方向，其结果见左边。

左边结果表示第六种可能的移动方向：

6号小虫由（6，6）移动（5，6）

5号小虫由（5，4）移到（6，3）

2号小虫由（2，3）移到（2，4）

在6号小虫上述的移动方向时，其它二只小虫也可能做另一种移动，即：

5号小虫由（5，4）移到（6，5）

1号小虫由（1，5）移到（1，4）

注：为了简化程序，第1、第2、第3分别用1TH、2TH、3TH表示。

4）附注

为了编写程序简单，特做如下约定：

（1）九只小虫从1至9顺序编号，按小虫所在行编号，即：第5行上的小虫编号为5。

（2）任二只小虫的i、j值不能同时只差1。

（3）输入时按小虫的编号顺序输入。

（三）思 考 题

行、列各为6的36个方格。请你把12个棋子放到方格里去，每个方格只能放一只棋子，使每一横行，每一纵列和两条主对角上都恰好有两只棋子。

83 生 命 游 戏

由剑桥大学的John Horton Conway发明的生命之游戏，在一个由方格组成的M×N的矩形阵上进行，每一个方格可以包含一个机体。每一个方格和八个方格相邻，我们用OCC（K）表示与方格K相邻的、被机体所占有的方格数。机体分布每隔一定时间变化一次，有的死亡，并产生一些新的机体。从某一种机体的分布情况到新生成的机体分布情况应满足下述二个规则：

1）若2≤OCC(K)≤3，则方格K中的机体仍有活到下一次生成（若K非空），否则死亡。

2）若OCC(K)=3，则在一个空方格K中诞生一个新的机体。

请构造一个程序，模拟机体生成分布的变化，并将该变化打印出。

（一）算 法 分 析

（1）为了保证每个K与八个方格相邻，对于阵列的四边不做观察，只对（M－2）×（N－2）范围内进行观察。

（2）由于要在已有的机体分布情况下计算下一次新的机体分布，为此要设二个M×N×2的矩阵A、B。一个矩阵记录偶次数（包括初态）变化，另一个矩阵存放奇数次变化，依次输出。每个矩阵中，一个M×N记录是否有机体存在，用0、1表示，另一个M×N记录其相应的OCC（K）。

（3）设A(i，j，1)、B(i，j，1)中存放是否有机体的标志，A(i，j，2)、B(i，j，2)中存放OCC(K)，取值为1～8，则有：

（a）若2≤A(i，j，2)≤3，且A(i，j，1)=1，则令B(i，j，1)=1；

（b）若A(i，j，2)=3，且A(i，j，1)=0，则令B(i，j，1)=1；

（c）对于A(i，j，2)的其余值，令B(i，j，1)=0。

（4）对于每个B(i，j，1)，依次检查其相邻的八个格中等于1的个数，并填入B(i，j，2)，作为OCC(K)的值。

（5）下一次变化时，同样由B产生A。

（二）程 序 设 计

1）设计思路

首先要给出机体的初始配置情况，为了减少输入上的麻烦，事先造一个数据文件INDATA，程序读入INDATA做为机体分布的初始状态。改变初始状态只须修改INDATA

即可。

输入用户指定的变化次数，然后按算法（3）、（4）计算机体的变化情况，并输出结果。

2）框图　见图83

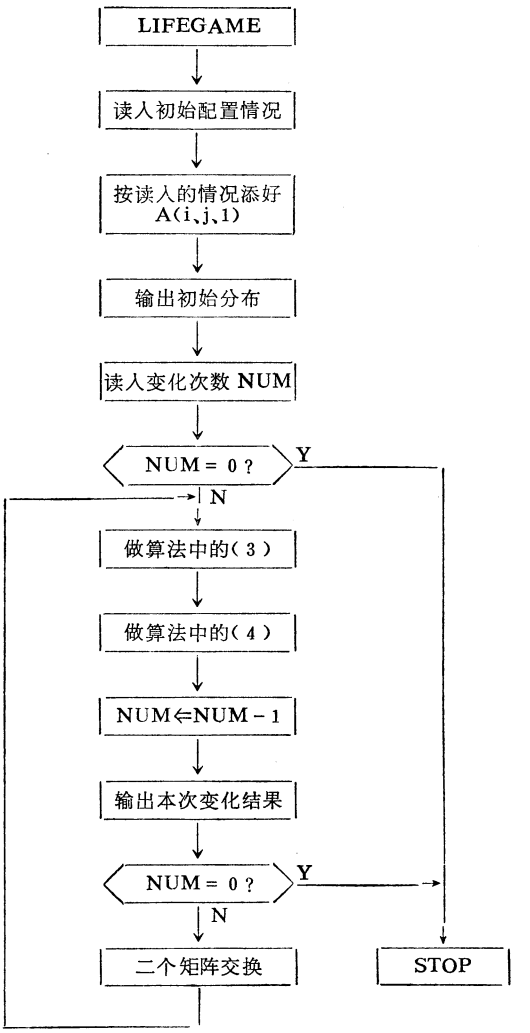


图 83

3）源程序及运行结果

```
C
C       PROGRAM  LIFEGAME
C       READ  INITATE  STATE  AND  THE  NUMBER  OF  CHANGES  WHICH
          MAKE  THF  STATE
```

```
C       CHANGEBLE, AN  QUTPUT  RESULT  EVERY  TIME.
C
        INTEGER  COUNT, NUM, I, J, K
        INTEGER  FIRST (20, 40, 2) , SECOND (20, 40, 2)
        OPEN (5, FILE='INDATA', ACCESS='READ')
        OPEN (3, FILE='RESULT', ACCESS='WRITE')
        READ (5, 5)  (((FIRST (I, J, K) , K=1, 1) , J=1, 40) , I=1, 20)
                                                          输入初始状态
5       FORMAT (40I1)
C
C       OUTPUT  INITATE  STATE
C
        WRITE (3, 7)
7       FORMAT (10X, 26HTHIS  IS  THE  INITIATE  STATE)
        WRITE (3,10)  (((FIRST (I,J,K) ,K=1, 1) ,J=1,40) , I=1, 20)
10      FORMAT (40I2)
        CALL  OCC (FIRST)
        READ (1, 15) NUM                                  输入变化次数
        COUNT=0
15      FORMAT (I4)
        IF (COUNT. EQ. NUM)  GO  TO  150
17      DO  40  I=2, 19
        DO  40  J=2, 39
        IF (FIRST (I, J, 2). GE. 2. AND. FIRST (I,
          J, 2) . LE. 3. AND. FIRST (I, J, 1). EQ. 1)
          GOTO  55
        IF (FIRST (I, J, 2) . EQ. 3. AND. FIRST (I,        } 算法 (3)
          J, 1). EQ. 0) GO  TO  55
        SECOND (I, J, 1) = 0
        GO  TO  40
55      SECOND (I, J, 1) = 1
40      CONTINUE
        CALL  OCC (SECOND)
        COUNT = COUNT + 1
C
C       OUTPUT  RESULT
C
        WRITE (3, 45) COUNT
45      FORMAT (//10X, 22HTHIS  IS  THE
          RESUL  T  OF, I4, 7HTH  TIME)
        WRITE (3, 10) (((SECOND (I, J, K) ,   } 输出奇数次变化结果
          K=1, 1), J=1, 40) , I=1, 20)
        IF (COUNT. EQ. NUM) GO  TO  150
```

```
      DO  80  I=2, 19                                    ┐
      DO  80  J=2, 39                                    │
      IF (SECOND(I, J, 2) . GE. 2. AND.                  │
        SECOND(I, J, 2) . LE. 3. AND.                    │
        SECOND(I, J, 1). EQ. 1) GO  TO  65               │
      IF (SECOND(I, J, 2) . EQ. 3. AND.                  ├ 算法（3）
        SECONDI(, J, 1) . EQ. O) GOTO  65                │
      FIRST(I, J, 1) = 0                                 │
      GOTO  80                                           │
65    FIRST (I, J, 1) = 1                                │
80    CONTINUE                                           ┘
      CALL  OCC (FIRST)
      COUNT = COUNT + 1
C
C     OUTPUT  RESULT
C
      WRITE (3, 45) COUNT                                ┐
      WRITE (3, 10)  (((FIRST(I, J, K) ,                 │
        K = 1, 1), J = 1, 40), I = 1, 20)                ├ 输出偶数次变化结果
      IF (COUNT. NE. NUM) GO  TO  17                     │
150   CLOSE (-1)                                         ┘
      STOP
      END
C
C     SUBROUTINE  ONE
C
      SUBROUTINE  OCC (MATRIX)                  本子程序的功能是算法（4）
      INTEGER  MATRIX(20, 40, 2)
      INTEGER  I, J, K, C
      DO  200  I = 2, 19
      DO  200  J = 2, 39
      C = 0
      IF (MATRIX(I-1, J-1, 1) . EQ. 1)  C = C + 1
      IF (MATRIX(I-1, J, 1) . EQ. 1)    C = C + 1
      IF (MATRIX(I-1, J+1, 1) . EQ. 1)  C = C + 1
      IF (MATRIX(I, J+1, 1) . EQ. 1)    C = C + 1
      IF (MATRIX(I+1, J+1, 1) . EQ. 1)  C = C + 1
      IF (MATRIX(I+1, J, 1) . EQ. 1)    C = C + 1
      IF (MATRIX(I+1, J-1, 1) . EQ. 1)  C = C + 1
      IF (MATRIX(I, J-1, 1) . EQ. 1)    C = C + 1
      MATRIX(I, J, 2) = C
200   CONTINUE
      RETURN
```

END

```
01010101010101010101010101010101010101
11001100110011001100110011001100110011001100
11100011100011100111000111000111000111000
00000000001111111111000000011111111000000
11101111011110111101111011110111101111000
00001000010000100001000010000100001000000
11110000111100000111100000111100000111000
10101010101010101010101010101010101010101
01010101010101010101010101010101010101010
00000000000000000000000000000000000000000
11111111111111111111111111111111111111111
11111111111111111100000000000000000000000
10000000000000000000111111111111111111111
01010101010101010101010101010101010101010
00000000011111111100000000111111110000000
00000010000001000000010000000100110011110
11000010000100000100000111000001000001000
11110000010000110000011000011010000100001
00001010101010101010101010000001100011000
10001000010000100001000010000100001000100
```

THIS IS THE INITIATE STATE

输入文件 INDATA，其作为机体分布的初始状态。

```
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 0
0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 1 1 1 0
1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0
1 1 1 1 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0 1 0 0 0 1
0 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 1 1 0 0
1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0
```

以下是机体变化的结果。

THTS IS THE RESULT OF 1TH TIME 第一次变化后

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 1 0 0 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 1 0 1 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 1 1 0 0 0 0 1 0 0
0 0 1 0 1 1 0 1 1 0 0 1 0 1 0 1 0 1 0 0 1 1 0 1 1 0 1 0 1 0 0 0 0 0 0 0 1 1 0 0
0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0
0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0
0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 1 0 1 0 0 0 1 0 0 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 0 0 0 1 1 1 1 1 0 0 0 0 1 0 0
0 0 1 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 0 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 1 0
0 0 1 0 1 1 0 0 1 0 1 0 0 0 1 0 0 0 0 1 1 0 1 0 0 0 0 1 1 1 0 1 1 0 0 0 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

THIS IS THE RESULT OF 2TH TIME 第二次变化后

```
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
1 0 0 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 0 0 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0
1 0 1 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0
1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0
0 0 1 1 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1
0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 1
0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 0 0 0 0 1 1 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0 0 1 1 1 0
1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0
1 0 1 1 0 1 1 0 1 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1
0 0 1 0 1 1 1 0 1 0 1 1 0 1 1 0 1 0 0 0 1 0 1 0 0 0 0 0 1 1 1 1 1 0 0 0 1 0 1 0
1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0
```

THIS IS THE RESULT OF 3TH TIME 第三次变化后

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
0 1 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 1 0 1 0 0 0 1 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 1 1 0
0 1 1 0 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 1 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 0 0 1 0
0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 1 1 0 0 0 1 1 0 0 0 1 0 1 0 1 0 0 1 1 1 1 0 0 1 0
0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 1 0 0 0 1 1 1 1 1 0
0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 0 0 1 1 0 0 1 0 0 1 0 0 1 1 0
0 0 1 1 0 0 1 0 1 0 0 0 0 1 1 1 1 1 0 0 0 1 1 1 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 1 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

THIS IS THE RESULT OF 4TH T IME 第四次变化后

```
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0
0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0
1 1 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 1 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0
0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 1 1 0 1 1 1 1 0 1
0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0
1 1 0 1 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1
1 0 1 0 1 1 1 0 0 0 1 1 1 1 1 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 1 0 1 1
0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 1 0 1 1 0 0 0 0 1 0 1 0 1 1 1
0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0
0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 1 0 0 1 0 1 1 1 1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0
1 0 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 1 1 0 1 1 0 1 1 0 0 1 0
1 1 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 1
0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0
```

THIS IS THE RESULT OF 5TH TIME 第五次变化后

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 1 1 0 1 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 1 0
0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 1 0
0 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 1 1 1 0
0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 1 1 1 1 1 0 0 0 0 0 1 0 0 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 1 1 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0
0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 1 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

注：为了简化程序，将第 1 、第 2 、第 3 分别用1TH、2TH、3TH表示。

（三）思 考 题

请你自己出道类似这样题的游戏。

十二 巧 妙 填 数

84 使每两个棋子跳在一起

十二只象棋，排成一个圆圈如图84-1所示。每次移动一个棋子，移动时必须跳过两只棋子，然后与另一只棋子迭合。允许移动六次，使棋子两个两个地迭合在一起，并分别做到：奇数的棋子迭放在偶数上。

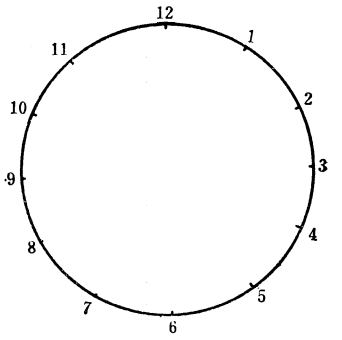


图 84-1

（一）笔算步骤和结果

答案的顺序是：1→4，5→8，9→12，3→6，7→10，11→2。

（二）程 序 设 计

1）设计思路

建立一个数组K（12），将1～12的数字放在相应的数组元素里，然后用循环语句依次判数组元素是偶数时，放过；是奇数时，记下该奇数I，将其加3，便是要跳在的偶数I 3上，即I 3⇐I＋3。完成这一对奇、偶吻合后，立即打印出该奇、偶对，并将这对数组元素冲成零。再换一个数时，判该数组元素若是已冲零或是偶数便放过。若是奇数，便重复上述做法。直到将数组元素全冲成零为止。每落实一对，计数器J加1，也打印出次数J，J＝6时便应结束。

2）框图 见图84-2

3）源程序及运行结果

```
      DIMENSION K(12)
      DO 10 I=1, 12
10    K(I) =I
      J = 0
      DO 20 I=1, 12
      IF (K(I) . EQ. 0) GOTO 20
      IF (K(I) /2*2. EQ. K(I)) GOTO 20
      IF (J. EQ. 6) STOP 'RIGHT'
      J=J+1
      IF (I+3. LE. 12) GOTO 25
      I3=I+3-12
      WRITE (10, 30) J, I, I3
      GOTO 40
25    I3=I+3
      WRITE (10 30) J, I, I3
30    FORMAT (10X, 2HJ=, I1, 3H:, I2, 4H--->, I2)
40    K(I) = 0
```

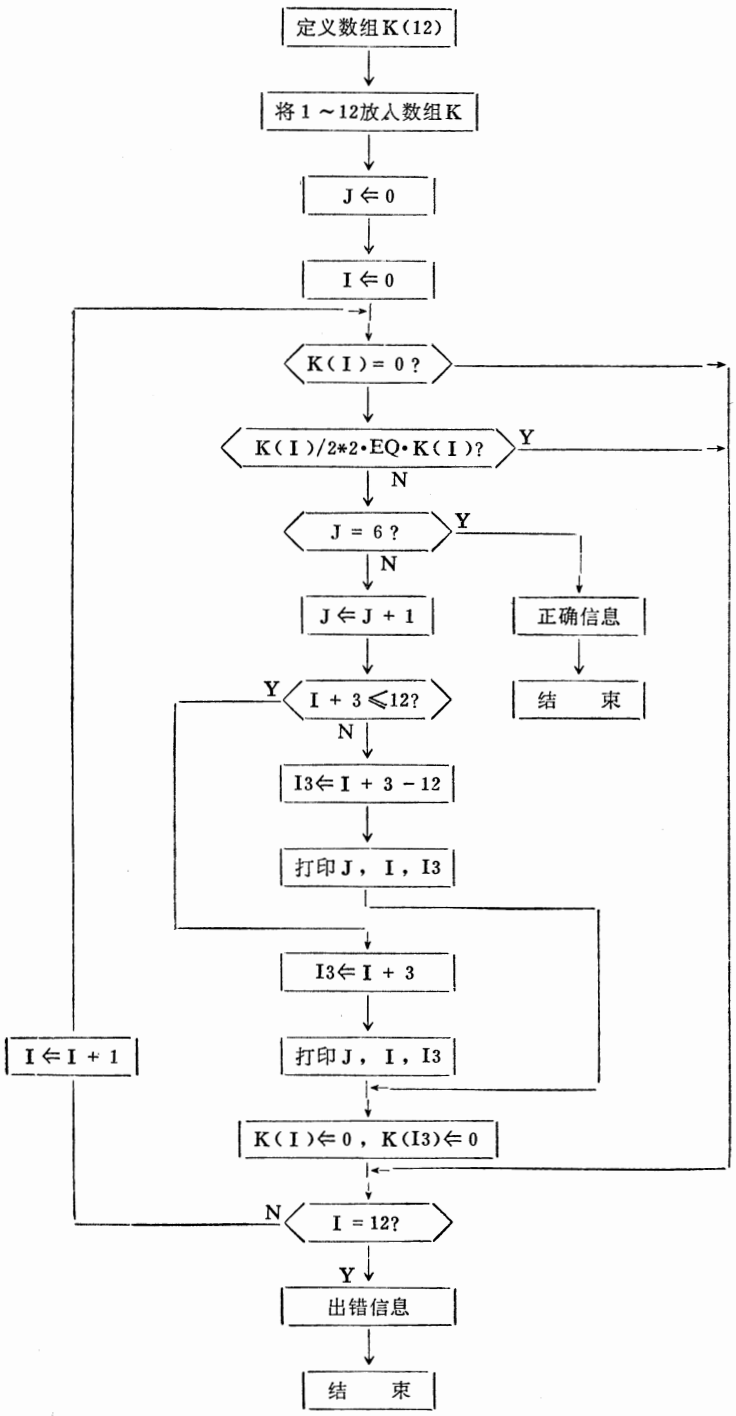

定义数组K(12)
将1～12放入数组K
J⇐0
I⇐0
K(I)=0?
K(I)/2*2·EQ·K(I)?
Y
N
J=6?
Y
N
J⇐J+1
正确信息
结　束
Y
I+3≤12?
N
I3⇐I+3-12
打印J，I，I3
I3⇐I+3
I⇐I+1
打印J，I，I3
K(I)⇐0，K(I3)⇐0
N
I=12?
Y
出错信息
结　束

图 84-2

```
          K(I3) = 0
20        CONTINUE
          STOP 'STOP'
          END
```

J = 1 ： 1 ---> 4
J = 2 ： 3 ---> 6
J = 3 ： 5 ---> 8
J = 4 ： 7 ---> 10
J = 5 ： 9 ---> 12
J = 6 ： 11 ---> 2

（三）思 考 题

如果按主题目的条件，将 7 到 12 只 棋 子在上， 1 到 6 只棋子在下，应如何进行程序设计。

85 按 要 求 填 表

把 1， 2， 3， 4， 5， 6 填入图85-1表格内，要使得每一列右边的数字比左边的数字大，每一行下面的数字比上面的大，按此要求，你有几种填法？

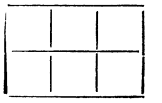


图 85-1

（一）算法分析和结论

由题目要求 1 必定在左上角， 6 一定在右下角， 5 可能在两个位置即：

1		
	5	6

(a)

1		5
		6

(b)

而 4 对于（a）有两个可能的位置：

1		
4	5	6

1		4
	5	6

因此对（a）有三种填法：

1	2	3
4	5	6

(1)

1	2	4
3	5	6

(2)

1	3	4
2	5	6

(3)

同样对（b）有两种填法：

1	3	5
2	4	6

(4)

1	2	5
3	4	6

(5)

所以共有五种符合条件的填法（如上（1）～（5）图）。

（二）程 序 设 计

1）设计思路

定义数组A（2，3）表示此表。先把1和6分别放入左上角和右下角。其余数字，利用二重循环来寻找填法，每试探一种填法时，按题意判是否符合条件，符合条件时，立即打印出来，否则，再重新寻找，直到二重循环完为止。请见程序运行时打印的五种填法。

2）框图　见图85-2

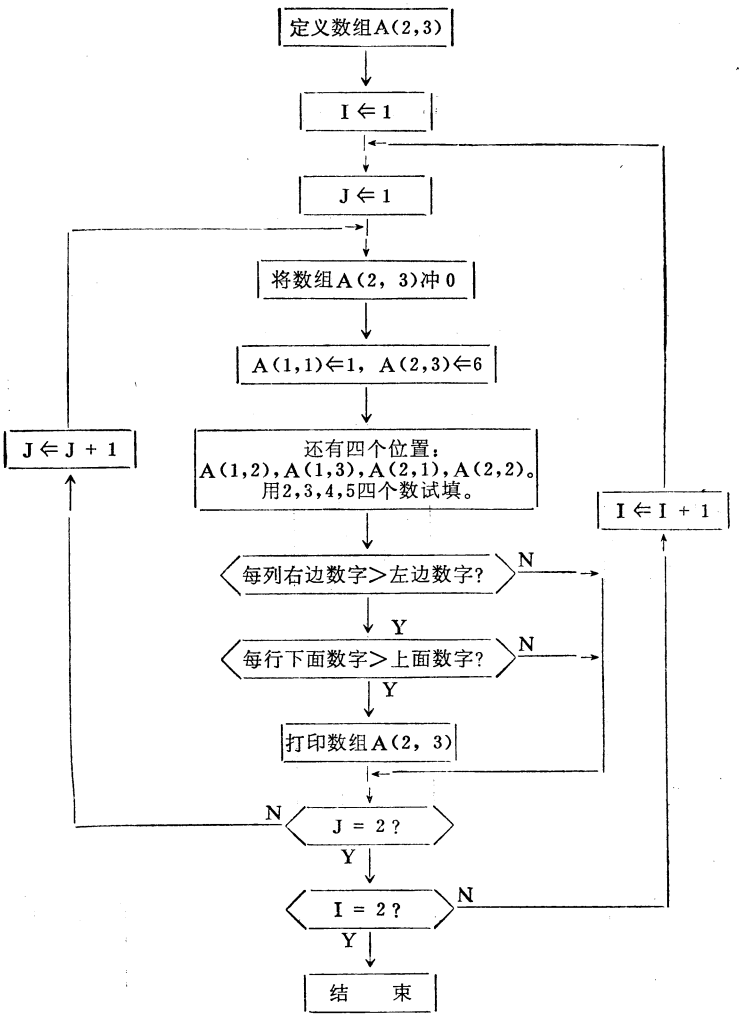


图 85-2

3）源程序及运行结果

```
C       PROGRAM TB
        INTEGER A( 2，3 )
        DO 100 I= 1，2
```

```
      DO 100 J = 1, 2
      DO 5 I1 = 1, 2
      DO 5 J1 = 1, 3
5     A (I1, J1) = 0
      N = 0
      A(1, 1) = 1
      A(2, 3) = 6
      GOTO (10, 20) I
10    A(1, 2) = 2
      GOTO 30
20    A(2, 1) = 2
30    GOTO (40, 50) J
40    A(1, 3) = 5
      GOTO 60
50    A(2, 2) = 5
60    DO 90 K = 1, 2
      DO 90 L = 1, 3
      IF (A(K, L) . NE. 0) GOTO 90
      A(K, L) = 3
      NO = 1
63    IF (K. EQ. 1) GOTO 65
      IF (A(K, L) . GT. A(K-1, L)) GOTO 70
      GOTO 80
65    IF(A(K, L) . LT. A(K+1, L)) GOTO 70
      A(K+1, L) = 4
      GOTO 80
70    GOTO (71, 73, 75) L
71    IF (A (K, L) . LT. A(K, L+1)) GOTO 85
      A(K, L+1) = 4
      GOTO 80
73    IF (A(K, L) . GT. A(K, L-1) . AND. A(K, L) . LT. A(K, L+1) )
         GOTO 85
      A(K, L+1) = 4
      GOTO 80
75    IF (A(K, L) . GT. A(K, L-1)) GOTO 78
      IF (NO. NE. 1) GOTO 100
      GOTO 90
78    IF (NO. NE. 1) GOTO 85
      IF (N. NE. 0) GOTO 88
      A(2, 1) = 4
      N = 1
      GOTO 80
88    A(2, 1) = 3
```

```
      A(K, L) = 4
80    NO = NO + 1
      GOTO 63
90    CONTINUE
85    WRITE (10, 96) ((A(M1, M2), M2 = 1, 3), M1 = 1, 2)
96    FORMAT (1X, 3I6/)
      IF (A(1, 3). EQ. 3) GOTO 98
      GOTO 100
98    A(1, 3) = 4
      A(2, 1) = 3
      GOTO 85
100   CONTINUE
      STOP
      END
```

程序运行时打印结果

```
1   2   5
3   4   6        } 第一种填法

1   2   3
4   5   6        } 第二种填法

1   2   4
3   5   6        } 第三种填法

1   3   5
2   4   6        } 第四种填法

1   3   4
2   5   6        } 第五种填法
```

（三）思 考 题

请改写程序，不首先固定1、6的位置，把这五种填表方法搜寻出来。

86 钟表盘上数字分组

将钟盘分为六部分（形状、大小不限），只要使各部分数字之和相等就行。

（一）笔算步骤和结果

因为盘上数字之和为 1 + 2 + 3 + … + 12 = 78，若分为六等份，每份的数字之和应为 78 ÷ 6 = 13。

（二）程 序 设 计

1）设计思路

定义数组IA（12）放入相应的1～12数字，用内、外循环语句来寻找每两个相加为13的数字，冲入数组IB（6），找到便立即打印输出，并将该对数组元素冲成零。

2）框图 见图86-1

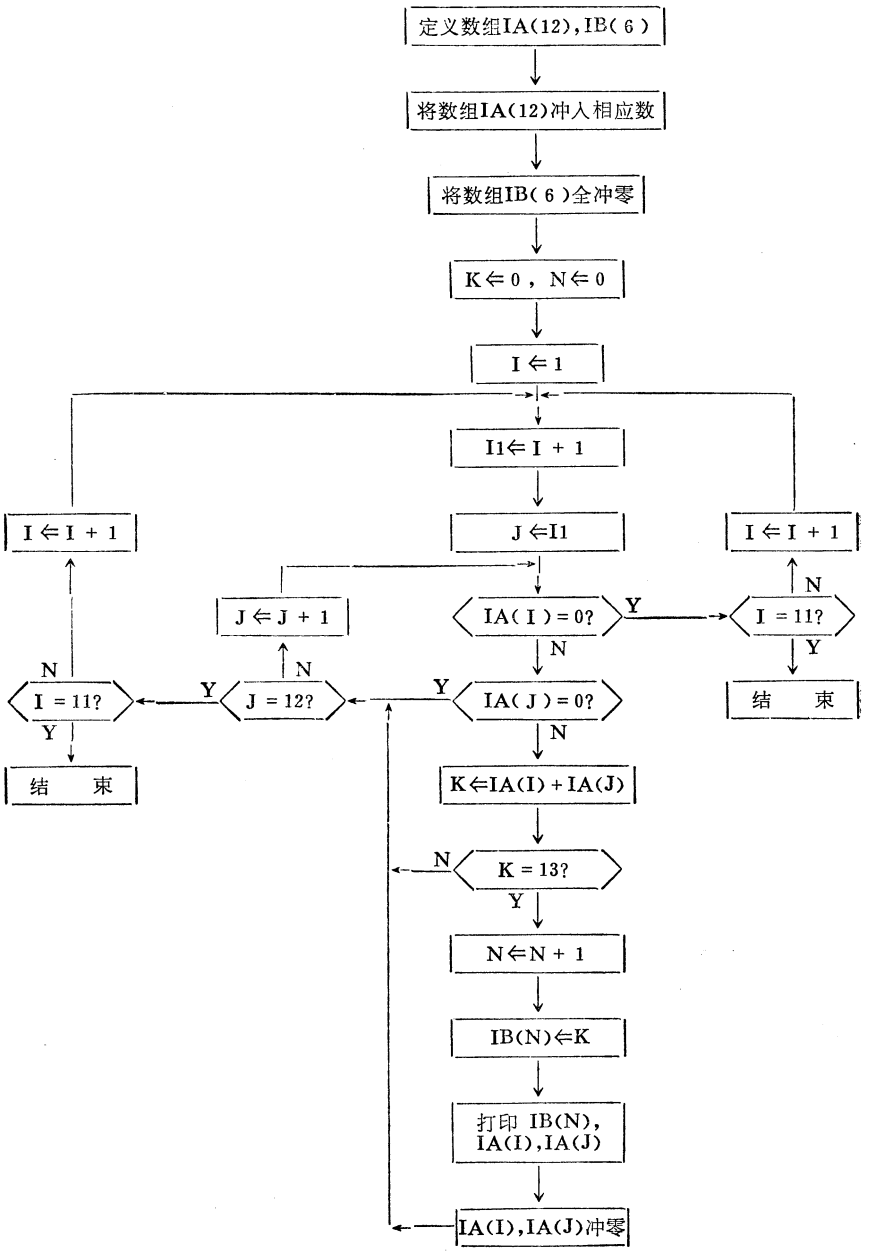


图 86-1

3）源程序及运行结果

```
      DIMENSION IA(12) , IB(6)
      DO 10 I=1, 12
10    IA(I) =I
      DO 20 I=1, 6
20    IB(I) = 0
      K = 0
      N = 0
```

```
      DO 30 I=1, 11
      I1=I+1
      DO 40 J=I1, 12
      IF (IA(I) . EQ. 0) GOTO 30
      IF (IA(J) . EQ. 0) GOTO 40
      K=IA(I) +IA(J)
      IF (K. NE. 13) GOTO 40
      N=N+1
      IB (N) =K
      WRITE (10, 50) IB(N) , IA(I) , IA(J)
50    FORMAT (5X, I2, 1H=, I2, 1H+, I2)
      IA(I) = 0
      IA(J) = 0
40    CONTINUE
30    CONTINUE
      STOP
      END
```

```
13= 1 +12
13= 2 +11
13= 3 +10
13= 4 + 9
13= 5 + 8
13= 6 + 7
```

程序运行结果

（三）思 考 题

战士们做了一个靶子，靶子分五格，每一格中有击中的分数如图86-2。小李射了若干枪，每次都击中靶子，正好共得100分。请你算一算，小李射了几枪，击中哪几格？并进行程序设计。

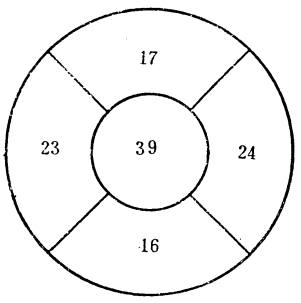


图 86-2

87 变换各数的位置

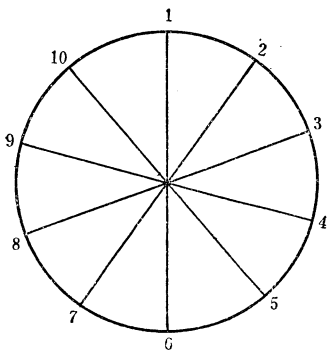


图 87-1

在5条直径线的两端分放着从1到10的顺序数，如图87-1示，但这样分置法，只有一种情况两邻接数之和等于相对位置上的两邻接数之和，即：

$$10+1=5+6$$

而 $1+2\neq 6+7$

或 $2+3\neq 7+8$

现在请你变换一下这些数的位置，使任何两个邻接数之和等于相对位置上的两邻接数之和。预料这题目不仅有一个答案，合题目要求的排列有好几种，试求出一个能确定全部解答的方法来。

（一）笔算步骤和结果

如果在任意一条直径线的两端分置A和a两数，又在邻接直径线上两端分置B和b两数，那末按要求A + B = a + b。由此A - a = b - B，即各相对位置上的两数之差，彼此应相等。从这个线索中找到全部解答。

现在为了便于解题，须把从1到10各数分成5对，每对之差相等。经研究分析，知道合条件成对的数只有两组：

（1） 差数 = 1

1 — 2
4 — 3
5 — 6
8 — 7
9 —10

如图 87-2

（2） 差数 = 5

1 — 6
7 — 2
3 — 8
9 — 4
5 —10

如图 87-3

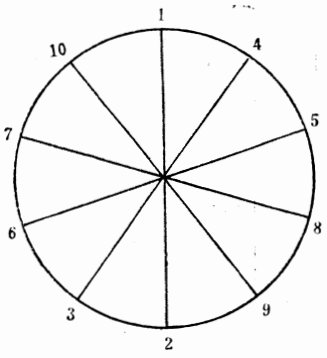


图 87-2

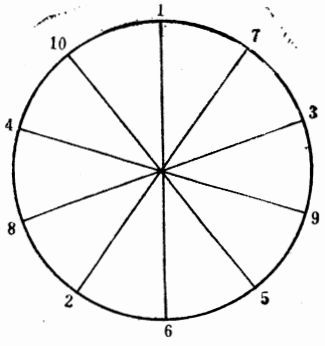


图 87-3

（二）程 序 设 计

1）设计思路

设D代表差数，从1到5循环，A（2，5）存放结果，同一列上的两个元素表示同一直径线上的两个数，第一行与第二行元素首尾相接形成一个圈，正好表示了相邻关系；又设数组B（10）存放1～10十个数，从小到大排列。每次循环开始时，先从B中取出最小一数，加上差数D，再取对应的这个数，放到数组B的同一列中，若是奇数列，小的在前；偶数列，大的在前。再比较相邻两对之和是否相等，若不等则重新循环。

2）框图 见图87-4

3）源程序及运行结果

```
C       PROGRAM BHSW
        INTEGER A(0 : 2, 0 : 5), B(10), C1, C2, D   NOVA/4机定义数组A
        DO 100 D = 1, 5                                  D是二数之差
        DO 15 I = 1, 10
15      B(I) = I
        DO 80 N = 1, 5
        N1 = 0
```

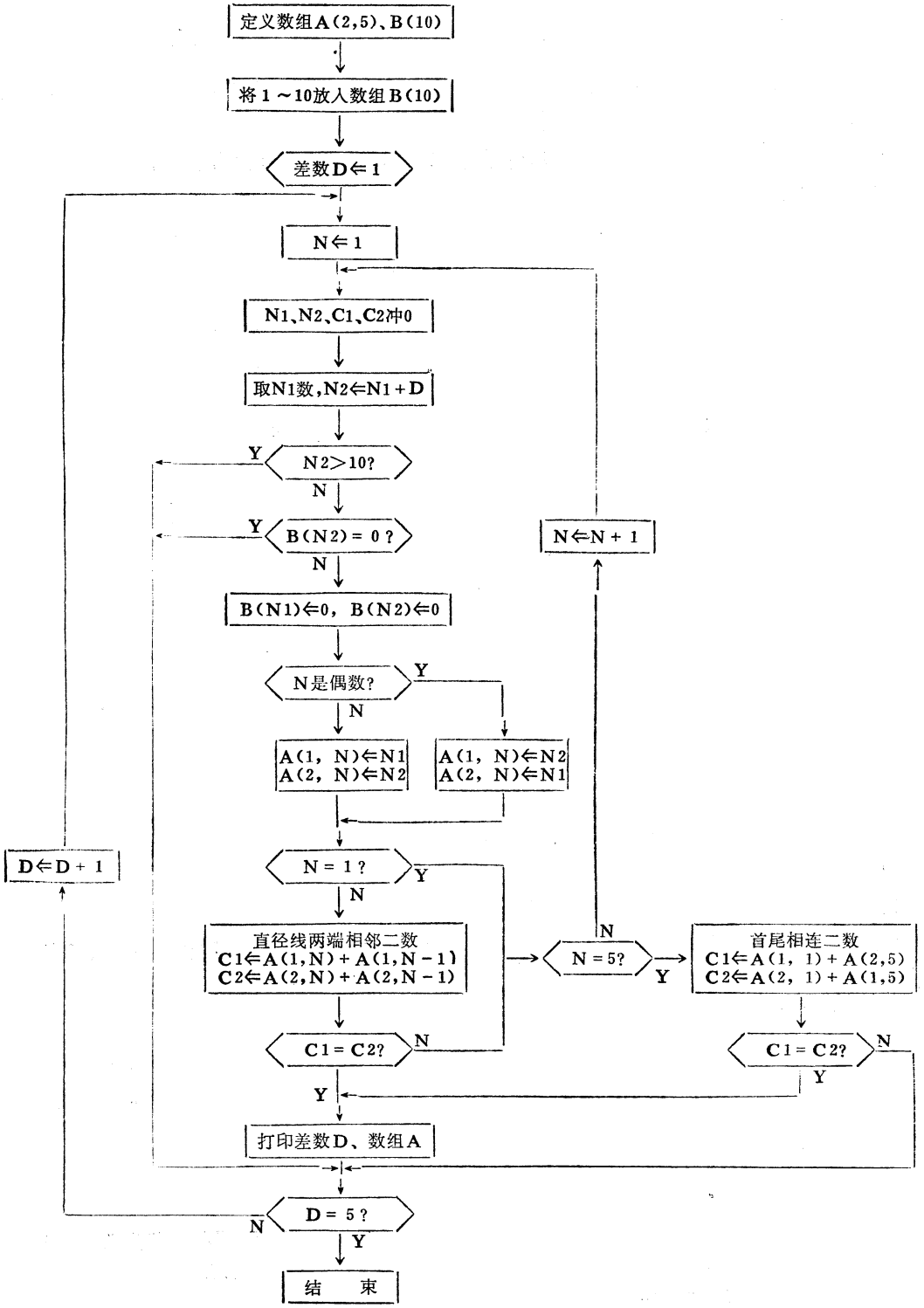

定义数组A(2,5)、B(10)
将1～10放入数组B(10)
差数D⇐1
N⇐1
N1、N2、C1、C2冲0
取N1数,N2⇐N1+D
N2>10?
Y
N
B(N2)=0?
Y
N
B(N1)⇐0, B(N2)⇐0
N是偶数?
Y
N
A(1, N)⇐N1
A(2, N)⇐N2
A(1, N)⇐N2
A(2, N)⇐N1
N=1?
Y
N
直径线两端相邻二数
C1⇐A(1,N)+A(1,N-1)
C2⇐A(2,N)+A(2,N-1)
C1=C2?
N
Y
N=5?
N
Y
首尾相连二数
C1⇐A(1, 1)+A(2,5)
C2⇐A(2, 1)+A(1,5)
C1=C2?
N
Y
N⇐N+1
D⇐D+1
打印差数D、数组A
D=5?
N
Y
结 束

图 87-4

```
      N2 = 0
      C1 = 0
      C2 = 0
      DO 20 J = 1, 10
      IF (B(J) . EQ. 0) GOTO 20
      N1 = B(J)                                   先找数N1
      GOTO 25
20    CONTINUE
      GOTO 100
25    N2 = N1 + D                                 N1数另一端数为N2
      IF (N2. GT. 10) GOTO 100
      IF (B(N2) . EQ. 0) GOTO 100
      B(N1) = 0
      B(N2) = 0
      IF (N/2*2. EQ. N) GOTO 30                   判N为奇、偶数,不同处理。
      A(1, N) = N1                                N为奇数，小的数在上行。
      A(2, N) = N2
      GOTO 40
30    A(1, N) = N2                                N为偶数，大的数在上行。
      A(2, N) = N1
40    IF (N. EQ. 1) GOTO 80
      C1 = A(1, N) + A(1, N-1)                    直径线上两端相邻二数。
      C2 = A(2, N) + A(2, N-1)
      IF (C1. NE. 02) GOTO 100
80    CONTINUE
      C1 = A(1, 1) + A(2, 5)
      C2 = A(2, 1) + A(1, 5)
      IF (C1. NE. C2) GOTO 100
      WRITE (10, 6) D, ((A(I, J), J = 1, 5), I = 1, 2)
6     FORMAT (12X, 3HD = :, I1, 10X, 5I10/18X, 5I10)
100   CONTINUE
      STOP
      END
```

程序运行后打印结果

```
D = 1 :   1    4    5    8    9      差数为 1 的数对
          2    3    6    7   10
D = 5 :   1    7    3    9    5      差数为 5 的数对
          6    2    8    4   10
```

（三）思 考 题

按图87-5顺时针方向，把九个数字分成三段，组成三个数。这三个数居然是一个乘法等式：$28 \times 157 = 4396$

请根据此规定找出图87-6乘法等式（乘积值并不限为4396），编写程序。

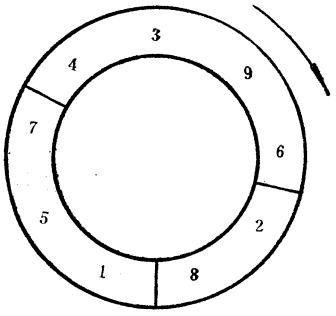


图 87-5

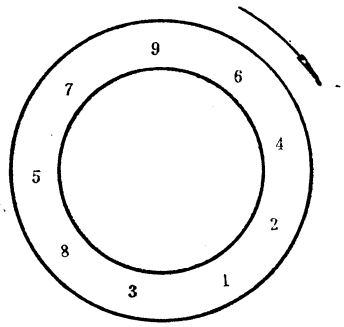


图 87-6

88 巧 妙 填 数

将 1 ～ 9 这九个数字填入九个空格子中。这样，每一横行的三个数字组成一个三位数。如果要使第二行的三位数是第一行的两倍，第三行的三位数是第一行的三倍，应怎样填数。如图88-1。

（一）算法分析和结论

由于第三排是第一排的三倍，所以第一排的第一位只能在数 1 ～ 3 中间选，第一排其余二个数可任选。

首先取第一排的一种组合，然后分别求出其值的二倍、三倍，并检查二倍、三倍的数中是否符合要求。若符合要求，则为一组摆法，否则另选一种第一排的组合。

如此找法可有如下几种填法：

1	9	2
3	8	4
5	7	6

图 88-1

2	7	3
5	4	6
8	1	9

图 88-2

2	1	9
4	3	8
6	5	7

图 88-3

3	2	7
6	5	4
9	8	1

图 88-4

（二）程 序 设 计

1）设计思路

FORTRAN 语言一般不具有位处理功能，因此要稍加处理。本程序设计是为了将数字填成图88-1和图88-2形式。设第一排的三个数分别取 I1、I2、I3，则三个数形成的三位数的值为 A = 100 * I1 + 10 * I2 + I3。

若 MA = 2 * A，即第一排的二倍，则可用下述方法分别取出其中的三个数字。

X1 = MA - INT (MA/10) * 10　　（个位数）

X2 = (MA - INT (MA/100) * 100 - X1)/10　　（十位数）

X3 = INT (MA/100)　　（百位数）

求出后可分别与I1、I2、I3比较以及相互间比较。若有相同者，则要调整I1、I2、I3的值，若无相同者，则为一种摆法。

本程序中设置三重循环。第一重循环用于选取百位数的值，可在 1 ～ 3 中取；第二重循环用于选取十位数的值；第三重循环选取个位数，然后安算法和设计思路中的方法进行比

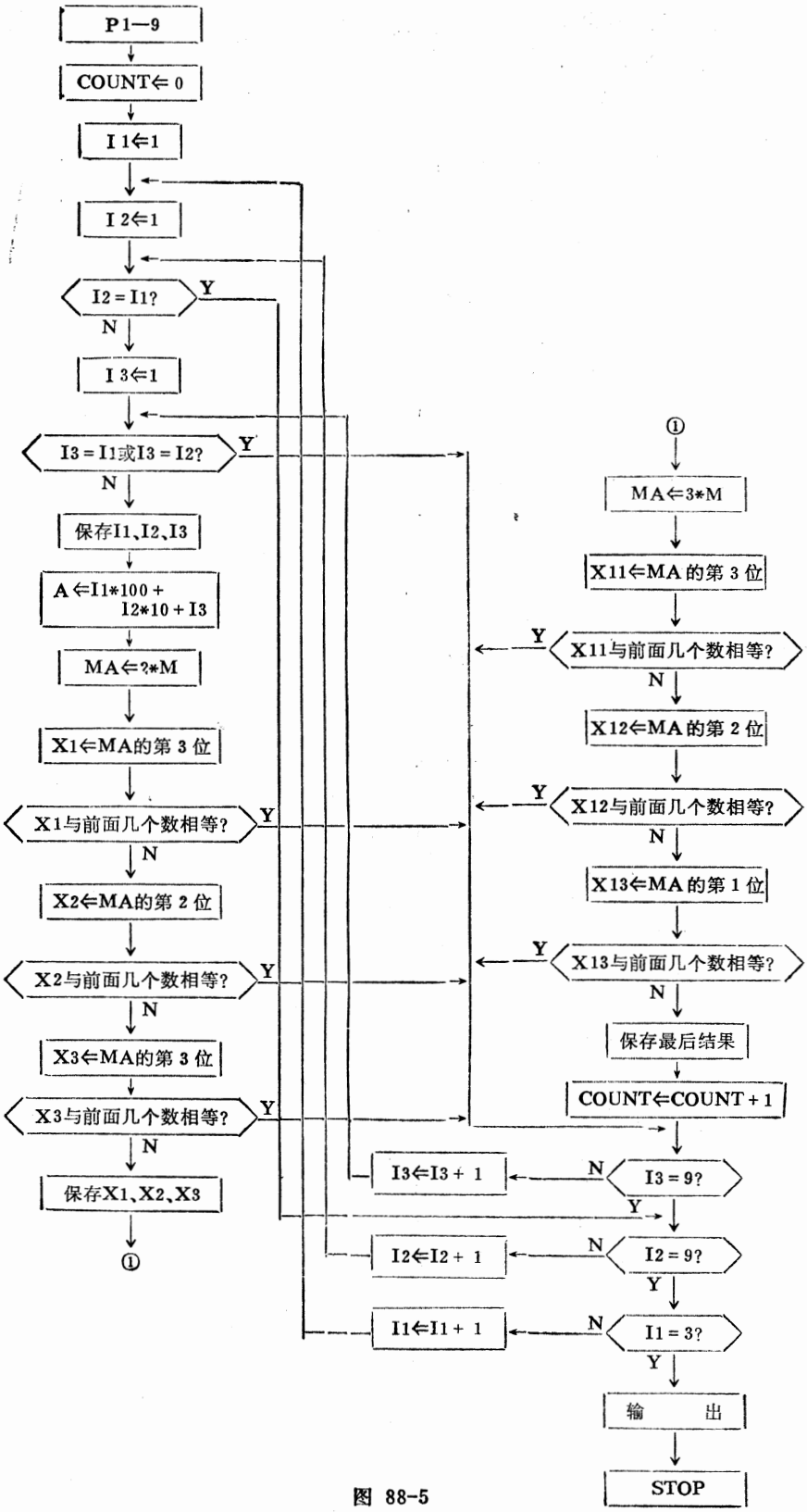

P1—9
COUNT⇐0
I1⇐1
I2⇐1
I2=I1?
Y
N
I3⇐1
I3=I1或I3=I2?
Y
N
保存I1、I2、I3
A⇐I1*100+I2*10+I3
MA⇐?*M
X1⇐MA的第3位
X1与前面几个数相等?
Y
N
X2⇐MA的第2位
X2与前面几个数相等?
Y
N
X3⇐MA的第3位
X3与前面几个数相等?
Y
N
保存X1、X2、X3
①
①
MA⇐3*M
X11⇐MA的第3位
X11与前面几个数相等?
Y
N
X12⇐MA的第2位
X12与前面几个数相等?
Y
N
X13⇐MA的第1位
X13与前面几个数相等?
Y
N
保存最后结果
COUNT⇐COUNT+1
I3=9?
N
Y
I3⇐I3+1
I2=9?
N
Y
I2⇐I2+1
I1=3?
N
Y
I1⇐I1+1
输　出
STOP

图 88-5

较。找到一种摆法后，则放入数组RE中，且摆法计数器加1。

2）框图　见图88-5

3）源程序及运行结果

```
C  ***  PROGRAM NAME: P1—9
C  ***
       INTEGER I1, I2, I3, A, M, X1, X2, X3, COUNT, MA,X11, X12, X13,
       INTEGER RE (20, 3)
       COUNT = 0
       DO 10 I1 = 1, 3
       DO 100 I2 = 1, 9
       IF (I2. EQ. I1) GO TO 100
       DO 200 I3 = 1, 9
       IF ((I3. EQ. I1). 0R. (I3. EQ. I2) ) GO TO 200
       RE (COUNT* 3 + 1, 1) = I1
       RE (COUNT* 3 + 1, 2) = I2
       RE (COUNT* 3 + 1, 3) = I3
       A = I1*100 + I2*10 + I3
C  ***  CHECK 2 *A AND 3 *A
       MA = 2 *A
C  ***  THIRD BIT OF 2 *A
       X1 = MA - INT (MA/10) *10
       IF ( (X1. EQ. I1) . OR. (X1. EQ. I2) . OR. (X1. EQ. I3) . OR.
           (X1. EQ. 0) ) GO TO 200
C  ***  SECOND BIT
       X2 = (MA - INT (MA/100) *100 - X1) /10
       IF ((X2. EQ. I1). OR. (X2. EQ. I2) OR. (X2. EQ. I3) OR. (X2. EQ.
           0)) GO TO 200
C  ***  FIRST BIT
       X3 = INT (MA/100)
       IF ((X3. EQ. I1) . OR. (X3. EQ. I2) . OR. (X3. EQ. I3)) GO TO 200
       IF ((X1. EQ. X2) . OR. (X1. EQ. X3) . OR. (X2. EQ. X3) . OR.
           (X3. EQ. 0)) GO TO 200
       RE (COUNT* 3 + 2, 1) = X3
       RE (COUNT* 3 + 2, 2) = X2
       RE (COUNT* 3 + 2, 3) = X1
       MA = 3 *A
C  ***  THIRD BIT OF 3 *A
       X11 = MA - INT (MA/10) *10
       IF ((X11. EQ. I1) . OR. (X11. EQ. I2) . OR. (X11. EQ. I3. ) . OR.
           (X11. EQ. X1) . OR.
     1   (X11. EQ. X2) . OR. (X11. EQ. X3) . OR. (X11. EQ. 0)) GO TO 200
C  ***  SECOND BIT
```

```
      X12= (MA-INT (MA/100) *100-X1) /10
      IF ((X12. EQ. I1) . OR. (X12. EQ. I2) . OR. (X12. EQ. I3) . OR.
          (X12. EQ. X1) . OR.
     1   (X12. EQ. X2) . OR. (X12. EQ. X3) . OR. (X12. EQ. 0)) GO TO 200
C  ***  FIRST BIT
      X13=INT (MA/100)
      IF ((X13. EQ. I1) . OR. (X13. EQ. I2) . OR. (X13. EQ. I3) . OR.
          (X13. EQ. X1) . OR.
     1   (X13. EQ. X2) . OR. (X13. EQ. X3) . OR. (X13. EQ. 0)) GO TO 200
      IF ( (X11. EQ. X12) . OR. (X11. EQ. X13) . OR. (X12. EQ. X13) )
         GO TO 200
      RE (COUNT* 3 + 3, 1) =X13
      RE (COUNT* 3 + 3, 2) =X12
      RE (COUNT* 3 + 3, 3) =X11
      WRITE (1, 199) I1, I2, I3, X3, X2, X1, X13, X12, X11
199   FORMAT (5X,I2, 2X, I2, 2X, I2, 2X, I2, 2X, I2, 2X, I2, 2X, I2, 2X,
        I2, 2X, I2)
      COUNT=COUNT+1
200   CONTINUE
100   CONTINUE
 10   CONTINUE
      DO 300 M=1, COUNT
      WRITE (1, 310) (RE(M* 3-2, J), J=1, 3)
      WRITE (1, 310) (RE(M* 3-1, J), J=1, 3)
      WRITE (1, 310) (RE(M* 3, J), J=1, 3)
310   FORMAT (5X, 3I5)
300   CONTINUE
350   STOP
      END
```

```
1    9    2
3    8    4      } 摆法 1
5    7    6

2    7    3
5    4    6      } 摆法 2
8    1    9
```

（三）思 考 题

请编写程序将图88-3和图88-4的填法打印出。

十三　三对新婚夫妻蜜月旅行

89　皇后王子公主逃命

在许多经典程序的难题中，路易斯·卡罗尔最得意的一个题目如下。

皇后和她的儿子、女儿被监禁在一座高大城堡的最高一间房子里，窗外有一个滑车，滑车上有一根绳索，绳索两端各有一只箩筐，两箩筐重量相等。窗外的箩筐是空的，地上的箩筐里有一块重量为30公斤的石头，石头当做平衡锤。

滑车有足够大的摩擦力，在箩筐里如被放下的任何一个人，只要其体重比另一箩筐恰好重 6 公斤，则箩筐开始滑动，滑动时是绝对安全的。如果重量差超过 6 公斤，那么他们下降的速度会使他们在城堡脚下的箩筐与地面碰撞过猛而受伤。当然，一个箩筐向下降落时，另一个箩筐就升高至窗口。

皇后的体重是78公斤，公主重42公斤，王子重36公斤，那么他们能安全到达地面的最简单（即指出最少的步数）的办法是怎样安排的？

箩筐的大小足够容纳任何两个人，或者是一个人和那块石头。没人协助他们逃走，他们也无法通过拖曳绳子而逃离。换句话说，只是在箩筐的重量超过另一箩筐的重量时，滑车才能滑动。

（一）笔算步骤和结果

对这一问题作一模拟分析，最少九次可以使他们三人都安全滑到地面，其做法是：

1）王子滑下，石头上来；

2）公主滑下，王子上来；

3）石头滑下；

4）皇后滑下，石头与公主上来；

5）石头滑下；

6）王子滑下，石头上来；

7）石头滑下；

8）公主滑下，王子上来；

9）王子滑下，石头上来。

（二）程 序 设 计

1）设计思路

定义数组A（4）、B（4），用数据初值语句将四个重量分别置入数组B，暂存之，又赋值给数组A。但在NOVA/4机调试时，因出现一维数组下界下溢，数组元素变为零的现象，故在该机定义了数组元素下限为零（见源程序）。开始前三人全在上边，用数组A里的数值表示，石头在下边，用A（4）冲0表示。以后，由上边下来，便将该数组元素冲成0，若又需要该数组元素上去（由地面上城堡），再赋给原数值。数组K1（2）、K2（2），分别表示上下二筐，每筐可容二物，故每筐用两个数组元素表示。上下筐装物时，依据上下筐重量之差等于6来进行。人为适当干预，以免死循环。每上下一次计数器KK加1，打印出上下各是谁，（用数字1、2、3、4分别表示皇后、公主、王子、

石头）。九次完成，见打印结果。

2）框图　见图89

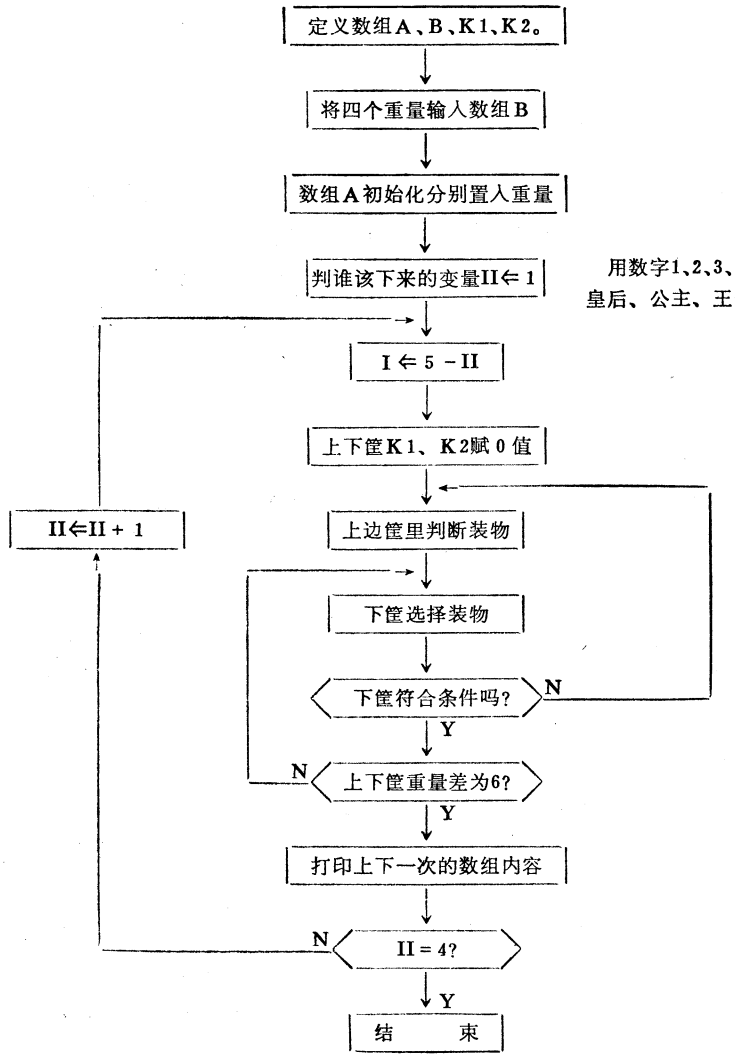


图 89

3）源程序及运行结果

```
C       PROGRAM QPPB3
        INTEGER A, B
        DIMENSION A(0 : 4) , B(0 : 4), K1
            (2) , K2(2)                    NOVA/4机器的数组说明，A、B装重物，
                                           K1、K2分别为上下筐。
        COMMON/QP/B
        DATA B/ 0 , 78, 42, 36, 30/        将四个重量和0放入数组B
```

```
      DO 10 I= 0, 3                                  }
10    A(I) = I                                       } 初始化
      A(4) = 0                                       }
      KK = 0                                         上下次数计数器KK
20    DO 70 II= 1, 4
      I= 5 -II
      DO 25 J= 1, 2
      K1(J) = 0                                      }
25    K2(J) = 0                                      } 上下筐清 0
      IF (A(I) . NE. 0. AND. A(I) . NE.
          4) K1(1) =I                                装上筐
      DO 70 I1= 1, 4                                 DO 70是下筐选择装何物
      IF (KK. LT. 8) GOTO 26
      K2(1) = 4
      GOTO 50
26    IF(A(I1) . EQ. 0) K2(1) =I1
      IF (I. EQ. 1. AND. A(1) . NE. 0) GOTO 30
      GOTO 50
30    DO 40 J= 1, 4                                  }
      IF (. NOT. (A(J) . EQ. 0. AND. J.              } 若肥重的皇后下来,
          NE. I)) GOTO 40                            } 下筐需装二物与她平衡。
      K2(2) =J                                       }
      GOTO 50                                        }
40    CONTINUE
50    KA=B(K1(1)) +B(K1(2))                          }
      KB=B(K2(1)) +B(K2(2))                          } 判上下筐装物后是否平衡
      IF (IABS (KA-KB) . EQ. 6) GOTO 80              }
      IF ( (K1(1) . EQ. 4) . AND. (K2(1) . EQ. 0)) GO TO 80
                                                     若不平衡，先换下筐物，若仍不满足条
                                                     件，再换上筐物。
70    CONTINUE
      GOTO 120
80    WRITE (10, 98) K1, K2                          }
98    FORMAT (1X, 2I3, 6H    DOWN, 12X,              } 打印满足条件的上下筐里装的物
          2I3, 4H    UP)                           }
      DO 96 L= 1, 2
      IF (K1(L) . NE. 0) A(K1(L)) = 0                谁下来，便置 0，即该数组元素 A 冲
                                                     0。
      IF (K2(L) . NE. 0) A(K2(L)) =K2(L)
                                                     若又需上去，便赋值，仍用数组元素
                                                     A。
96    CONTINUE
      KK=KK+1
```

```
      IF (KK. GE. 10) GOTO 120
      IF (KK/2*2. EQ. KK. AND. KK. NE. 8) GOTO 200
```

算法分析中3、5、7次是石头下，执行过程中是偶数4、6次，石头上去。

```
      IF (KK. EQ. 3) GOTO 210
      DO 110 M=1, 3
      IF (A(M) . NE. 0) GOTO 20
110   CONTINUE
      GOTO 120
200   K1(1) = 4
      K1(2) = 0
      K2(1) = 0
      K2(2) = 0
      GOTO 80
210   K1(1) = 1
      K1(2) = 0
      K2(1) = 2
      K2(2) = 4
      GOTO 80
120   STOP
      END
```

判全下完否

下面是程序运行时，依次打印结果

上筐				下筐			
3	0	DOWN	(王子3，下)	4	0	UP	(石头4，上)
2	0	DOWN	(公主2，下)	3	0	UP	(王子3，上)
4	0	DOWN	(石头4，下)	0	0	UP	(空筐，上)
1	0	DOWN	(皇后1，下)	2	4	UP	(公主、石头，上)
4	0	DOWN	(石头4，下)	0	0	UP	(空筐，上)
3	0	DOWN	(王子3，下)	4	0	UP	(石头4，上)
4	0	DOWN	(石头4，下)	0	0	UP	(空筐，上)
3	0	DOWN	(公主2，下)	3	0	UP	(王子3，上)
3	0	DOWN	(王子3，下)	4	0	UP	(石头4，上)

STOP

（三）思 考 题

若皇后、王子、公主既不舍命，又不舍财，除了皇后、王子、公主、石头外，在城堡上，还有她们豢养的玩物，一头重24公斤的狗，一只18公斤的猴子，和一篮子12公斤的猫。对两个箩筐之间重量差的限制同主题目规定的一样，不过现在在城堡和地面必须都要有人，以帮助这些动物进、出箩筐。箩筐足够容纳所需的重物。你能否找到少至12步的解法？或更多步的解法，编写程序，打印输出结果。要注意的是，上述这两个问题，从箩筐里走出的最后那个人，必须迅速地走开，否则，箩筐和石头会碰在他（她）们的头上！

90 猎人监送狗兔白菜

猎人冬森携带猎犬狩猎，某日捕获一只野兔。猎人顺便买了棵大白菜，来到家门前的小渡河。他狩猎时将船系在岸边，但船太小，一次只能带一样，因为狗能吃兔，兔要吃白菜，所以狗和兔、兔和白菜不能在无人监视的情况下留在一起，请你设法使猎人既要将三者完整无损渡河带回家，又要往返监送次数最少。

（一）笔算步骤和结果

分析后可以采用下述方案，往返七次即可。

1）猎人把兔从本岸送到对岸；

2）猎人划空船返回；

3）猎人将菜送到对岸；

4）猎人再把兔放上船返回本岸；

5）猎人将狗送到对岸；

6）猎人划空船返回；

7）猎人再把兔送到对岸。

至此，猎人将三者监送安全到家，美餐一顿。

若不限次数，还可用其它方案完成，如下图90-1所示，图中数字 1 ～ 7 是如上所求出的最短路径（七次），图中还可依箭头所示找出其它路径。

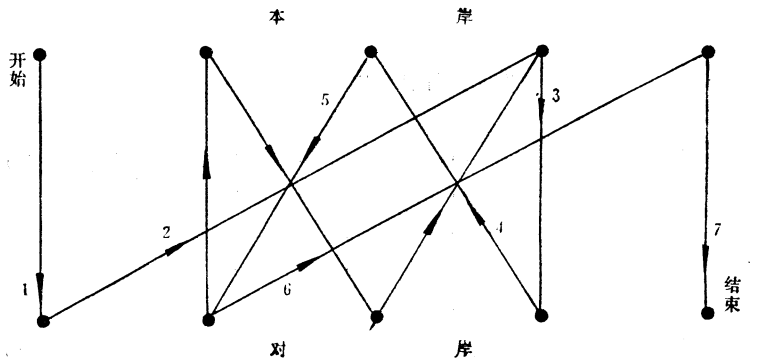


图 90-1

（二）程 序 设 计

1）设计思路

用数字来表示三者：狗为 1，兔为 2，菜为 3，它们每两个数之间的数量差若为 1，则它们不相容，如果为 2，则相容。用数组 IA 表示河的本岸，以数组 IB 表示河的对岸，以暂存单元IC表示小船。从数组IA出发，取出一个数到IC，看剩下的数是否相容，如果不相容则取另外的数到IC，如果相容，便将IC中的数送给IB，到IB后看IB中是几个数，如果IB中是一个数，将IC冲为零。再重复开始的步骤取下一个IA 中的数。如果 IB 中是两个不相容的数就将原在IB中的数给 IC，IC 再给 IA，再重复开始的步骤，取下一个IA的数。如果IB中是三个数，则已安全渡河。

为了记录过河的步骤，我们将每一次成功（即留在IA中的数和留在IB中的数都是相容的）的IC的结果送入数组IP，在结束之前打印出IP，可观察往返情况。

2）框图　见图90-2

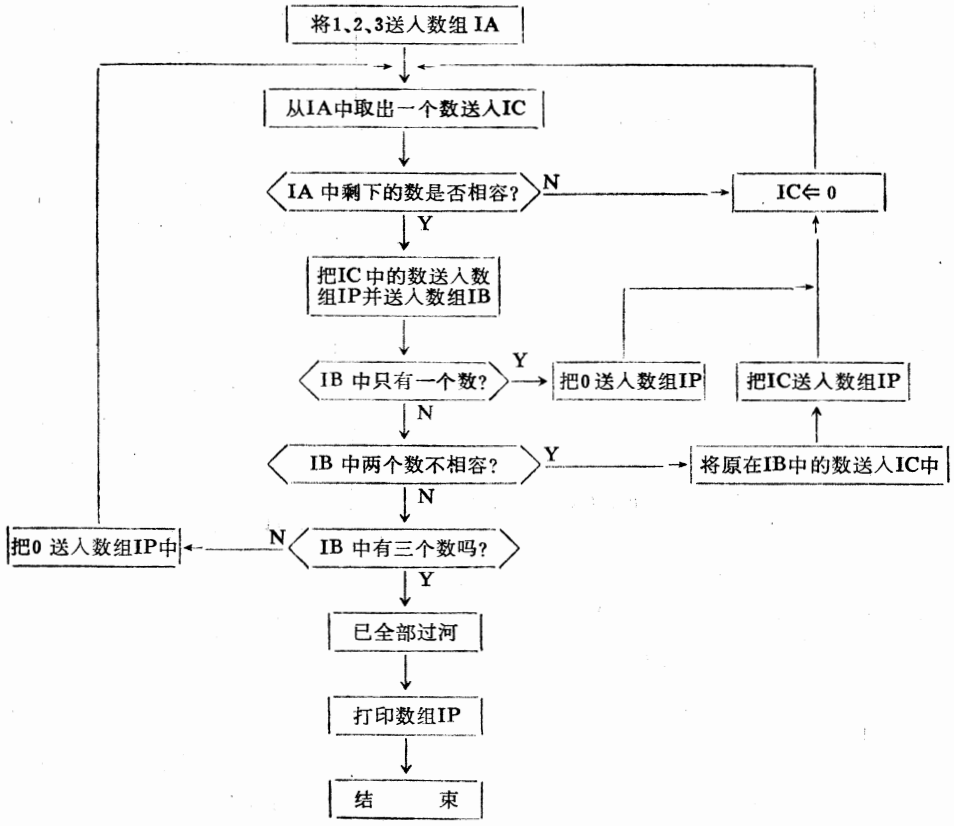


图 90-2

3）源程序及运行结果

```
C       PROGRAM DOG. RABBIT. CABBAGE
        DIMENSION IA(3) , IB(3) , ID(3) , IG(3) , IP(10) , P(6)
        REAL PP*8
        DATA P(1) /' (7X'/, P(2) /', '/, P(3) /'A3'/, P(4) /', '/, P(5)
           /'A7'/
     1  , P(6) /') '/, IA/1, 2, 3/
        N = 0
15      DO 75 I = 1, 3                          I = 1、2、3 分别表示狗、兔、白
                                                菜
        K = 0
        IF (IA(I) . EQ. 0) GO TO 75
        IC = IA(I)
```

```
      DO 60 J = 1, 3
      IF (J. EQ. I. OR. IA(J). EQ. 0) GO
         TO 60
      K = K + 1
      ID (K) = IA(J)
60    CONTINUE
      IF (K. NE. 2. OR. ID(2) - ID(1).
         EQ. 2) GO TO 85
75    CONTINUE
      GO TO 15
85    IA(I) = 0
      IX = I
      N = N + 1
      IP(N) = IC
      IB(IX) = IC
      M = 0
      DO 130 IW = 1, 3
      IF (IB(IW) . EQ. 0) GO TO 130
      M = M + 1
      IG(M) = IB(IW)
130   CONTINUE
      IF (M. EQ. 1) GO TO 145
      GO TO 170
145   IC = 0
      N = N + 1
      IP(N) = IC
      IE = IB(IX)
      IY = IX
      GO TO 75
170   IF (M. EQ. 3) GO TO 230
      IF (IG(2) - IG(1) . EQ. 2) GO TO 215
      IC = IE
      N = N + 1
      IP(N) = IC
      IA(IY) = IE
      IE = IB(IX)
      IB(IY) = 0
      IY = IX
      GO TO 75
215   IC = 0
      N = N + 1
      IP(N) = IC
      GO TO 75
```

从本岸IA里取走一个判 IA 里剩余者相容否

85句是将IA取走者冲 0

由IA送入IC船上

渡到对岸IB

辨别IB里有几个即M等于几

对岸 IB里有 1 个则 IC 船冲 0，表示空返回。

判IB里有三个吗

判IB里两个相容否

IB里两个不相容，送一个上IP船，返回。

IB里两个相容，IC冲0，即船空返回。

```
230       PRINT*, 'NUMEBER:', N
          DO  260  IU=1, N
          IF (IU/2.EQ.IU/2.0) GO TO 250判奇偶数
          A='-->'                    奇数去对岸
          GO  TO  255
250       A='<--'                    偶数返回
255       IF (IP(IU) .EQ.0) PP='0'
          IF (IP(IU) .EQ.1) PP='DOG'
          IF (IP(IU) .EQ.2) PP='RABBIT'
          IF (IP(IU) .EQ.3) PP='CABBAGE'
260       WRITE (6, P) A, PP
          STOP
          END
```

IB里有三个，即全部渡到对岸。打印全过程

运行结果打印如下

```
NUMEBER:        7                    往返7次
          -->PABBIT                  兔往对岸
          <--0                       船空返回
          -->CABBAGE                 白菜送往对岸
          <--RABBIT                  兔返回
          -->DOG                     狗往对岸
          <--0                       船空返回
          -->RABBIT                  兔往对岸
```

（三）思　考　题

戴尔公爵非常喜爱赛马表演，可他性情怪癖。有一次，他把四匹白马布置在赛马场的南端，把四匹黑马布置在北端，然后命令：

（1）每次只许从两端共牵出五匹马来表演，既不能多，也不能少；

（2）表演后南端马牵往北端，北端马牵往南端。

如果按照戴尔公爵这个奇特的规定，最少需要表演几次才能使四匹白马都到北端，而四匹黑马都到南端？

提示：最少表演四次才能使黑白马位置对换，以A_1、A_2、A_3、A_4表示白马，B_1、B_2、B_3、B_4代表黑马，表演方法之一是：

	南端	北端
原始位置	$A_1A_2A_3A_4$	$B_1B_2B_3B_4$
第一次	$A_1A_2B_2B_3B_4$	$B_1A_3A_4$
第二次	B_1B_2	$A_3A_4A_1A_2B_3B_4$
第三次	$A_1A_2B_4$	$A_3A_4B_1B_2B_3$
第四次	$B_1B_2B_3B_4$	$A_1A_2A_3A_4$

91　跳　　棋

将三只黑子放在1、2、3方格中，3只白子放在5、6、7方格中，如图91-1。利用空格4，将白子移到黑子位置，黑子移到白子位置。同时应遵守规则：棋子可以走入邻

接的空格中，也可以跳到邻接对方棋子后面的空格。黑白两种棋子只能向前走，不能后退反向走。问走多少步便可互换位置满足要求？

1	2	3	4	5	6	7
×	×	×		○	○	○

图 91-1

（一）笔算步骤和结果

试跳后，15步便可互换位置。如图91-2所示，设白子（用 0 表示）只能往上走，黑子（用×表示）只能往下走，（因为条件上规定两种子只能往相对方向走）。走法按图91-2先后次序：黑白白黑黑黑白白白黑黑黑白白黑（共15步）。

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
×	×	×	×	×	×		0	0	0	0	0	0	0	0	0
×	×	×	×	×		×	×	×	×	×	×		0	0	0
×		0	0	0	0	0		0	0	0	0	0		0	0
	×	×	×		×	×	×	×	×	×		×	×		
0	0		0	0	0	0	0		0	0	0	0	0		×
0	0	0		×	×	×	×	×	×		×	×	×	×	×
0	0	0	0	0	0	0	0	0		×	×	×	×	×	×

图 91-2

（二）程 序 设 计

1）设计思路

编程序模拟图91-2的走法。用数字 1 表示黑子“×”，2 表示白子“0”。将竖着走的图91-2旋转成横着走的图91-3形式，定义数组 D（11）表示11格，首尾各两个空格是算法需要而设的。总结图91-2跳棋规律如下：

1	2	3	4	5	6	7	8	9	10	11
		1	1	1		2	2	2		

前　　后

图 91-3

先令 1 子走一步；

（1）判空位，再判何子能跳到空位，跳优先于走，跳后置标记，1 子跳后，标记 J 1⇐1，2 子跳后，标记 J 2⇐1；

（2）不能跳时，若两种棋子都能走入空位时，如标记 J 1 为 1，则 1 子走，若标记 J 2 为 1，则 2 子走；

（3）不能跳、又只有一种子能走时，当然该子则走。

编两个子程序，子程序 P T B 是判跳步，子程序 P S Z 是判谁走。每跳或走一步，都输出打印，15步走完，便全打印出，请见源程序运行后打印结果。

2）框图　见91-4

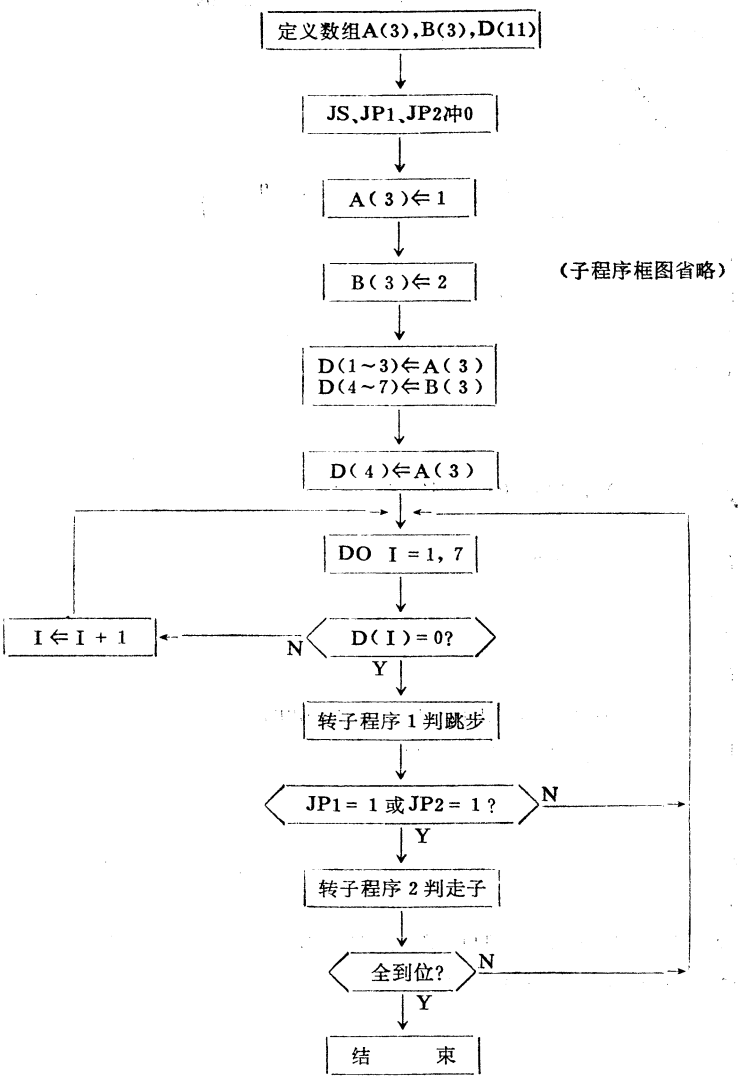


图 91-4

3）源程序及运行结果

```
C       PROGRAM  TQ
        INTEGER  D(11)
        JS=0
        J11=0
        J22=0
        DO  5  I=1, 11
 5      D(I) =0
        DO  10  I=3, 5
10      D(I) =1
        DO  20  I=7, 9
```

```
 20      D(I) =2
         WRITE  (10, 21)  JS,  (D(K) , K=3, 9)
 21      FORMAT (/3X, I2, 1H:, 7I3)
         D(6) =D(5)
         D(5) =0
         JS=JS+1
         WRITE (10, 21) JS,  (D(K) , K=3, 9)
 25      IF  (D(6) . NE. 0)  GOTO  35
         DO  30  J=3, 5
         IF  (D(J) . NE. 2)  GOTO  35
         IF  (D(J+4) . NE. 1)  GOTO  35
 30      CONTINUE
         STOP
 35      DO  40  I=3, 9
         IF  (D(I) . EQ. 0)  GOTO  45
 40      CONTINUE
         STOP  "ERR 1"
 45      J1=0
         J2=0
         MT=0
         CALL  PTB (D, I, J1, J2, JS, MT)
         IF (MT. EQ. 1) GOTO  55
         J11=J1
         J22=J2
         GOTO  25
 55      L=0
         CALL  PSZ (D, I, J11, J22, JS, L)
         IF (L. EQ. 1) STOP  'ERR 2'
         GOTO  25
         END
         SUBROUTINE  PTB (D, I, J1, J2, JS, MT)          判跳步子程序
         INTEGER  D (11)
         IF  (D(I-2) . EQ. 1. AND. D(I-1) . EQ. 2)  GOTO  2
         GOTO  10
 2       D(I) =D(I-2)
         D(I-2) =0
         J1=1
         JS=JS+1
 3       WRITE  (10, 5)  JS,  (D(K) , K=3, 9)
 5       FORMAT (/3X, I2, 1H:, 713)
         GOTO  20
 10      IF  (D(I+2) . EQ. 2. AND. D(I+1) . EQ. 1) GOTO  12
         GOTO  15
```

```
12      D(1) =D(I+2)
        D(I+2) =0
        J2=1
        JS=JS+1
        GOTO 3
15      MT=1
20      RETURN
        END
        SUBROUTINE PSZ (D, L, J11, J22, JS, L)          判谁走的子程序
        INTEGER D (11)
        IF (D(I-1) . NE. 1) GOTO 10
        IF (D(I+1) . NE. 2) GOTO 30
        IF (J11. EQ. 1) GOTO 30
        IF (J22. EQ. 1) GOTO 40
        L=1
        GOTO 20
10      IF (D(I+1) . EQ. 2) GOTO 40
        L=1
        GOTO 20
30      D(I) =D(I-1)
        D(I-1) =0
        JS=JS+1
33      WRITE (10, 35) JS, (D(K) , K=3, 9)
35      FORMAT (/3X, I2, 1H:, 7I3)
        GOTO 20
40      D(I) =D(I+1)
        D(I+1) =0
        JS=JS+1
        GOTO 33
20      RETURN
        END
```

程序运行过程的打印输出

```
0 :  1  1  1  0  2  2  2
1 :  1  1  0  1  2  2  2
2 :  1  1  2  1  0  2  2
3 :  1  1  2  1  2  0  2
4 :  1  1  2  0  2  1  2
5 :  1  0  2  1  2  1  2
6 :  0  1  2  1  2  1  2
7 :  2  1  0  1  2  1  2
8 :  2  1  2  1  0  1  2
9 :  2  1  2  1  2  1  0
10:  2  1  2  1  2  0  1
```

序号 0 是原始状态

序号 1 是第 1 步，以下序号都是步数。

```
11：  2  1  2  0  2  1  1
12：  2  0  2  1  2  1  1
13：  2  2  0  1  2  1  1
14：  2  2  2  1  0  1  1
15：  2  2  2  0  1  1  1
```

（三）思 考 题

幼儿园的张阿姨照料八个孩子，四个男孩，四个女孩，每天按时坐在一条长橙上，听故事。四个男孩总坐在一起，四个女孩也如此，因为同性别之间互相打闹，不安心听讲，阿姨指定他（她）们换位置，以便彼此隔开，但谁也不听。于是，张阿姨想出个主意，按往常一样，坐在能容10个人的长橙上做游戏：

相邻两个孩子手拉手走到两个空位上坐下，其余也是相邻的两个孩子手拉手走到空位上坐下。经四次调换位置，把男、女孩互相隔开，大家都笑了。

请你做程序设计，如果四次调换位置，不能把男女孩互相隔开，最少几次能做到呢？并不限次数，只要能间隔开，程序便算成功！

提示——十个位置，在一端空两个位置，每次二人手拉手走去就座，说明二人左右位置保持原样，不能倒换。设男孩用O表示，女孩用＊表示。阿姨令他（她）们四次换位过程如图91-5所示（箭头表示移动方向）：

```
原始0步   OOOO＊＊＊＊
            →
第一步    O    O＊＊＊＊OO
               ←
第二步    O＊＊O    ＊＊OO
                  ←
第三步    O＊＊O＊O＊    O
          →
第四步        ＊O＊O＊O＊O
```

图 91-5

92 三对新婚夫妻蜜月旅行

三对新婚夫妇蜜月旅行。他们来到一条河边，发现通往对岸的渡口处，只有一只可供两人乘坐的无人摆渡的小木船。在过河时岸上的三位丈夫提出：任何女人不得跟任何男人（除非自己的丈夫）乘船或留在河岸上（当然允许彼此相遇见面）。请你设计出一种渡河方案，既能在符合丈夫的要求将三对夫妇送到对岸，又要使小船来回划动的次数为最少。

（一）笔算步骤和结果

小船来回一共只要划九次，便能将三对夫妇全部渡过河去，顺利进行蜜月旅行。

具体渡法，如图92-1所示，其中男人用A、B、C表示，女人用A_1、B_1、C_1表示。箭头⇌表示移动方向，目的从左岸渡到右岸。

（二）程 序 设 计

1）设计思路

我们用双数代表妻子，如2、4、6。用单数代表丈夫，如1、3、5。2N-1与2N是一对夫妻关系（N＝1，2，3）。如1、2；3、4；5、6是三对夫妻关系。

次数	左岸	右岸
0	A A₁ B B₁ C C₁	
1	A A₁ B B₁ C C₁ →	A A₁
2	A₁ B B₁ C C₁	A A₁ ←
3	A₁ → B B₁ C C₁	A A₁ B₁
4	B B₁ C C₁	A A₁ B₁ ←
5	B B₁ → C C₁	A A₁ B B₁
6	B₁ C C₁	A A₁ B B₁ ←
7	B₁ C C₁ →	A A₁ B C C₁
8	B₁ C₁	A A₁ B C C₁ ←
9	B₁ → C₁ →	A A₁ B B₁ C C₁
10		A A₁ B B₁ C C₁

图 92-1

目的从左岸渡到右岸。设两个数组，数组L（7）表示左岸人员状况，为了程序中算法需要，右岸多设一个数组元素（7）。数组IR（6）表示右岸人员状况。先给数组L(7)赋值，L（I）⇐I （I=1，2,…6),L（7）⇐0，表示准备阶段人员全在左岸。将数组IR全冲零，表示没行动时右岸没人。令数组IG（2）表示由左岸往右岸时船上的二人，令数组IC（2）为送船返回时船上乘坐的人。开始时先令双数中一个（某妇人）上船，立即判左岸（即数组L）有否与自己相容的人（找丈夫），如果有，则丈夫也上船，同舟共济，否则（说明丈夫在右岸等着她呢!），这时另一妇女上船，数组IG容二人渡到右岸，IG上的二人谁下船，要判右岸有否自己的丈夫，有，则下船；否，则留在船上返回送船。假若IG里就有自己的丈夫，而右岸又没有不带丈夫的妻子，便令自己丈夫下船，妻子返回送船。无论哪种情况，返回的船上即数组IG总是一人，且是妇女（数字为双数）……。如此往返判断，九次便可按规则全部渡过河。每次摆渡时都形象地打印出，以箭头表示往返的方向。用J表示渡河的次数，J=0表示准备状态，打印出人员全在左岸，即数组L中有1～6人。J为1～9表示摆渡次数，J=10表示左岸已空，不用再渡，这时可由打印结果观察出右岸即数组IR有1～6人。

2）框图　见图92-2

3）源程序及运行结果

```
      DIMENSION  L(7) , IG(2) , IR(6) , IC(2)
      DO  111  I=1, 6
      L(I) =I
111   IR(I) =0
      J=0
      L(7) =0
      WRITE (6, 222) J, L
222   FORMAT (5X, 2HJ=, I2, 3X, 3HL=, 7 (I2, 5X) )
      DO  22  I=2, 20, 2
```

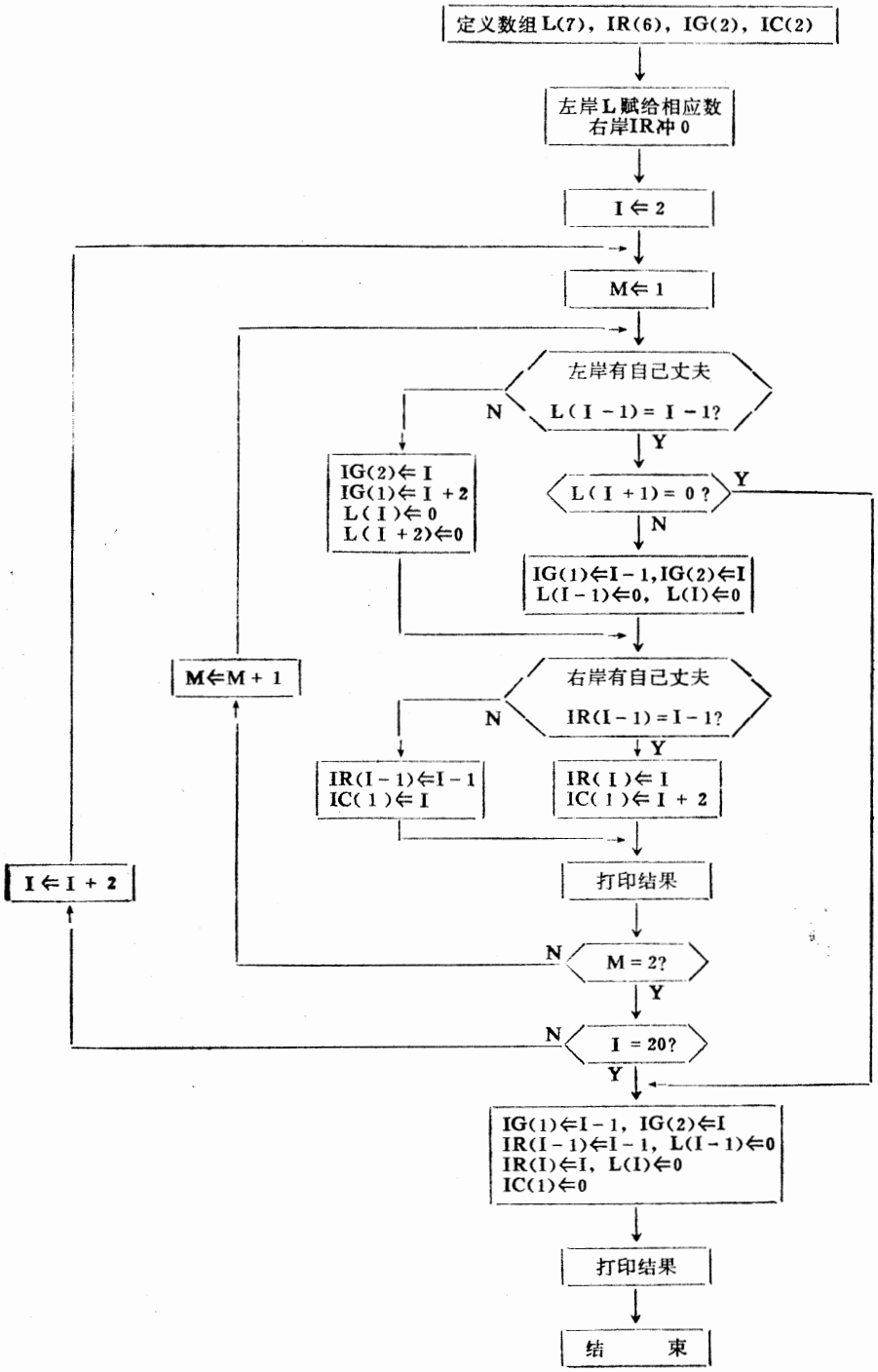

定义数组L(7)，IR(6)，IG(2)，IC(2)
左岸L赋给相应数
右岸IR冲0
I⇐2
M⇐1
左岸有自己丈夫
L(I-1)=I-1?
N
Y
IG(2)⇐I
IG(1)⇐I+2
L(I)⇐0
L(I+2)⇐0
L(I+1)=0?
Y
N
IG(1)⇐I-1,IG(2)⇐I
L(I-1)⇐0, L(I)⇐0
M⇐M+1
右岸有自己丈夫
IR(I-1)=I-1?
N
Y
IR(I-1)⇐I-1
IC(1)⇐I
IR(I)⇐I
IC(1)⇐I+2
I⇐I+2
打印结果
M=2?
N
Y
I=20?
N
Y
IG(1)⇐I-1, IG(2)⇐I
IR(I-1)⇐I-1, L(I-1)⇐0
IR(I)⇐I, L(I)⇐0
IC(1)⇐0
打印结果
结　束

图 92-2

```
      DO  11  M=1, 2
      IF (L(I-1) . EQ. I-1)  GO  TO  66
      IG(1) =I+2
      IG(2) =I
      L(I) =0
      L(I+2) =0
      GO  TO  77
66    IF (L(I+1) . EQ. O)  GO  TO  88
      IG(1) =I-1
      IG(2) =I
      L(I-1) =0
      L(I) =0
77    IF (IR(I-1) . EQ. I-1)  GO  TO  33
      IR(I-1) =I-1
      IC(1) =I
      GO  TO  44
33    IR(I) =I
      IC(1) =I+2
44    J=J+2
      N=J-1
      WRITE (6, 55) N, L, IG, J, IR, IC
11    CONTINUE
22    CONTINUE
88    IG(1) =I-1
      IG(2) =I
      IR(I-1) =I-1
      L(I-1) =0
      IR(I) =I
      L(I) =0
      IC(1) =0
      J=J+2
      N=J-1
      WRITE (6, 55) N, L, IG, J, IR, IC
55    FORMAT (/5X, 2HJ=, I2, 3X, 3HL=, 7 (I2, 5X) /12X, 3HIG=,
     1  2 (I2, 5X) , 5H---->//5X, 2HJ=, I2, 3X, 3HIR=, 6 (I2, 5X) /
     2  12X, 3HIC=, 2 (I2, 5X) , 5H<----)
      STOP
      END
```

程序运行过程打印结果如下

```
J= 0    L= 1    2    3    4    5    6    0
J= 1    L= 0    0    3    4    5    6    0
        IG= 1   2    ---->
```

J = 2	IR = 1	0	0	0	0	0	
	IC = 2	0	<----				
J = 3	L = 0	0	3	0	5	6	0
	IG = 4	2	---->				
J = 4	IR = 1	2	0	0	0	0	
	IC = 4	0	<----				
J = 5	L = 0	0	0	0	5	6	0
	IG = 3	4	---->				
J = 6	IR = 1	2	3	0	0	0	
	IC = 4	0	<----				
J = 7	L = 0	0	0	0	5	0	0
	IG = 6	4	---->				
J = 8	IR = 1	2	3	4	0	0	
	IC = 6	0	<----				
J = 9	L = 0	0	0	0	0	0	0
	IG = 5	6	---->				
J = 10	IR = 1	2	3	4	5	6	
	IC = 0	0	<----				

（三）思　考　题

有 4 个医务人员护送 4 个精神不正常的人去治疗，用一只小船过河，而船上最多能容纳 3 个人，假设两种人都可划船，但不能放空船；且在任何地方（两岸或船上），如果医务人员的人数少于患者，将被患者伤害。问怎样才能将这 8 个人安全渡过河去？

93 古老的神话

流传很久的一个古老神话——三牛三虎来到河边，船只能容 1 至 2 只（牛和虎、或单独、或同类全可）。若两岸虎多于牛则虎吃牛，如何设法使牛不受伤害而安全渡河？

（一）笔算步骤和结果

设A、B、C表示三虎，X、Y、Z表示三牛。

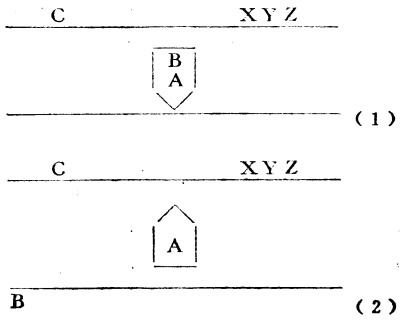
又设只有一虎A会划船，只有一牛X会划船。

渡河方案及其设想根据如下：

（1）第一步A和B过河。

（2）第二步A单独返回。

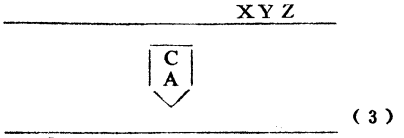
（3）第三步A和C过河。

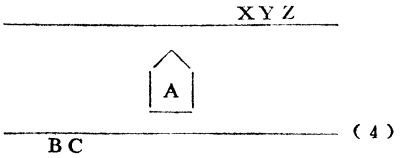


第三步显然不能是两牛过河，否则渡口会发生虎多牛少情况。如果设想为一牛一虎过河，则到了对岸，牛少于虎，也不行。因而只能是两虎过河。

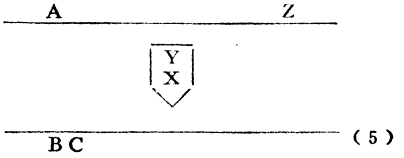
（4）第四步A单独返回。

（5）第五步X和Y过河。





显然，第五步不能是一虎一牛过河，否则到了对岸牛就少于虎。因此只有两牛（X和Y，或X和Z，这两种情况一样）过河。

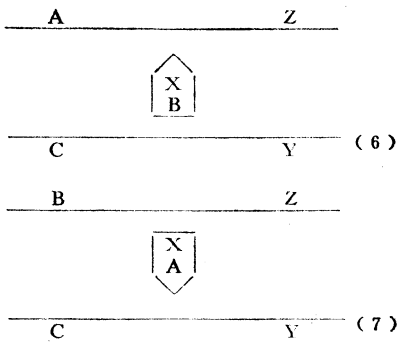


（6）第六步X和B返回。

如果X单独返回，对岸会出现两虎一牛情况，因而必须X和B（或C）一起返回。

（7）第七步A和X过河。

这一步的情况比较复杂。



（a）X和B过河，回复到第五步状态，无意义。

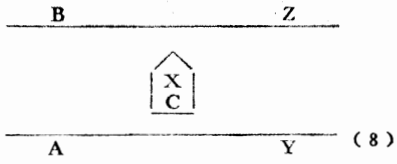
（b）B和Z过河，没有一个会划船，不行。

（c）A和B过河，到了对岸，虎多于牛，也不行。

（d）A和Z过河，设想下一步，无论是A单独返回或A和C一起返回，到渡口时虎多于牛；如为A和Y返回，只不过是Z和Y交换了一下位置，无意义。

（e）X和Z过河，设想下一步，X单独返回或X和C返回，到渡口时虎多于牛；X和Y返回，只是Y和Z交换位置无进展，因而只能是A和X过河。

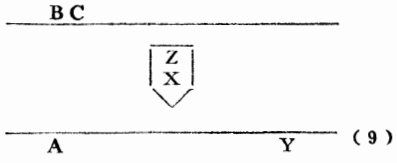
（8）第八步X和C返回。



（8）

A或X单独返回，A和C返回，都会出现虎多于牛的情况；A和Y返回，下一步只能是A和Z过河，没有进展，无意义。因而只能是X和C返回。

（9）第九步X和Z过河。



（9）

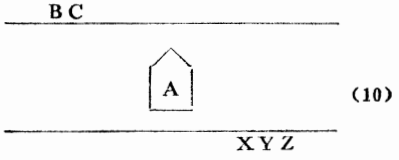
如果是X和B过河，只是B和C交换了一下位置，无意义。因而只能是X和Z过河。

（10）第十步A单独返回。

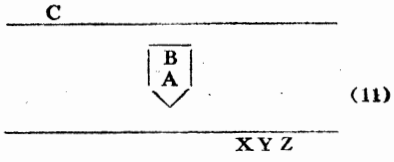
（11）第十一步A和B（或C）过河。

（12）第十二步A（或X）单独返回。

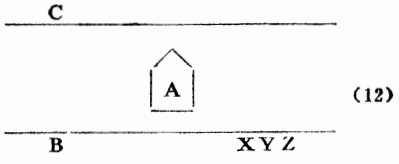
（13）第十三步A（或X）和C过河。



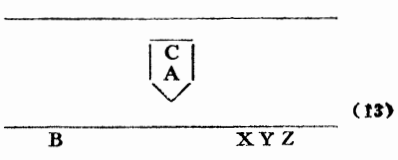
（10）



（11）



（12）



（13）

另外，设X和B（或C）先过河，可得到另一个渡河方案：

第一步X和B过河。　　第二步X单独返回。
第三步A和C过河。　　第四步A单独返回。
第五步X和Y过河。　　第六步X和B返回。
第七步A和X过河。　　第八步X和C返回。
第九步X和Z过河。　　第十步A单独返回。
第十一步A和B过河。　　第十二步A单独返回。
第十三步A和C过河。

（二）程 序 设 计

1）设计思路

用数字表示牛虎，令1、2、3表示虎，4、5、6表示牛，又设1、4会划船。定义数组A（6）为此岸，初始状态置入虎牛。数组B（6）表示彼岸，初始状态清零，数组C（2）表示小船。A（I）＝0时，表示I已过河，B（I）＝0时，表示I没过河

（还留在此岸）。每过渡一次，计数器N加1。当N为奇数时，此岸到彼岸。当N为偶数时，由彼岸返回。当N为15时，渡河结束。（因原始状态和全部牛虎过河以后各算1次。）

从A中取出一个或两个送到C中，然后，判断A中剩余的是否虎比牛少，及C中是否有一个会划船的，B＋C是否虎比牛少。如都满足，则把C送B。再从B中取一或两个送C，重复上述过程，直到牛、虎都过河为止。

为方便和通用，我们把上述的三个条件，独立放在一个子程序中，当需要判别时，调用这个子程序即可，这样改变子程序，即可解决所有相似的问题。子程序中标志单元RET为0时，可过河或返回。当RET为1时，表示不满足过河条件，返主再重选过河者。过河时尽量船载两个，返回时尽量载一个。

2）框图　见图93

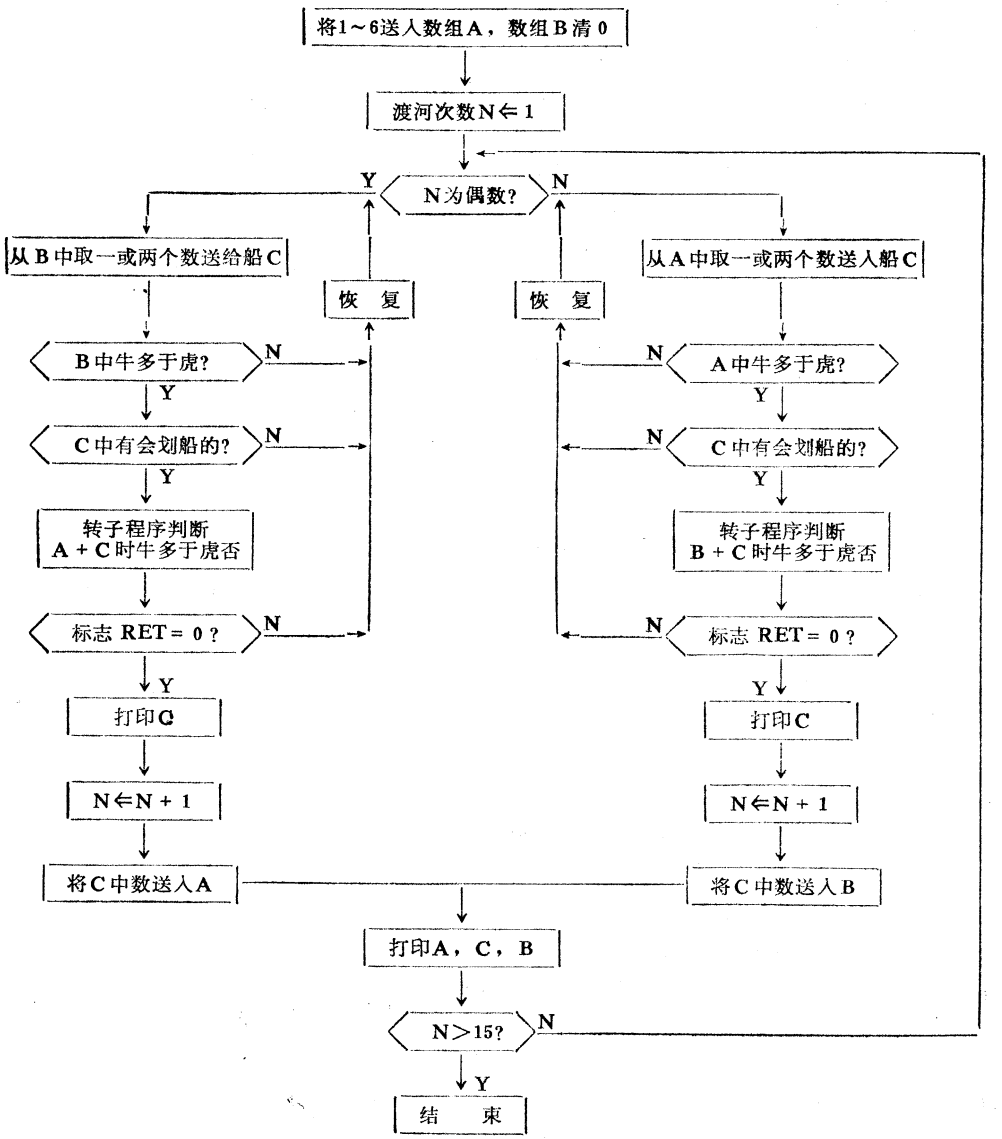


图 93

3）源程序及运行结果

```
C       PROGRAM  NHGH
        INTEGER  RET, A(6) , C(2) , B(6)
        DO  10  I=1, 6                                  I=1、2、3为虎 I=4、5、6为牛
10      A(I) =I                          }
        N=1                              }置初始状态
12      DO  5  I=1, 2                    }
5       C(I) =0                          }
        IF  ((N/2) *2. EQ. N)  GOTO  50
        DO  40  I=1, 6                                  N为奇数是过河
        IF (A(I) . EQ. 0) GOTO  40       }选择第一个过河
        C(1) =I                          }
        DO  40  J=1, 6                   }选择第二个过河
        IF (J. EQ. I) GOTO  40           }
        IF (A(J) . EQ. 0) GOTO  40       }                选择满足条件的两个过河
        C(2) =J                          }
        CALL  SUBPD (A, C, N, RET)       }
        IF (RET. EQ. 0) GOTO  120  满足过河条件是RET为0
40      CONTINUE
        STOP  "ERR1"                     未找到满足条件则出错
50      DO  100  J=1, 6
        IF (A(J) . NE. 0) GOTO  100      选择返回的第一个
        C(1) =J
        IF (J. EQ. 1. OR. J. EQ. 4) GOTO  60   若被选中的会划
        DO  55  I=1, 6                         船，转子判断条件
        IF (A(I) . NE. 0. OR. I. EQ. J)  }
          GOTO  55                       }
        C(2) =I                          }第一个不会划船，  N为偶数是返回
        CALL  SUBPD (A, C, N, RET)       }再选一个判断。
        IF  (RET. EQ. 0) GOTO  140       }
55      CONTINUE
        GOTO  100
60      CALL  SUBPD (A, C, N, RET)
        IF  (RET. EQ. 0) GOTO  140
100     CONTINUE
        STOP "ERR 2"                     未找到满足条件者则出错
        GOTO  360
120     WRITE  (10, 310) C               }
        GOTO  200                        }找到满足条件者，打印过河。
140     WRITE (10, 320) C                }
200     N=N+1                            计数器N加1
        IF (N. GT. 15) GOTO  360         N>15结束。
```

```
      DO  220  IK=1, 6
220   B(IK) =0                                  把已过河的送入B数组，便于打印。
      DO  250  IK=1, 6
      IF (A(IK) . NE. 0) GOTO  250
      B(IK) =IK
250   CONTINUE
      WRITE (10, 330) A, C, B                   打印结果
      IF (N. NE. 9) GOTO  340
      C(1) =4
      C(2) =6
      CALL  SUBPD (A, C, N, RET)
      IF (RET. EQ. 0) GOTO  120
310   FORMAT (25X, 2I3, 8H    ---->)
320   FORMAT (31X, 6H<----, 2I3)
330   FORMAT (3X, 6I3, 4X, 2I3, 4X, 6I3/)
340   DO  350  KL=1, 6                          若已全部过河则结束
      IF   (A(KL) . NE. 0) GOTO  12
350   CONTINUE
360   STOP
      END
      SUBROUTINE  SUBPD (A, C, N, RET)
      INTEGER  E, C, A, RET
      DIMENSION  A(6), E(6), C(2)    E(6) 为A(6) 的工作数组
      DO  5  I=1, 6
5     E(I) =A (I)
      IF (C(1) . EQ. 1. OR. C(1). EQ. 4)  GOTO  10   C(1)中有会划船的吗
      IF (C(2) . NE. 1. AND. C(2) . NE. 4)  GOTO  150
                                    若C(1) 中都不会划船，不满足条件返回。
10    IF ( (N/2) *2. EQ. N)  GOTO  30           N为奇数，过河，工作单元清 0 。
      DO  20  I=1, 2
      IF   (C(I) . NE. 0)   E(C(I) ) =0
20    CONTINUE
      GOTO  45
30    DO  40  J=1, 2                            N为偶数返回，对应单元置 J 。
      IF (C(J) . NE. 0)  E(C(J)) =C(J)
40    CONTINUE
45    L1=0                                      工作单元  A岸虎数 L 1
      L2=0                                                A岸牛数 L 2
      M1=0                                                B岸虎数M 1
      M2=0                                                B岸牛数M 2
      DO  100  K=1, 6
      IF   (E(K) . EQ. 0)   GOTO  80
      IF   (E)K) . LE. 3)   GOTO  50
```

```
      L2 = L2 + 1
      GOTO  100
50    L1 = L1 + 1
      GOTO  100
80    IF  (K. LE. 3)  GOTO  90
      M2 = M2 + 1
      GOTO  100
90    M1 = M1 + 1
100   CONTINUE
      IF  (L1. GT. L2. AND. L2. NE. 0. OR. M1. GT. M2. AND. M2. NE.
     0)  GOTO  150
      RET = 0
      GOTO  200
150   RET = 1
200   IF  (RET. NE. 0)  GOTO  220
      DO  210  I = 1, 6
210   A(I) = E(I)
220   RETURN
      END
```

计算A、B两岸的牛虎数

判断虎数多于牛数否

标志RET = 0，满足条件。

RET = 1，不满足条件。

当RET = 0 时改变A的状态

程序运行打印结果如下

（此岸）								（彼岸）					
						1	2---->						
0	0	3	4	5	6	1	2	1	2	0	0	0	0
							<----1	0					
1	0	3	4	5	6	1	0	0	2	0	0	0	0
						1	3---->						
0	0	0	4	5	6	1	3	1	2	3	0	0	0
							<----1	0					
1	0	0	4	5	6	1	0	0	2	3	0	0	0
						4	5---->						
1	0	0	0	0	6	4	5	0	2	3	4	5	0
							<----2	4					
1	2	0	4	0	6	2	4	0	0	3	0	5	0
						1	4---->						
0	2	0	0	0	6	1	4	1	0	3	4	5	0
							<----3	4					
0	2	3	4	0	6	3	4	1	0	0	0	5	0
						4	6---->						
0	2	3	0	0	0	4	6	1	0	0	4	5	6
							<----1	0					
1	2	3	0	0	0	1	0	0	0	0	4	5	6
						1	2---->						
0	0	3	0	0	0	1	2	1	2	0	4	5	6

```
                                    <----1   0
1   0   3   0   0   0       1   0       0   2   0   4   5   6

                                    1   3----->
0   0   0   0   0   0       1   3       1   2   3   4   5   6
```

（三）思　考　题

两大人和两小孩来到河边，渡口只有一条小船，一次只能渡过一个大人或者两个孩子。

他们四个人都会划船，但不会游泳。请你想一想，他们四个人怎样才能最快地渡过河去？

十四　第五届国际数学竞赛试题

94　教练员智拆两队

业余体校篮球队10个队员，是来自两个学校的学生，每校五人为一组。每组的5个人，由于经常在一起训练学习，配合默契。但教练员发现各有所长，试图打乱原组，调整阵容，但两组都不同意，说这样参加比赛时配合不好。教练员急中生智说："你们原组配合就那么好吗？我看不然，咱们进行测验，在篮球场两个半场，各画5个圆圈成一字形，每组的5个人各站在5个圆圈里，本组的5个人互相交换位置，看有多少种排列方法。不许重复、丢拉，并且看哪个组排的速度快，胜者不拆散，回去准备，明天测验。"

次日两组队员站在指定位置上，身穿1～5号字样衣服，以便两名裁判员记录各组排列变化。有一组进行中途便混乱了，相互拉扯说应该站的位置，无法进行。另一组好象比赛时穿插切入，首尾相连，宛如游龙，排列完毕，但裁判员说正确位置还不过百。教练员得胜。

试问正确排列位置不过百还不全对，那么究竟正确排列共有多少种？

（一）笔算步骤和结果

这是一个全排列问题，5个队员进行不同位置的全排列共有120种，即$P_5=5\times4\times3\times2\times1=120$（种）

请看计算机打印出的每种变换情况。

（二）程序设计

1）设计思路

5个队员用数字1～5表示，定义数组IA(5)，置入5个队员排列位置。利用5重循环进行排列，经判断正确时，输出数组IA(5)，为了便于观察变化规律，打印时每行一组。计数器II记录共有多少种变化。

2）框图　见图94

3）源程序及运行结果

```
DIMENSION  IA(5)
II=0
DO  10  I=1, 5
IA(1) =I
DO  10  J1=1, 5
IF (J1. EQ. IA(1))  GOTO  10
IA(2) =J1
DO  10  J2=1, 5
IF (J2. EQ. IA(1) . OR. J2. EQ. IA(2))  GOTO  10
IA (3) =J2
DO  10  J3=1, 5
IF (J3. EQ. IA(1) . OR. J3. EQ. IA(2) . OR. J3. EQ. IA(3))
    GOTO  10
```

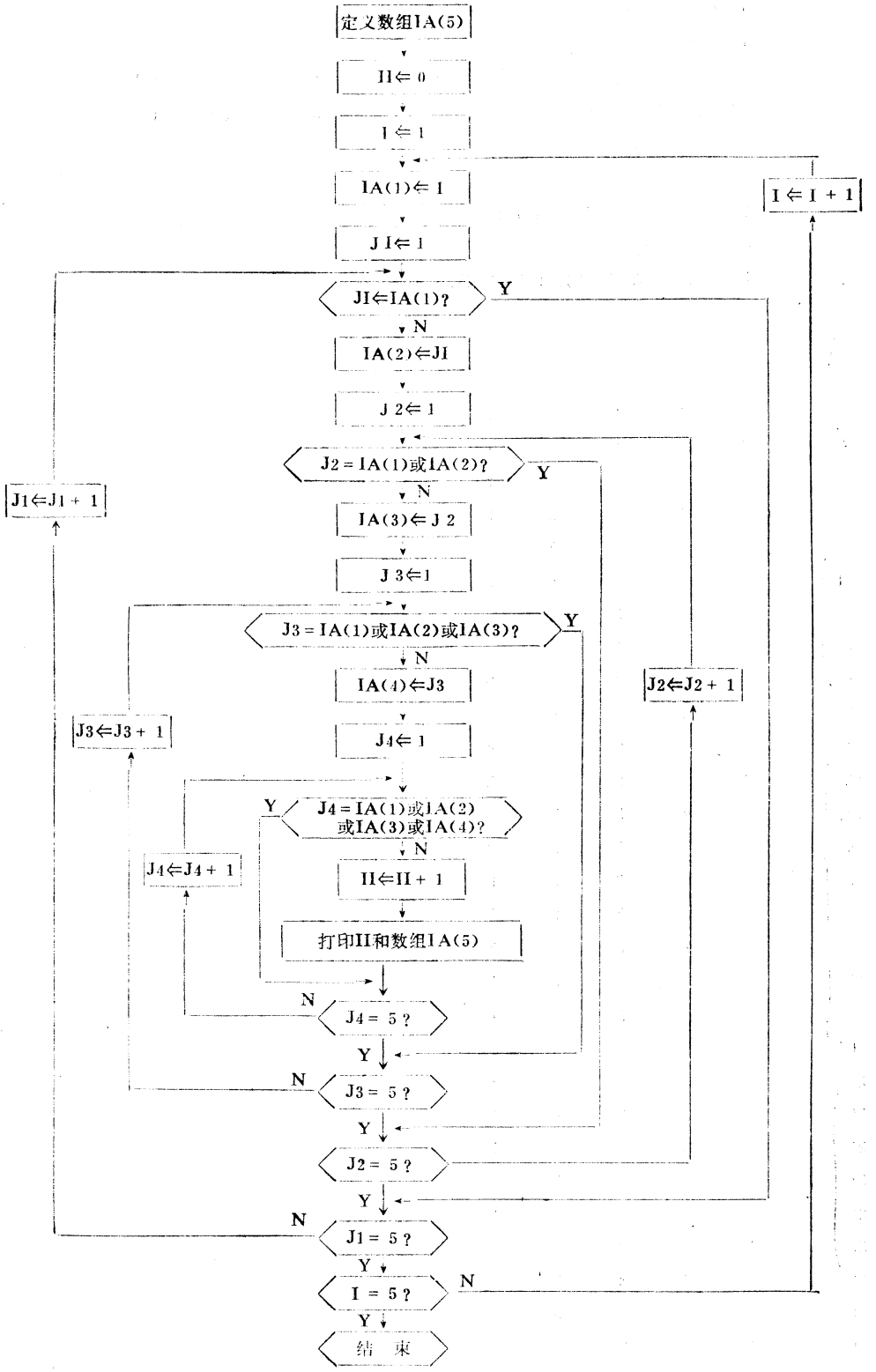

定义数组IA(5)
II⇐0
I⇐1
IA(1)⇐I
JI⇐1
JI⇐IA(1)?
Y
N
IA(2)⇐JI
J2⇐1
J2=IA(1)或IA(2)?
Y
N
IA(3)⇐J2
J3⇐1
J3=IA(1)或IA(2)或IA(3)?
Y
N
IA(4)⇐J3
J4⇐1
Y
J4=IA(1)或IA(2)或IA(3)或IA(4)?
N
II⇐II+1
打印II和数组IA(5)
N
J4=5?
Y
N
J3=5?
Y
J2=5?
Y
N
J1=5?
Y
I=5?
N
Y
结束
I⇐I+1
J2⇐J2+1
J3⇐J3+1
J4⇐J4+1
J1⇐J1+1

图 94

```
      IA(4) = J3
      DO 10 J4 = 1, 5
      IF (J4. EQ. IA(1) . OR. J4. EQ. IA(2)) GOTO 10
      IF (J4. EQ. IA(3) . OR. J4. EQ. IA(4)) GOTO 10
      IA(5) = J4
      II = II + 1
      WRITE (1, 1) II, (IA(J) , J = 1, 5)
10    CONTINUE
1     FORMAT (6X, I3, 5X, 5 (I2, 1X) )
      STOP
      END
```

1	1	2	3	4	5
2	1	2	3	5	4
3	1	2	4	3	5
4	1	2	4	5	3
5	1	2	5	3	4
6	1	2	5	4	3
7	1	3	2	4	5
8	1	3	2	5	4
9	1	3	4	2	5
10	1	3	4	5	2
11	1	3	5	2	4
12	1	3	5	4	2
13	1	4	2	3	5
14	1	4	2	5	3
15	1	4	3	2	5
16	1	4	3	5	2
17	1	4	5	2	3
18	1	4	5	3	2
19	1	5	2	3	4
20	1	5	2	4	3
21	1	5	3	2	4
22	1	5	3	4	2
23	1	5	4	2	3
24	1	5	4	3	2
25	2	1	3	4	5
26	2	1	3	5	4
27	2	1	4	3	5
28	2	1	4	5	3
29	2	1	5	3	4
30	2	1	5	4	3
31	2	3	1	4	5

32	2	3	1	5	4
33	2	3	4	1	5
34	2	3	4	5	1
35	2	3	5	1	4
36	2	3	5	4	1
37	2	4	1	3	5
38	2	4	1	5	3
39	2	4	3	1	5
40	2	4	3	5	1
41	2	4	5	1	3
42	2	4	5	3	1
43	2	5	1	3	4
44	2	5	1	4	3
45	2	5	3	1	4
46	2	5	3	4	1
47	2	5	4	1	3
48	2	5	4	3	1
49	3	1	2	4	5
50	3	1	2	5	4
51	3	1	4	2	5
52	3	1	4	5	2
53	3	1	5	2	4
54	3	1	5	4	2
55	3	2	1	4	5
56	3	2	1	5	4
57	3	2	4	1	5
58	3	2	4	5	1
59	3	2	5	1	4
60	3	2	5	4	1
61	3	4	1	2	5
62	3	4	1	5	2
63	3	4	2	1	5
64	3	4	2	5	1
65	3	4	5	1	2
66	3	4	5	2	1
67	3	5	1	2	4
68	3	5	1	4	2
69	3	5	2	1	4
70	3	5	2	4	1
71	3	5	4	1	2
72	3	5	4	2	1
73	4	1	2	3	5
74	4	1	2	5	3

75	4	1	3	2	5
76	4	1	3	5	2
77	4	1	5	2	3
78	4	1	5	3	2
79	4	2	1	3	5
80	4	2	1	5	3
81	4	2	3	1	5
82	4	2	3	5	1
83	4	2	5	1	3
84	4	2	5	3	1
85	4	3	1	2	5
86	4	3	1	5	2
87	4	3	2	1	5
88	4	3	2	5	1
89	4	3	5	1	2
90	4	3	5	2	1
91	4	5	1	2	3
92	4	5	1	3	2
93	4	5	2	1	3
94	4	5	2	3	1
95	4	5	3	1	2
96	4	5	3	2	1
97	5	1	2	3	4
98	5	1	2	4	3
99	5	1	3	2	4
100	5	1	3	4	2
101	5	1	4	2	3
102	5	1	4	3	2
103	5	2	1	3	4
104	5	2	1	4	3
105	5	2	3	1	4
106	5	2	3	4	1
107	5	2	4	1	3
108	5	2	4	3	1
109	5	3	1	2	4
110	5	3	1	4	2
111	5	3	2	1	4
112	5	3	2	4	1
113	5	3	4	1	2
114	5	3	4	2	1
115	5	4	1	2	3
116	5	4	1	3	2
117	5	4	2	1	3

118	5	4	2	3	1
119	5	4	3	1	2
120	5	4	3	2	1

（三）思 考 题

6个儿童站成一排表演小合唱，其中某一个领唱的儿童不站在排头，也不站在排尾，一共有多少种站法？

分析1 因为某一个儿童不站在排头，也不站在排尾，站在这两个位子上的，就只能在其余5个儿童里选，有A_5^2种选法，而对于其中的每一种选法，站在中间位子上的4个儿童又有P_4种不同的站法，因此，一共有$A_5^2 \cdot P_4 = 4P_5$种站法。

分析2 6个儿童站成一排，如果没有什么限定条件，那么可以有P_6种站法。这些站法里有符合条件的，也有不符合条件的。不符合条件的站法中，这个儿童站在排头的有P_5种，他站在排尾的有P_5种，一共有$2P_5$种。因此，符合要求的站法一共有$P_6 - 2P_5 = 4P_5$种。

分析3 因为限定某一个儿童不站在排头，也不站在排尾，这个儿童就只能站在中间四个位置上的一个上，有4种站法；而对于每一种这样的站法，其余5个儿童可以全排列，故有P_5种站法，所以一共有$4P_5$种站法。

95 乒乓球队员分组训练

甲队有七名乒乓球队员，在培训中每三人分成一组，以便每组中的三个人相互比赛时余一人当裁判。问共有几种分组方法。

（一）笔算步骤和结果

这是一个组合问题，将七名队员分别编上号1、2…7。每次比赛时，由三人组成，共有 $C_7^3 = \dfrac{7!}{3!(7-3)!} = 35$（种）分组方法。

（二）程 序 设 计

1）设计思路

七名运动员编号为1～7，定义数组IA（7）置放1～7这七个数字，每次取三名为一组，利用三重循环来确定，要判断只要这三名队员不重复即可，每组合乎要求时，便打印出一行不同的三个数字，表明一组中的不同的三名队员，每打印一组，记数器II加1，并在每组前打印顺序员，以便观察各种组合。

2）框图 见图95

3）源程序及运行结果

```
DIMENSION  IA(7)
II = 0
DO  10  I = 1, 7
IA(1) = I
DO  10  J1 = 1, 7
IF (J1. LE. IA(1))  GOTO  10
IA(2) = J1
DO  10  J2 = 1, 7
IF (J2. LE. IA(1). OR. J2. LE. IA(2))  GOTO  10
```

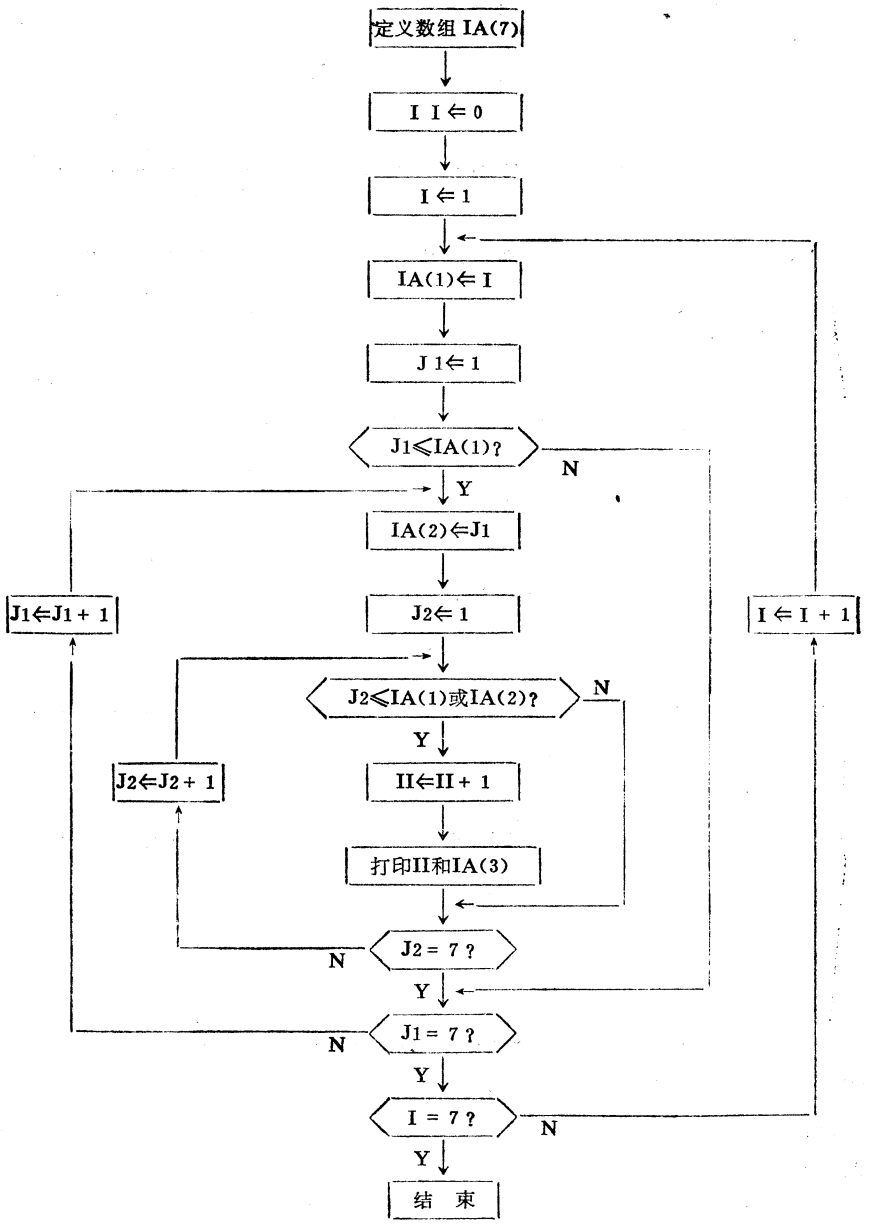


图 95

```
      IA(3) = J2
      II = II + 1
      WRITE (1, 1) II, (IA(J), J = 1, 3)
10    CONTINUE
      FORMAT (6X, I3, 7X, 3 (I2, 1X))
      STOP
      END
```

1	1	2	3
2	1	2	4
3	1	2	5
4	1	2	6
5	1	2	7
6	1	3	4
7	1	3	5
8	1	3	6
9	1	3	7
10	1	4	5
11	1	4	6
12	1	4	7
13	1	5	6
14	1	5	7
15	1	6	7
16	2	3	4
17	2	3	5
18	2	3	6
19	2	3	7
20	2	4	5
21	2	4	6
22	2	4	7
23	2	5	6
24	2	5	7
25	2	6	7
26	3	4	5
27	3	4	6
28	3	4	7
29	3	5	6
30	3	5	7
31	3	6	7
32	4	5	6
33	4	5	7
34	4	6	7
35	5	6	7

（三）思 考 题

1）学校开运动会，一共有8个篮球队参加篮球比赛。如果每队都要和其他的队比赛一次，那么全校一共要比赛篮球多少场？

笔算分析——因为篮球比赛每次需要两个队，而这两个队的选择与顺序无关，也就是甲队同乙队比赛与乙队同甲队比赛是同一场比赛，而不是两场。所以这个问题可以把它归结为组合问题来求解。这里的元素是指篮球队，从8个元素里每次取出2个元素的组合，对应着一次比赛，请编写程序并打印出结果。因此，组合的种数C_8^2就是一共比赛的

次数。

2）为了提高区别“排列”、“组合”这两类问题的能力，这里可以提出一个问题，在上述思考题1）里，如果8个篮球队比赛的结果，产生一个第一名和一个第二名，那么一共有多少种可能得名次的方法？

笔算分析——甲得第一名乙得第二名和乙得第一名甲得第二名是两种可能得名次的方法。所以这个问题就应当归结为排列问题来求解。从8个元素里每次取出2个元素的一个排列，对应一种可能得名次的方法，请编写程序并打印出这种排列所有的可能，因此，一共有A_8^2种可能的方法。

96 砝码问题

法国数学家德·梅齐亚克在他的名著《数字组合游戏》（1962年出版）中提出了一个这样问题：一位商人有一个重40磅的砝码。一天，商人不小心将砝码掉地碎成四块。后来，商人称得每块的重量都是整磅数，而且发现用这四块碎片，可以称1至40磅之间的任意整磅数的重物。请问这四块砝码碎片各重多少？

（一）笔算步骤和结果

天平的两个称盘分别为砝码盘和称量盘，在砝码盘上只能放砝码，而在称量盘上除了可放重物外还可以加放砝码。

有一系列砝码均为正整数磅A、B、C……，把它们适当地分放在两个称盘上，就能称出从1到n的所有整数磅的重物。德·梅齐里亚克指出：如果有一块新砝码P，只要它的重量是原有砝码的重量总和n的两倍加1，即：$P=2n+1$，那么，当这个新砝码P加入砝码组A、B、C……之后，就能称出从1至$3n+1$的所有整数磅的重物。

在商人跌碎的四块碎片中，如果要使碎片A和B能称出最大范围的重量，那么A必须是1磅，B必须是3磅，这两块碎片就能称出1、2、3、4磅的重物。

假如第三块碎片C的重量等于$2n+1$，即$C=2\times4+1=9$（磅），那么，A、B、C、D这四块碎片就能称出1至40磅的所有重物。即$3\times13+1=40$（磅）

所以，商人跌碎的四块砝码分别重量是1、3、9、27磅。

（二）程序设计

1）设计思路

分析砝码重量依次为3^0、3^1、3^2、3^3，定义数组M(5)置入重0、1、3、9、27磅五个砝码重量（设有0重砝码）。利用循环语句寻找天平两端平衡时是如何搭配的。

外层循环是物重，依次由1～39放重物的天平端还允许搭配砝码，由笔算分析知最多搭配三个不同重量的砝码（包括0砝码），另一端，除称40磅重物需放上四个砝码外，其余也和重物端一样，最多放三个砝码（包括0砝码），因此外层物体重量循环由1～39（磅），出循环后，再打印出40磅重物，使程序简化。天平两端各放三个砝码，寻找天平平衡时的等式，将40种平衡状态的每一种全打印出。等号左端表示天平放物重的一端，其第1个数字为物体重量，其余为砝码重，等号右端完全放砝码，三个数字则是砝码重量。如$16+0+3+9=0+1+27$，表示16磅重物一端需搭配0、3、9磅重的砝码，等号右端恰巧可用0、1、27磅砝码称量。

实际只有四个不同重量的砝码，所以天平两端（即等号两端），除0外其它砝码重量

不可能相同，在条件语句中加以逻辑判断，如果说砝码重与另一砝码若是相同，则必然是0砝码。物重循环参数I，物重端搭配的三个砝码重量的循环参数分别为I1，I2，I3。另一端三个砝码重量的循环参数分别为J1，J2，J3。

2）框图　见图96

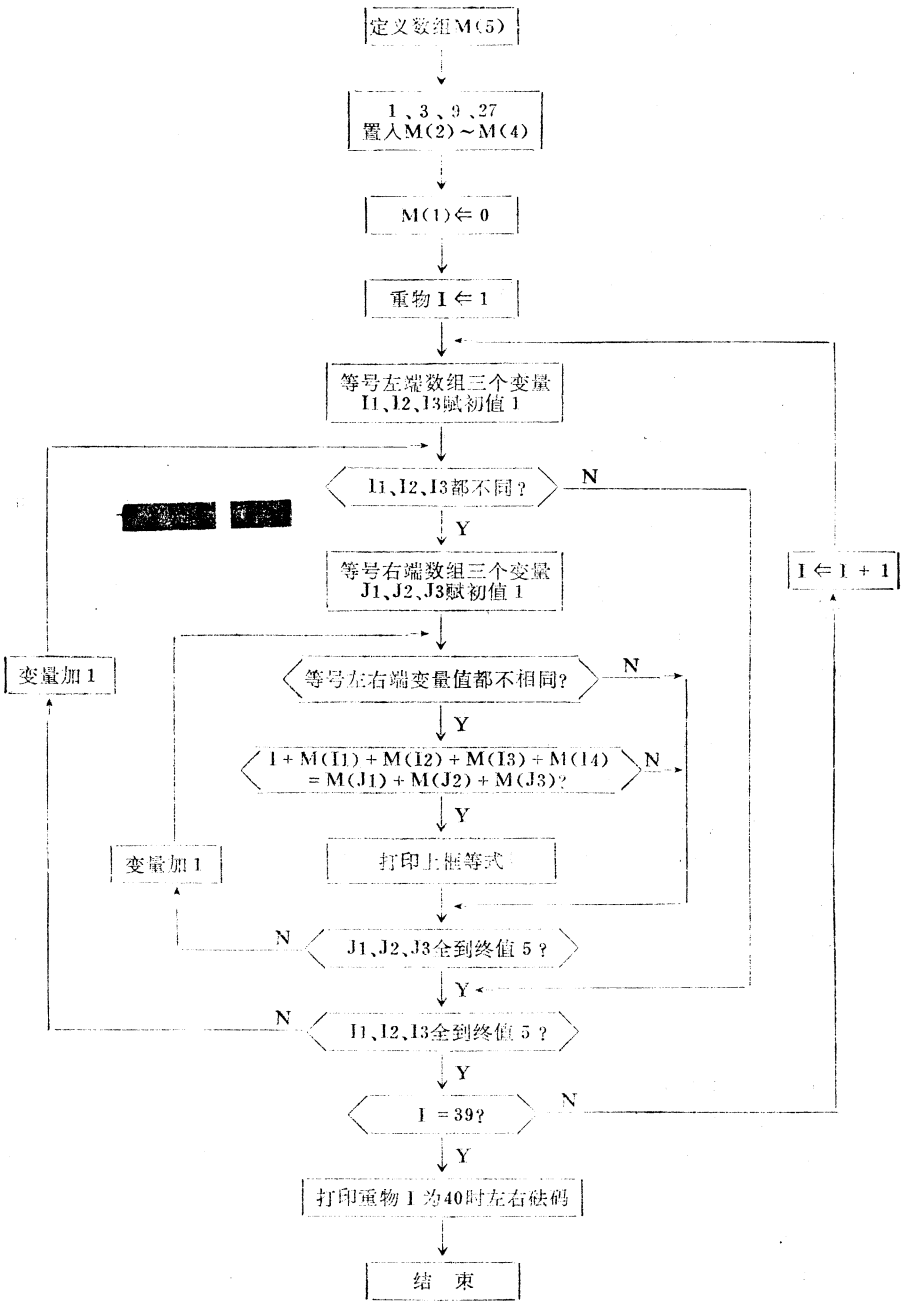


图 96

3）源程序及运行结果

```
      DIMENSION  M(5)
      DO  10  I=1, 4
      I1=I+1
10    M(I1) =3**(I1-2)
      M(1) =0
      DO  50  I=1, 39
      DO  30  I1=1, 5
      DO  30  I2=1, 5
      DO  30  J3=1, 5
      DO  30  J1=1, 5
      DO  30  J2=1, 5
      DO  30  J3=1, 5
      IF ((I1. EQ. I2. OR. I1. EQ. I3. OR. I1. EQ. J1. OR. I1. EQ. J2. OR.
     1    I1. EQ. J3) . AND. I1. NE. 1)  GOTO  30
      IF ( (I2. EQ. I3. OR. I2. EQ. J1. OR.  I2. EQ. J2. OR. I2. EQ. J3 ) .
     1  AND. I2. NE. 1) GOTO  30
      IF ((I3. EQ. J1. OR. I3. EQ. J2. OR. I3. EQ. J3 ) .  AND.  I3. NE. 1)
         GOTO  30
      IF ((J1. EQ. J2. OR. J1. EQ. J3) . AND. J1. NE. 1)  GOTO  30
      IF ((J2. EQ. J3) . AND. J2. NE. 1) GOTO  30
      IF ( I+M(I1) +M(I2) +M( I3) . EQ. M(J1) +M( J2 ) +M(J3 ))
         GOTO  40
30    CONTINUE
40    WRITE (1,  1) I, M(I1) , M(I2) , M(I3) , M(J1) , M(J2) , M(J3)
1     FORMAT (10X, 3 (I2, 1H+) , I2, 1H=, 2 (I2, 1H+) , I2)
50    CONTINUE
      WRITE (1, 2)
2     FORMAT (10X, '40+ 0 + 0 + 0 = 1 + 3 + 9 +27')
      STOP
      END
```

```
1 + 0 + 0 + 0 = 0 + 0 + 1
2 + 0 + 0 + 1 = 0 + 0 + 3
3 + 0 + 0 + 0 = 0 + 0 + 3
4 + 0 + 0 + 0 = 0 + 1 + 3
5 + 0 + 1 + 3 = 0 + 0 + 9
6 + 0 + 0 + 3 = 0 + 0 + 9
7 + 0 + 0 + 3 = 0 + 1 + 9
8 + 0 + 0 + 1 = 0 + 0 + 9
9 + 0 + 0 + 0 = 0 + 0 + 9
```

10 + 0 + 0 + 0 = 0 + 1 + 9
11 + 0 + 0 + 1 = 0 + 3 + 9
12 + 0 + 0 + 0 = 0 + 3 + 9
13 + 0 + 0 + 0 = 1 + 3 + 9
14 + 1 + 3 + 9 = 0 + 0 + 27
15 + 0 + 3 + 9 = 0 + 0 + 27
16 + 0 + 3 + 9 = 0 + 1 + 27
17 + 0 + 1 + 9 = 0 + 0 + 27
18 + 0 + 0 + 9 = 0 + 0 + 27
19 + 0 + 0 + 9 = 0 + 1 + 27
20 + 0 + 1 + 9 = 0 + 3 + 27
21 + 0 + 0 + 9 = 0 + 3 + 27
22 + 0 + 0 + 9 = 1 + 3 + 27
23 + 0 + 1 + 3 = 0 + 0 + 27
24 + 0 + 0 + 3 = 0 + 0 + 27
25 + 0 + 0 + 3 = 0 + 1 + 27
26 + 0 + 0 + 1 = 0 + 0 + 27
27 + 0 + 0 + 0 = 0 + 0 + 27
28 + 0 + 0 + 0 = 0 + 1 + 27
29 + 0 + 0 + 1 = 0 + 3 + 27
30 + 0 + 0 + 0 = 0 + 3 + 27
31 + 0 + 0 + 0 = 1 + 3 + 27
32 + 0 + 1 + 3 = 0 + 9 + 27
33 + 0 + 0 + 3 = 0 + 9 + 27
34 + 0 + 0 + 3 = 1 + 9 + 27
35 + 0 + 0 + 1 = 0 + 9 + 27
36 + 0 + 0 + 0 = 0 + 9 + 27
37 + 0 + 0 + 0 = 1 + 9 + 27
38 + 0 + 0 + 1 = 3 + 9 + 27
39 + 0 + 0 + 0 = 3 + 9 + 27
40 + 0 + 0 + 0 = 1 + 3 + 9 + 27

（三）思 考 题

若主题限制天平一端只许放物体，另一端只许放砝码，则至少需要几个不同重量的砝码？请编写程序，并将每一种称量方式都打印出。

97 五本书被三个人借

三个人从五本不同的书里每人借一本，一共有多少种不同的借法？

（一）笔算步骤和结果

应该这样理解这个问题，把五本不同的书分给三个人每人一本，问有多少种分法。如此分析，这是一个排列问题，即从五本书里每次取三本进行排列，共有多少种排列。

$A_5^3 = 5 \times 4 \times 3 = 60$（种），则共有60种不同的借法。

（二）程 序 设 计

1）设计思路

定义数组IA（3），表示三个人，将五种不同的书用数字1～5表示，利用三重循环，每个循环控制变量1～5，表示五本不同的书，往数组IA（3）冲放三个不相同的数字（不同的书），满足条件时，打印出该种借书方法，记数器II加1，记数器做为顺序号也打印出。

2）框图　见图97

3）源程序及运行结果

```
      DIMENSION  IA(3)
      II=0
      DO  10  I=1, 5
      IA(1) =I
      DO  10  J1=1, 5
      IF (J1. EQ. IA(1))   GOTO  10
      IA(2) =J1
      DO  10  J2=1, 5
      IF (J2. EQ. IA(1) . OR. J2. EQ. IA(2))   GOTO  10
      IA(3) =J2
      II=II+ 1
      WRITE (1, 1)   II,   (IA(J) , J=1, 3)
10    CONTINUE
1     FORMAT (6X, I3, 1H, , 5X, 3 (I2, 1X))
      STOP
      END
```

```
1,     1    2    3
2,     1    2    4
3,     1    2    5
4,     1    3    2
5,     1    3    4
6,     1    3    5
7,     1    4    2
8,     1    4    3
9,     1    4    5
10,    1    5    2
11,    1    5    3
12,    1    5    4
13,    2    1    3
14,    2    1    4
15,    2    1    5
16,    2    3    1
17,    2    3    4
```

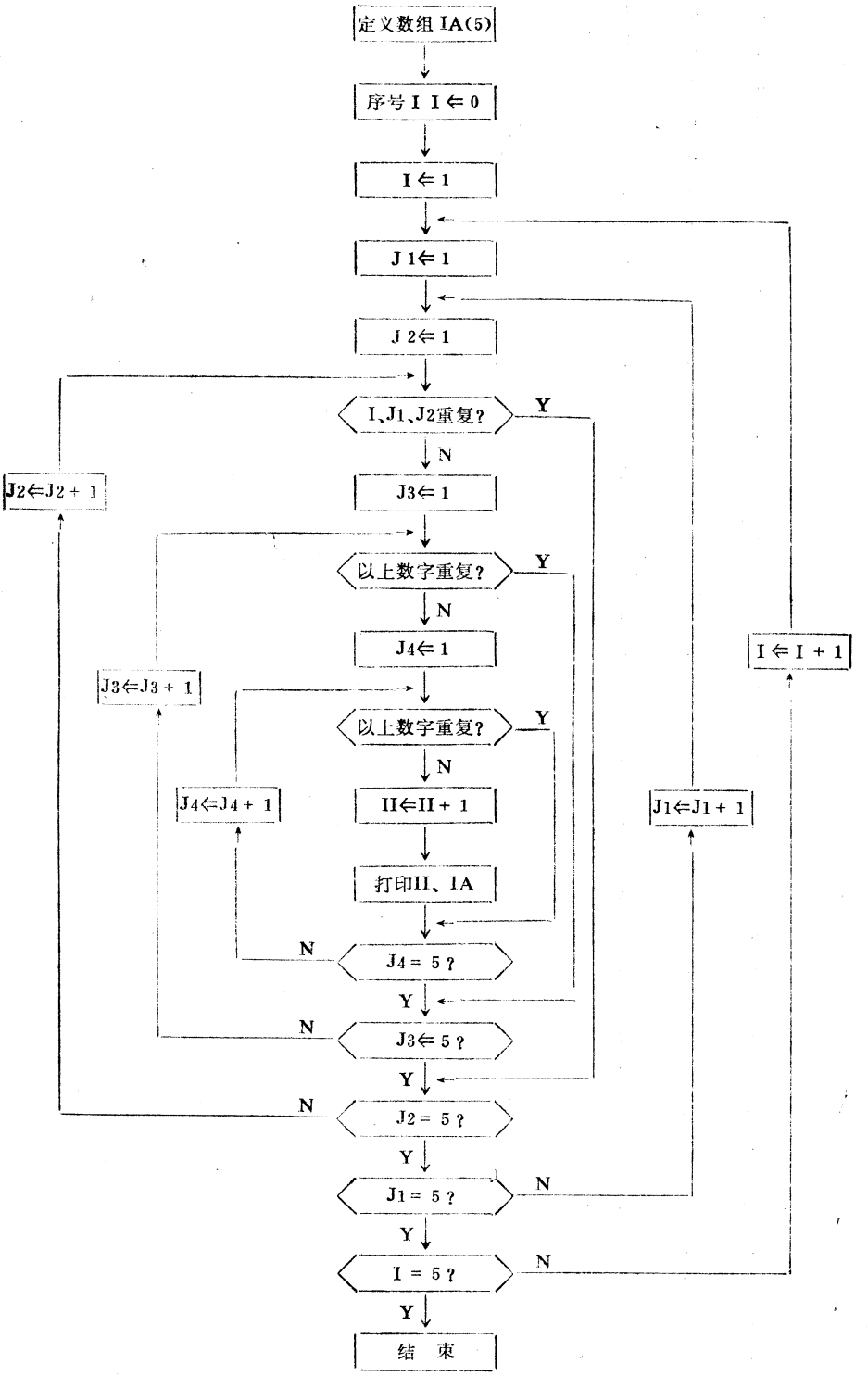

定义数组 IA(5)
序号 I I ⇐ 0
I ⇐ 1
J 1⇐ 1
J 2⇐ 1
I、J1、J2重复?
Y
N
J2⇐J2 + 1
J3⇐ 1
以上数字重复?
Y
N
J4⇐ 1
J3⇐J3 + 1
以上数字重复?
Y
N
J4⇐J4 + 1
II⇐II + 1
J1⇐J1 + 1
I ⇐ I + 1
打印II、IA
N
J4 = 5 ?
Y
N
J3⇐ 5 ?
Y
N
J2 = 5 ?
Y
J1 = 5 ?
N
Y
I = 5 ?
N
Y
结 束

图 97

18,	2	3	5
19,	2	4	1
20,	2	4	3
21,	2	4	5
22,	2	5	1
23,	2	5	3
24,	2	5	4
25,	3	1	2
26,	3	1	4
27,	3	1	5
28,	3	2	1
29,	3	2	4
30,	3	2	5
31,	3	4	1
32,	3	4	2
33,	3	4	5
34,	3	5	1
35,	3	5	2
36,	3	5	4
37,	4	1	2
38,	4	1	3
39,	4	1	5
40,	4	2	1
41,	4	2	3
42,	4	2	5
43,	4	3	1
44,	4	3	2
45,	4	3	5
46,	4	5	1
47,	4	5	2
48,	4	5	3
49,	5	1	2
50,	5	1	3
51,	5	1	4
52,	5	2	1
53,	5	2	3
54,	5	2	4
55,	5	3	1
56,	5	3	2
57,	5	3	4
58,	5	4	1
59,	5	4	2
60,	5	4	3

（三）思 考 题

一条铁路线上有8个车站，一共需要准备多少种不同的车票？

笔算分析——因为一种车票是只能在某两个车站中使用的，而且从甲站到乙站的车票不同于乙站到甲站的车票。这样，这个问题可以归结为从8个元素每次取2个元素的排列问题，每两个元素的一种排列对应一种车票，因此，所要求的车票种数就等于所有的排列种数A_8^2了。

$A_8^2=8\times7=56$。请编程序并打印出所有结果。

98 巧 排 分 组

1到15的全部整数正好可以分成5组，每组3个数：

1, 8, 15 } d=7　　4, 9, 14 } d=5　　2, 6, 10 } d=4　　3, 5, 7 } d=2　　11, 12, 13 } d=1

分配在各组的数是，每一组第2第1两数之差与第3第2两数之差相等，差数以d表示。如8－1＝7和15－8＝7，又9－4＝5和14－9＝5（凡邻接数之间为常数的一组数，都可以编成序列，叫做算术级数）。将15个顺序整数分为五组，每组包含上述差数的分配法，不只一种。如第一组数（1、8、15）不变动，其余的12个数（2、3、4、5、6、7、9、10、11、12、13、14）可以另取三个数一组，编成四组，而保持原来的差：d＝5；d＝4；d＝2；d＝1。

现在我们就把这15个数重新编组。

（一）笔算步骤和结果

根据题意分析，不难得出另一种解：

1, 8, 15 } d=7　　2, 7, 12 } d=5　　6, 10, 14 } d=4　　9, 11, 13 } d=2　　3, 4, 5 } d=1

（二）程 序 设 计

1）设计思路

定义数组K（15）置放1～15这15个数字。先将第1组（1，8，15）固定不变，直接赋值。然后利用循环语句，按题设d的数值去寻找其余的四个组，每找到符合条件的组合时，便打印输出，请见程序运行后打印结果。

2）框图　见图98

3）源程序及运行结果

```
      DIMENSION  K(15)
      DO   5   I=1, 15
5     K(I) = 0
      K(1) = 1
      K(2) = 8
      K(3) =15
      DO  10  K4=2, 4
```

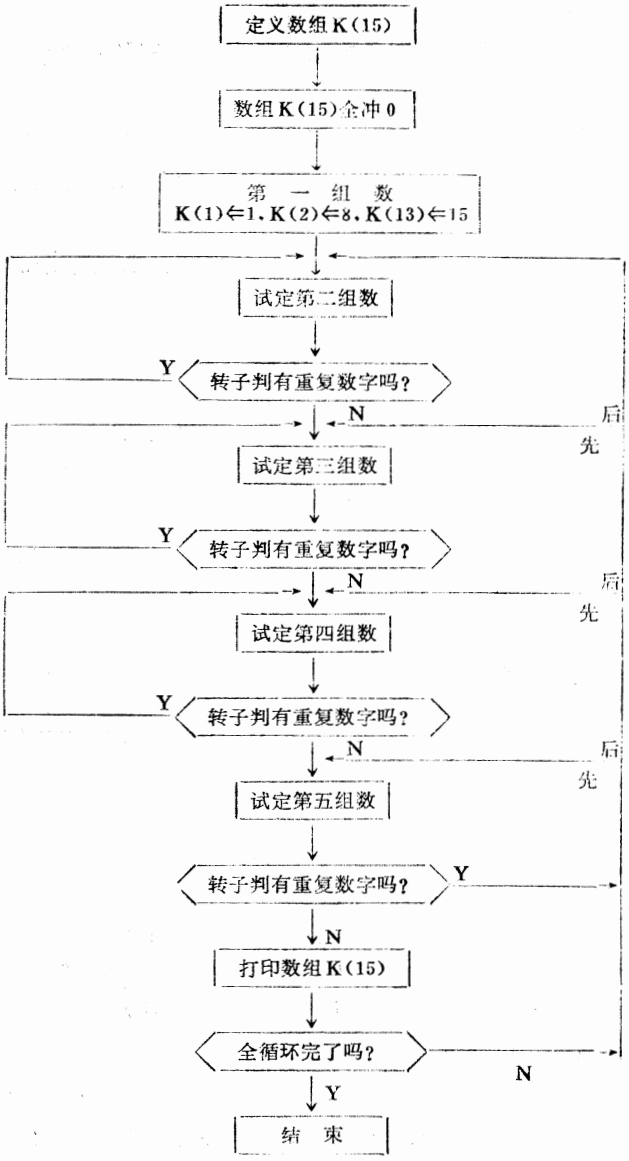


图 98

```
K(4) =K4
K(5) =K(4) + 5
K(6) =K(5) + 5
IY=0
CALL  PS (K, 5, IY)
IF (IY. EQ. 1) GO  TO  10
DO  20  K7=2, 6
K(7) =K7
K(8) =K(7) + 4
K(9) =K(8) + 4
IY= 0
```

```
      CALL  PS (K, 8, IY)
      IF (IY. EQ. 1) GO  TO  20
      DO  30  K10=2, 10
      K(10) =K10
      K(11) =K(10) +2
      K(12) =K(11) +2
      IY=0
      CALL  PS (K, 11, IY)
      IF (IY. EQ. 1) GO  TO  30
      DO  40  K13=2, 12
      K(13) =K13
      K(14) =K(13) +1
      K(15) =K(14) +1
      IY=0
      CALL  PS (K, 14, IY)
      IF (IY. EQ. 1) GO  TO  40
      WRITE (5, 50)  (K(L) , L=1, 15)
50    FORMAT (/5X, 3I4)
40    CONTINUE
30    CONTINUE
20    CONTINUE
10    CONTINUE
      STOP
      END
      SUBROUTINE  PS (K, IX, IY)
      DIMENSION  K(15)
      DO  10  I=1, IX
      I1=I+1
      IX1=IX+1
      DO  10  J=I1, IX1
      IF (K(I) . EQ. K(J)) GO  TO  20
10    CONTINUE
      GO  TO  30
20    IY=1
30    RETURN
      END
```

程序运行后打印结果

```
1    8    15
2    7    12
6    10   14   符合条件第一组数
9    11   13
3    4    5
```

1	8	15	符合条件第二组数
4	9	14	
2	6	10	
3	5	7	
11	12	13	

（三）思 考 题

请将 1 到15这些数，试分配成另一种 d 值的算术级数组。

99 第五届国际数学竞赛试题

在某次数学竞赛中，A、B、C、D、E五名学生，被取为前五名，请据下述说法判断出他们的具体名次，即谁是第几名？

条件 1：你如果认为A、B、C、D、E就是这些人的第一至第五名的名次排列，便大错。因为：

（1）没猜对任何一个优胜者的名次；

（2）也没猜对任何一对名次相邻的学生（即两个名次紧挨着的学生）。

条件 2：你如果按D、A、E、C、B来排列五人名次的话，结果：

（1）说对了其中两个人的名次；

（2）还猜中了两对名次相邻的学生的名次顺序。

（一）笔算步骤和结果

题中的问题是要把A、B、C、D、E的顺序作一排列，使这个排列恰好符合两个猜测者所猜测的情况。但是，我们知道A、B、C、D、E的排列可作 $P_5=120$（种）。罗列所有各种情形，再一一作鉴别，当然也可以，然而毕竟失之太繁。

如果从第二个人的猜测出发，在五个元素（如 x_1、x_2、x_3、x_4、x_5）的全排列中，固定两个元素的位置（对应于猜中了的两个名次），其他三个元素的位置全部重新排列，看在何种情况下恰好能保持其中有两对相邻元素仍取原来的顺序（对应于猜中了两对名次相邻的学生的名次顺序）。那末这样需考虑的可能情形只有 $C_5^2=10$（种），从下面的解答可以看到，其中符合上述要求的可能情形只能是

$$DA***;\quad ***CB。$$

当然，这种方法在本质上仍然属于筛选。但是，一次筛去的不是一个而是一批。事实上，经过这样的筛选以后，所需研究的排列便只剩下了 $2\times P_3=12$（种）。再根据两个猜测者所猜测的情况，对这12种排列一一加以筛选，就是实际可行的了。

〈**解法一**〉 设有五个元素的全排列

$$x_1x_2x_3x_4x_5$$

我们使其中的两个元素的位置不变，其他三个元素的位置全部重新排列，并观察其中有无两对相邻的元素仍取原来排列中的元素所取的顺序：

$$x_1x_2;\quad x_2x_3;\quad x_3x_4;\quad x_4x_5$$

容易验证，若按下列形式排列，不存在两对相邻的元素同时取上列的顺序：

x_1**x_3**；
x_1***x_5；
$*x_2*x_4*$；
$**x_3x_4*$；

x_1**x_4*；
$*x_2x_3**$；
$*x_2**x_5$；
$**x_3*x_5$。

于是，仅有下面两种排列，就是开头两个或最后两个元素位置不变的排列，可能存在某两对相邻的元素同时取上列的顺序：

$$X_1X_2***；\quad ***X_4X_5。$$

根据上面的结论，由第二个猜测的结果分析，实际比赛结果只可能是：

$$DA***；\quad ***CB。$$

第一种情形有下述可能性：

DACBE ；　DACEB ；
DABCE ；　DABEC ；
DAEBC ；　DAECB 。

通过和第一个人的猜测比较，第1，2，3，4，5种都是不可能的；第6种就是第二个人的猜测，同样是不可能的。

第二种情形有下述可能性：

ADECB ；　AEDCB ；
DAECB ；　DEACB ；
EADCB ；　EDACB 。

与前种情形的分析一样，第1，2，3，4种都是不可能的，第5种与第二个人的猜测比较，只有一对相邻元素CB的顺序相同，同样是不可能的。

由此，只剩下第6种排列，即E、D、A、C、B。容易验证，它恰好是问题的解。

〈**解法二**〉 从第二个人的猜测可推知，被猜中的两个名次必定是相邻的，否则将至多猜中一对名次顺序。因此只要从相邻的D、A，A、E，E、C和C、B四对名次顺序出发去考虑。

我们先考虑中间的两对A、E和E、C，如果其中任何一对在正确的位置上，例如A、E这对：

D、Ⓐ、Ⓔ、C、B。

其中有圈的假定是猜中的，此时，A在D后，C在E后，B在C后都不符合条件，否则将不止猜中了两个名次。同样，假定E、C是猜中的，也有这样的结论，所以，这两对都不是猜中的。

其次考虑D、A这对：

Ⓓ、Ⓐ、E、C、B。

其中A在D后，B在C后符合顺序关系，而且只有一种调法如下：

Ⓓ、Ⓐ、C、B、E。

这样又不符合C不在第三位上的条件，所以D、A这对也不是猜中的。

最后，考虑C、B这对：

D、A、E、Ⓒ、Ⓑ。

1）B在C后，又E在A后符合顺序关系，但经过调动后，则得

A、E、D、Ⓒ、Ⓑ，

又不能符合A不在第一位上的条件，所以这不是正确的答案。

2）B在C后，又A在D后也符合顺序关系，经过调动后，则得

E、D、A、Ⓒ、Ⓑ。

这样的排列就完全符合条件了。

（二）程 序 设 计

1）设计思路

定义数组MA(6)、MB(6)。数组MA(6)表示条件2给出的五人次序，即D、A、E、C、B，例如MA(1)表示D，MA(2)表示A，…，MA(5)表示B，程序中找出的数组元素值，是他们的名次。数组MB(6)表示条件1给出的五人次序（并非名次），即A、B、C、D、E。打印数组MB的元素值，便是A～E这五人的名次。程序中算法需要多设一个数组元素，即6。

其它说明，详见程序的旁注。

2）框图 见图99

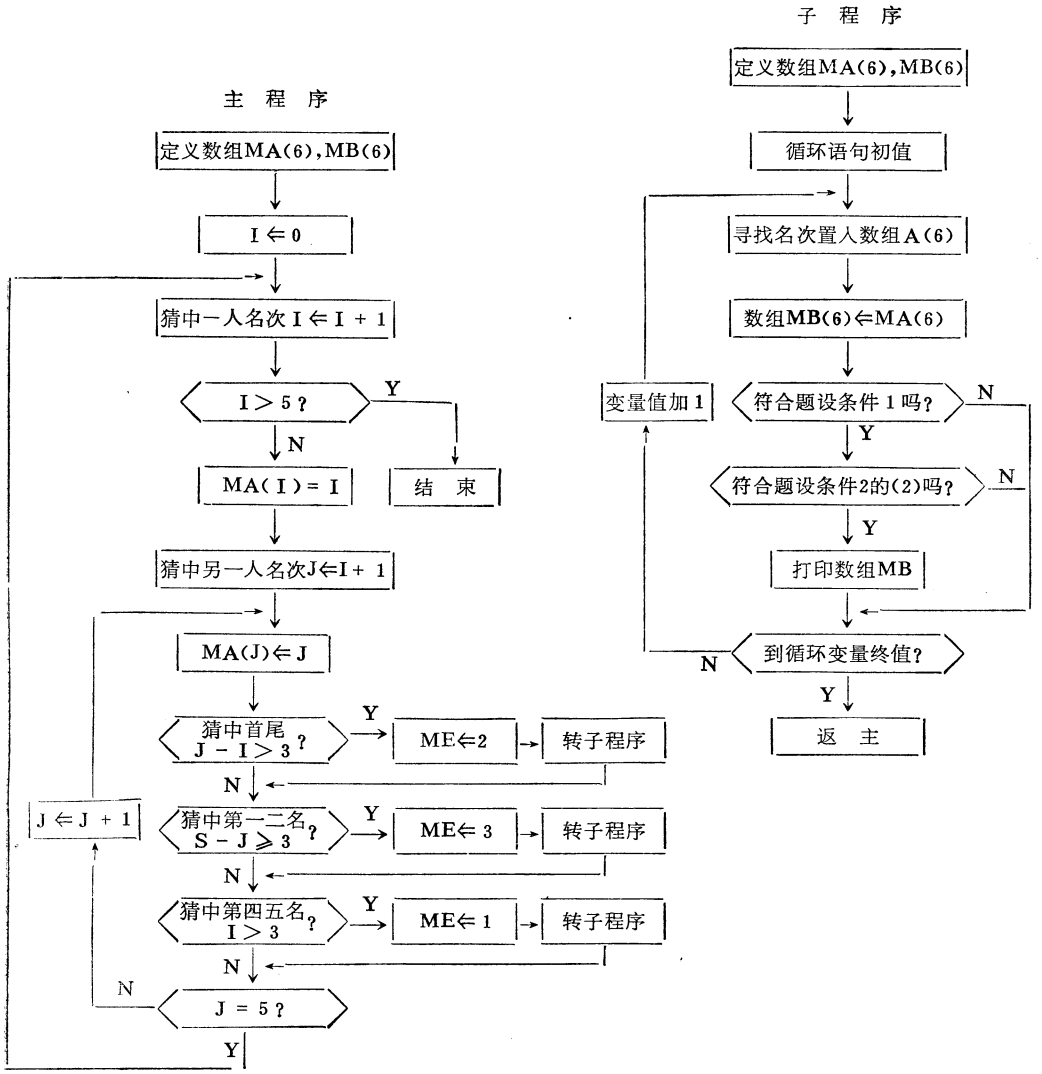


图 99

3）源程序及运行结果

```
      DIMENSION  MA(6), MB(6)
      I=0
80    I=I+1          据条件2，猜中的二人里其中之一是第I名
      IF (I. GT. 5)  GOTO 1000
      MA(I) =I
      J1=I+1
      DO 70 J=J1, 5          由条件2，猜中的二人里，另一人为第J名
      MA(J) =J
      IF (J-I. GT. 3)  GOTO 90          猜中首尾
120   IF (5-J. GE. 3)  GOTO 100          猜中第1、2名
130   IF (I. GT. 3)  GOTO 110          猜中第4、5名
70    CONTINUE
      GOTO 80
90    ME=2          除猜中二人外，控制其余三人的名次。
      CALL ORDER (MA, MB, ME, MH)
      GOTO 120
100   ME=3          若猜中第1、2名，控制其它三人的名次。
      CALL ORDER (MA, MB, ME, MH)
      GOTO 130
110   ME=1          若猜中第4、5名，控制其它三人的名次。
      CALL ORDER (MA, MB, ME, MH)
      GOTO 70
1000  STOP
      END
      SUBROUTINE ORDER(MA, MB, ME, MH)
      DIMENSION MA(6), MB(6)
      ME1=ME+1
      ME2=ME+2
      DO 10 MP=ME1, ME2          MP为未猜中的三人，找出他们由左至右第一位
                                 位置。
      MA(1) =MP
      DO 20 K=ME, ME2          K为未猜中的三人，找出他们由左至右第二位位置。
      IF (K. EQ. ME+1)  GOTO 20
      MA(ME+1) =K
      DO 30 MC=ME, ME1          MC为未猜中的三人，找出他们由左至右第三位
                                位置。
      MA(ME+2) =MC
      IF (MP. EQ. MC)  GOTO 30          若成立，说明D、E排的名次对。
      IF (MP. EQ. K)  GOTO 20          若成立，说明D、A排的名次对。
      IF (K. EQ. MC)  GOTO 30          若成立，说明A、E排的名次对。
      MB(1) =MA(2)
```

```
      MB(2) =MA(5)
      MB(3) =MA(4)
      MB(4) =MA(1)
      MB(5) =MA(3)
                                  五人名次置放数组MB里。
      DO 40 MH=1, 5
      IF (MB(MH) .EQ. MH) GOTO 30          判条件1的排列，应没有一个是对的。
      IF (MB(MH) .EQ. MB(MH+1)-1) GOTO 30  判条件2没指出另一
                                           个优胜者的前1名。
40    CONTINUE
      WRITE (6, 70)
70    FORMAT (4X, 1HA, 3X, 1HB, 3X, 1HC, 3X, 1HD, 3X, 1HE)
      WRITE (6, 60) (MB(I), I=1, 5)
60    FORMAT (1X, 5I4)
50    CONTINUE
30    CONTINUE
20    CONTINUE
10    CONTINUE
      RETURN
      END
```

程序运行后打印结果

A	B	C	D	E	表示五个人
3	5	4	2	1	表示五个人对应的名次

（三）思 考 题

五个学生A、B、C、D、E参加一场比赛，五个人对比赛结果名次进行了猜测，每个人猜测两个学生（英文字母表示人，数字代表名次）：

甲猜：2—B，3—A；
乙猜：2—D，4—E；
丙猜：1—E，5—C；
丁猜：3—D，4—C；
戊猜：2—A，5—B。

结果每个人恰好猜对了一个，并且每个名次都有一个人猜对，问实际比赛结果如何？请进行程序设计，寻找出正确名次。

100 用金环付房租

西欧一旅游者在当地旅游时没带现钱，但有一串首尾不相连的金质链子，由23个环组成，旅馆允许在停留期间每天必须付一金环，做为住宿费，而旅客又不肯预先多付，旅馆可用他已付的金环再找给他。试问，若旅游者欲住23天，最少需要将链子剪断几次？在第几环剪开？

（一）笔算步骤和结果

分析后，最少剪断两次，从第四第七节剪断，剪断后的单环立即取下，便将长链分成五段，每段个数为：3、1、6、1、12，便可对付房租。

第一天付1环；第二天再付1环；第三天付3环，找回前两天付的两个1环；第四天、第五天付1、1环；第六天付6环，找回以前所付；…依次类推，直到第二十三天正好用完23节金链。

（二）程序设计

1）设计思路

定义数组MS（25）置放23个环，程序算法需要多设2个数组元素，具体步骤如下：

（1）先从剪断次数N着手，开始时N⇐N+1；

（2）当N暂定后，再试从1～23长链中从何处剪断，断链除单环外，赋给数组MA；

（3）转子程序判能应付连续住23天吗？若能应付，说明剪断次数N以及剪断位置全对，打印出N及除单环外的几段环数。若不能应付23天，再返回上述（1），重复寻找。

本程序原设计剪断次数N为6次，多次调试发现N＝2即可，故将本程序修改为N＝3时便停机。详见源程序注释。

2）框图　见图100

3）源程序及运行结果

```
      DIMENSION  MS(25)
      N=0          N为剪断次数
 70   N=N+1
      GOTO  (100, 200, 1000), N
100   DO  110  I=1, 23          从链中第I个环剪断，剪一次。
      MS(1)=I-1          I环以前置入数组MS（1），因剪断处拆下环为1，故 I-1。
      MS(2)=23-I         I环以后置入数组MS（2）。
      CALL  RING (N, MS)              转子判能应付住的天数吗
110   CONTINUE
      GOTO  70
200   DO  210  I1=1, 22          剪两次，从I1环处剪断。
      I21=I1+1
      DO  220  I2=I21, 23          再从I2环处剪断
      MS(1)=I1-1         I1环以前置入数组MS（1）
      MS(2)=I2-I1-1          I1至I2环间置入数组MS（2）
      MS(3)=23-I2          I2环以后置入数组MS（3）
      CALL  RING (N, MS)              转子判能应付住的天数吗
220   CONTINUE
210   CONTINUE
      GOTO  70
1000  STOP
      END
      SUBROUTINE  RING (N, MS)
      DIMENSION  MS(25)
      DO  10  I3=1, N          将剪断的每段环数由大到小依次排列于数组MS（25）中
      N1=N+1
      J1=I3+1
```

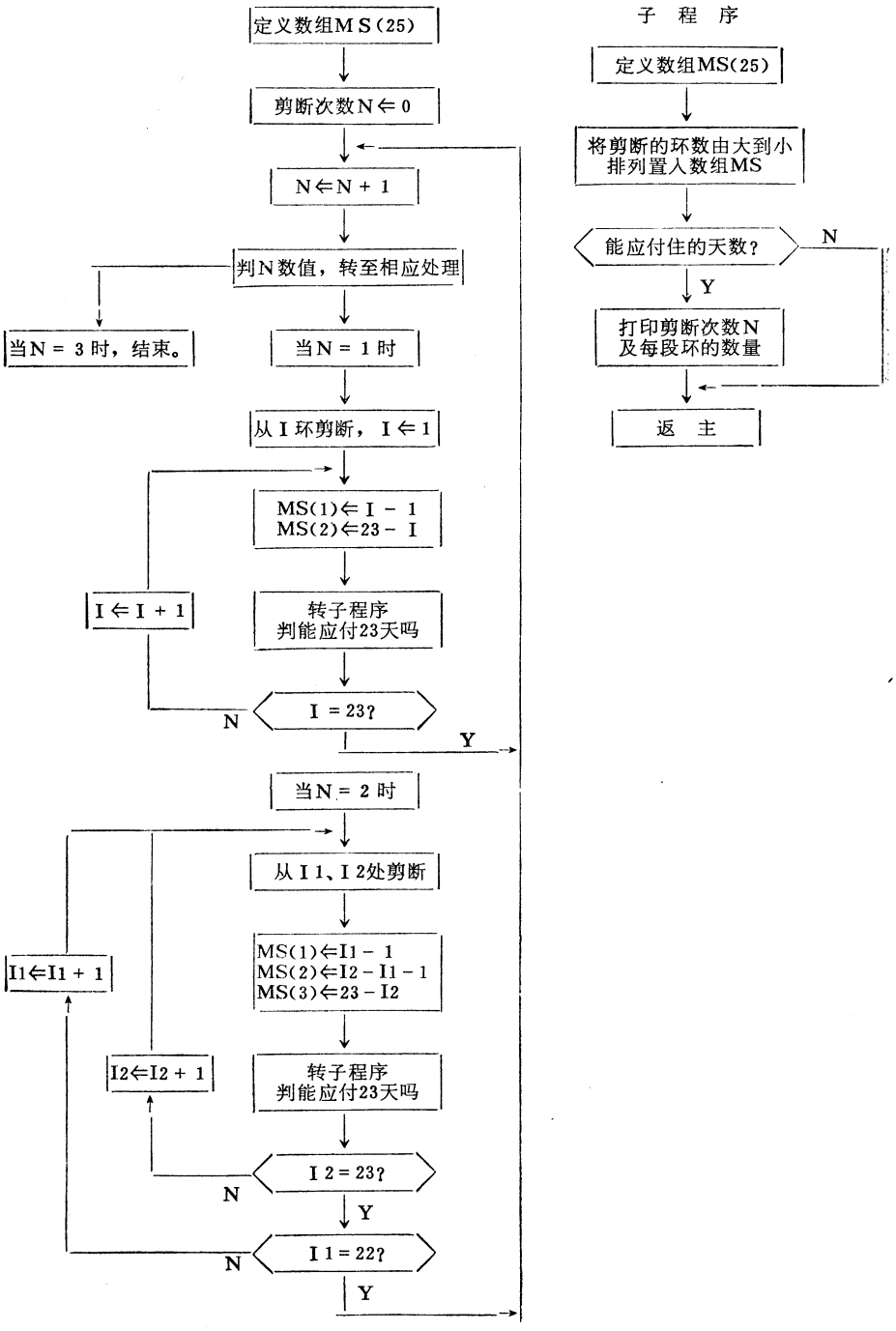

定义数组MS(25)
剪断次数N⇐0
N⇐N+1
判N数值，转至相应处理
当N=3时，结束。
当N=1时
从I环剪断，I⇐1
MS(1)⇐I-1
MS(2)⇐23-I
I⇐I+1
转子程序
判能应付23天吗
I=23?
N
Y
当N=2时
从I1、I2处剪断
MS(1)⇐I1-1
MS(2)⇐I2-I1-1
MS(3)⇐23-I2
I1⇐I1+1
I2⇐I2+1
转子程序
判能应付23天吗
I2=23?
N
Y
I1=22?
N
Y
子 程 序
定义数组MS(25)
将剪断的环数由大到小
排列置入数组MS
能应付住的天数?
N
Y
打印剪断次数N
及每段环的数量
返 主

图 100

```
      DO  20  J = J1, N1
      IF (MS(I3) . GE. MS(J)) GOTO  20
      MA = MS(13)
      MS(I3) = MS(J)
      MS(J) = MA
20    CONTINUE
10    CONTINUE
      DO  30  MQ = 2, 23          住的天数 2～23，有法付环吗，因住一天时剪断的1环
                                  必能付。
      K = MQ
      IF (K. LE. N)  GOTO  30     N为剪断处的环数（即单个环数）
      N1 = N + 1
      DO  40  MP = 1, N1
      IF (K. LT. MS(MP))  GOTO  40天数K＜环数，没法付，再换个小环数。
      K = K - MS(MP)              天数K≥环数，有法付，还余K天找小环数付。
      IF (K. LE. N)  GOTO  30
40    CONTINUE
      RETURN
30    CONTINUE
      WRITE (6, 50)  N            出DO30循环，说明剪断的环已能应付23天。
50    FORMAT (6X, 2HN=, I1, 1H: ) 打印剪断次数N
      N1 = N + 1
      DO  60  I4 = 1, N1
60    WRITE (6, 50)  MS (I4)      打印除单环外，其余段的环数
70    FORMAT (11X, I2)
      END
```

N = 2:　　程序运行结果剪断 2 次，除 2 个单环外，尚有以下三段环的个数：

12　　12环一段

6　　6 环一段

3　　3 环一段

（三）思　考　题

张丽做实验时，发现砝码被借出，老师给了她一长串67个铜环，首尾不相连。每个环都是一克重，只要想法断开其中两个环，将这两个断开的单环取下，便能称 1、2、3、…67克重的重量，请你设法编写程序，将程序运行结果告诉张丽，应该从哪两个环断开，如何利用各段的环来称量 1～67克重物？

提示——在第 6 和第22环断开，使长环分成五部分，即1、5、1、15、45，便能称1～67克重物，称法见下表：

放在天平左端的铜环数	放在天平右端的铜环数	放在天平右端待称的重物(克)
1		1
1 + 1		2
5	1 + 1	3
5	1	4
5		5
5 + 1		6
5 + 1 + 1		7
15	5 + 1 + 1	8
15	5 + 1	9
15	5	10
15 + 1	5	11
15 + 1 + 1	5	12
15	1 + 1	13
15	1	14
15		15
15 + 1		16
...	...	...
45 + 15 + 5 + 1 + 1		67

十五　人工智能“重排九宫”

101　巧排魔方阵

所谓魔方阵是指这样一种方阵，它的每一行和每一列以及两个对角线上的每一元素之和都是相等的。最普通的魔方阵由1到n^2个自然数构成。沿每行、每列及对角线的各数之和为：$1/2\times n(n^2+1)$，称为魔方阵常数。

例如：3×3的魔方阵如右图101-1它的每行、每列及对角线各数之和为$1/2\times 3\times(9+1)=15$。一个n阶魔方阵便是一个n×n的矩阵。每个格中放一个数，共可放n^2个数，请你把数1到数n^2分别放入各格中，且使每行的和、每列的和、以及二个主对角线上的和均相等。（题中要求n为奇数。）

8	1	6
3	5	7
4	9	2

图 101-1

（一）算　法　分　析

举个最简单的例子（3阶方阵）的画法，画一正方形，把它分为九个格，按图101-2的样子依斜线方向将1～9各数顺序填入图内。正方形外的数字填到图内与它相反的那面（仍在原直行或横行），结果得到图101-1的正方形。

由S·de La Loubére 于1687年从泰国传到法国一个算法。该算法的大意是：把第一行中间的位置添入1，然后垂直走到第n行并向左转一个位置，添入2。顺该位置的对角线走，边走边添数，直至第1列。由第1列的该位置起横向走到第n列并向上走一格，添入数。然后顺对角线走到第一行。若重复这样走时遇到已添入数的格子，则向下退一步并继续走。

若把排好的5阶方阵如图101-3画在一张硬纸上，把其两个对角对接起来卷成筒状，就可以发现仍是走对角线。

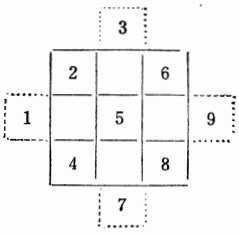


图 101-2

17	24	1	5	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	13	25	2	9

图 101-3

算法描述如下

（1）在第一行中间的位置中添入1，设该位置的列为j；

（2）在〔i_{max}，j－1〕的位置添入2；

（3）由〔i_{max}，j－1〕的位置向左上方沿其对角线走，即下一个位置是〔i－1，j－1〕。每走一步顺序添入数K，K＋1，K＋2，…；

（4）当到达〔i，1〕位置时，横向走到〔i，j_{max}〕，并移一行，即〔i－1，j_{max}〕为〔i，1〕的后继位置，在其中添入数后继续做算法（3）；

（5）当到达〔1，j〕时，垂直走到最下边一行，并向左移一列，即〔i_{max}，j－1〕是〔1，j〕的后继位置，添入数后继续做算法（3）；

（6）若沿对角线走到〔i，j〕位置时，其〔i－1，j－1〕的位置中已添入数，则向下移一行，添入数后继续做算法（3）；

（7）上述算法中$1<i<i_{max}$，$1<j<j_{max}$，而$i_{max}=n$，$j_{max}=n$。

上述是介绍往n阶魔方阵里填n^2个自然数的规律，其原理较复杂，不推论。

（二）程 序 设 计

1）设计思路

为了适合指定的N，采用可调数组FF(9)，数组元素FF(6) 赋于N值。当得到键盘输入的N值后，则按上述算法添写N＊N的二维数组。当添完时，检查各行、各列、主对角线上各元素之和是否相等，若不相等，打印出错（WRONG! ），正确时，打印出所添的N阶方阵，并将魔方阵常数SUM打印出。

本程序可按排N为三位数，我们由键盘输入N＝5、9、21等三个魔方阵，运行正确，请看打印结果。

2）框图　见图101-4

3）源程序及运行结果

```
C***  MAIN  PROGRAM
      INTEGER  AAA(10000)
      OPEN (5, FILE='PL1', ACCESS='WRITE')
      WRITE (1, 3)
 3    FORMAT (3X, 27HTYPE  THE  NUMBER  YOU  WANTED! , /)
 5    READ (1, 10) N
10    FORMAT (I3)
C***  PLACE  NUMBERS  INTO  MATRIX
30    CALL  PLACE (N, AAA(1))
      STOP
      END
C***  SUBROUTINE  1
      SUBROUTINE  PLACE (N, AA)
      INTEGER  AA(N, N)
      CALL  PLACE1(N, AA)
      RETURN
      END
C***  SUBROUTINE  2
      SUBROUTINE  PLACE1 (N, A)
      INTEGER  A(N, N)
      INTEGER  NUM, I, J, TEMP1, TEMP2
      INTEGER  FF(9)
```

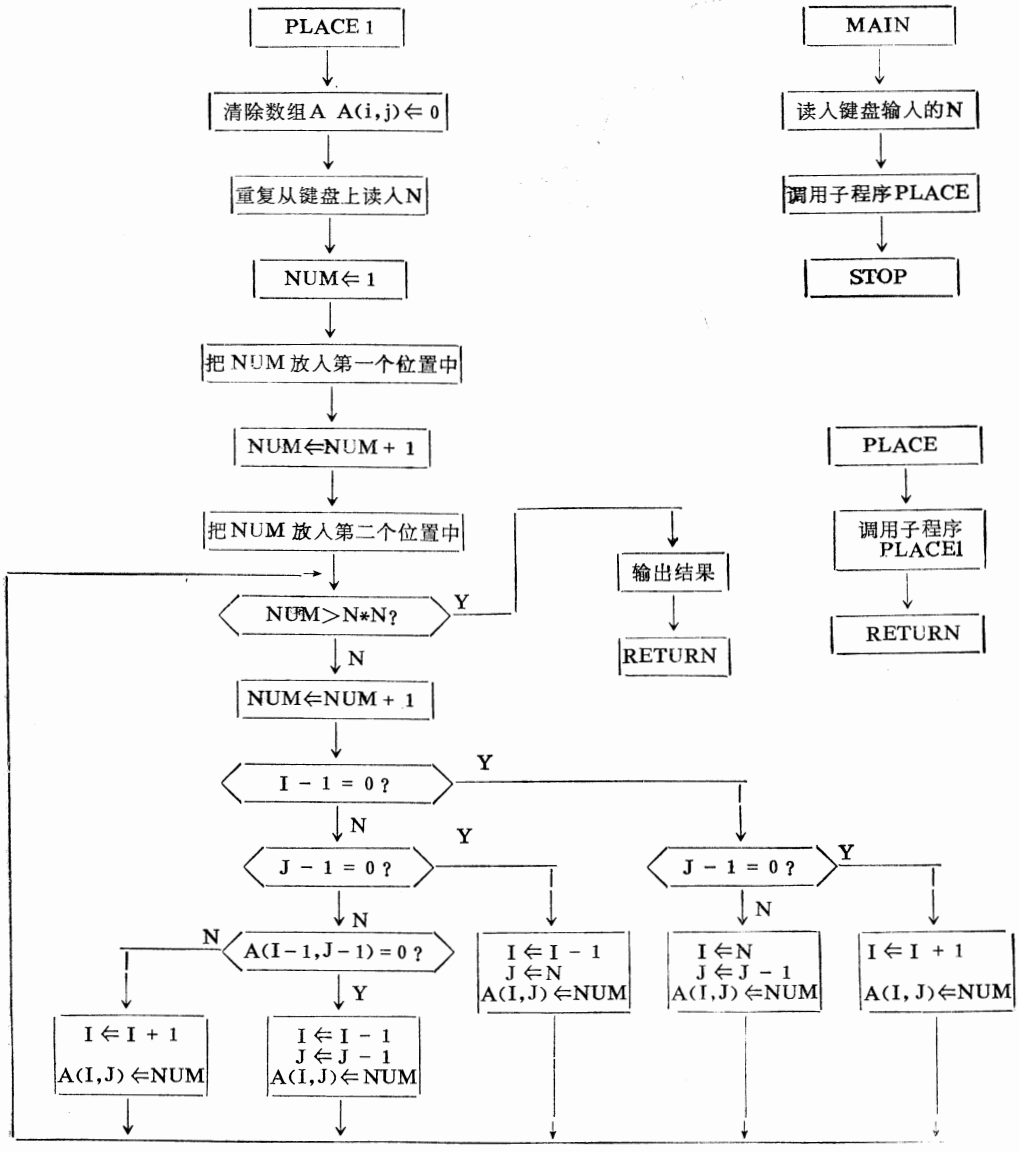


图 101-4

```
      DATA  FF(1), FF(2), FF(3), FF(4), FF(5),
     *   FF(7), FF(8), FF(9)/1H(, 1H/, 1H3,
     *  1HX, 1H,, 1HI, 1H5, 1H)/
C***  INITIAT  A
      DO  60  I=1, N
      DO  60  J=1, N
60    A(I, J)=0
      WRITE (1, 61)
61    FORMAT (3X, 40HTYPE THE NUMBER OF
     *   SPECIFICATION WHICH=N,1
```

给出可调数组的数组说明

数组A的各元素清0

请求再一次读入N

```
      READ (1, 62) FF(6)
62    FORMAT (A3)
C***  PLACE  FIRST  NUMBER
      J = IFIX(N/2) + 1                          定出中间一列的列号
      I = 1
      NUM = 1
      A(I, J) = NUM                              添入第一个数
C***  PLACE  SECOND  NUMBER
      NUM = NUM + 1                              \
      I = N                                      |
      J = J - 1                                  |添入第二个数
      A(I, J) = NUM                              /
70    IF (NUM. EQ. N*N) GO TO 210
      NUM = NUM + 1
      IF ((I - 1). EQ. 0) GO TO 80               \
      IF ((J - 1). EQ. 0) GO TO 150              |在I≠1, J≠1且对
      IF (A(I - 1, J - 1). NE. 0) GO TO 200      /角线的下一个位置未
      I = I - 1                                  \
      J = J - 1                                  |添入数，则添上相应
      A(I, J) = NUM                              /数
      GO TO 70
80    IF ((J - 1). EQ. 0) GO TO 100              \
      I = N                                      |
      J = J - 1                                  |算法 (5)
      A(I, J) = NUM                              |
      GO TO 70                                   /
100   I = I + 1                                  \
      A(I, J) = NUM                              |I = 1, J = 1时的处理
      GO TO 70                                   /
150   I = I - 1
      J = N                                      \
      A(I, J) = NUM                              |算法 (4)
      GO TO 70                                   /
200   I = I + 1                                  \
      A(I, J) = NUM                              |算法 (6)
      GO TO 70                                   /
210   WRITE (1, 220) N                           \
      WRITE (5, 220) N                           |
220   FORMAT (2X, 22HTHIS IS THE RESULT          |
         (N=, I3, 1H), )                         |输出结果
      WRITE (1, FF) ((A(I, J), J = 1, N), I = 1, N)
      WRITE (5, FF) ((A(I, J), J = 1, N), I = 1, N)
```

```
      I = 1
      TEMP1 = 0
      TEMP3 = 0
      DO  250  J = 1, N
      TEMP1 = TEMP1 + A(1, J)
250   TEMP3 = TEMP3 + A(J, J)
      IF (TEMP1. NE. TEMP3) GO TO 280
      DO  275  I = 2, N
      TEMP2 = 0
      DO  270  J = 1, N
270   TEMP2 = TEMP2 + A(I, J)
      IF  (TEMP1. NE. TEMP2)  GO  TO  280
275   CONTINUE
      WRITE(1, 277) TEMP1
277   FORMAT (2X, 4HSUM = , I4, /)
      GO  TO  300
280   WRITE  (1, 290)
290   FORMAT  (5X, 5HWRONG! )
300   RETURN
      END
```

检查结果TEMP1、TEMP2、TEMP3 分别是列、行、主对角线元素之和

检查结果正确，打印其中之一，表示之。

若检查行、列、对角线之和不等，则给出“WRONG”（错）指示

```
THIS  IS  THE  RESULT (N = 5)
      17   24    1    8   15
      23    5    7   14   16
       4    6   12   20   22
      10   12   19   21    2
      11   18   25    2    9
SUM = 65

THIS  IS  THE  RESULT (N = 9)
      47   58   69   80    1   12   23   34   45
      57   68   79    9   11   22   33   44   46
      67   78    8   10   21   32   43   54   56
      77    7   18   20   31   42   53   55   66
       6   17   19   30   41   52   63   65   76
      16   27   29   40   51   62   64   75    5
      26   28   39   50   61   72   74    4   15
      36   38   49   60   71   73    3   14   25
      37   48   59   70   81    2   13   24   35
SUM = 369

THIS  IS  THE  RESULT  (N = 21)
  231 208 185 162 139 116  93  70  47  24   1 440 417 394 371 348 325 302 279 256 233
  232 230 207 184 161 138 115  92  69  46  23  21 439 416 393 370 347 324 301 278 255
```

254	252	229	206	183	160	137	114	91	68	45	22	20	438	415	392	369	346	323	300	277
276	253	251	228	205	182	159	136	113	90	67	44	42	19	437	414	391	368	345	322	299
298	275	273	250	227	204	181	158	135	112	89	66	43	41	18	436	413	390	367	344	321
320	297	274	272	249	226	203	180	157	134	111	88	65	63	40	17	435	412	389	366	343
342	319	296	294	271	248	225	202	179	156	133	110	87	64	62	39	16	434	411	388	365
364	341	318	295	293	270	247	224	201	178	155	132	109	86	84	61	38	15	433	410	387
386	363	340	317	315	292	269	246	223	200	177	154	131	108	85	83	60	37	14	432	409
408	385	362	339	316	314	291	268	245	222	199	176	153	130	107	105	82	59	36	13	431
430	407	384	361	338	336	313	290	267	244	221	198	175	152	129	106	104	81	58	35	12
11	429	406	383	360	337	335	312	289	266	243	220	197	174	151	128	126	103	80	57	34
33	10	428	405	382	359	357	334	311	288	265	242	219	196	173	150	127	125	102	79	56
55	32	9	427	404	381	358	356	333	310	287	264	241	218	195	172	149	147	124	101	78
77	54	31	8	426	403	380	378	355	332	309	286	263	240	217	194	171	148	146	123	100
99	76	53	30	7	425	402	379	377	354	331	308	285	262	239	216	193	170	168	145	122
121	98	75	52	29	6	424	401	399	376	353	330	307	284	261	238	215	192	169	167	144
143	120	97	74	51	28	5	423	400	398	375	352	329	306	283	260	237	214	191	189	166
165	142	119	96	73	50	27	4	422	420	397	374	351	328	305	282	259	236	213	190	188
187	164	141	118	95	72	49	26	3	421	419	396	373	350	327	304	281	258	235	212	210
209	186	163	140	117	94	71	48	25	2	441	418	395	372	349	326	303	280	257	234	211

SUM = 4641

（三）思 考 题

有一个用 1～81这八十一个数排成的九次方阵（横、竖各九格），依次去掉它的最外面一层后，分别得到七次方阵、五次方阵、三次方阵，且正中一个数刚好是上面各方阵（三次、五次、七次、九次）的每行或每列、每条对角线上的各数之和的最大公约数，请问这是怎样的一个九次方阵？

102 在方阵中寻找规律

1	2	3	4	5	6	7	8	9	10
20	19	18	17	16	15	14	13	12	11
21	22	23	24	25	26	27	28	29	30
40	39	38	37	36	35	34	33	32	31
41	42	43	44	45	46	47	48	49	50
60	59	58	57	56	55	54	53	52	51
61	62	63	64	65	66	67	68	69	70
80	79	78	77	76	75	74	73	72	71
81	82	83	84	85	86	87	88	89	90
100	99	98	97	96	95	94	93	92	91

图 102-1

在 10 阶方阵里已填完 1～100 数字，如图 102-1所示，要求：

（1）计算每行各数字之和是多少，是否有规律？

（2）每一列各数字之和等于多少？

（3）从上到下，每四个小方格数字之和最小者是多少？最大者是多少？

（一）笔算步骤和结果

1）方阵中，从上至下，每行数字之和为 55，155，255，…，955。其规律是后一行比前一行递增 100，这是因为后一行的数比前一行的每个数增加10后形成的，只是次序排列不同而已。

2）计算后得出，每一列的和都是505，这是因为奇数行的数从左到右逐列增1，而偶数行的数却是从左到右逐列减1。加1减1互相抵消了，故使每列的和相等。

3）四个数字之和最大的正方形一定在下面几排，而和最小的正方形一定在上面几排。

（1）最大的和是362，这样的正方形有九个：

81	82	82	83	……	89	90
100	99	99	98		92	91

（2）最小的和是42，这样的正方形也有九个：

1	2	2	3	……	9	10
20	19	19	18		12	11

（二）程 序 设 计

1）设计思路

定义二维数组IA（10，10），放入1～100的数字，利用循环语句赋值，赋值后打印出该方阵，以便观察输入对否。计算每行之和和每列之和并打印出结果。

定义二维数组IB（9，9），计算每四个小方格里数字之和，放入IB（9，9）数组中，并将所有四个小方格里数字之和进行比较，将最小者放入MIN单元，最大者放入MAX单元，最后也打印出。

2）框图　见图102-2

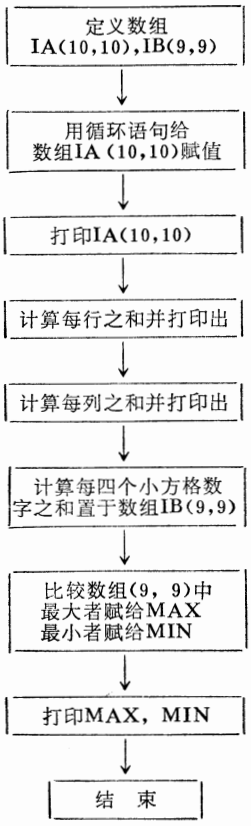


图 102-2

3）源程序及运行结果

```
      DIMENSION  IA(10, 10), IB(9, 9)
      I = 1
      DO  5  K = 1, 10
5     IA(I, K) = K
      K = 10
      DO  10  I = 2, 10
      IF  (I. NE. I/2*2)  GOTO  20
      L = 11
      DO  15  J = 1, 10
      K = K + 1
      L = 11 - J
15    IA(I, L) = K
      GOTO  10
20    DO  25  J = 1, 10
      IA(I, J) = K
25    K = K + 1
10    CONTINUE
      WRITE  (10, 28)  ((IA(I, J), J = 1, 10), I = 1, 10)
28    FORMAT  ((5X, 10(I4))/)
      DO  30  I = 1, 10
      IW = 0
      DO  35  J = 1, 10
35    IW = IW + IA(I, J)
      WRITE  (10, 40)  I, IW
40    FORMAT  (2X, 2HI = , I2, 5X, 4HSUM = , I3)
30    CONTINUE
      DO  50  J = 1, 10
      IW = 0
      DO  62  I = 1, 10
62    IW = IW + IA(I, J)
      WRITE (10, 65)  J, IW
65    FORMAT  (2X, 2HJ = , I2, 5X, 4HSUM = , I3)
50    CONTINUE
      DO  70  I = 1, 9
      DO  70  J = 1, 9
70    IB(I, J) = IA(I, J) + IA(I + 1, J) + IA(I, J + 1) + IA(I + 1, J + 1)
      MIN = 0
      MAX = 0
      DO  80  I = 1, 9
      DO  80  J = 1, 9
      IF  (MIN. GT. IB(I, J))GOTO  85
```

```
      MIN=IB(I, J)
      GOTO 80
85    IF (IB(I, J). LT. MAX) GOTO 80
      MAX=IB(I, J)
80    CONTINUE
      WRITE (10, 90) MIN, MAX
90    FORMAT (10X, 4HMIN=, I3/10X, 4HMAX=, I3)
      STOP
      END
```

1	2	3	4	5	6	7	8	9	10	打印出利用循环语句
20	19	18	17	16	15	14	13	12	11	输入的矩阵， 观察对否。
21	22	23	24	25	26	27	28	29	30	
40	39	38	37	36	35	34	33	32	31	
41	42	43	44	45	46	47	48	49	50	
60	59	58	57	56	55	54	53	52	51	
61	62	63	64	65	66	67	68	69	70	
80	79	78	77	76	75	74	73	72	71	
81	82	83	84	85	86	87	88	89	90	
100	99	98	97	96	95	94	93	92	91	

程序运行时打印结果

```
I = 1     SUM =  55   第一行各数之和
I = 2     SUM = 155   第二行各数之和
I = 3     SUM = 255   ……
I = 4     SUM = 355
I = 5     SUM = 455
I = 6     SUM = 555
I = 7     SUM = 655
I = 8     SUM = 755
I = 9     SUM = 855
I = 10    SUM = 955   第10行各数之和
J = 1     SUM = 505   第 1 列各数之和
J = 2     SUM = 505   ……
J = 3     SUM = 505
J = 4     SUM = 505
J = 5     SUM = 505
J = 6     SUM = 505
J = 7     SUM = 505
J = 8     SUM = 505
J = 9     SUM = 505
J = 10    SUM = 505   第10列各数之和
          MIN = 42    四个方格数之和最小者
          MAX = 362   四个方格数之和最大者
```

（三）思 考 题

1）试计算主题目方阵中每四个小方格构成的小正方形共有多少个。

2）验算每四个小方格里数字之和都是偶数吗？为什么？

103 对调四阶方阵中数字

如图103-1所示，是一个四阶方阵，每个方格里填有从1到16，共16个自然数。你能将16个数对调某几个数，使每一行、每一列以及两条对角线上的四个数的和都相等吗？

（一）笔算步骤和结果

因为从1到16这十六个数字之和为：$(1+16)/2\times16=136$ 所以每一行、每一列以及两条对角线上的四个数的和为：

$$136\div4=34$$（见图103-2）

而两条对角线上四个数之和均满足要求，剩下的8个数中，是以中心对称的两个数互换位置：2—15，3—14，5—12，8—9。得到图103-2的四阶方阵便满足了要求。

这是从观察中发现的，以此作为特例进行程序设计。

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

图 103-1

1	15	14	4
12	6	7	9
8	10	11	5
13	3	2	16

图 103-2

（二）程 序 设 计

1）设计思路

由观察出来的算法分析，是将四阶方阵中16个数中四对数互换位置，便可得到题目要求的新方阵中的各行、各列、两个主对角线上的元素之和均等于34。如果将观察出来的元素下标直接赋值交换，将使程序设计毫无意义。我们利用循环语句将数组NA（4，4）冲入相应的数做为原始排列，并打印出。而后利用循环语句巧妙地配合，将这四对元素交换位置，形成新矩阵，检验新矩阵的每行、每列、两个主对角线上的元素之和是否全等于34，并打印出。

2）框图 见图103-3

3）源程序及运行结果

```
C       THIS IS A PROBLEM OF CHANGING
        DIMENSION NA(4, 4)
        N = 1
        DO 1 I = 1, 4
        DO 1 J = 1, 4
        NA(I, J) = N
1       N = N + 1
        WRITE(6, 5) ((NA(I, J), J = 1, 4), I = 1, 4)
```

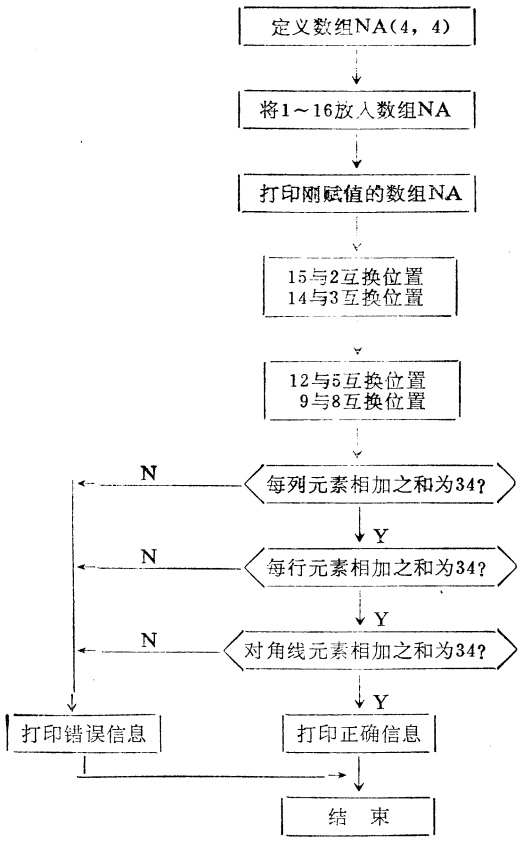


图 103-3

```
5       FORMAT ((1X, 4(I7))/)
        I = 1
        DO  2  J = 1, 2
        MM = 4 - J
        W = NA(I, J + 1)
        NA(I, J + 1) = NA(4, MM)                    15→2  14→3
2       NA(4, MM) = W                               2→15  3→14
        I = 2
        DO  3  J = 1, 4, 3
        MM = 5 - J
        W = NA(I, J)
        NA(I, J) = NA(I + 1, MM)                    12→5  9→8
3       NA(I + 1, MM) = W                           5→12  8→9
        DO  7  I = 1, 4
        DO  70  J = 1, 4
70      M(I) = M(I) + NA(I, J)                      列元素相加之和应等于34
        IF(M(I). NE. 34) GO  TO  8
7       CONTINUE
```

```
      DO   9   I=5, 8                                  ┐
      DO   10  J=1, 4                                  ├ 行元素相加之和应等于34
10    M(I)=M(I)+NA(J, I-4)                             │
      IF(M(I). NE. 34) GO TO 8                         ┘
 9    CONTINUE
      DO  30  J=1, 4                                   ┐
      M(9)=M(9)+NA(J, J)                               ├ 两条主对角元素相加
      K=5-J                                            ┘
30    M(10)=M(10)+NA(J, K)
      IF(M(9). EQ. 34. AND. M(10). EQ. 34) GO TO 11
 8    WRITE (6, 20)
20    FORMAT(1X, 26H*THE CHANGE IS NOT RIGHT. *)
11    WRITE (6, 4) ((NA(I, J), J=1, 4), I=1, 4)
 4    FORMAT ((1X, 4HP=34, 4X, 4(I7))/)
      STOP
      END
```

```
 1    2    3    4
 5    6    7    8        原矩阵排列
 9   10   11   12
13   14   15   16
```

```
P=34    1   15   14    4
P=34   12    6    7    9        对调后的矩阵
P=34    8   10   11    5
P=34   13    3    2   16
```

（三）思 考 题

本程序是据观察时发现的规律，作为特例处理的，请你改变成通用方法，将1～16数字重新排列在四阶矩阵里，使每行、每列、主对角线元素之和全相等。

104 人工智能“重排九宫”

从图104-1通过逐格移动变成图104-2的方法称为“重排九宫”。它是由我国古代传入欧洲曾风靡一时的“移动十五”简化而来的。重排九宫和过去流传的“华容道”（现称“船坞排挡”）游戏相类似，但它们并非是一种单纯的游戏，而是有一定实用意义的。实际上它们是“人工智能”的一个研究专题。很久以来，吸引了许多有关专家学者的注意和探索。

所谓“重排九宫”具体是指八个数字按序放在九个格子中，其中余下一个格子为空（见图104-1），任意移动与空格相邻的二个数字之一，则形成八个数在九个格子中的新的布局，并有一个新的空格，然后再移动与空格相邻的一个数字。如此继续下去，经过若干次移动后，可得图104-2所示的布局。

十九世纪数学家亨利·杜特尼曾研究过这个问题，并提出了公认的较好走法，共需36

步，即：1 2 5 4 3 1 2 3 7 6 1 2 3 7 6 1 2 3 7 5 4 8 1 2 3 6 5 7 6 5 8 4 7 8 5 6。序列中的数字表示顺序移动的格子中的数字。

为了让读者直截了当地看清楚这36步的走法，我们把36步分为六组，每组用一个图来对应：125431（图104-3），237612（图104-4），376123（图104-5），754812（图104-6），365765（图104-7），847856（图104-8）。

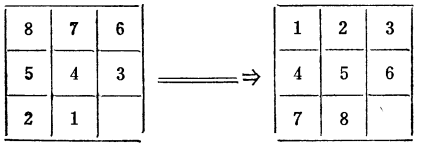


图 104-1　　图 104-2

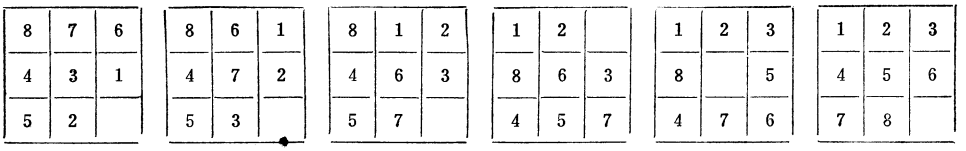


图 104-3　图 104-4　图 104-5　图 104-6　图 104-7　图 104-8

那么，36步是否是最少步数吗？并没到此止步。现代的技术已帮助我们找到比亨利·杜特尼提出的方法更佳的移动方法，仅需30步则完成重排九宫，且有10种走法。据说世界上对这个问题进行过测验，但似乎没有一个人找全了这10种走法。你是否能找找30步的走法？能找出几种这样走法？

这个问题当初在国外一家杂志上提出来以后，大约经过3个月的时间，虽然收到了大量读者来信，但没有一个读者能找到全部10种走法。

英国爱丁堡大学的一位数学家，曾在IBM7094计算机找出了10种走法。

我们仅知道结果有10种，但不知何种算法，更不知思路和源程序。我们探索算法进行程序设计，终于找到了这30步10种走法的最优解。详见程序运行后输出打印结果。

（一）算 法 分 析

1）首先把图104-1画成图104-9的形式。图104-9中的数字表示格子的编号，格子中带括号的数字表示图104-1中的数字分布。弧线箭头表示当某个位置为空时，相邻格子中可移到此空格的位置。如：若4号格为空，则可从5号、7号、1号三个格子中取其一种进行移动。

把图104-9中的弧线关系表示成表104的形式。

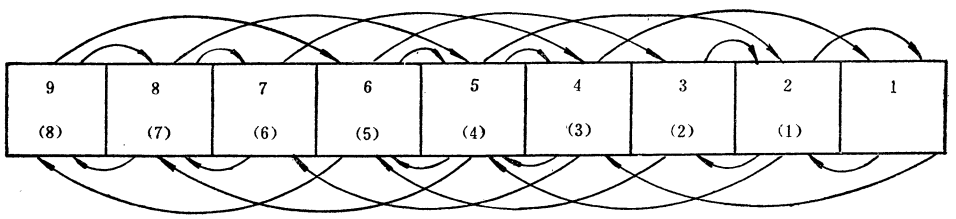


图 104-9

表 104

位　　置	相　邻　位　置	合　　计
1	2.4.	2
2	1.3.5.	3
3	2.6.	2
4	1.5.7.	3
5	2.4.6.8.	4
6	3.5.9.	3
7	4.8.	2
8	5.7.9.	3
9	6.8.	2

有了图104-9和表104后，就可以进行查找了。

2）移动的方法

对于每个空位置，例如位置1，查图104-9，可知有二种移动可能，任选一种，设为位置2，并记住选了哪种，然后把位置2中的数字送到位置1中。此时位置2为空，查表104，然后继续移动。

3）具体算法

我们采用搜索试探法。查找一种走法的算法为：

（1）为了保存走过的路径，设置数组PATH（40，2），PATH(I，1）中放第I步选择的相邻位置编号，PATH(I，2）中放该位置中的值。变量COUNT为移动次数计数器。数组MARK(40）用于保存与表104有关的值，在第I步选中一个相邻位置后，MARK(I)⇐MARK(I)+1，则再次退回第I步时，应选另一个相邻位置。

（2）选第一步移动的位置。

（3）对于第I步的空位置J(1≤J≤9)，由表104和MARK(I)，选取相应的相邻位置K，把K位置中的数送到J位置中，COUNT⇐COUNT+1。

（4）检查。

a）每得到一种新的分布后，首先检查是否与最终结果相同。若相同，则输出本次走法。

b）若与最终结果比较后不同，则要检查COUNT的值，若COUNT=30，则说明此路不通，要退一步（算法（5）），COUNT⇐COUNT-1，重新查找。若COUNT＜30，则继续走下一步，执行算法（3）。

（5）退一步的算法。退一步后，查表104，选未走过的位置继续走，执行算法(3)。若MARK(COUNT）表明其相邻的几个位置均走过（即MARK(COUNT）的值与表104最右边一列中合计数字相等），则要继续退一步，COUNT⇐COUNT－1，返回算法（5）处重新查找。

从上可以看出，此算法为指数型算法。若每一步有N种选择，则走30步要进行N^{30}次运算。

（二）程序设计

1）为了减少机时的使用时间，仅限于查找30步的10种走法。

2）除了PATH(40，2)、MARK(40）外，另用A(40，9）保存每次移动后九个位置中数的分布情况，用PLACE(9，5）保存表104，SB(10，41）保存找到的10次结果，C(9）保存最终应得的结果图104-2。

3）对这10种走法进行分类，例如，以下5种走法可分为二类。第一类的3种走法中，头

第一类	1 4	3 1 4 2 5 8 7 3 1 6 3 1 2 5 8 7 1 2 5 4 6 5 4 8 7 4 5 6
	1 4	5 8 7 5 3 6 5 3 4 1 6 5 3 4 1 2 8 7 4 1 2 8 7 4 1 2 5 6
	1 4	7 8 5 2 4 7 8 6 3 8 6 5 2 4 7 1 8 6 1 7 4 1 5 2 1 4 7 8
第二类	1 2 5 8 7 4	3 1 2 5 8 7 4 3 1 6 3 1 5 2 6 5 2 8 7 4 1 2 5 6
	1 2 5 8 7 4	8 5 2 8 3 1 8 2 5 7 4 3 1 6 3 1 2 5 7 4 1 2 5 8

2步1、4（如方框所圈）是不变的，仅从第3步找起。而第二类的2种走法中，头6步1、2、5、8、7、4是相同（如方框所圈），仅需从第7步找起。这样共分为四类。分别用D1(7)、D2(3)、D3(9)、D4(7）保存其头几步。

4）对于要查找的走法，首先按规定的头几步走，然后按算法找出后面的走法。

5）框图　见图104-10(见下页)、图104-11、图104-12。

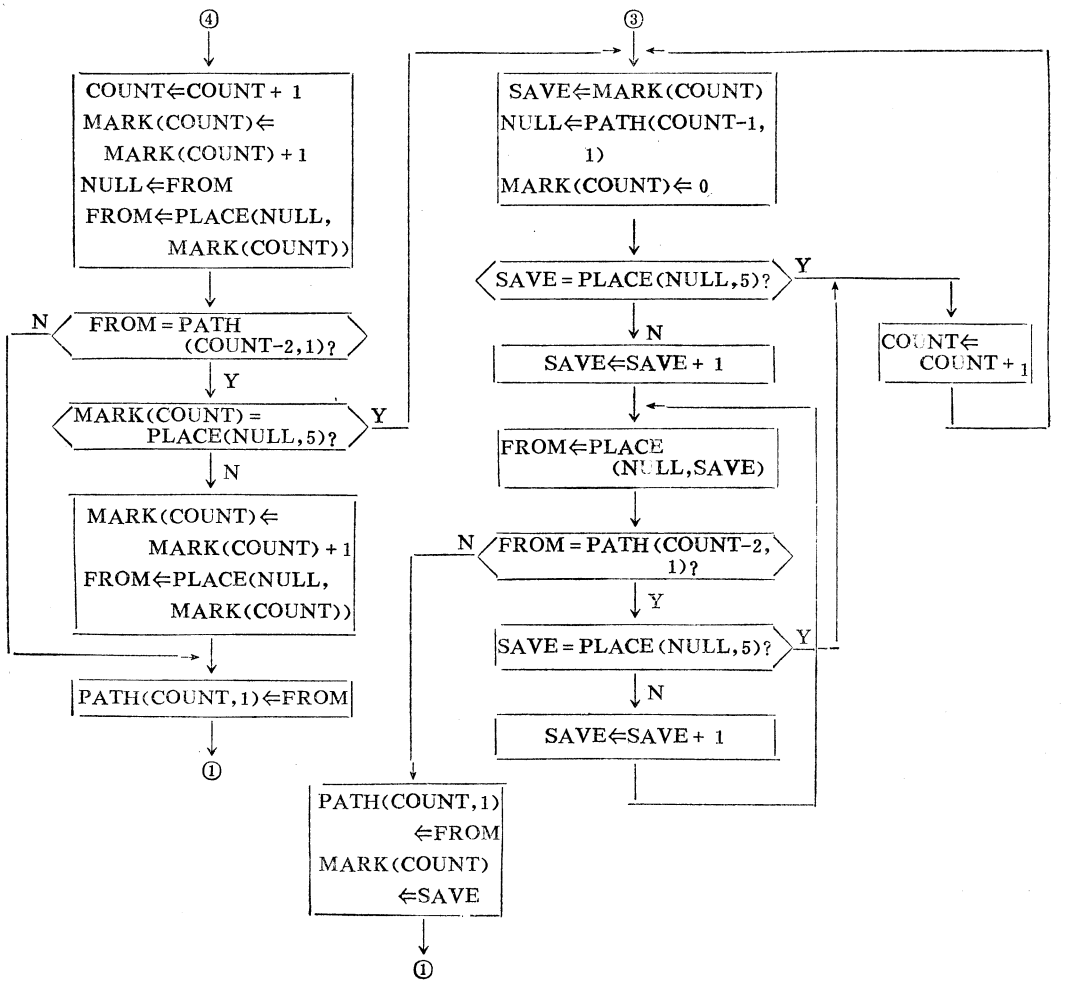


图 104-11

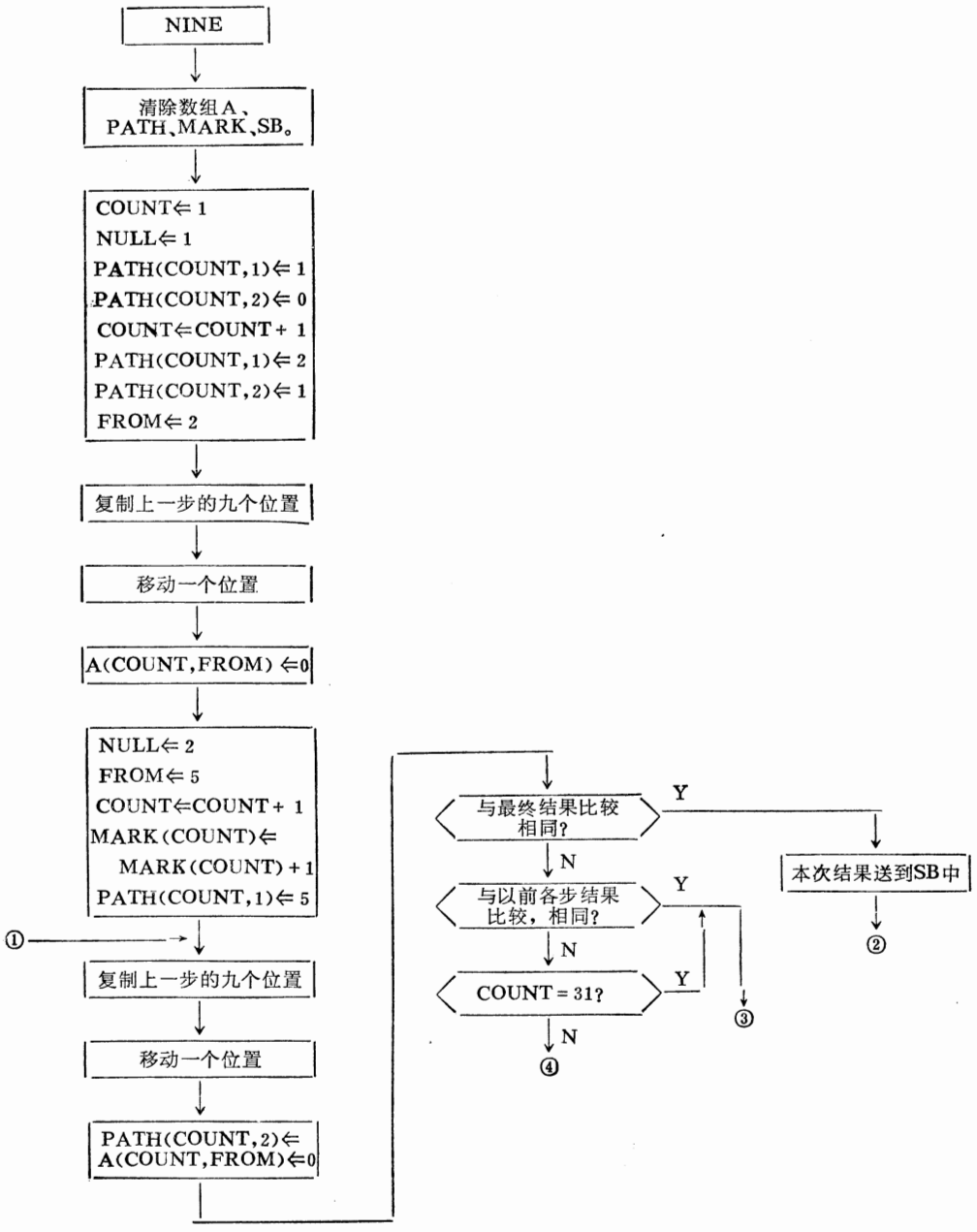

NINE
清除数组A、PATH、MARK、SB。
COUNT⇐1
NULL⇐1
PATH(COUNT,1)⇐1
PATH(COUNT,2)⇐0
COUNT⇐COUNT+1
PATH(COUNT,1)⇐2
PATH(COUNT,2)⇐1
FROM⇐2
复制上一步的九个位置
移动一个位置
A(COUNT,FROM)⇐0
NULL⇐2
FROM⇐5
COUNT⇐COUNT+1
MARK(COUNT)⇐
MARK(COUNT)+1
PATH(COUNT,1)⇐5
①
复制上一步的九个位置
移动一个位置
PATH(COUNT,2)⇐
A(COUNT,FROM)⇐0
与最终结果比较相同?
Y
本次结果送到SB中
②
N
与以前各步结果比较，相同?
Y
③
N
COUNT=31?
Y
N
④

图 104-10

6）源程序及运行结果

```
C***  PROGRAM  NAME : NINE
C***  THIS  PROGRAM  REARAIGE  THE  9  PLACES
C***
      INTEGER  A(40, 9), PATH(40, 2), MARK(40), PLACE(9, 5), C(9),
     SB(10, 41)
      INTEGER  D1(7), D2(3), D3(9), D4(7)
      INTEGER  COUNT, NULL, FROM, I, J, SAVE, NUM
C***  INITIALIZATION
      DATA  C(1), C(2), C(3), C(4), C(5), C(6), C(7), C(8), C(9)/0, 8, 7, 6,
     5, 4, 3, 2, 1/
      DATA  PLACE(1, 1), PLACE(2, 1), PLACE(3, 1), PLACE(4, 1),  PLACE
     (5, 1), PLACE(6, 1),
     1  PLACE(7, 1), PLACE(8, 1), PLACE(9, 1), PLACE(1, 2),  PLACE(2, 2),
     PLACE(3, 2),
     2  PLACE(4, 2), PLACE(5, 2), PLACE(6, 2), PLACE(7, 2),  PLACE(8, 2),
     PLACE(9, 2),
     3  PLACE(1, 3), PLACE(2, 3), PLACE(3, 3),  PLACE(4, 3), PLACE(5, 3),
     PLACE(6, 3),
     4  PLACE(7, 3), PLACE(8, 3), PLACE(9, 3), PLACE(1, 4),  PLACE(2, 4),
     5  PLACE(3, 4), PLACE(4, 4), PLACE(5, 4), PLACE(6, 4), PLACE(7, 4),
     PLACE(8, 4),
     6  PLACE(9, 4), PLACE(1, 5), PLACE(2, 5), PLACE(3, 5), PLACE(4, 5),
     PLACE(5, 5),
     7  PLACE(6, 5), PLACE(7, 5), PLACE(8, 5), PLACE(9, 5)/2, 1, 2, 1, 2, 3,
      4, 5, 6, 4, 3, 6, 5, 4, 5, 8, 7, 8, 0, 5, 0, 7, 6, 9, 0, 9, 0, 4* 0, 8, 4 *0,
     2, 3, 2, 3, 4, 3, 2, 3, 2/
      DATA  A(1, 1), A(1, 2), A(1, 3), A(1, 4), A(1, 5), A(1, 6), A(1, 7), A
     (1, 8), A(1, 9)/0, 1, 2, 3, 4, 5, 6, 7, 8/
      DATA  D1(1), D1(2), D1(3), D1(4), D1(5), D1(6), D1(7), D2(1), D2(2),
     D2(3),
     1  D3(1), D3(2), D3(3), D3(4), D3(5), D3(6), D3(7), D3(8), D3(9), D4(1),
     D4(2),
     2  D4(3), D4(4), D4(5), D4(6), D4(7)/1, 2, 3, 6, 9, 8, 5, 1, 2, 5, 1, 4, 5,
     8, 9, 6, 3, 2, 5, 1, 4, 5, 6, 3, 2, 5/
C***
      OPEN(4, FILE = 'NA', ACCESS = 'WRITE')
```

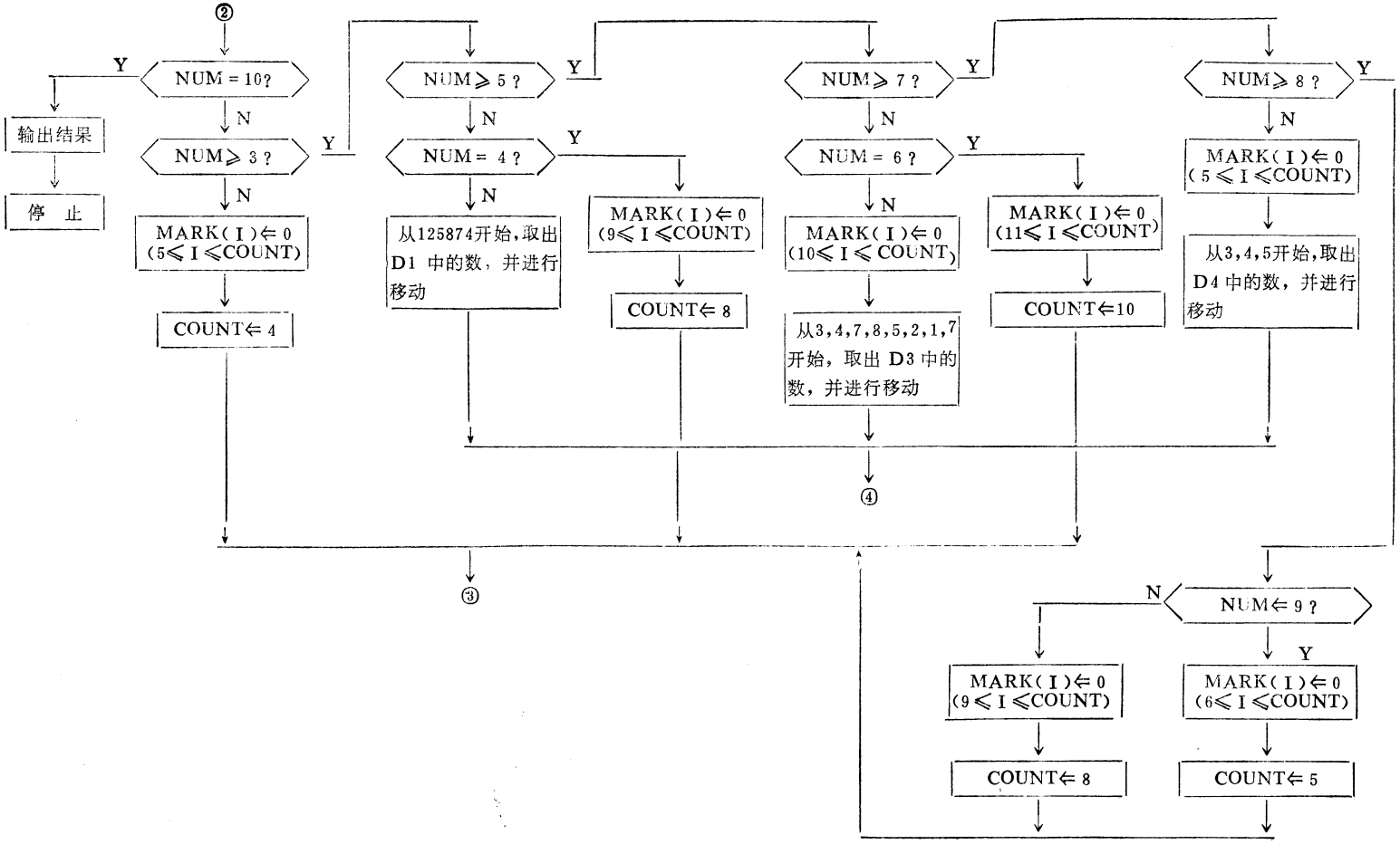

②
NUM = 10?
Y
输出结果
停 止
N
NUM≥ 3 ?
Y
N
MARK(I)⇐ 0
(5≤ I ≤COUNT)
COUNT⇐ 4
NUM≥ 5 ?
Y
N
NUM = 4 ?
Y
N
从125874开始，取出D1 中的数，并进行移动
MARK(I)⇐ 0
(9≤ I ≤COUNT)
COUNT⇐ 8
NUM≥ 7 ?
Y
N
NUM = 6 ?
Y
N
MARK(I)⇐ 0
(10≤ I ≤COUNT)
从3,4,7,8,5,2,1,7开始，取出 D3 中的数，并进行移动
MARK(I)⇐ 0
(11≤ I ≤COUNT)
COUNT⇐10
NUM≥ 8 ?
Y
N
MARK(I)⇐ 0
(5≤ I ≤COUNT)
从3,4,5开始，取出D4 中的数，并进行移动
④
③
N
NUM⇐ 9 ?
Y
MARK(I)⇐ 0
(9≤ I ≤COUNT)
COUNT⇐ 8
MARK(I)⇐ 0
(6≤ I ≤COUNT)
COUNT⇐ 5

图 104-12

```
C***  CLEAR  A(I, J), PATH(I, J), WARK(I)      清除数组A、PATH、MARK
      DO  5  I=1, 10
      DO  5  J=1, 41
5     SB(I, J)=0
      DO  10  I=2, 40
      DO  10  J=1, 9
10    A(I, J)=0
      DO  13  I=1, 40
13    MARK(I)=0
      DO  15  I=1, 40
      DO  15  J=1, 2
15    PATH(I, J)=0
C***  ''NUM''REPRESENTS  THE  WAYS  FOUND  变量NUM用于对已找到的方
                                           法计数
      NUM=0
C***  FIRST  WAY                                找第一种方法
C***  FIRST  STEPS, BEGINNING  FROM  1, 4
      COUNT=1
C***  ''NULL''IS  THE  NUMBER  OF  PLACE  INTO  WHICH  TO  BE  MO-
        VED                      NULL表示空位置
      NULL=1
C***  ''PATH (COUNT, 1)''  SAVES  THE  NUMBER  OF  THE  PLACE  AT
        ''CONUT''TH
C***  TIME  MOVE
C***  ''PATH(COUNT, 2)''  SAVES  THE  CONTENT  OF  THE  PLACE  AT
C***  ''CONUT''TH  TIME  MOVE
      PATH(COUNT, 1)=1
      PATH(COUNT, 2)=0
      COUNT=COUNT+1
      PATH(COUNT, 1)=2
      PATH(COUNT, 2)=1
C***  ''FROM''IS  THE  NUMBER  OF  PLACE  FROM  WHICH  TO  BE  MO-
        VED              FROM表示要移动的相邻位置
      FROM=2
C***  FIRST  STEP, MOVE  IT
C***
        DO  20  J=1, 9
20    A(COUNT, J)=A(COUNT-1, J)
      A(COUNT, NULL)=A(COUNT, FROM)
      A(COUNT, FROM)=0
C***  SECOND  STEP, MOVE  IT
      NULL=2
      FROM=5
```

```
      COUNT = COUNT + 1
      MARK(COUNT) = MARK(COUNT) + 1
      PATH(COUNT, 1) = 5
25    DO 30 J = 1, 9
30    A(COUNT, J) = A(COUNT - 1, J)                    ┐
      A(COUNT, NULL) = A(COUNT, FROM)                  │
      PATH(COUNT, 2) = A(COUNT, FROM)                  │ 移动位置，添入PATH
      A (COUNT, FROM) = 0                              ┘
C***  COMPARE IT WITH THE FINAL RESULT                  ┐
      DO 40 J = 1, 9                                   │ 与最终结果比较
      IF (A(COUNT, J). NE. C(J)) GO TO 50              ┘
40    CONTINUE
C***  FOUND A METHOD                                   已找到一种方法
C***
      WRITE (1, 41) COUNT, (PATH(J, 2), J = 1, COUNT)
41    FORMAT (2X, 'NUMBER = ', I2, 4X, 34I2)
C***  COUNTER + 1, AND SAVE THE STEPS MOVED
      NUM = NUM + 1
      DO 45 J = 1, COUNT
45    SB(NUM, J) = PATH(J, 2)                          ┐
      SB(NUM, 41) = COUNT                              │ 保存本次结果
      GO TO 200                                        ┘
C***  COMPARE IT WITH THE STATES BEFORE,
C***  IF SAME, RETURN TO LAST STEP                     ┐
C***  OR TO CONTINUE, TO NEXT STEP                     │ 不是最终结果，
50    DO 70 I = 1, COUNT - 1                           │ 与前面的各次结果
      DO 60 J = 1, 9                                   │ 比较
      IF (A(I, J). NE. A(COUNT, J)) GO TO 70           ┘
60    CONTINUE
C***  GO TO RETURN A STEP
      GO TO 100
70    CONTINUE
      IF (COUNT. EQ. 31) GO TO 100
C***  NEXT STEP
80    COUNT = COUNT + 1                                ┐ 走下一步
      MARK (COUNT) = MARK(COUNT) + 1                   │
      NULL = FROM                                      │
      FROM = PLACE (NULL, MARK(COUNT))                 │
      IF (FROM. NE. PATH(COUNT - 2, 1))                │
         GO TO 90                                      │
C***  IF SAME, RETURN A STEP ONCE AGAIN                 │
      IF (MARK(COUNT). EQ. PLACE(NULL, 5))             │
         GO TO 100                                     ┘
```

```
        MARK(COUNT) = MARK(COUNT) + 1
        FROM = PLACE(NULL, MARK(COUNT))
 90     PATH(COUNT, 1) = FROM
        GO TO 25
 C***   RETURN LAST STEP
100     SAVE = MARK(COUNT)
        NULL = PATH(COUNT - 1, 1)
        MARK(COUNT) = 0
        IF (SAVE. EQ. PLACE(NULL, 5)) GO TO 120
        SAVE = SAVE + 1
110     FROM = PLACE (NULL, SAVE)
        IF (FROM. NE. PATH(COUNT - 2, 1))
           GO TO 190
        IF (SAVE. EQ. PLACE(NULL, 5)) GO TO 120
        SAVE = SAVE + 1
        GO TO 110
120     COUNT = COUNT - 1
        GO TO 100
190     PATH(COUNT, 1) = FROM
        MARK(COUNT) = SAVE
 C***   ONECE AGAIN
        GO TO 25
200     IF (NUM. EQ. 10) GO TO 450
        IF (NUM. GE. 3) GO TO 220
 C***   2TH AND 3TH WAYS
 C***   RETURN IMMIDIATELY TO COUNT = 3
 C***   CONTINUE GO FROM 2, 5
        DO 210 I = 5, COUNT
210     MARK(I) = 0
        COUNT = 4
        GO TO 100
220     IF (NUM. GE. 5) GO TO 250
 C***   4TH AND 5TH WAYS
 C***   BEGIN FROM 125874, BEFOR(INCLUD) LABEL 240
        IF (NUM. EQ. 4) GO TO 245
 C***   4TH WAY
        DO 225 I = 8, COUNT
225     MARK(I) = 0
        COUNT = 1
        DO 240 I = 2, 7
        COUNT = COUNT + 1
 C***   COPY A LINE
        DO 230 J = 1, 9
230     A(COUNT, J) = A(COUNT - 1, J)
```

退回到上一步

判断是否已找到10种

判断是否已找到 3 种

继续找第 2 、 3 种

直接退回到COUNT = 4

判断是否已找到 5 种

找第 4 种

```
C***  MOVE  A  PLACE                                  ┐
      FROM = D1(I)                                    │ 取出D1，进行移动
      NULL = D1(I - 1)                                │
      A(COUNT, NULL) = A(COUNT , FROM)                │
C***  SAVE  THE  PLACE  AND  VALUE  MOVED             │
      PATH (COUNT, 1) = FROM                          │
      PATH (COUNT, 2) = A(COUNT, FROM)                ┘
240   A (COUNT, FROM) = 0
      GO  TO  80
C***  5TH  WAY                                        找第 5 种
245   DO  600  I = 9, COUNT
600   MARK(I) = 0                                     ┐ 直接退回到COUNT = 5
      COUNT = 8                                       │
      GO  TO  100                                     ┘
C***  6TH  AND  7TH  WAYS
250   IF  (NUM. GE. 7) GO  TO  300                    判断是否找到第 7 种
C***  BEGIN  FROM  3, 4, 7, 8, 5, 2, 1, 7
      IF  (NUM. EQ. 6) GO  TO  700
C***  6TH  WAY                                        ┐
      DO  260  I = 10, COUNT                          │
260   MARK(I) = 0                                     │
      COUNT = 1                                       │
      DO  280  I = 2, 9                               │
      COUNT = COUNT + 1                               │
      DO  290  J = 1, 9                               │ 找第 6 种
290   A(COUNT, J) = A(COUNT - 1, J)                   │ 取出D3，
      FROM = D3(I)                                    │ 并进行移动
      NULL = D3 (I - 1)                               │
      A(COUNT, NULL) = A(COUNT, FROM)                 │
      PATH(COUNT, 1) = FROM                           │
      PATH(COUNT, 2) = A(COUNT, FROM)                 │
280   A(COUNT, FROM) = 0                              │
      GO  TO  80                                      ┘
C***  7TH  WAY
700   DO 720 I = 11, COUNT
720   MARK(I) = 0                                     ┐ 找第 7 种
      COUNT = 10                                      │ 直接退回到
      GO  TO  100                                     ┘ COUNT = 7
C***  8TH  WAY
300   IF  (NUM. GE. 8) GO  TO  350
C***  BEGIN  FROM  3, 4, 5                            ┐
      DO  310  I = 5, COUNT                           │
310   MARK(I) = 0                                     │
      COUNT = 1                                       ┘
```

```
        DO  320  I=2, 4
        COUNT=COUNT+1
        DO  330  J=1, 9
330     A(COUNT, J)=A(COUNT-1, J)
        FROM=D4(I)
        NULL=D4(I-1)
        A(COUNT, NULL)=A(COUNT, FROM)
        PATH(COUNT, 1)=FROM
        PATH(COUNT, 2)=A(COUNT, FROM)
320     A(COUNT, FROM)=0
        GO  TO  80
 C***   8TH  AND  9TH  WAYS
350     IF  (NUM. EQ. 8)  GO  TO  800
        IF  (NUM. EQ. 9)  GO  TO  900
 C***   9TH  WAY
800     DO  820  I=9, COUNT
820     MARK(I)=0
        COUNT=8
        GO  TO  100
 C***   10TH  WAY
 C***   BEGIN  FROM  3, 4, 5
900     DO  920  I=6, COUNT
920     MARK(I)=0
        COUNT=5
        GO  TO  100
 C***   OUTPUT  THE  RESULTS  AND  STOP
 C***   OUTPUT  THE  RESU  LTS
450     DO  455  I=1, NUM
        WRITE(1, 460) I, (SB(I, J), J=2, SB(I, 41)),
           SB(I, 41)-1
455     WRITE(4, 460) I, (SB(I, J), J=2, SB(I, 41)),
           SB(I, 41)-1
460     FORMAT(5X, I2, 'TH  WAY', 5X, 30I2, 5X, '
           TOTAL  STEPS=', I2)
470     CLOSE(-1)
        STOP
        END
```

找第8种，取出D4，进行移动

找第9种直接退回到COUNT=8

找第10种，直接退回到COUNT=5

输出结果

最后输出的10种结果，其数字表示顺序移动的格子中的数

```
1TH WAY 1 4 3 1 4 2 5 8 7 3 1 6 3 1 2 5 8 7 1 2 5 4 6 5 4 8 7 4 5 6 TOTAL STEPS=30
2TH WAY 1 4 5 8 7 5 3 6 5 3 4 1 6 5 3 4 1 2 8 7 4 1 2 8 7 4 1 2 5 6 TOTAL STEPS=30
3TH WAY 1 4 7 8 5 2 4 7 8 6 3 8 6 5 2 4 7 1 8 6 1 7 4 1 5 2 1 4 7 8 TOTAL STEPS=30
4TH WAY 1 2 5 8 7 4 3 1 2 5 8 7 4 3 1 6 3 1 5 2 6 5 2 8 7 4 1 2 5 6 TOTAL STEPS=30
5TH WAY 1 2 5 8 7 4 8 5 2 8 3 1 8 2 5 7 4 3 1 6 3 1 2 5 7 4 1 2 5 8 TOTAL STEPS=30
6TH WAY 3 4 7 8 5 2 1 7 4 3 7 4 8 6 3 8 6 5 2 1 4 7 8 6 5 2 1 4 7 8 TOTAL STEPS=30
```

7TH WAY 3 4 7 8 5 2 1 7 8 5 2 1 7 8 5 6 4 3 8 5 6 4 3 6 4 2 1 4 5 8 TOTAL STEPS = 30
8TH WAY 3 4 5 2 1 5 4 3 5 4 7 8 2 1 4 7 8 6 3 8 6 2 1 4 7 5 8 6 5 8 TOTAL STEPS = 30
9TH WAY 3 4 5 2 1 5 7 6 4 3 5 7 6 8 2 1 7 6 8 4 3 5 6 8 4 2 1 4 5 6 TOTAL STEPS = 30
10TH WAY 3 4 5 8 7 5 1 3 4 6 5 1 3 2 8 7 1 3 2 4 6 5 3 2 4 8 7 4 5 6 TOTAL STEPS = 30

注：为了简化程序，第1、第2、第3用1TH、2TH、3TH表示。

（三）思 考 题

1）请考虑如何改进算法，提高查找速度。

2）30步是最少的步数吗？请你试试能否找出比这还要少的步数。

105 魔 方 游 戏

把涂有不同颜色的魔方阵容弄乱，再用手转动使其复原，不仅仅是一种游戏，而且是一种智力测验，是考验思维、反应锐敏能力，锻炼手脑协调配合的好工具。在国外广为流传，极为盛行，甚至用它做比赛，著书传授。近年来也影响到我国，出售各种玩具的魔方，有的计算机软件专业的学生选它作为毕业论文题。

我们利用现代化的工具电子计算机，在终端屏幕上来做这种游戏。

本程序是模拟魔方的转动情况，它可以任意地转动魔方，以便复原。每个面都可以顺时针或逆时针转动，为了改变观看角度，也可以绕轴向转动，三个轴向是：穿过顶平面到底平面的轴、穿过左平面到右平面的轴、穿过前平面到后平面的轴。

各种转动的表示方法：

R +：右平面顺时针转动90°；

R −：右平面逆时针转动90°；

R 2：右平面转动180°；

F +：前平面顺时针转动90°；

F −：前平面逆时针转动90°；

F 2：前平面转动180°；

L +：左平面顺时针转动90°；

L −：左平面逆时针转动90°；

L 2：左平面转动180°；

D +：底平面顺时针转动90°；

D −：底平面逆时针转动90°；

D 2：底平面转动180°；

U +：顶平面顺时针转动90°；

U −：顶平面逆时针转动90°；

U 2：顶平面转动180°；

B +：后平面顺时针转动90°

B −：后平面逆时针转动90°；

B 2：后平面转动180°；

MFR# （#为1、2、3）：绕着顶平面到底平面的轴转动魔方，把前平面转到右平面1次、2次或3次；

MFU# （#为1、2、3）：绕着右平面到左平面的轴转动魔方，把前平面转到顶平面1次、2次或3次。

MUR# （#为1、2、3）：绕着前平面到后平面的轴转动魔方，把顶平面转到右平面1次、2次或3次。

程序首先把魔方置为初态，然后问您要进行多少次转动，以便把魔方搞乱，程序显示出搞乱的魔方。逐次输入您欲进行的转动。每转动一次，程序均显示所得的结果。若输入为S，则表示要求停止。若输入的不是S或上述表示法之一，则程序认为您输入有错，请求重新输入。

（一）算法分析 （省 略）参阅程序旁注

（二）程序设计

1）设计思路

程序用一个数组CUBE（5、5、5）表示魔方，各平面分别用1、2、3、4、5、6表示（图105-1是魔方的表示法）。

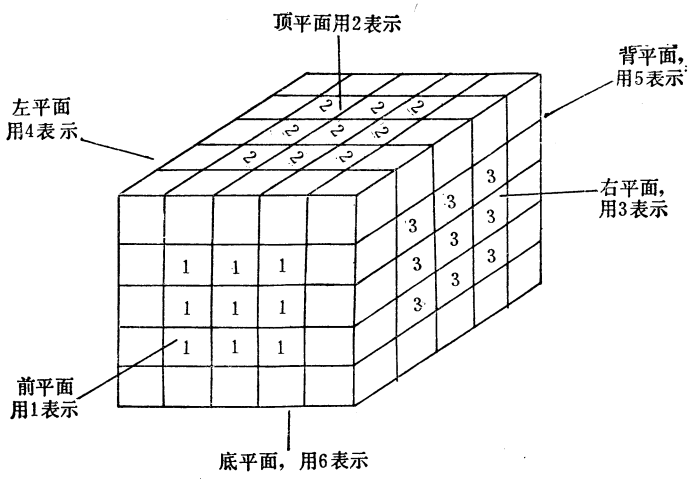


图 105-1

转动平面时，按要求调整数组CUBE中元素的值即可。

转动一个平面90°，要对数组中元素做下述调整：

（1）把该平面的9个元素进行调整（图105-2）；

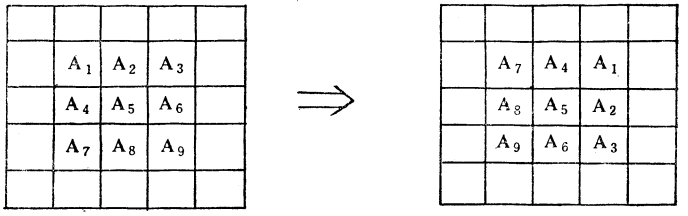


图 105-2

（2）把与该平面相邻的四个平面的元素进行调整（图105-3）。

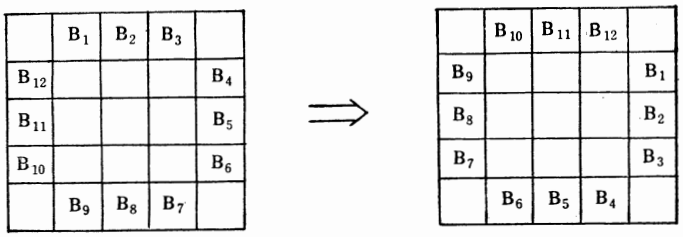


图 105-3

按轴向转动时，把其转动分解为平面的转动，加上数组第3层元素的转动。

程序中设二个较大的子程序，一个用于转动魔方，一个用于输出显示魔方。显示输出用字符F、R、L、U、D、B分别表示各面上的小平面。

2）框图

给出主程序（图105-4及图105-5）以及子程序ROTATE（图105-6）、ROT1（图105-7）、ROT2（图105-8），其它三个子程序较简单，不再一一画出，可参阅源程序中的注释。

3）源程序及运行结果

```
C***  PROGRAM  NAME:CUBE
C***  PROGRAM  RUBIK'S  CUBE
C***  METHOD  OF  PLAING  THE  CUBE      魔方程序玩的方法:
C***  1, INPUT  A  NUMBER  REDOMLY
          FOR  MIX  THE  CUBE             1. 随机输入一个数，搞乱魔方;
C***  2, INPUT  THE  NUMBER  YOU
          WANT  TO  TURN  THE  CUBE       2. 输入你要转动的方向和次数;
C***  3, IF  YOU  WANT  TO  STOP  THE
          PLAING, TYPE  THE  'S'  PLEASE  3. 若要停止，则打入"S";
C***  4, IF  YOU  WANT  TO  TURN  THE
          CUBE  IN  ORDER  TO  CHANGE
          THE  VIEW                       4. 若要转动观看角度,则先 打入"M"。
C***  POINT, TYPE  THE  'M'  PLEASE
C***
      INTEGER  CUBE(5, 5, 5) , A(5, 5)
      CHARACTER  X(4)
      INTEGER  I, J, N, R0, R1, X4
C***  INITIATE  THE  CUBE
      DO  10  I = 1, 5                    \
      DO  10  J = 1, 5                     | CUBE (I, J, K) ⇐0
      DO  10  K = 1, 5                     |
10    CUBE(I, J, K) = 0                   /
```

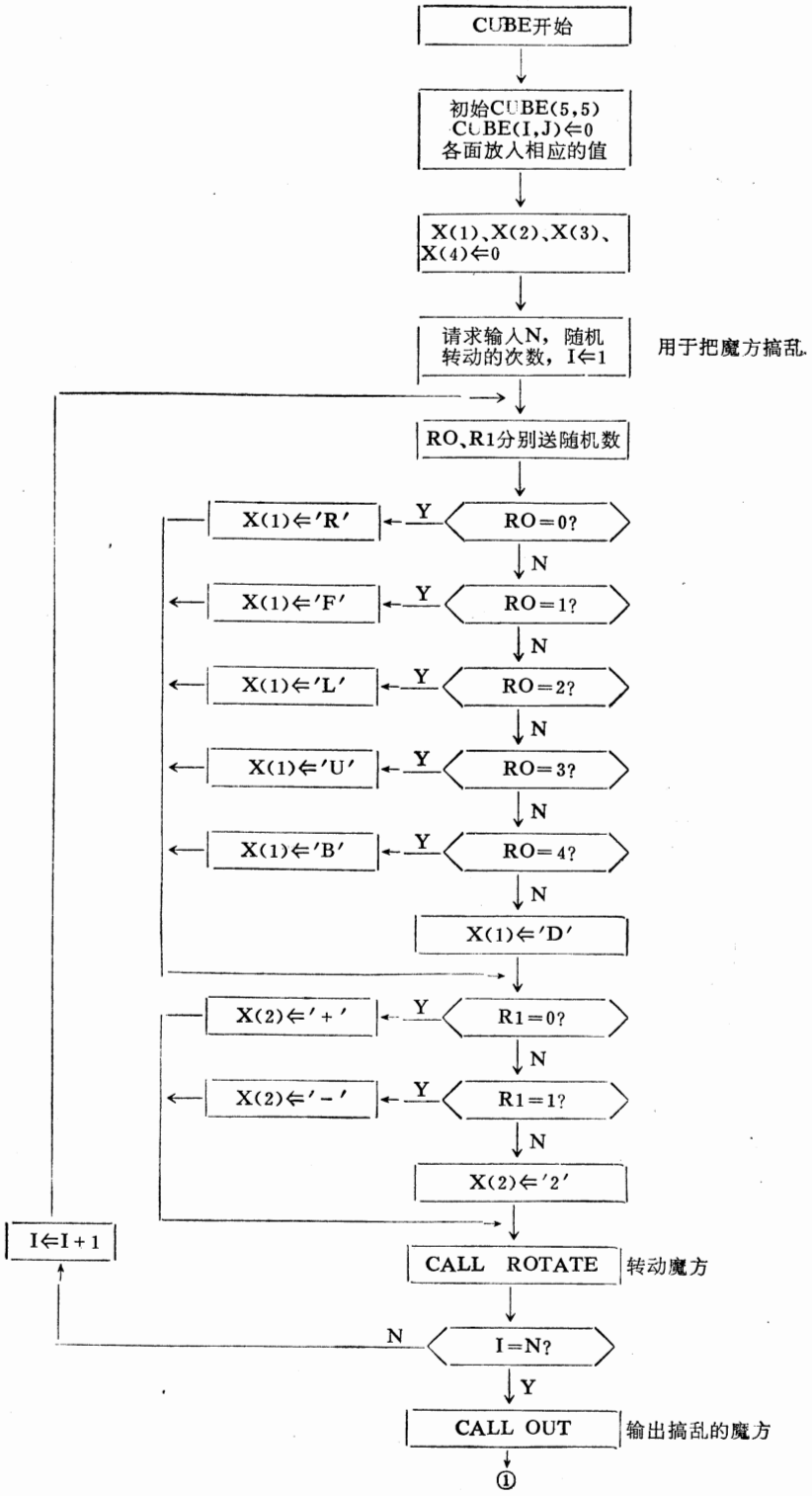

CUBE开始
初始CUBE(5,5)
CUBE(I,J)⇐0
各面放入相应的值
X(1)、X(2)、X(3)、
X(4)⇐0
请求输入N，随机
转动的次数，I⇐1
用于把魔方搞乱.
RO、R1分别送随机数
RO=0?
Y
X(1)⇐'R'
N
RO=1?
Y
X(1)⇐'F'
N
RO=2?
Y
X(1)⇐'L'
N
RO=3?
Y
X(1)⇐'U'
N
RO=4?
Y
X(1)⇐'B'
N
X(1)⇐'D'
R1=0?
Y
X(2)⇐'+'
N
R1=1?
Y
X(2)⇐'-'
N
X(2)⇐'2'
CALL ROTATE
转动魔方
I=N?
N
Y
I⇐I+1
CALL OUT
输出搞乱的魔方
①

图 105-4

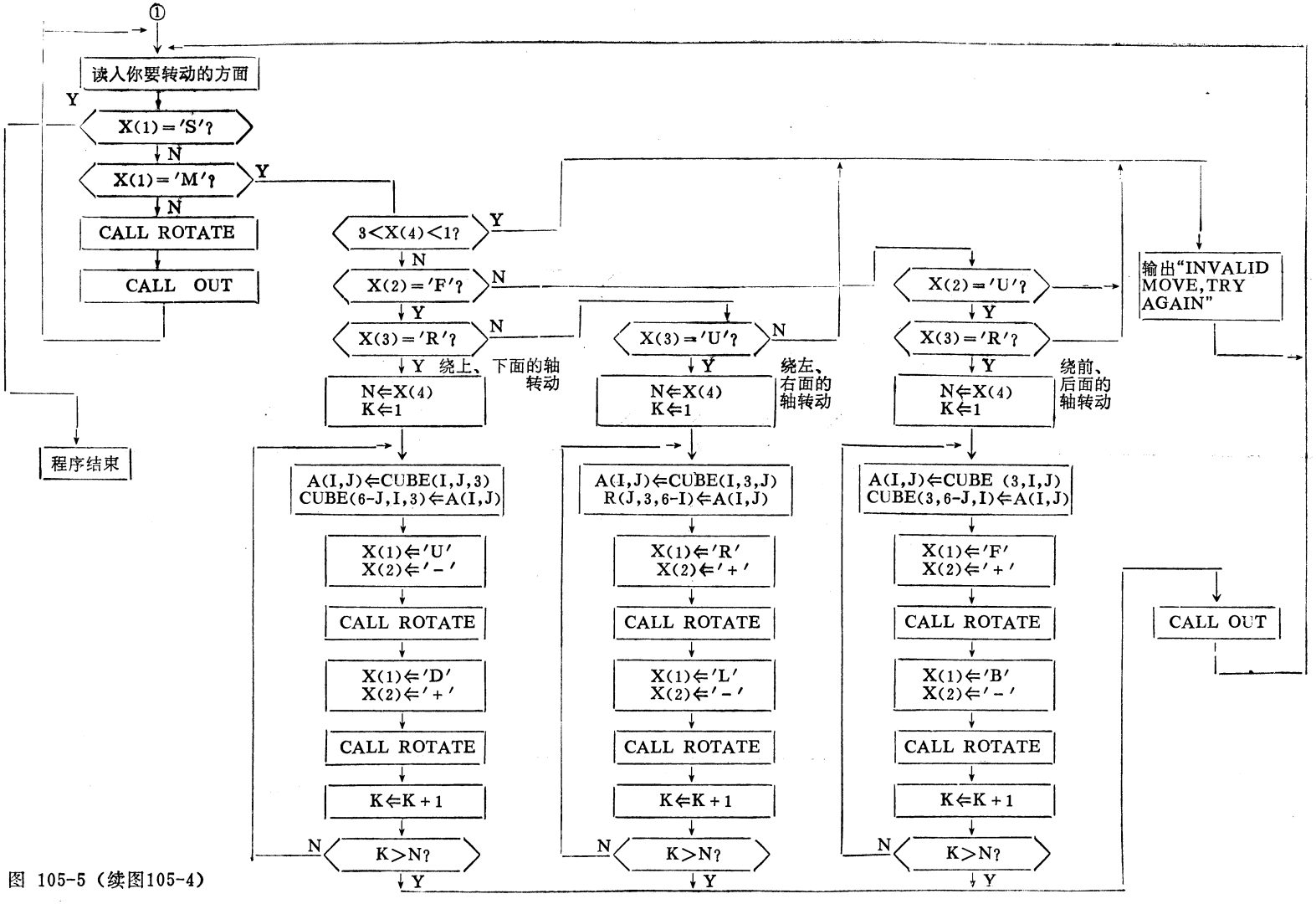
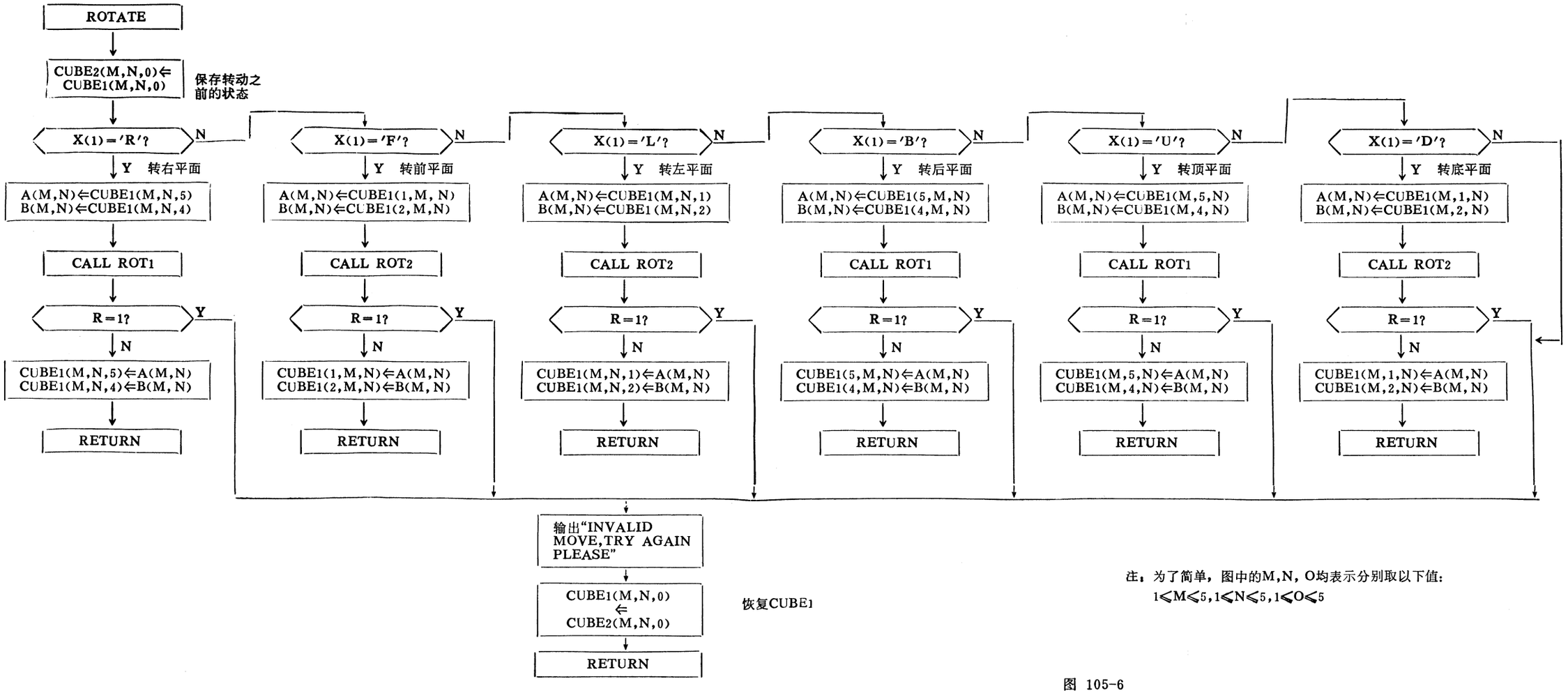


图 105-5（续图105-4）



注：为了简单，图中的M,N，O均表示分别取以下值：
$1 \leqslant M \leqslant 5, 1 \leqslant N \leqslant 5, 1 \leqslant O \leqslant 5$

图 105-6

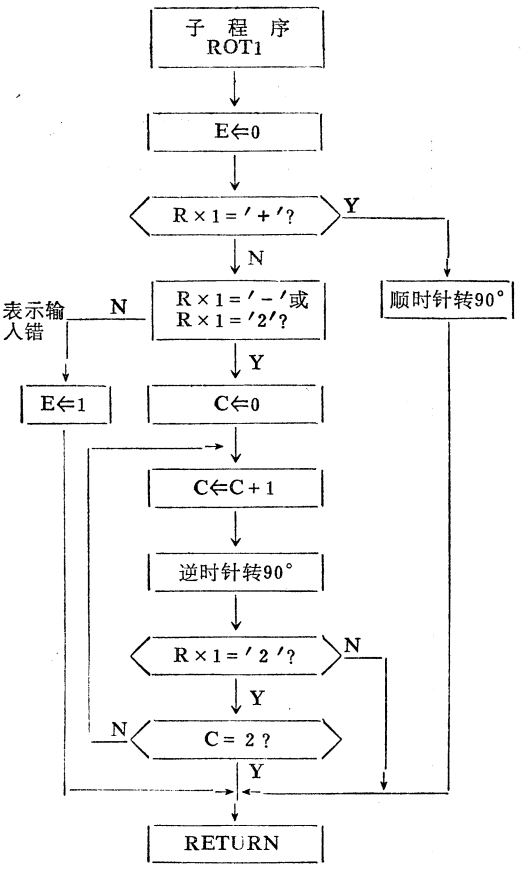


图 105-7

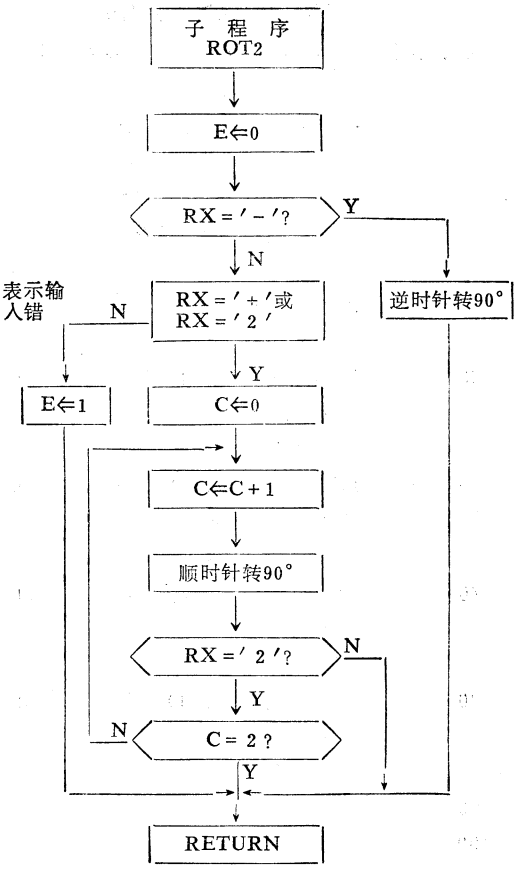


图 105-8

```
C***  FACE=1, UP=2, RIGHT=3, LEFT=4,
          BACK=5, DOWN=6
      DO  20  I=2, 4
      DO  20  J=2, 4
      CUBE(1, I, J) =1
      CUBE(I, 1, J) =6
      CUBE(I, J, 1) =4
      CUBE(5, I, J) =5
      CUBE(I, 5, J) =2
20    CUBE(I, J, 5) =3
      DO  25  I=1, 4
25    X(I) =' '
C***  MIX  RUBIK'S  CUBE
      WRITE (1, 30)
30    FORMAT (3X, 38HMIX  RUBIK'S  CUBE. TYPE  A  NUMBER  PLEASE)
      READ (1, 40) N
40    FORMAT (I2)
```

按规定添入相应的数字，

读入N

```
      YYL=FLOAT(N) /10
C***  CALCULAT  RANDOM  NUMBER
      DO  140  I=1, N                          随机转动N次
      CALL  RAND(YYL)                          调用随机数子程序，产生R0和R1，
      RO=INT (6*YYL)                           然后按R0和R1的值转动不同的平
      CALL  RAND(YYL)                          面和角度
      R1=INT (3*YYL)
      IF (RO. NE. 0) GO  TO  50
      X (1) ='R'
      GO  TO  100
50    IF (RO. NE. 1) GO  TO  60
      X (1) ='F'
      GO  TO  100
60    IF (RO. NE. 2) GO  TO  70
      X (1) ='L'
      GO  TO  100
70    IF (RO. NE. 3) GO  TO  80
      X (1) ='U'
      GO  TO  100
80    IF (RO. NE. 4) GO  TO  90
      X (1) ='B'
      GO  TO  100
90    X (1) ='D'
100   CONTINUE
      IF (R1. NE. 0) GO  TO  110
      X (2) ='+'
      GO  TO  130
110   IF (R1. NE. 1) GO  TO  120
      X (2) ='-'
      GO  TO  130
120   X (2) ='2'
130   CALL  ROTATE (CUBE, X)
140   CONTINUE
      CALL  OUT (CUBE)                         输出搞乱的魔方
C***  CHANGE  RUBIK'S  CUBE
145   WRITE (1, 150)
150   FORMAT (1X, 'INPUT  TURN  DIRECTION  YOU  WANTED')
      READ (1, 160) X(1), X(2), X(3), X(4)     读入你给出的转动方向和次数
160   FORMAT (4A1)
      IF (X(1). EQ. 'S') GO  TO  170
      IF (X(1). EQ. 'M') GO  TO  370
      CALL  ROTATE (CUBE, X)                   若不是"S"或"MXXX"，
                                               则转动相应的平面
      CALL  OUT (CUBE)                         输出
```

```
      GO TO 145
370   CALL CONVERT (X(4), X4)                    因为是“MXXX”，所以把第4字符
                                                 转换成相应的值
      WRITE (1, 371) X4
371   FORMAT (3X, 'X4=', I2)
      IF (X4. GT. 3. OR. X4. LT. 1) GO TO 474
      IF (X(2). NE. 'F') GO TO 410
      IF (X(3). NE. 'R') GO TO 450
C***  ROTATE THE CUBE ABOUT THE AXIS PASSING THYOUGH
        THE CENTER
C***  SQUARES OF THE UP (TOP) AND DOWN (BOTTOM) FACES
      DO 400 K=1, X4
      DO 380 M=1, 5
      DO 380 N=1, 5
380   A(M, N) =CUBE(M, N, 3)
      DO 390 M=1, 5
      DO 390 N=1, 5
390   CUBE (6-M, N, 3) =A (M, N)
      X(1) ='U'                                  绕穿过顶平面和底平面的轴转动
      X(2) ='-'
      CALL ROTATE (CUBE, X)
      X(1) ='D'
      X(2) ='+'
      CALL ROTATE (CUBE, X)
400   CONTINUE
      CALL OUT (CUBE)
      GO TO 145
410   IF (X(2). NE. 'U') GO TO 474
      IF (X(3). NE. 'R') GO TO 474
C***  ROTATE THE CUBE ABOUT THE AXIS PASSING THROUGH
        THE CENTER
C***  SQUARES OF THE FRONT AND BACK FACES
      DO 440 K=1, X4
      DO 420 M=1, 5
      DO 420 N=1, 5
420   A(M, N) =CUBE(3, M, N)
      DO 430 M=1, 5
      DO 430 N=1, 5
430   CUBE(3, 6-M, N) =A(M, N)
      X(1) ='F'                                  绕穿过前平面和后平面的轴转动
      X(2) ='+'
      CALL ROTATE (CUBE, X)
      X(1) ='B'
      X(2) ='-'
```

```
        CALL  ROTATE (CUBE, X)
440     CONT  INUE
        CALL  OUT (CUBE)

        GO  TO  145
450     IF (X (3). NE. 'U') GO  TO  474
 C***   ROTATE  THE  CUBE  ABOUT  THE  AXIS  PASSING  THROUGH
           THE  CENTER
 C***   SQUARES  OF  THE  RIGHT  AND  LEFT  FACES

        DO  475  K=1, X4
        DO  460  M=1, 5
        DO  460  N=1, 5
460     A(M, N) =CUBE(M, 3, N)
        DO  470  M=1, 5
        DO  470  N=1, 5
470     CUBE(M, 3, 6-N) =A(M, N)
        X(1) ='R'
        X(2) ='+'
        CALL  ROTATE (CUBE, X)
        X(1) ='L'
        X(2) ='-'
        CALL  ROTATE (CUBE, X)
475     CONTINUE
        CALL  OUT (CUBE)

        GO  TO  145
 C***   OUTPUT  MESSAGE  "INVALID
           INPUT"
474     WRITE (1, 477)
477     FORMAT (5X, 30HINVALID  MOVE,
           TRY  AGAIN  PLEASE, /)

        GO  TO  145

170     WRITE (1, 180)
180     FORMAT (5X, 21HTHANK  YOU  FOR
           PLAYING)
        STOP

        END
 C***   THE  SUBROUTINE  ROTATING
           RUBIK'S  CUBE  ACCOUNDING
           TO  YOUR  TYPING
        SUBROUTINE  ROTATE (CUBE1, RX)
        INTEGER  CUBE1(5, 5, 5), CUBE2(5, 5, 5), A(5, 5), B(5, 5)
        INTEGER  M, N, K, R
        CHARACTER  RX (4)
```

绕穿过左平面和右平面的轴转动

输出："INVALID INPUT"，输入无效，请重新输入。

程序输出"谢谢"，然后停止。

子程序ROTATE，按要求转动魔方

```
C***  SAVE  THE  CUBE1
      DO  200  M=1, 5
      DO  200  N=1, 5
      DO  200  K=1, 5
200   CUBE2(M, N, K) =CUBE1(M, N, K)
      IF (RX (1). NE. 'R') GO  TO  230
C***  MOVE  RIGHT  FACE
      DO  210  M=1, 5
      DO  210  N=1, 5
      A(M, N) =CUBE1(M, N, 5)
210   B(M, N) =CUBE1(M, N, 4)
      CALL  ROT1 (R, A, B, RX (2))
      IF (R. EQ. 1) GO  TO  480
      DO  220  M=1, 5
      DO  220  N=1, 5
      CUBE1(M, N, 5) =A(M, N)
220   CUBE1(M, N, 4) =B(M, N)
      GO  TO  510
C***  MOVE  FRONT  FACE
230   IF (R×(1). NE. 'F') GO  TO  260
      DO  240  M=1, 5
      DO  240  N=1, 5
      A(M, N) =CUBE1(1, M, N)
240   B(M, N) =CUBE1(2, M, N)
      CALL  ROT2 (R, A, B, RX(2))
      IF (R. EQ. 1) GO  TO  480
      DO  250  M=1, 5
      DO  250  N=1, 5
      CUBE1(1, M, N) =A(M, N)
250   CUBE1(2, M, N) =B(M, N)
      GO  TO  510
C***  MOVE  LEFT  FACE
260   IF (RX(1). NE. 'L') GO  TO  290
      DO  270  M=1, 5
      DO  270  N=1, 5
      A(M, N) =CUBE1(M, N, 1)
270   B(M, N) =CUBE1(M, N, 2)
      CALL  ROT2(R, A, B, RX (2))
      IF (R. EQ. 1) GO  TO  480
      DO  280  M=1, 5
      DO  280  N=1, 5
      CUBE1(M, N, 1) =A(M, N)
280   CUBE1(M, N, 2) =B(M, N)
      GO  TO  510
```

保存CUBE1

转动右平面

转动前平面

转动左平面

```
C***  MOVE  BACK  FACE
290       IF (RX(1). NE. 'B') GO  TO  320
          DO  300  M=1, 5
          DO  300  N=1, 5
          A(M, N) =CUBE1(5, M, N)
300       B(M, N) =CUBE1(4, M, N)
          CALL  ROT1 (R, A, B, RX(2))
          IF (R. EQ. 1) GO  TO  480
          DO  310  M=1, 5
          DO  310  N=1, 5
          CUBE1(5, M, N) =A(M, N)
310       CUBE1(4, M, N) =B(M, N)
          GO  TO  510
C***  MOVE  UP  FACE
320       IF (RX(1). NE. 'U') GO  TO  350
          DO  330  M=1, 5
          DO  330  N=1, 5
          A(M, N) =CUBE1(M, 5, N)
330       B(M, N) =CUBE1(M, 4, N)
          CALL  ROT1 (R, A, B, RX(2))
          IF (R. EQ. 1) GO  TO  480
          DO  340  M=1, 5
          DO  340  N=1, 5
          CUBE1(M, 5, N) =A(M, N)
340       CUBE1(M, 4, N) =B(M, N)
          GO  TO  510
C***  MOVE  DOWN  FACE
350       IF (RX(1). NE. 'D') GO  TO  480
          DO  360  M=1, 5
          DO  360  N=1, 5
          A(M, N) =CUBE1(M, 1, N)
360       B(M, N) =CUBE1(M, 2, N)
          CALL  ROT2 (R, A, B, RX(2))
          IF (R. EQ. 1) GO  TO  480
          DO  365  M=1, 5
          DO  365  N=1, 5
          CUBE1(M, 1, N) =A(M, N)
365       CUBE1(M, 2, N) =B(M, N)
          GO  TO  510
C***  OUTPUT  MESSAGE  "INVALID
          INPUT"
480       WRITE (1, 490)
490       FORMAT (5X, 30HINVALID  MOVE,
          TRY  AGAIN  PLEASE, /)
C***  RESTORE  THE  CUBE1
```

转动后平面

转动顶平面

转动底平面

输出“输入无效，请重新输入”。

```
      DO  500  M = 1, 5
      DO  500  N = 1, 5
      DO  500  K = 1, 5                          } 因为输入无效，所以恢复CUBE1。
500   CUBE1(M, N, K) = CUBE2(M, N, K)
510   RETURN
      END
C***  SUBROUTINE
      SUBROUTINE  ROT1 (E, RA1, RB1,
        RX1)                                     子程序ROT1
      INTEGER  RA1(5, 5), RB1(5, 5), RC1(5, 5), RD1(5, 5)
      CHARACTER  RX1
      INTEGER  E, P, Q, C
      E = 0
      IF (RX1. NE. '+') GO  TO  565
C***  TURN  CLOCKWISE  ONE  QUARTER
      DO  550  P = 1, 5
      DO  550  Q = 1, 5
      RC1 (P, Q) = RA1 (P, Q)
550   RD1 (P, Q) = RB1 (P, Q)                    } 顺时针转90°
      DO  560  P = 1, 5
      DO  560  Q = 1, 5
      RA1(Q, 6-P) = RC1(P, Q)
560   RB1(Q, 6-P) = RD1(P, Q)
      GO  TO  610
565   IF (RX1. EQ. '-'. OR. RX1. EQ. '2') GO  TO  570
      E = 1                                      置输入有错标志
      GO  TO  610
570   C = 0
580   C = C + 1
C***  TURN  COUNTERCLOCKWISE  ONE  QUARTER
      DO  590  P = 1, 5
      DO  590  Q = 1, 5
      RC1(P, Q) = PA1(P, Q)
590   RD1(P, Q) = RB1(P, Q)                      } 逆时针转90°
      DO  600  P = 1, 5
      DO  600  Q = 1, 5
      RA1(6-Q, P) = RC1(P, Q)
600   RB1(6-Q, P) = RD1(P, Q)
      IF (RX1. NE. '2') GO  TO  610
      IF (C. NE. 2) GO  TO  580
610   RETURN
      END
C***  SUBROUTINR  ROT2
```

```
      SUBROUTINE ROT2(E, RA, RB, RX)
      INTEGER RA(5, 5) , RB(5, 5) , RC(5, 5) , RD(5, 5)
      CHARACTER RX
      INTEGER P, Q, C, E
      E = 0
      IF (RX. NE. '-') GO TO 660
C***  TURN COUNTER CLOCKWISE ONE QUARTER
      DO 640 P=1, 5                          \
      DO 640 Q=1, 5                          |
      RC(P, Q) =RA(P, Q)                     |
640   RD(P, Q) =RB(P, Q)                     | 逆时针转90°
      DO 650 P=1, 5                          |
      DO 650 Q=1, 5                          |
      RA(Q, 6-P) =RC(P, Q)                   /
650   RB(Q, 6-P) =RD(P, Q)
      GO TO 700
660   IF (RX. EQ. '+'. OR. RX. EQ. '2') GO TO 670
      E = 1                                  置输入有错标志
      GO TO 700
670   C = 0
675   C = C + 1
C***  TURN CLOCKWISE ONE QUARTER
      DO 680 P=1, 5                          \
      DO 680 Q=1, 5                          |
      RC(P, Q) =RA(P, Q)                     |
680   RD(P, Q) =RB(P, Q)                     |
      DO 690 P=1, 5                          | 顺时针转90°
      DO 690 Q=1, 5                          |
      RA(6-Q, P) =RC(P, Q)                   |
690   RB(6-Q, P) =RD(P, Q)                   |
      IF (RX. NE. '2') GO TO 700             /
      IF (C. NE. 2) GO TO 675
700   RETURN
      END
C***  THIS SUBROUTINE CONVERTS THE VALUE OF EACH BLOCK
         TO THE
C***  REPRESANTATION BY CHARACTER, TO CALL IT BEFOR DRAW
         THE CUBE
C***
      SUBROUTINE MAP (TEMP, TD)              子程序MAP，把一个平面的九个小平
      INTEGER TEMP(5, 5) , TEMP1 (9)         面的数字转换成相应的字符，即
      CHARACTER TD (9)                       F、U、R、L、B、D。
      TEMP1(1) =TEMP(4, 2)
```

```
      TEMP1(2) =TEMP(4, 3)
      TEMP1(3) =TEMP(4, 4)
      TEMP1(4) =TEMP(3, 2)
      TEMP1(5) =TEMP(3, 3)
      TEMP1(6) =TEMP(3, 4)
      TEMP1(7) =TEMP(2, 2)
      TEMP1(8) =TEMP(2, 3)
      TEMP1(9) =TEMP(2, 4)
      DO 870 I=1, 9
      GO TO (810, 820, 830, 840, 850, 860) , TEMP1 (I)
810   TD (I) ='F'
      GO TO 870
820   TD (I) ='U'
      GO TO 870
830   TD (I) ='R'
      GO TO 870
840   TD (I) ='L'
      GO TO 870
850   TD (I) ='B'
      GO TO 870
860   TD (I) ='D'
870   CONTINUE
      RETURN
      END
C***  SUBROUTINE GENERATE RANDOM
        NUMBER                          子程序RAND，其产生一个0—1
      SUBROUTINE RAND (YFL)             间的随机数。
      IF (YFL. EQ. 0. 0) YFL=0.314159
      Y=YFL*YFL
1     Y=Y*10.0
      IF (Y-1.0) 1, 2, 2
2     YFL=Y-FLOAT (IFIX (Y))
      RETURN
      END
C***  PROGRAM NAME: OUT
C***
C***  SUBROUTINE CONVERT WHICH CONVERTS THE VALUE OF X(3)
      SUBROUTINE CONVERT (Y1, Y2)       子程序CONVERT，把字符1、2、3
      INTEGER Y2                        分别转换成相应的数值1、2、3。
      CHARACTER Y1
      IF (Y1. EQ. '1') GO TO 2
      IF (Y1. EQ. '2') GO TO 4
      IF (Y1. EQ. '3') GO TO 6
```

```
      Y2=10
      GO TO 8
 2    Y2=1
      GO TO 8
 4    Y2=2
      GO TO 8
 6    Y2=3
 8    RETURN
      END
C***  OUTPUT THE RESULT OF CUBE'S MOVED
      SUBROUTINE OUT (DRAW)                    子程序OUT，输出魔方图形。
      INTEGER DRAW (5, 5, 5) , FACE (5, 5)
      CHARACTER A(9) , B(9) , C(9) , D(9) , E(9) , F(9)
      INTEGER I
      DO 890 I=1, 9
      A (I) =' '
      B (I) =' '
      C (I) =' '                               清除六个数组
      D (I) =' '
      E (I) =' '
890   F (I) =' '
      DO 900 I=1, 5
      DO 900 J=1, 5
900   FACE(I, J) =0
C***  UP FACE
      DO 910 I=2, 4                            取出顶平面
      DO 910 K=2, 4
910   FACE(I, K) =DRAW(I, 5, K)
      CALL MAP (FACE, A)                       转换成相应的字符
C***  RIGHT FACE
      DO 920 I=2, 4                            取出右平面
      DO 920 J=2, 4
920   FACE(I, J) =DRAW(I, J, 5)
      CALL MAP (FACE, B)                       转换成相应的字符
C***  FRONT FACE
      DO 930 J=2, 4                            取出前平面
      DO 930 K=2, 4
930   FACE(J, K) =DRAW(1, J, K)
      CALL MAP (FACE, C)                       转换成相应的字符
C***  LEFT FACE
      DO 940 I=2, 4                            取出左平面
      DO 940 J=2, 4
940   FACE(I, J) =DRAW(I, J, 1)
```

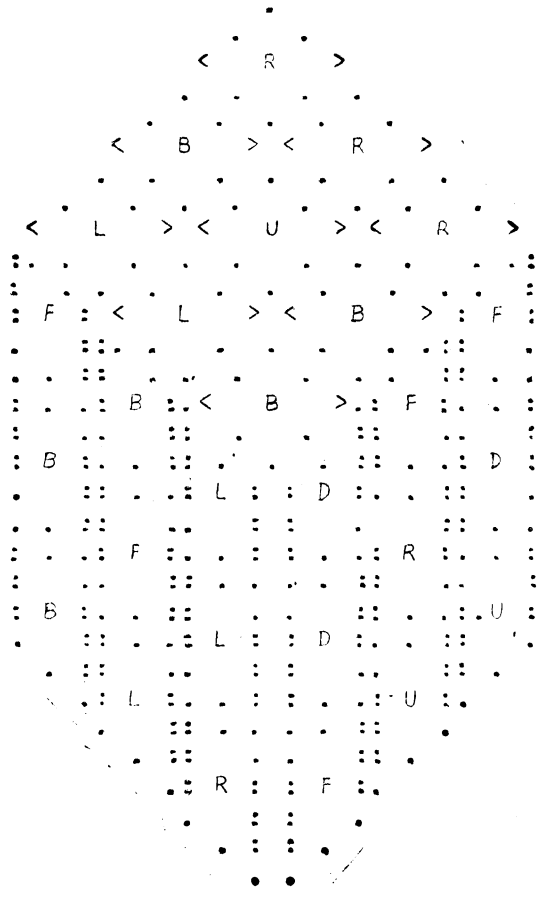
```
      CALL  MAP (FACE, D)                    转换成相应的字符
C***  DOWN  FACE
      DO  950  I = 2, 4                      ┐
      DO  950  K = 2, 4                      ├ 取出底平面
950   FACE(I, K) = DRAW(I, 1, K)             ┘
      CALL  MAP (FACE, F)                    转换成相应的字符
C***  BACK  FACE
      DO  960  J = 2, 4                      ┐
      DO  960  K = 2, 4                      ├ 取出后平面
960   FACE(J, K) = DRAW(5, J, K)             ┘
      CALL  MAP (FACE, E)                    转换成相应的字符
      WRITE (1, 970) A(1)
970   FORMAT (20X, '.', 38X, '.', 1X, '.', /, 18X, '.     .', 34X, '. : :
     . ', /, 16X,                            以下为输出图形
     1 '<', 3X, A1, 3X, '>', 30X, '.     : :    . ')
      WRITE (1, 980) D(1), E(3)
980   FORMAT (15X, 2 ('.     .     '), 25X, '. . ', A1, ': :', A1, ':. ',
     1 /, 13X, '.  .  .  .     . ', 24X, '. : :    .  .     : :. ')
      WRITE (1, 990) A(4), A(2)
990   FORMAT (11X, 2 ('<    ', A1, '    > '), 19X, '.     : :. . . .
     : :    . ')
      WRITE (1, 1000) D(2), E(2)
1000  FORMAT (10X, 2 ('.     .     .  '), '. ', 18X, '. : ', A1, ' : . .
     1 : : . . :', A1, ':. ')
      WRITE (1, 1010)
1010  FORMAT (8X, 2 ('.     .  .  '), '.     . ', 14X, '. : :     . .
     1 : :    . .     : :. ')
      WRITE (1, 1020) A(7), A(5), A(3), D(4), E(6)
1020  FORMAT (6X, 3 ('<    ', A1, '    > '), 9X, '.     : :. . : ', A1,
     1 ' : : ', A1, ' :. . : :    . ')
      WRITE (1, 1030) D(3), E(1)
1030  FORMAT (5X, ':. ', 2 ('.     .     .     '), '.     . . :', 9X, '
     1 :', A1, ':. . : :. .     : :. . : ', A1, ' :')
      WRITE (1, 1040)
1040  FORMAT (5X, ':     . ', 2 ('.     .     .  '), '. . :', 9X, ':
     1 ..    : :. . . . : :    ..    :')
      WRITE (1, 1050) C(1), A(8), A(6), B(3), D(5), E(5)
1050  FORMAT (5X, ' : ', A1, ' : ', 2 ('<    ', A1, '    > '), ': ',
     1 A1, ' :', 9X, ': . . : ', A1, ' :. . : :. . : ', A1, ' :. . :')
      WRITE (1, 1060)
1060  FORMAT (5X, '.     : :. ', 2 (' .     '), '.     .    . . : :    .
     1 ', 9X, '. . : :    ..  : :    ..    : :. . ')
      WRITE (1, 1070) D (7), E (9)
```

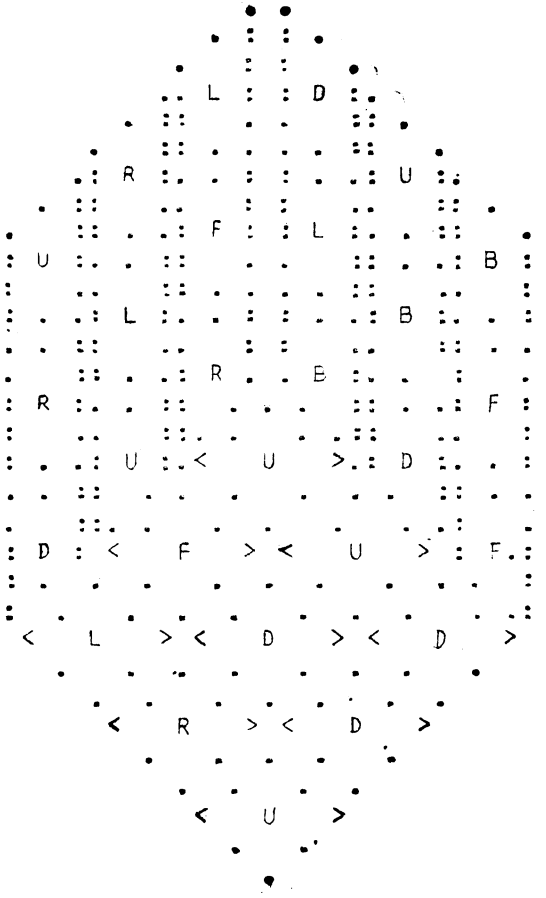
```
1070    FORMAT (5X, '. . : : . .  .    .  . .  : :. . ', 9X, '.     : :
     1 . .  :  ', A1, ' . .  ', A1, ' :. .    :   . ')
        WRITE (1, 1080) C (2) , A (9) , B (2) , D (6) , E (4) ,
1080    FORMAT (5X, ':. . : ', A1, ' :. <   ', A1, '   >. : ',A1, ' :
     1 . .  :', 9X, ':  ', A1, ' :. . : :  . . .   : :. . : ', A1, ' :')
        WRITE (1, 1090)
1090    FORMAT (5X, ':   ..   : :  .   .  : :   ..   :', 9X, ':
     1 ..   : :. . . . : :   ..   :')
        WRITE (1, 1100) C (4) , B (6) , D (8) , F (1) , E (8)
1100    FORMAT (5X, ': ', A1, ' :. . : :.  .  .  : : . . : ', A1, ' :',
     1  9X, ':. . :' , A1, ' :. <   ', A1, '   >. : ', A1, ' :. .  :')
        WRITE (1, 1110) C (3) , B (1)
1110    FORMAT (5X, '.    : :. . : ', A1, ' : : ', A1, ' : . . : :
     1 . '. 9X, '. . : :. . .   . . .  : :. . ')
        WRITE (1, 1120)
1120    FORMAT (5X, '. . : :   ..   : :  .    : :. . ', 9X, '.    : :
     1 . . :  . . .   . . :   . ')
        WRITE (1, 1130) C (5) , B (5) , D (9) , F (2) , F (4) , E (7)
1130    FORMAT (5X, ':. . : ', A1, ' :. . : :. . : ',A1, ' :. . :', 9X, '
     1  : ', A1, ' : <   ', A1, '   > <   ', A1, '   > : ', A1, '. :')
        WRITE (1, 1140)
1140    FORMAT (5X, ':   ..   : :. . . . : :   ..   :', 9 X, ':.  ',
     1  2 ('.   . :.  ') , '. .  :')
        WRITE (1, 1150) C (7) , B (9)
1150    FORMAT (5X, ': ', A1, ' :. . : :   .  .    : :. . :. ', A1, '
     1:', 9X, ':  ', 2  ('.   .  .  ') , '.   . . :')
        WRITE (1, 1160) C (6) , B (4) , F (3) , F (5) , F (7)
1160    FORMAT (5X, '.    : :. . : ', A1, ' : : ', A1, ' :. . : :  . ',
     1  10X, 3 ('<',A1, '   > ') )
        WRITE (1, 1170)
1170    FORMAT (7X, '. : :   ..   : :   ..   : :. ', 14X, 2 ('.   .
     1 .  ') , '. . ')
        WRITE (1, 1180) C (8) , B (8)
1180    FORMAT (9X, '. : ', A1, ' :. . : :. . : ', A1, ' :. ', 18X, 2 ('
     1 .  .  .  .  ') , '. ')
        WRITE (1, 1190) F (6) , F (8)
1190    FORMAT (10X, '.    : :. . . . : :   . ', 20X, 2 ('<   ', A1, '
           > ') )
        WRITE (1, 1200)
1200    FORMAT (12X, '. : :   .  .    : :. ', 24X, '.   .  .  .  .  .')
        WRITE (1, 1210) C (9) , B (7)
1210    FORMAT (14X, '. : ', A1, ' : : ', A1, ' :. ', 28X, 2 ('.  .  '))
        WRITE (1, 1220) F (9)
```

```
1220      FORMAT (15X, '.     : :    . ', 30X, '<    ', A1, '  >')
          WRITE (1, 1230)
1230      FORMAT (17X, '. : :. ', 34X, '.     . ')
          WRITE (1, 1240)
1240      FORMAT (19X, '. . ', 38X, '. ')
          RETURN
          END
```

MIX RUBIK'S CUBE, TYPE A NUMBER PLEASE （请输入搞乱魔方的数）

9 （输入为“9”）

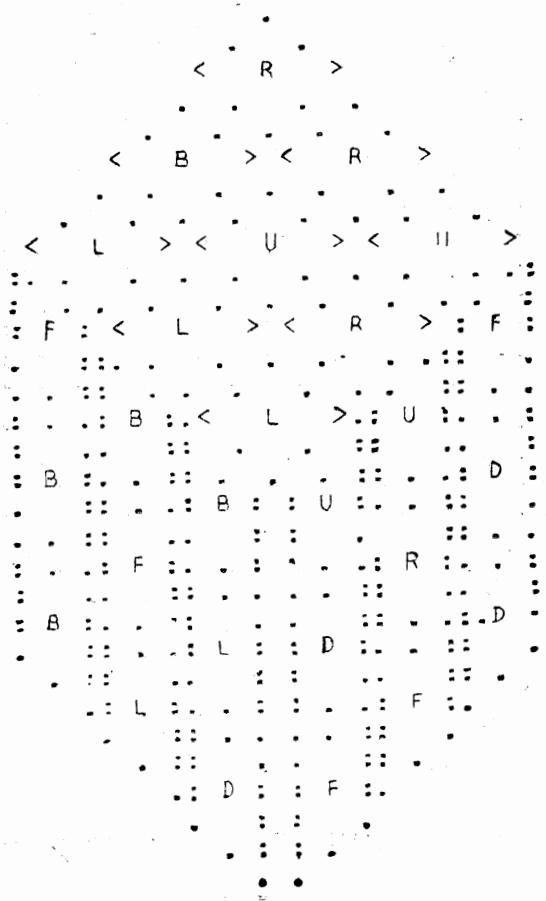


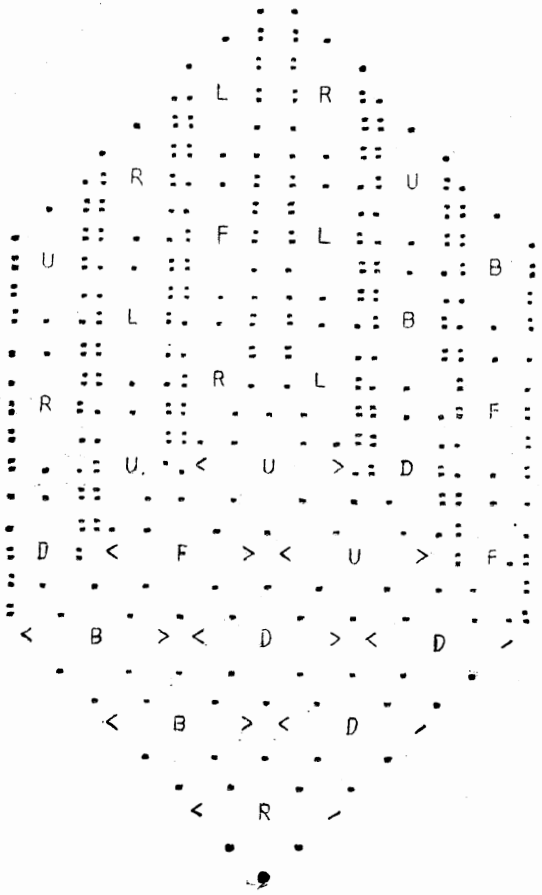


左边二图为输出图形

INPUT TURN DIRECTION YOU WANTED（请输入要转动的方向）

R2（输入为“R2”，即转动右平面180°）

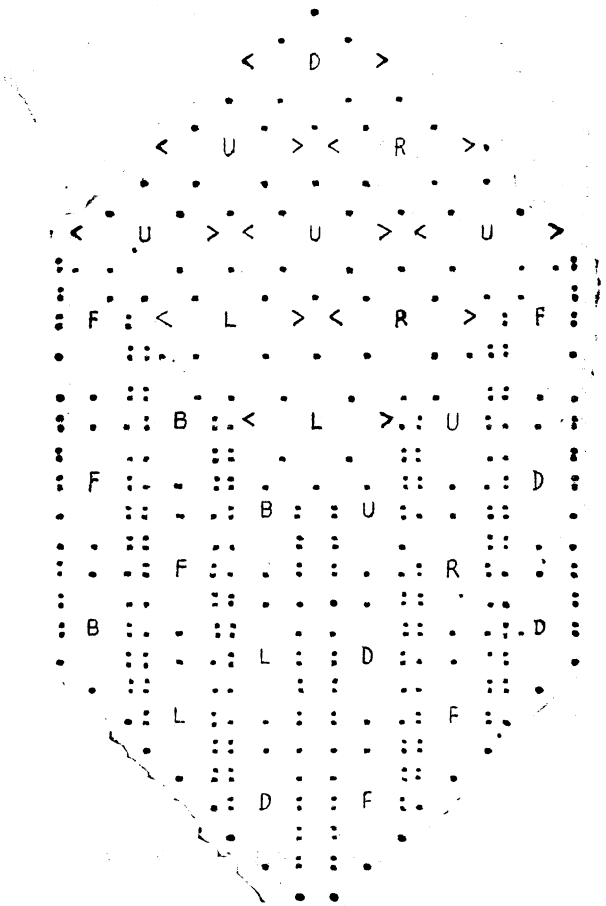


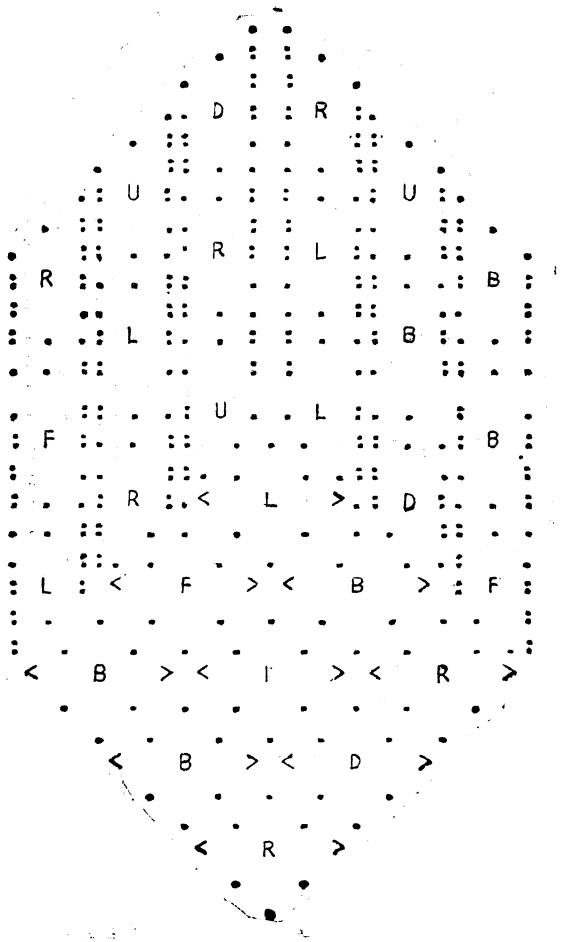


左边二图为输出结果

INPUT TURN DIRECTION YOU WANTED （输入要转动的方向）

L2（输入为“L2”，即转动左平面180°）





左边二图为输出结果

```
INPUT  TURN  DIRECTION  YOU  WANTED
S（输入“S”）
THANK  YOU  FOR  PLAYING                    （谢谢）
024535  STOP  0                             程序停止
```

（三）思　考　题

请设计出另外形式的输出图案。

十六　游览名胜十九处

106　如何画一笔画

一笔画从图论的角度看是一个欧拉图。当图中仅有二点为奇结点，其余为偶结点时，则可以一个奇结点为始点，走遍各偶结点及其路径，然后以另一奇结点离开此图。

（一）算法分析

由图论理论可知，任一欧拉图可以分成若干个流通子图，在找到走遍各路径的几个子图后，则可以遵循一定的规则不重复地走遍各通路。设有二个流通子图，则二个子图必有么共结点V，则可在第一个子图上沿结点L走到V，然后走遍第二子图又回到V，再续继走第一个子图的其余部分。这样就可不重复地走遍全部图。多个子图的情况依此类推。

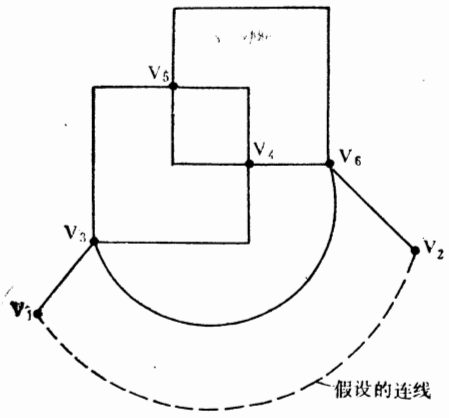


图 106-1

1）图106-1中奇点为V1、V2，则设由V_1出发，到达V_3。我们假设V_1、V_3之间有一连线，则图106-1为一个欧拉图。首先查找含有V_3、V_2的流通子图。具体的查找办法是：从V_3出发，找到与V_3相连的一点，若该点等于V_2，则可找到一个含有V_2的流通子图G_1。设图106-1中找到的是V_1、V_3、V_6、V_2、V_1。

2）检查是否还有未走过的连线，若没有，则说明各流通子图已经找到。否则选已经找到的子图上具有未走过路径的点作为新的子图的始点，图106-1中设V_3，则可找到G_2为V_3、V_4、V_5、V_3，为一个流通子图。同样有V_5、V_4、V_6、V_5为一子图。

3）按前述规则走遍全图。即从V_1出发，走到V_3后，该点与G_2相接，于是转过走G_2。当走到V_5时，又与G_3相接，则继续转G_3。G_3走遍后由V_5回到G_2，G_2子图由V_3又回到G_1，则可把G_1余下的路走完。示意性解释见图106-2。

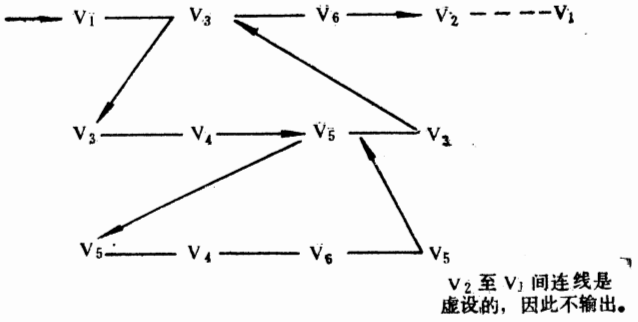


图 106-2

（二）程序设计

1）设计思路

为了完成查找子图路径的任务，我们设立几个数组，数组RE用于表达点与点之间的连线，比如，若RE（I，J）=2，则说明点I到点J有二条连线。由于是无向图，所以应有到点J有二条连线。由于是无向图，所以应有RE(I，J)=RE(J，I)。每当选中一条路径后，比如I点到J点，则要把RE(I，J)、RE(J，I)的值分别减1。

子程序LOOK用查找下一点。给定点I后，LOOK在RE的I行上查找不等于0的列，

若找到的列为J，则说明点I到点J有连线。

SA是一维数组，用于存放一个子图中的各点，当找到一个流通子图的全部点后，则把SA的内容送数组A中保存。二维数组A的每一行是一个流通子图的路径。

各子图的路径合并办法参考图106-2。程序中也是这样实现的。最后结果放入一维数组PATH中供输出。

本程序要求从键盘输入数据。因此你事先应把要求的图画出，找出结点间的连线，并按机器的要求逐一输入。输入结束用二个编号为0的结点表示。

对于图106-3由键盘输入，当程序运行后输出的结果详见打印。

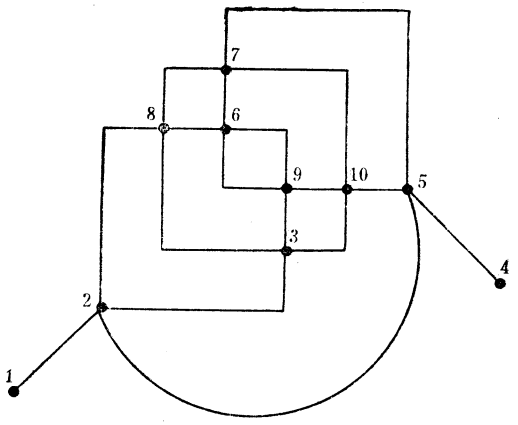


图 106-3

2) 框图 见图106-4

3) 源程序及运行结果

```
C***  PROGRAM  NAME: GRAPH                          程序名: GRAPH
C***  THIS  PROGRAM  SEARCH  THE  PATHES  FOR  YOU
C***  THE  GRAPH  MUST  BE  CAN  BE  DRAW  BY
C***
      INTEGER  RE(20, 20) , A(20, 21) , SA(20) , PATH(100)
      INTEGER  NODE, I, J, ODD1, ODD2, STEP, END, N1, N2, PC, G,
        COUNT
      OPEN   (4, FILE='GH', ACCESS='W')
      DO  5  J=1, 21
      DO  5  I=1, 20
5     A (I, J) =0
      DO  8  I=1, 100
8     PATH(I) =0                                   清除数组A、PATH、SA、RE。
      DO  10  J=1, 20
      SA(J) =0
      DO  10  I=1, 20
10    RE(I, J) =0
      WRITE (4, 15)
15    FORMAT (5X, 'HOW  MANY  NODES  ARE  THERE  IN  YOU  GRAPH?
        TELL  ME  PLEASE', /)
      READ(1, 30)  NODE                            输入结点数目
33    FORMAT (5X, I2)
30    FORMAT (I2)
      IF   (NODE. EQ. O)   GO  TO  350
      WRITE (1, 40)
40    FORMAT   (5X, 'WHAT  TWO  NODES  ARE  THE  ODD  ONE? TELL
```

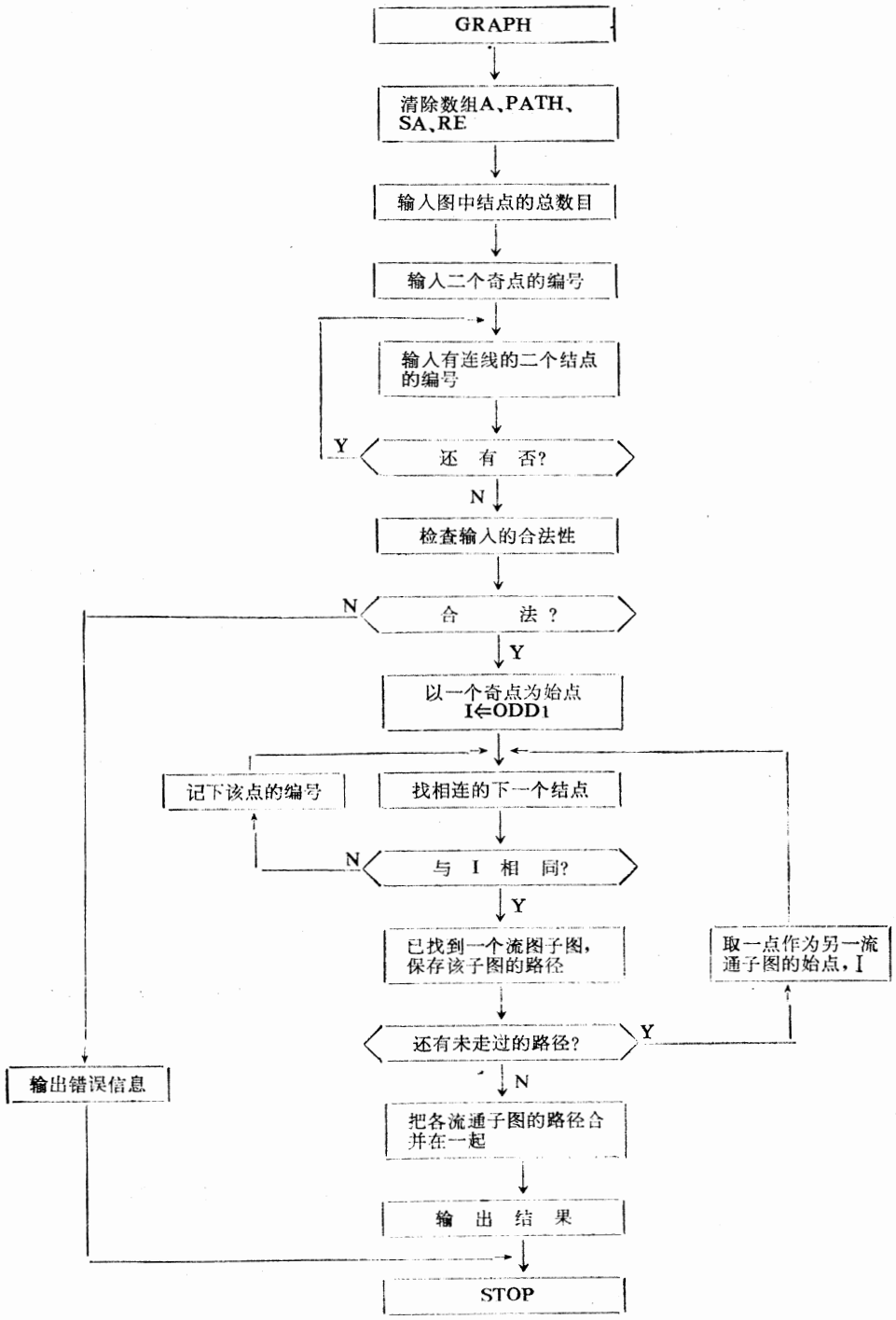

GRAPH
清除数组A、PATH、SA、RE
输入图中结点的总数目
输入二个奇点的编号
输入有连线的二个结点的编号
Y
还 有 否?
N
检查输入的合法性
N
合 法 ?
Y
以一个奇点为始点 I⇐ODD1
记下该点的编号
找相连的下一个结点
N
与 I 相 同?
Y
已找到一个流图子图，保存该子图的路径
取一点作为另一流通子图的始点，I
还有未走过的路径?
Y
N
输出错误信息
把各流通子图的路径合并在一起
输 出 结 果
STOP

图 106-4

```
      ME ALSO PLEASE', 1 /)                          输入二个奇结点
      READ (1, 50) ODD1, ODD2
50    FORMAT (I2, 1X, I2)
      RE(ODD1, ODD2) =RE(ODD1, ODD2) +1
      RE(ODD2, ODD1) =RE(ODD2, ODD1) +1
      WRITE (1, 60)
60    FORMAT (5X, 'TYPE THE ALL PATHES BETWEEN THE PAIR
      OF TWO NODES', /)                              输入图中二点之间的路径
70    READ (1, 50) I, J
      IF (I. EQ. O. OR. J. EQ. O) GO TO 90
      IF (I. EQ. J) GO TO 80
      RE(I, J) =RE(I, J) +1
      RE(J, I) =RE(J, I) +1
      GO TO 85
80    RE(I, J) =RE(I, J) +1
85    WRITE (1, 86)
86    FORMAT (5X, 'TYPE THE NEXT PATH', /)
      GO TO 70
90    DO 100 I=1, 20                                 检查输入数据的合法性
      COUNT=0
      DO 100 J=1, 20
      COUNT=COUNT+RE(I, J)
      IF ( (COUNT/2) -INT (COUNT/2). GT.
      0) GO TO 330
100   CONTINUE
      I=ODD1
      CALL LOOK (I, N1, END, RE)                     奇点1作为始点
      RE(N1, I) =RE(N1, I) -1
      RE(I, N1) =RE(I, N1) -1
      SA(1) =ODD1
      SA(2) =N1
      G=1
      PC=3
      I=N1
C***  SEARCH NEXT POINT                              找出一个流通子图
110   CALL LOOK (I, J, END, RE)
      IF (E. EQ. 1) GO TO 330
      SA (PC) =J
      RE(I, J) =RE(I, J) -1
      RE(J, I) =RE(J, I) -1
      IF (J. EQ. SA(19) GO TO 130
      PC=PC+1
      I=J
```

```
          GO  TO  110
      C***  SAVE  THE  PATH                    保存已找到子图的路径
  130     DO  150  STEP=1, PC
  150     A(G, STEP) =SA(STEP)
          A(G, 21) =PC
      C***  LOOK  FOR  OTHER  PATH
          DO  160  STEP=1, 20                  检查是否还有别的流通子图，
          COUNT=SA(STEP)                       若有，则返回110，若没有，则
          DO  160  J=1, 20                     顺序执行以下的语句。
          IF (RE(COUNT, J) . EQ. 0) GO  TO  160
          I=COUNT
          SA (1) =I
          PC=2
          G=G+1
          GO  TO  110
  160     CONTINUE
      C***  OUT                                输出结果
      C***
          PATH(1) =A(1, 1)
          ODD1=2
          DO  200  STEP=1, G                   按照算法中的规则，
          DO  197  COUNT=2, A(STEP, 21)        把几个流通子图的路
          PATH(ODD1) =A(STEP, COUNT)           径合在一起
          ODD1=ODD1+1
          IF  (A(STEP, COUNT). NE. A(STEP+1,
            1) ) GO  TO  197
          SA(STEP) =COUNT
      C   ODD1=ODD1-1
          GO  TO  200
  197     CONTINUE
  200     CONTINUE
          DO  300  N1=1, G-1
          ODD1=ODD1-1
          DO  300  STEP=SA(G-N1) , A(G-N1, 21)
          PATH(ODD1) =A(G-N1, STEP)
  300     ODD1=ODD1+1
          WRITE (4, 305)                       输出最后结果
  305     FORMAT (/, 2X, 'THE  PATH  SHOULD
            BE'/)
          WRITE (4, 310)  (PATH(I) , I=1, ODD1-2)
  310     FORMAT (I5)
          GO  TO  350
  330     WRITE (1, 340)
  340     FORMAT (5X, 'NO  PATH'/)
```

```
350       CLOSE (-1)
          STOP
          END
C***  SEARCH THE PATH FROM THE POINT L TO POINT C.
C***  E=0 IF THE PATH EXIST, OTHERWISE E=1        子程序LOOK
C***
          SUBROUTINE LOOK (L, C, E, RE1)          用于查找给定点L是否与其它
                                                  点相连，若相连则把另一点值
                                                  赋给C、且E=0，否则E=
                                                  1。
          INTEGER L, C, E, TEMP, RE1(20, 20)
          DO 430 TEMP=1, 20
          IF (RE1(L, TEMP) . NE. 0) GO TO 400
          E=1
          GO TO 430
400       C=TEMP
          E=0
          GO TO 440
430       CONTINUE
440       RETURN
          END
```

输入：

```
HOW MANY NODES ARE THERE IN YOU GRAPH? TELL ME
 PLEASE                                    (共有多少个结点？)
10          (输入值)
WHAT TWO NODES ARE THE ODD ONE? TELL ME ALSO
 PLEASE                                    (请输入二个奇结点的编号)
1  4
TYPE THE ALL PATHES BETWEEN THE PAIR OF TWO NODES
1  2          (输入值)
TYPE THE NEXT PATH                    (输入二个点之间的路径)
2  8        (输入值)
TYPE THE NEXT PATH                              以下同
2  3
TYPE THE NEXT PATH
2  5
TYPE THE NEXT PATH
3  10
TYPE THE NEXT PATH
3  9
TYPE THE NEXT PATH
9  10
```

```
TYPE  THE  NEXT  PATH
10  5
TYPE  THE  NEXT  PATH
5  4
TYPE  THE  NEXT  PATH
7  10
TYPE  THE  NEXT  PATH
7  5
TYPE  THE  NEXT  PATH
7  8
TYPE  THE  NEXT  PATH
8  6
TYPE  THE  NEXT  PATH
6  7
TYPE  THE  NEXT  PATH
6  9
TYPE  THE  NEXT  PATH
6  9
TYPE  THE  NEXT  PATH
8  3
TYPE  THE  NEXT  PATH
0  0          (表示输入结束)
```

输出：

```
THE  PATH  SHOULD  BE          (一条可能的路径是：)
1
2
3
9
6
8
7
10
9                              这是求图106-3给出的输入及其相应的输出
6
7
5
10
3
8
2
5
4
```

4）附注

本题要求欲求的结点数不超过20个，若超过20个，修改相应参数便可适应。

（三）思 考 题

本题的算法中不是查找最小的流通子图。请你用查找各个最小流通子图的办法编写本题的程序。

107 游览名胜十九处

几位程序设计工作者，来到了著名的名胜古迹十九处，游览区布局奇特新颖，在前门画有十九处的位置和通道，见图 107-1 所示，入口 1 和出口17也都是名胜处，而且指定入口买票进来，从后门17出去，更耐人寻味的是在入口处挂有醒目的大金字牌，上写："亲爱的旅游者，你能不走重复路游览这十九处吗？"

许多游客在纸上画来画去，有的画出来了，放声大笑，有的绕来绕去，唉声叹气。这几位程序设计者，对此牌提出的问题也颇感兴趣，边游览边思索，能否编个程序，让计算机告诉每个游客，如何走才能解决牌上提出的问题？

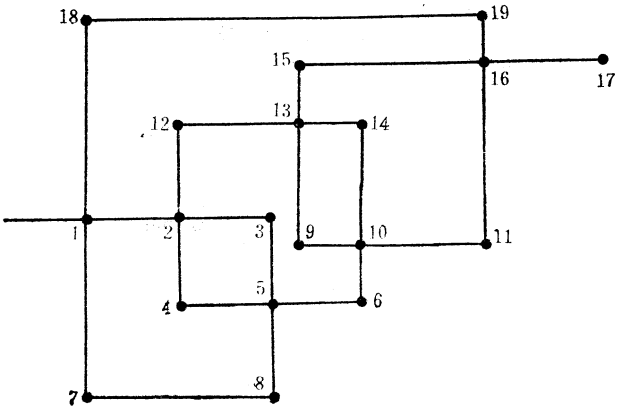


图 107-1

表 107 路 径 表

	x	y	其它邻接点			
1	10	12	18	2	7	0
2	10	20	12	3	4	0
3	10	28	2	5	0	0
4	16	20	2	5	0	0
5	16	28	4	3	6	8
6	16	40	5	10	0	0
7	20	12	1	8	0	0
8	20	28	7	5	0	0
9	12	32	13	10	0	0
10	12	40	9	11	6	14
11	12	48	10	16	0	0
12	6	20	2	13	0	0
13	6	32	12	15	14	9
14	6	40	13	10	0	0
15	4	32	13	16	0	0
16	4	48	15	19	17	11
17	4	60	16	0	0	0
18	2	12	1	19	0	0
19	2	48	18	16	0	0

（一）算法分析和结论

这几位程序设计者，回答是肯定的，可以编写程序，让计算机指导你游览时不走重复路而又能逛完这十九处，算法分析省略（可参看设计思路和程序旁注），结论是肯定的，具体如何走呢？请看程序运行过程：游客每走一处，形象地打印出一步步星号标出的路径图。（当然也可以在终端屏幕上显示出在图 107-1 上所走的路径星号标出）。

（二）程 序 设 计

1）设计思路

把上图画在座标方格纸上，用该数字所在处的X、Y坐标表示其方位。例如名胜 1 、 2 、 3 的X值相同，名胜 1 、 7 、18的Y值相同。X、Y值可由所取的比例不同而异。本程序使用的值是将游览区给的里数折合成百米单位列入表107。

由表 107 可见名胜 1 与名胜 2 、18、 7 三处相通，…。取以上这些数值可造一个路径表，如表 107 中的一行是与一个名胜处有关的信息，其

第1列表示其X值，第2列表示其Y值，第3～6列表示与其相通的名胜，若为0则表示没有。例如，第2列表示名胜2的有关信息。其处于X＝10、Y＝20处，与其相通的有名胜12、3、4三处。程序中把路径表放在数组A 1中。

为了便于打印输出，用一个20×60的二维数组，用于存放十九处。若直接用显示器的指令，则可输出其它方式。

设从1号点出发，每次从键盘上读入一个你指定的数。可以看出，每次敲入的数的X、Y值中总应有一个与上次敲入的数的X、Y值相同。否则输入非法，要求你重新敲数。当检查输入合法后，程序则把你走的路用星号标记下，便于你观看，并确定下一步。详见打印出图1～图25，它是走一遍十九处的完整的例子。

若走错了，机器会告诉您相应信息，并请求重新输入。若走对了，机器会说您对了。

2）框图　见107-2

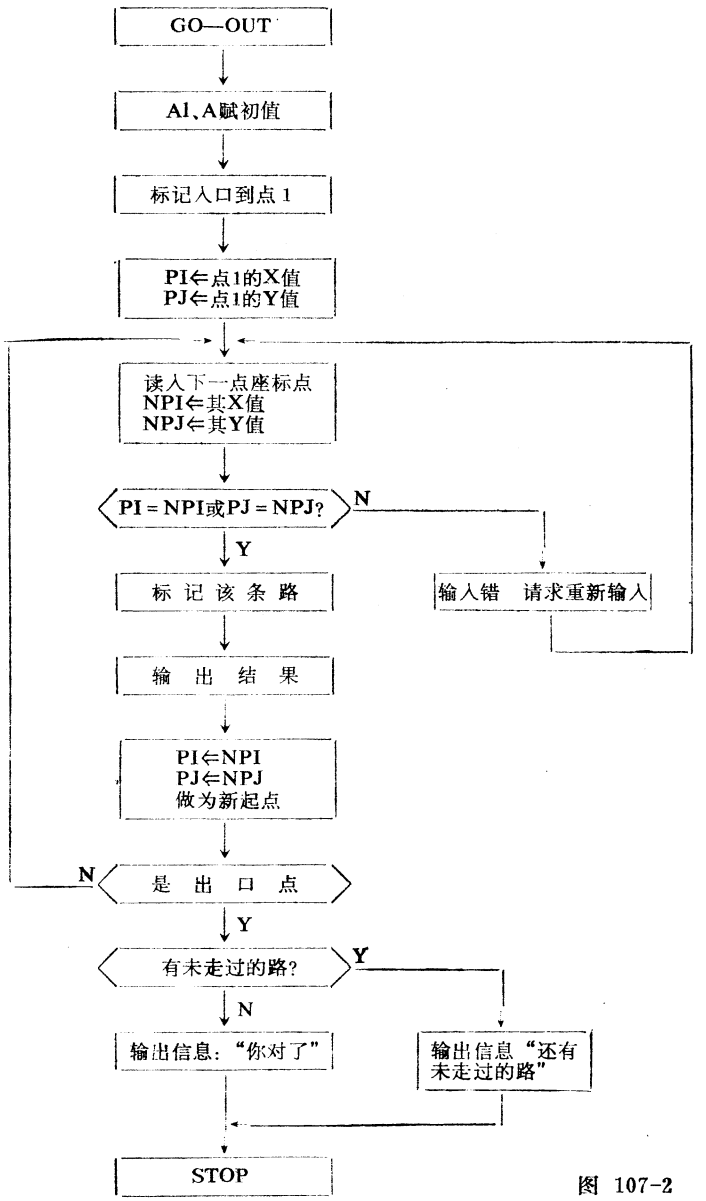


图 107-2

3）源程序及运行结果

```
C ***  PROGRAM  NAME:  GO—OUT
C ***
      INTEGER  I, J, PI, PJ, NPI, NPJ, FI, NODE, FJ
      INTEGER  A1 (19, 6)
      CHARACTER  A (20, 60) , GO
      DATA  A1 (1, 1) , A1 (2, 1) , A1 (3, 1) ,
     A1 (4, 1) , A1 (5, 1) , A1 (6, 1) , A1 (7,
     1) , A1 (8, 1) , A1 (9, 1) , A1 (10, 1) ,
     A1 (11, 1) , A1 (12, 1), A1 (13, 1), A1 (14,
     1) , A1 (15, 1) , A1 (16, 1) , A1 (17, 1), A1
      (18, 1) , A1 (19, 1) , A1 (1, 2), A1 (2, 2),
     A1 (3, 2) , A1 (4, 2) , A1 (5, 2) , A1 (6,
     2) , A1 (7, 2) , A1 (8, 2) , A1 (9, 2) , A1
      (10, 2) , A1 (11, 2) , A1 (12, 2) , A1 (13,
     2) , A1 (14, 2), A1 (15, 2), A1 (16, 2) , A1
      (17, 2) , A1 (18, 2) , A1 (19, 2) , A1 (1,
     3) , A1 (2, 3) , A1 (3, 3) , A1 (4, 3) , A1
      (5, 3) , A1 (6, 3) , A1 (7, 3) , A1 (8, 3) ,
     A1 (9, 3) , A1 (10, 3) , A1 (11, 3), A1 (12,
     3) , A1 (13, 3) , A1 (14, 3) , A1 (15, 3) ,
     A1 (16, 3) , A1 (17, 3) , A1 (18, 3) , A1
      (19, 3) /3*10, 3*16, 2*20, 3*12, 3*6,
     3*4, 2*2, 12, 20, 28, 20, 28, 40, 12, 28, 32,
     40, 48, 20, 32, 40, 32, 48, 60, 12, 48, 18, 12,
     2, 2, 4, 5, 1, 7, 13, 9, 10, 2, 12, 2*13, 15,
     16, 1, 18/                                          } 形成路径表的数组A 1
      DATA  A1 (1, 4) , A1 (2, 4) , A1 (3, 4) , A1
      (4, 4) , A1 (5, 4) , A1 (6, 4) , A1 (7, 4) ,
     A1 (8, 4) , A1 (9, 4) , A1 (10, 4) , A1 (11,
     4) , A1 (12, 4) , A1 (13, 4) , A1 (14, 4) ,
     A1 (15, 4) , A1 (16, 4) , A1 (17, 4) , A1
      (18, 4) , A1 (19, 4) , A1 (1, 5), A1 (2, 5),
     A1 (3, 5) , A1 (4, 5) , A1 (5, 5) , A1 (6,
     5) , A1 (7, 5) , A1 (8, 5) , A1 (9, 5) , A1
      (10, 5) , A1 (11, 5) , A1 (12, 5) , A1 (13,
     5) , A1 (14, 5) , A1 (15, 5) , A1 (16, 5) ,
     A1 (17, 5) , A1 (18, 5) , A1 (19, 5), A1 (1,
     6) , A1 (2, 6) , A1 (3, 6) , A1 (4, 6) , A1
      (5, 6) , A1 (6, 6) , A1 (7, 6) , A1 (8, 6) ,
     A1 (9, 6) , A1 (10, 6) , A1 (11, 6), A1 (12,
     6) , A1 (13, 6) , A1 (14, 6) , A1 (15, 6) ,
     A1 (16, 6) , A1 (17, 6) , A1 (18, 6) , A1
      (19, 6) /2, 3, 5, 5, 3, 10, 8, 5, 10, 11, 16,
```

```
      13, 15, 10, 16, 19, 0, 19, 16, 7, 4, 0, 0, 6,
      4*0, 6, 0, 0, 14, 0, 0, 17, 3*0, 4*0, 8,
      4*0, 14, 0, 0, 9, 0, 0, 11, 3*0/
      DO 5 I=1, 20
      DO 5 J=1, 60
5     A (I, J) =' '
      WRITE (1, 10)
10    FORMAT (5X, 'WHAT GRAPH DO YOU PERFER? (1, 2, 3, 4,
      5) ', /)
C***  SELECT A GRAPH
20    READ (1, 30) NODE                          把A置成初态
30    FORMAT (I1)
100   A(10, 11) ='1'
      A(10, 19) ='2'
      A(10, 27) ='3'
      A(16, 19) ='4'
      A(16, 27) ='5'
      A(16, 39) ='6'
      A(20, 11) ='7'
      A(20, 27) ='8'
      A(12, 31) ='9'
      A(12, 38) ='1'
      A(12, 39) ='0'
      A(12, 46) ='1'
      A(12, 47) ='1'
      A(6, 18) ='1'
      A(6, 19) ='2'
      A(6, 30) ='1'
      A(6, 31) ='3'
      A(6, 38) ='1'
      A(6, 39) ='4'
      A(4, 30) ='1'
      A(4, 31) ='5'
      A(4, 46) ='1'
      A(4, 47) ='6'
      A(4, 58) ='1'
      A(4, 59) ='7'
      A(2, 10) ='1'
      A(2, 11) ='8'
      A(2, 46) ='1'
      A(2, 47) ='9'
C***  DRAW THE GRAPH
      DO 105 J=1, A1(1, 2)-3
```

```
105       A(A1 (1, 1) , J) ='·'
          DO  190  I=1, 19
          PI=A1(I, 1)
          PJ=A1(I, 2)
          DO  180  J=3, 6
          IF   (A1(I, J) . EQ. 0) GO  TO  180
          NPI=A 1 (A1(I, J) , 1)
          NPJ=A 1 (A1(I, J) , 2)
          IF (PI. EQ. NPI) GO  TO  170
C***      DRAW  COLUMU
          IF (PI. GT. NPI) GO  TO  160
          DO  150  FI=PI+1, NPI-1
150       A(FI, PJ) ='·'
          GO  TO  180
160       DO  110  FI=NPI+1, PI-1
110       A(FI, PJ) ='·'
          GO  TO  180
C***      DRAW  LINE
170       IF (PJ. GT. NPJ) GO  TO  120
          DO  130  FJ=PJ+1, NPJ-3
130       A(PI, FJ) ='·'
          GO  TO  180
120       DO  140  FJ=NPJ+1, PJ-3
140       A(PI, FJ) ='·'
180       CONTINUE
190       CONTINUE
          WRITE (1, 195)  ( (A(I, J) , J=1, 60) , I=1, 20)
195       FORMAT (5X, 60A 1)
C***      FIRST  STEP
600       PI=A1(1, 1)                                          ┐
          PJ=A1(1, 2)                                          │ 把入口处到标号
          DO  610  J=1, PJ-3                                   │
610       A(PI, J) ='*'                                        │ 1处标为星号
          WRITE (1, 195)  ( (A(I, J) . J=1, 60) . I=1, 20)     ┘
C***      INPUT  NEXT  POINT
620       WRITE (1, 625)                                       ┐
625       FORMAT (5X, 'NEXT  NODE  YOU  GO  AHEAD')            │ 读入你要走的下一
          READ (1, 630) NODE                                   │ 个点的标号，并取
630       FORMAT (I2)                                          │ 出其X、Y值。
          NPI=A1(NODE, 1)                                      │
          NPJ=A1(NODE, 2)                                      ┘
632       FORMAT (5X, 'PI=', I3, ', NPI=', I3, ', PJ=', I3, ', NPJ=', I3)
          IF ( (PI. EQ. NPI) . OR.  (PJ. EQ. NPJ) ) GO  TO  650   输入检查
```

```
        WRITE (1, 640)
640     FORMAT (5X, 'INCORRECT INPUT, ONCE
        AGAIN PLEASE')
        GO TO 620
650     IF (PI. NE. NPI) GO TO 690
 C***   THE SAME LINE
        IF (PJ. GT. NPJ) GO TO 670
        DO 660 J=PJ+1, NPJ-3
660     A(PI, J) ='*'
        GO TO 740
670     DO 680 J=NPJ+1, PJ-3
680     A(PI, J) ='*'
        GO TO 740
 C***   THE SAME COLUMU
690     IF (PI. GT. NPI) GO TO 710
        DO 700 I=PI+1, NPI-1
700     A(I, PJ) ='*'
        GO TO 740
710     DO 720 I=NPI+1, PI-1
720     A(I, PJ) ='*'
 C***   NEW START POINT
740     PI=NPI
        PJ=NPJ
        WRITE (1, 745) ((A(I, J), J=1, 60), I=1, 20)
745     FORMAT (5X, 60A1)
        IF ((NPI. NE. A1(17, 1)). OR. (NPJ. NE.
          A1(17, 2))) GO TO 620
        DO 750 I=1, 20
        DO 750 J=1, 60
        IF (A(I, J) . EQ. '. ') GO TO 780
750     CONTINUE
        WRITE (1, 760)
760     FORMAT (5X, 'YOU ARE RIGHT, OTHER
          GRAPH DO YOU WANT? ')
        READ (1, 30) NODE
        IF (NODE. EQ. 0)  GO TO 820
        GO TO 100
780     WRITE (1, 790)
790     FORMAT (5X, 'THERE ARE SOME PATH
          YOU DID NOT GO THROUGH, OTHER
          GRAPH DO YOU WANT? ')
        READ (1, 800) GO
800     FORMAT (A1)
        IF  (GO. EQ. 'Y') GO TO 20
```

错，重新输入

根据输入数据把相应的通路用星号标记上

以走到的点作为新的起点

检查是否走到出口

检查是否有未走到的路径

正确

给出信息“有的路未走到”

```
820         STOP
            END
```

以下是程序运行时打印结果（我们编上图号，从图 1 至图25）

```
WHAT  GRAPH  DO  YOU  PERFER?  (1, 2, 3, 4, 5)
1
```

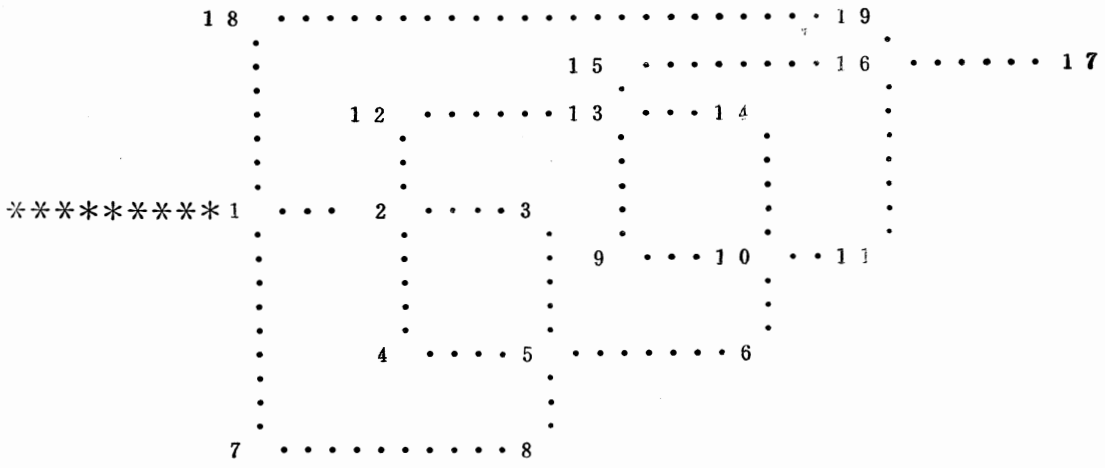


图 1　从名胜 1 开始

```
NEXT  NODE  YOU  GO  AHEAD
0       (输入 0)
INCORRECT  INPUT, ONCE  AGAIN  PLEASE    要求重新输入，因输入错。
NEXT  NODE  YOU  GO  AHEAD
2       (输入 2，对了)
```

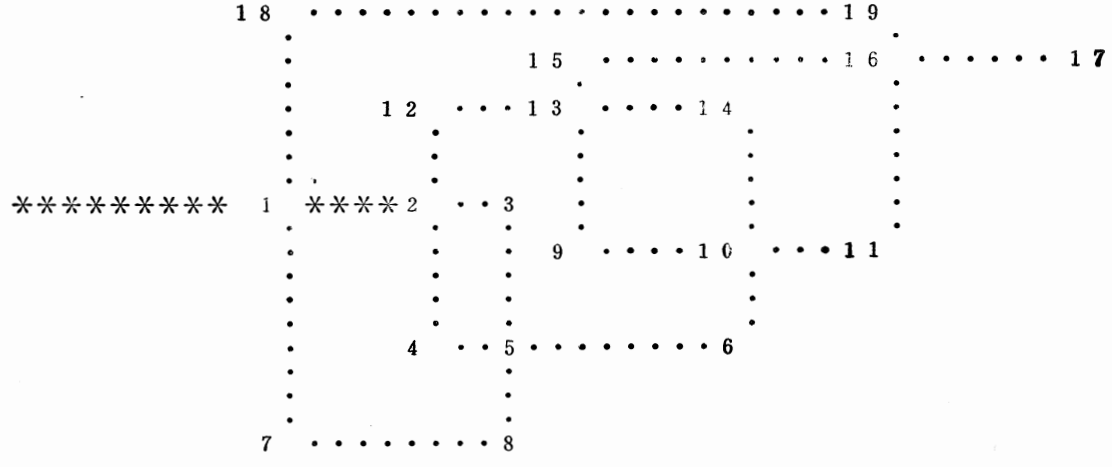


图 2　指定下一步走到 2

NEXT NODE YOU GO AHEAD
12

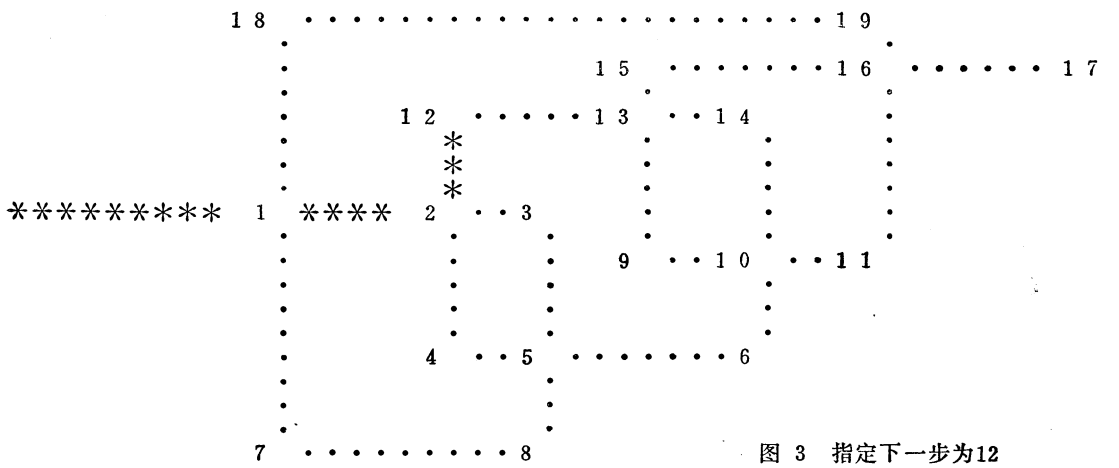


图 3 指定下一步为12

NEXT NODE YOU GO AHEAD
13

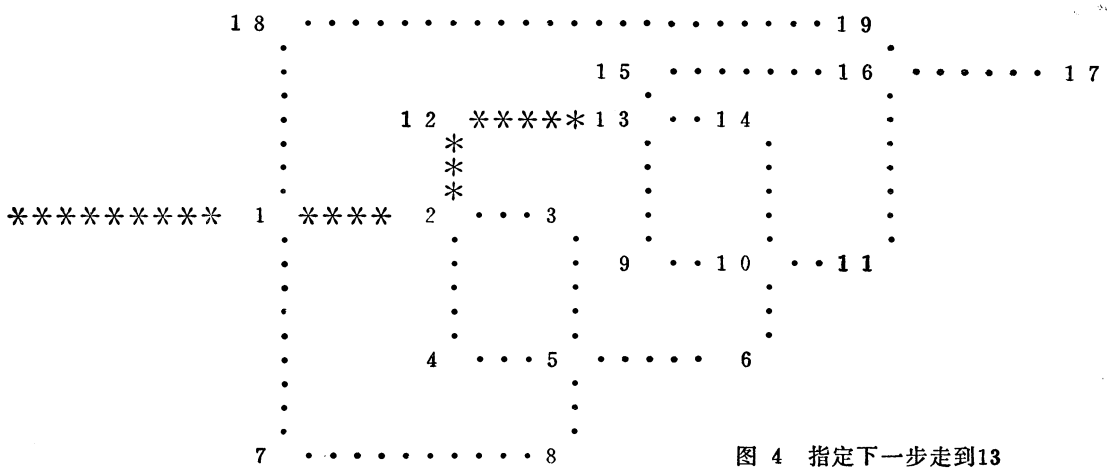


图 4 指定下一步走到13

NEXT NODE YOU GO AHEAD
15

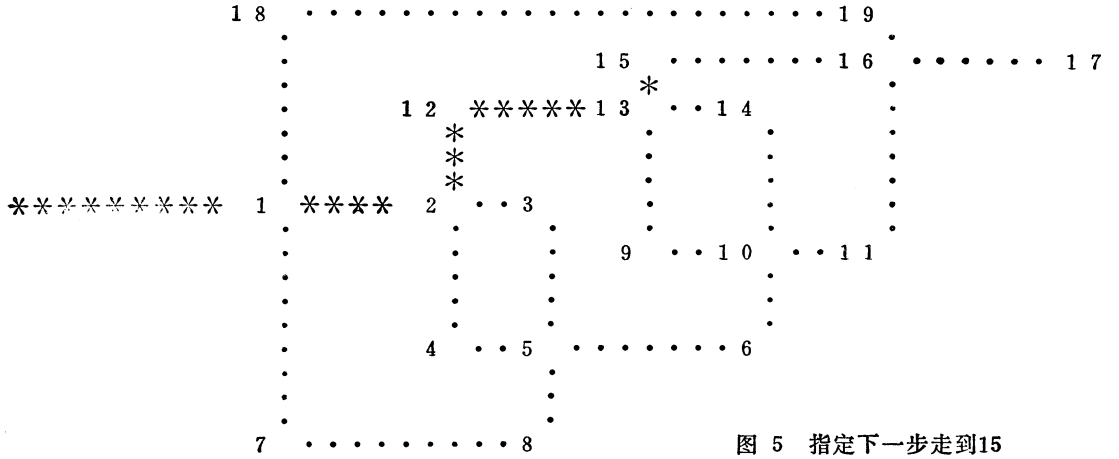


图 5 指定下一步走到15

NEXT NODE YOU GO AHEAD
16

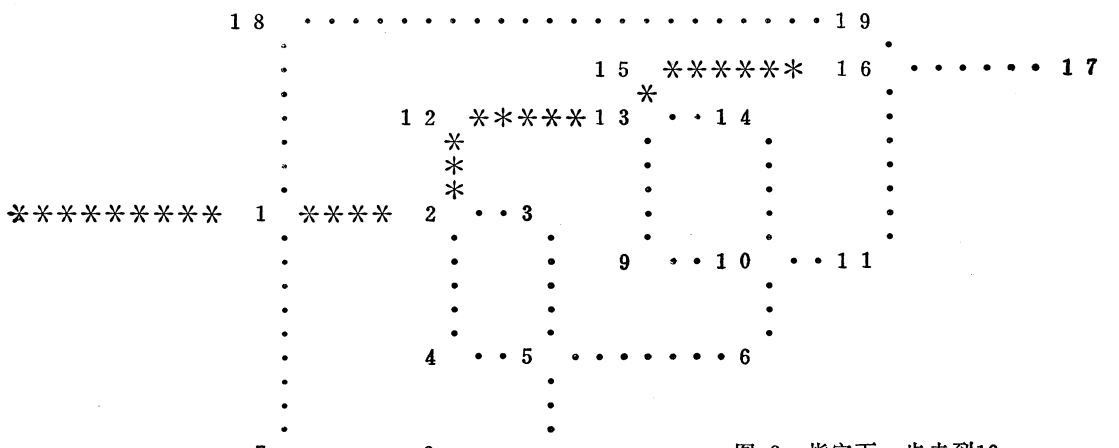


图 6 指定下一步走到16

NEXT NODE YOU GO AHEAD
11

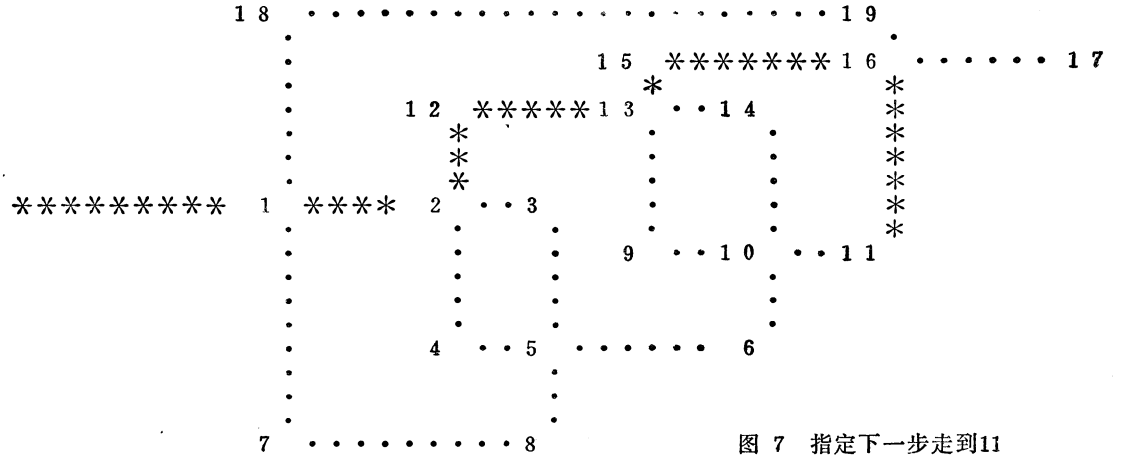


图 7 指定下一步走到11

NEXT NODE YOU GO AHEAD
10

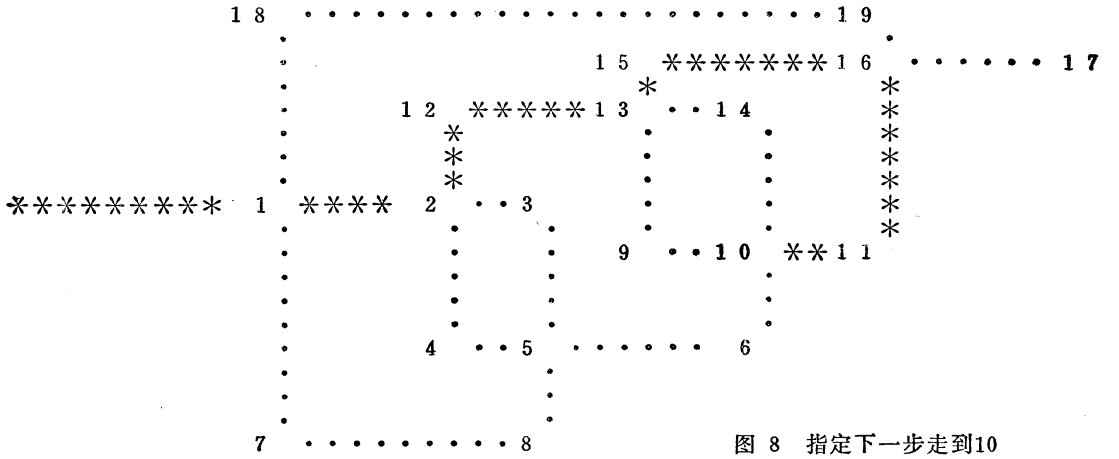


图 8 指定下一步走到10

NEXT NODE YOU GO AHEAD
9

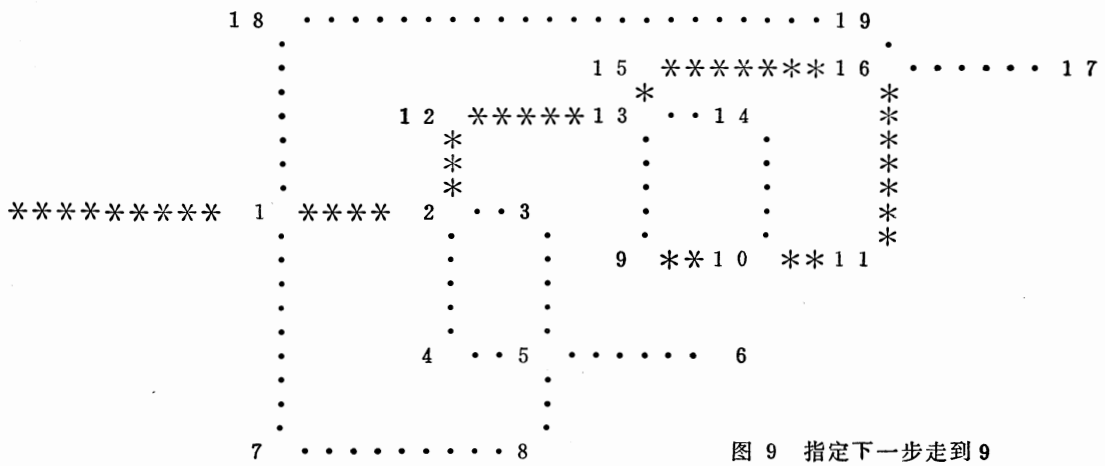


图 9 指定下一步走到9

NEXT NODE YOU GO AHEAD
13

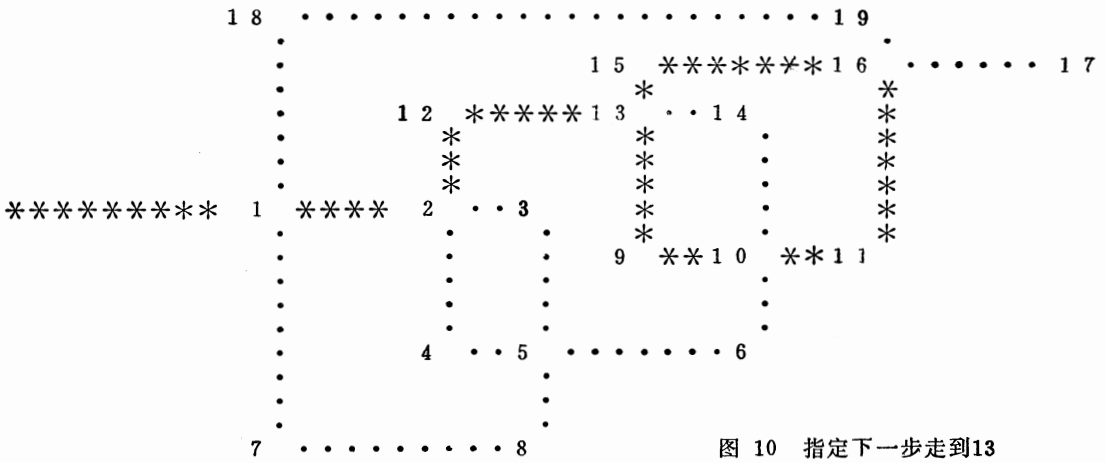


图 10 指定下一步走到13

NEXT NODE YOU GO AHEAD
14

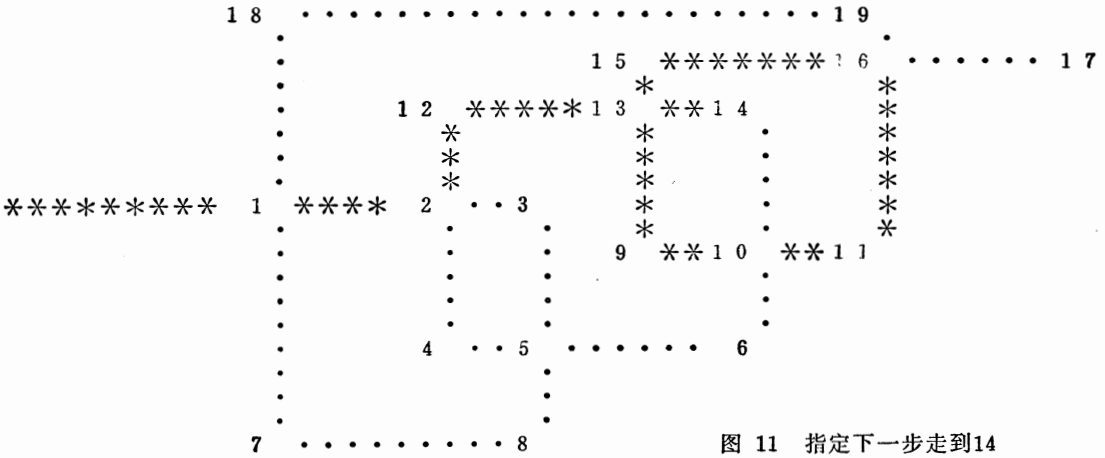


图 11 指定下一步走到14

NEXT NODE YOU GO AHEAD
10

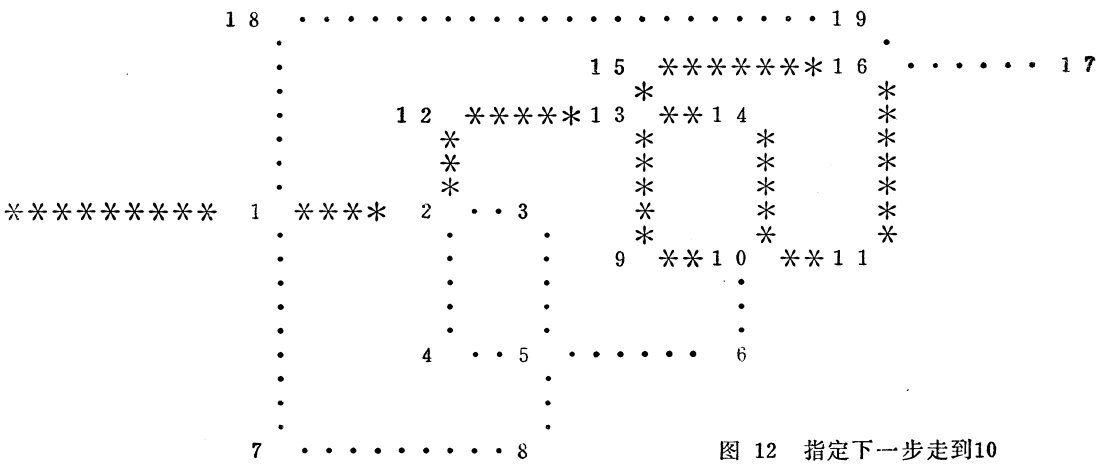


图 12 指定下一步走到10

NEXT NODE YOU GO AHEAD
6

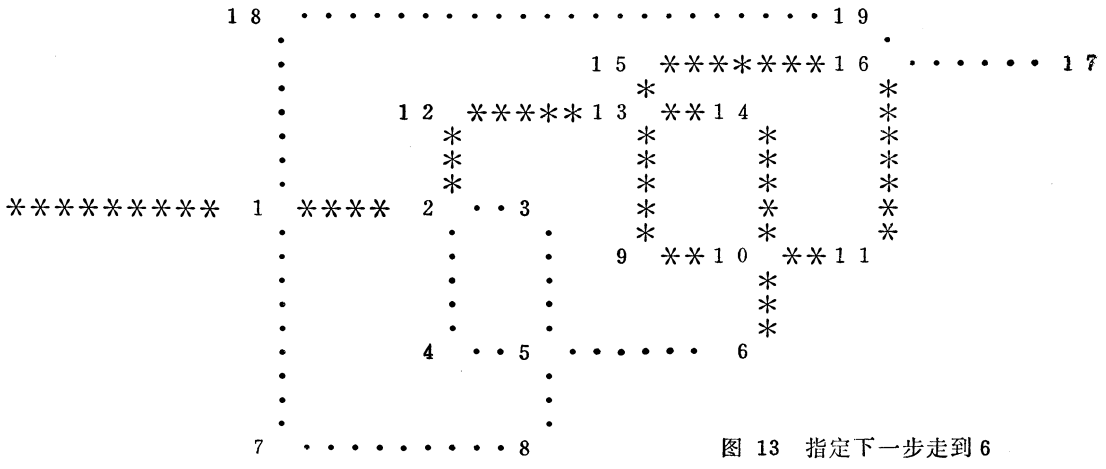


图 13 指定下一步走到 6

NEXT NODE YOU GO AHEAD
5

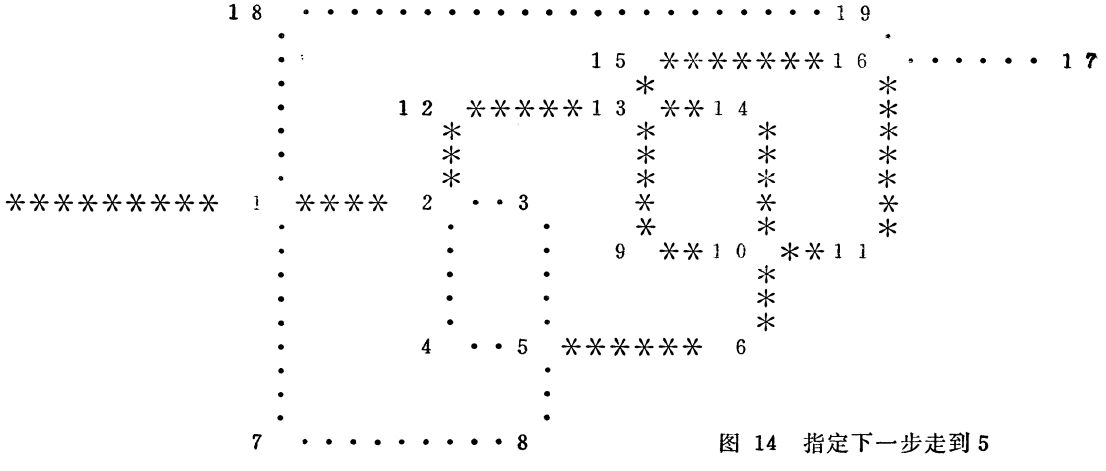


图 14 指定下一步走到 5

NEXT NODE YOU GO AHEAD
3

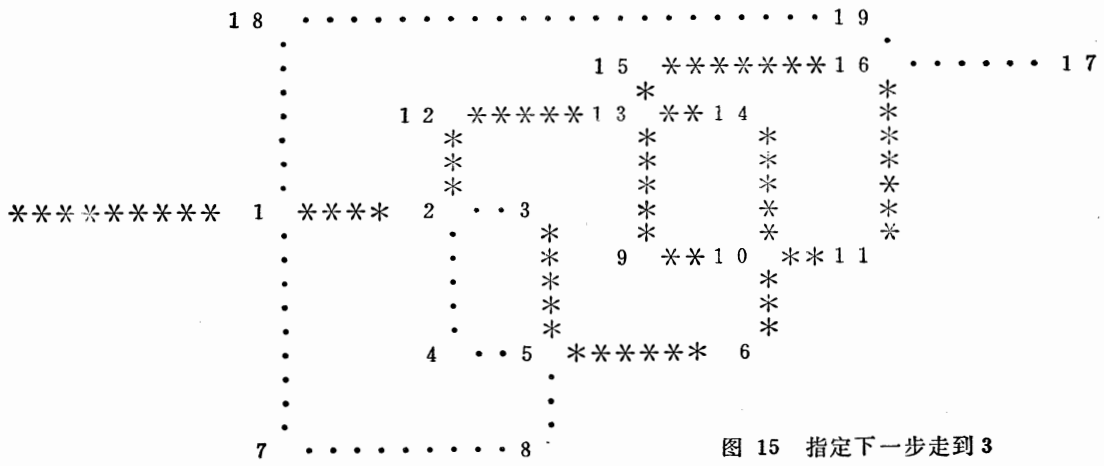


图 15 指定下一步走到 3

NEXT NODE YOU GO AHEAD
2

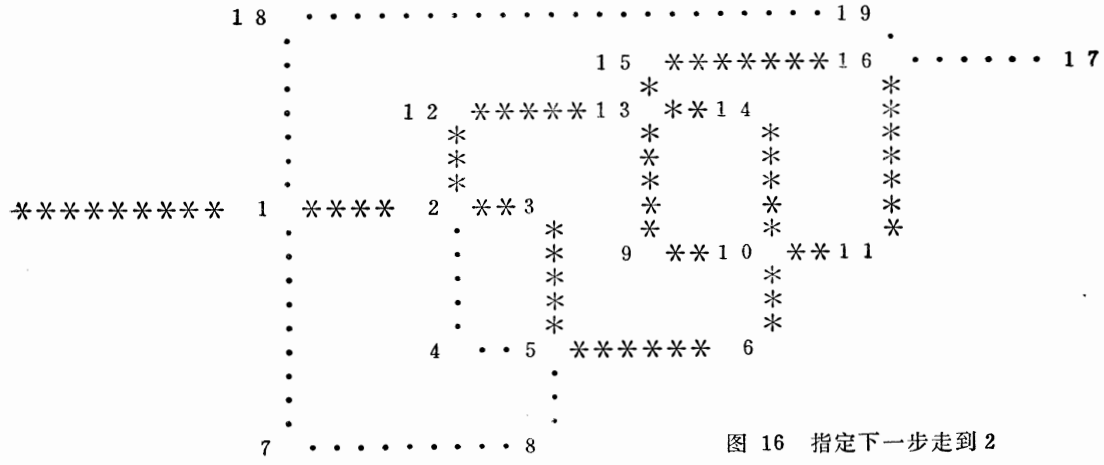


图 16 指定下一步走到 2

NEXT NODE YOU GO AHEAD
4

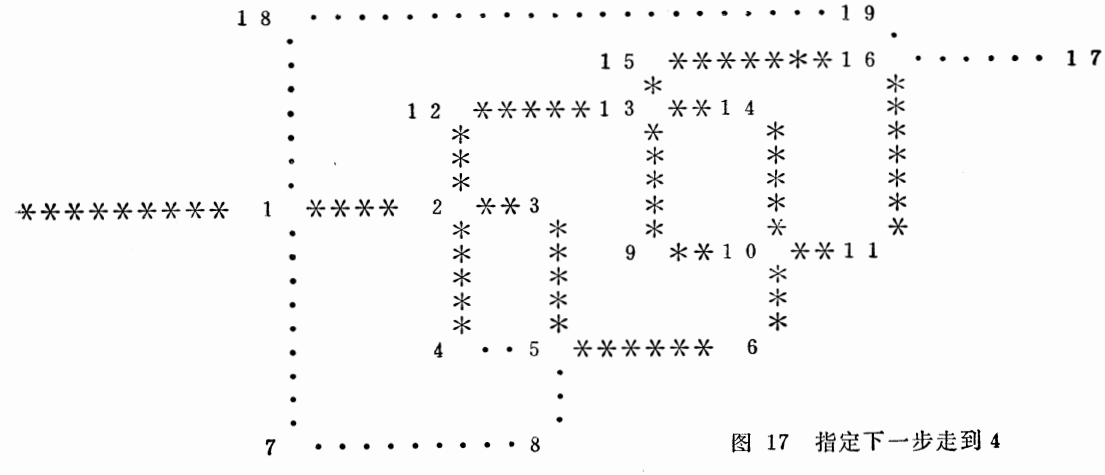


图 17 指定下一步走到 4

NEXT NODE YOU GO AHEAD
5

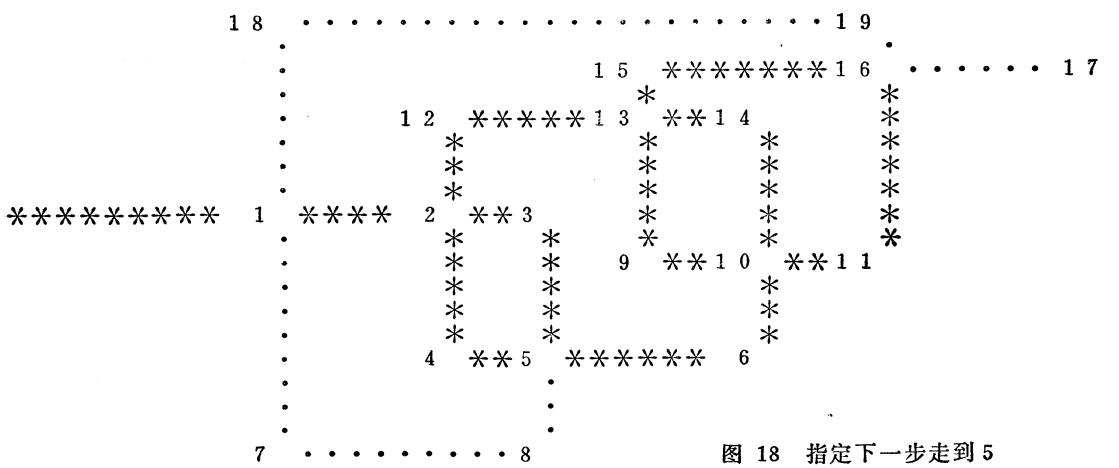


图 18 指定下一步走到 5

NEXT NODE YOU GO AHEAD
8

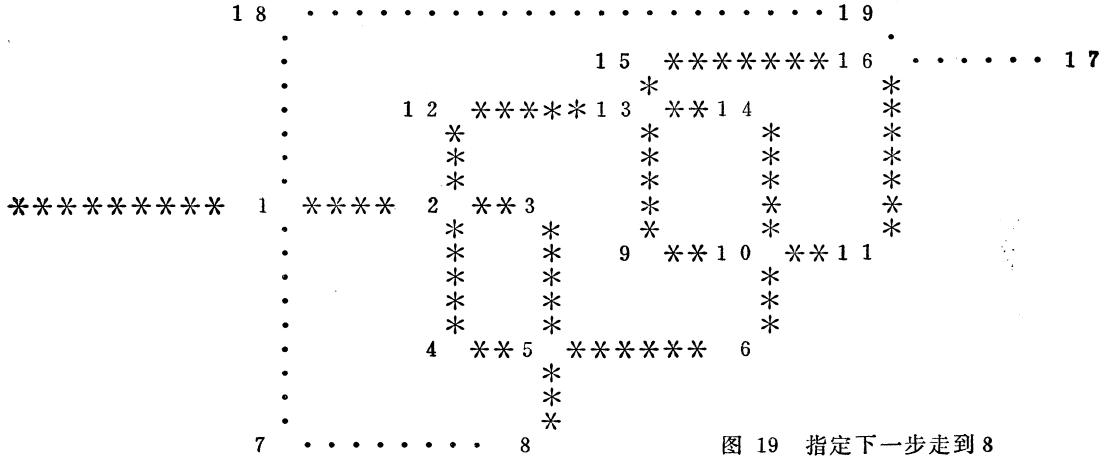


图 19 指定下一步走到 8

NEXT NODE YOU GO AHEAD
7

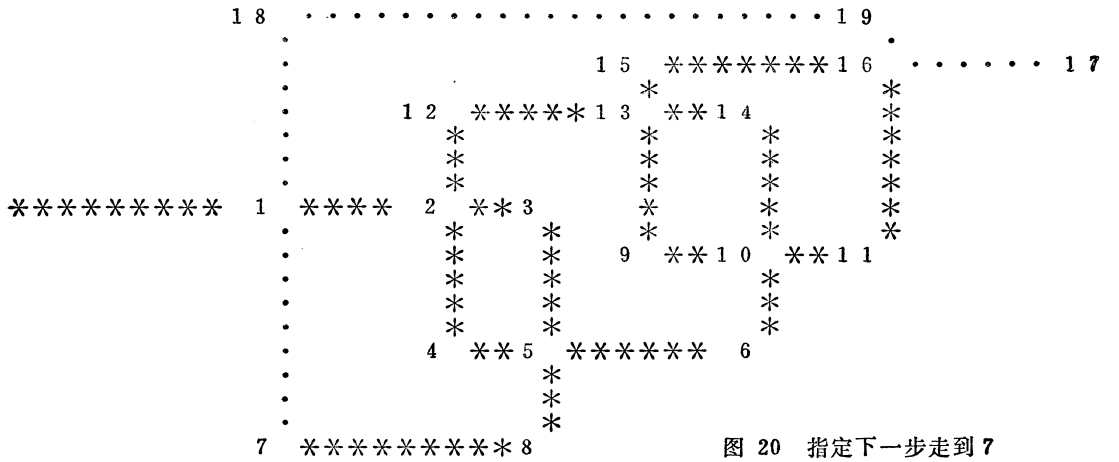


图 20 指定下一步走到 7

NEXT NODE YOU GO AHEAD
1

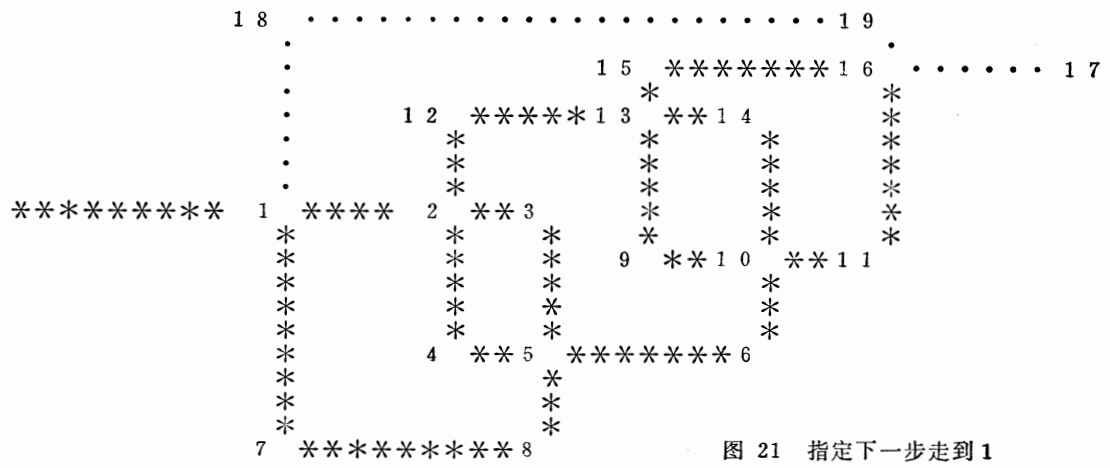


图 21 指定下一步走到1

NEXT NODE YOU GO AHEAD
18

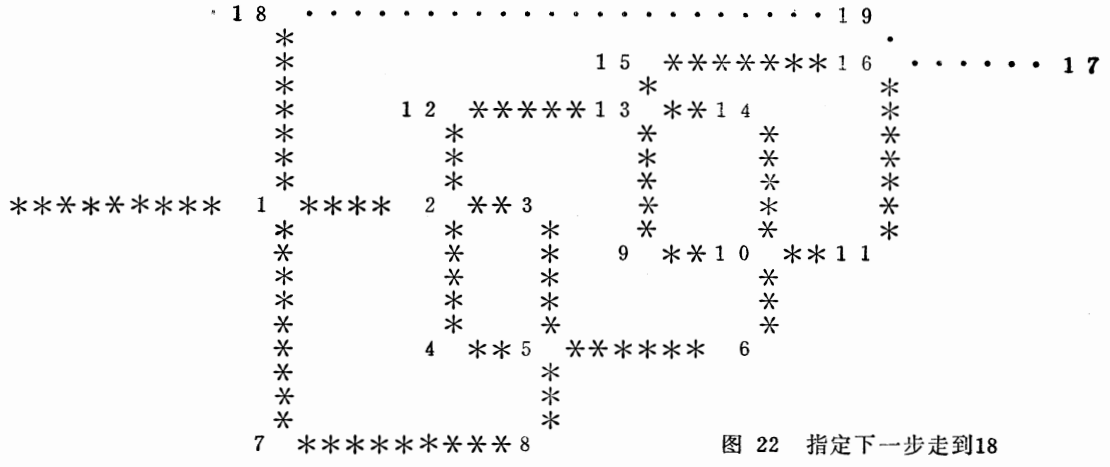


图 22 指定下一步走到18

NEXT NODE YOU GO AHEAD
19

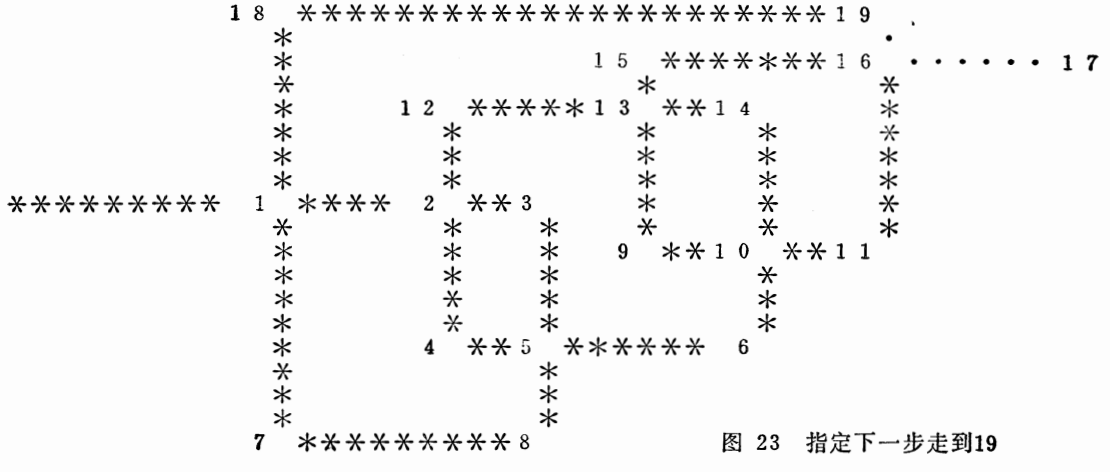


图 23 指定下一步走到19

NEXT NODE YOU GO AHEAD
16

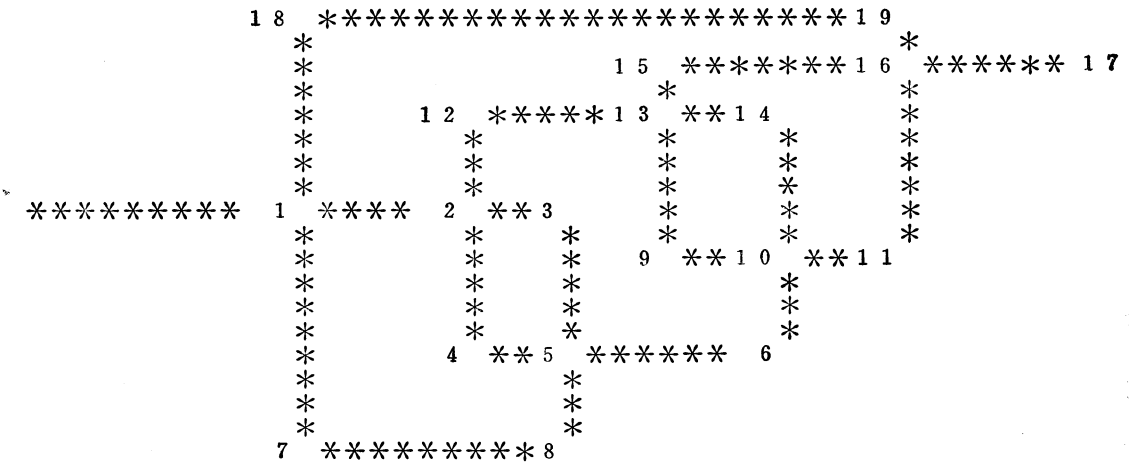


图 24 指定下一步走到16

NEXT NODE YOU GO AHEAD
17

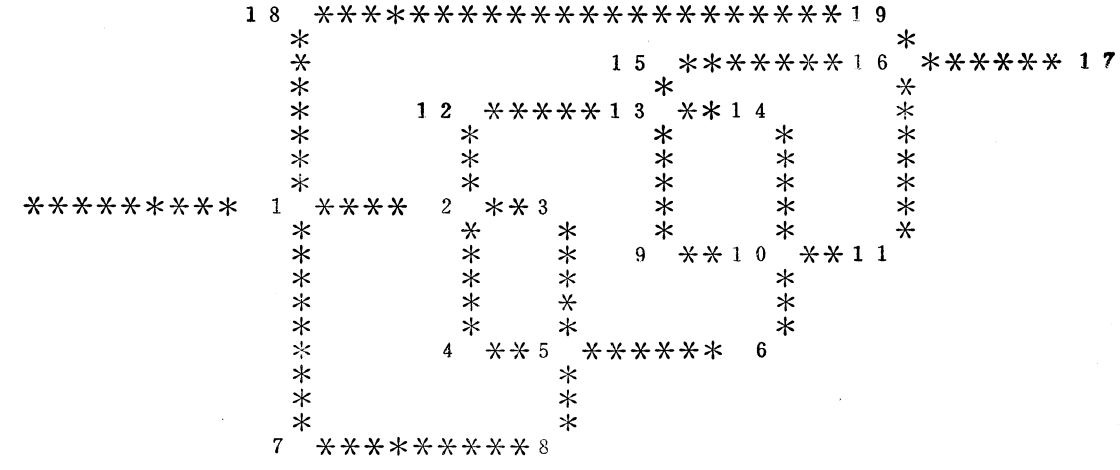


图 25 指定下一步走到17

YOU ARE RIGHT, OTHER GRAPH DO YOU WANT?

机器给出“您走对了，还想走其它图吗？”

（三）思 考 题

1）这类问题的解不是唯一的，你能找出几种可能解法?你能证明共有多少种走法吗?

2）在本书的第106题“如何画一笔画”中，给出了怎样由计算机自行确定走法的算法和程序。可把这个图送入“如何画一笔画”程序中，让它帮你找找看。

3）本程序可以扩充成多种图形可供你选择使用，我们是为了解决游览该名胜古迹十九大处都能游览到，而又不走重复路所给出的游览图形。若有几个图形，应做如何改动?进行程序设计。

十七　条条大路都能走

108　任意六人必有三人两两相识

老师为了鼓励学生练习上机操作，让学生自选算题上机练习。学生徐彼悦选道趣味题如下。请你批改这作业。

证明从任何 6 个人中，或者可以找到 3 个人，两两互不相识；或者可以找到 3 个人，他们两两相识。

（一）笔算步骤和结果

从 6 人中任选一人称为A，设与A熟悉的人中为 k，分别讨论如下。

1）当 k = 1 或 2 时，除去与A相识的人外，至少还有 3 个人。假如他们两两相识，就得到结论的第二种情况；否则至少有两人不相识。再加上A就归结为第一种情况。

2）当 k ⩾ 3 时，即与A相识的人至少有三人，如果他们两两不相识，就得到第一种情况；否则至少有二人互相认识，这二人再加上A，又归结为第二种情况。

（二）程序设计

1）设计思路

他用离散数学“图论”来做。六个人做为 6 个顶点，如图 108-1 所示，利用邻接矩阵表示，设矩阵 k (6，6)。

随机输入这 6 个点（人），哪三个点（人）两两互相认识便连成实线（如图108-1）成三角形（是无向图），设1、3、5这三个人认识，顶点标志为 1，即 k (1，3) = k (3，1) = 1，k (1，5) = k (5，1) = 1，k (3，5) = (5，3) = 1。

其余三个点（人）2、4、6，互不相识，顶点以 0 表示，没有连线，即k (1，2) = k (2，1) = … = k (5，6) = k(6，5) = 0

则 k (6，6) 的邻接矩阵，便成如图108-2所示

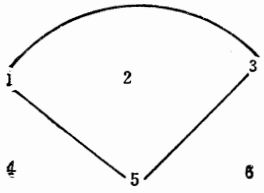


图 108-1

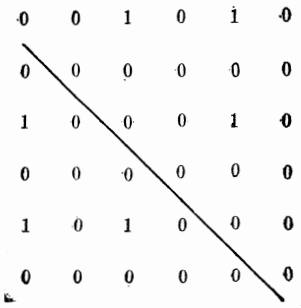


图 108-2

由于矩阵对称，所以程序中只搜寻下三角线（如图108-2）斜线以下这15个元素即可，在这下三角形里，只要找到三个元素 k (3，1) = k (5，1) = k (5，3) = 1，则构成三角形，验证出三个人两两互相认识。

寻找下三角形时，先固定行I，在该行上找遍各列J的元素有1否，有，则记录下标。当该行上找到两个元素为1时，如k（5，1）=1，k（5，3）=1，这时再看k（3，1）=1吗？若成立，则证毕。否则，出错!

程序中除上例子外，又一例子是设1、4、6三人中两两相识，请见运行后打印出正确结论。

2）框图　见图108-3

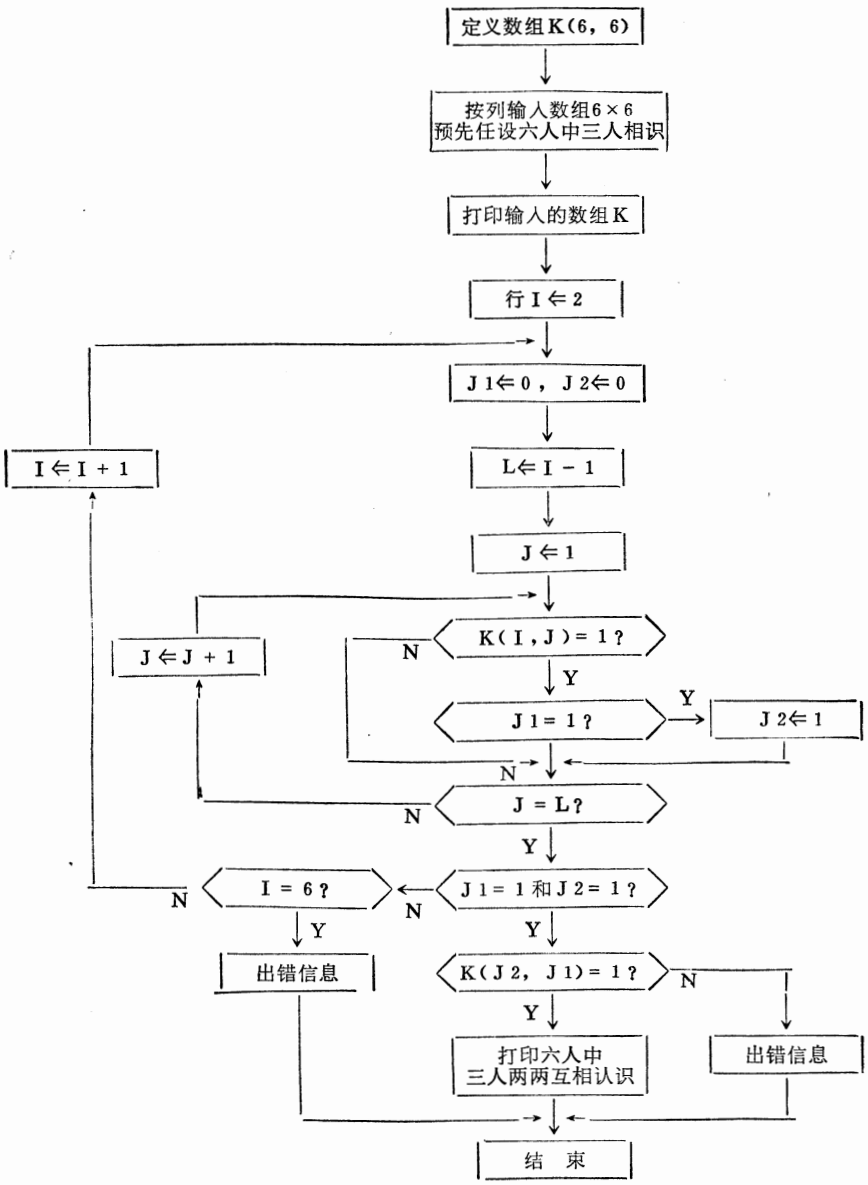


图 108-3

3）源程序及运行结果

```
      DIMENSION  K(6, 6)
      WRITE   (10)  (15H  INPUT  ARRAY  K :)
      READ (11, 10)  ( (K(I, J) , J = 1, 6) , I = 1, 6)
10    FORMAT (6I1)
      WRITE (10, 15)  ( (K(I, J) , J = 1, 6) , I = 1, 6)
15    FORMAT (5X, 6I4)
      DO  20  I = 2, 6
      J1 = 0
      J2 = 0
      L = I - 1
      DO  30  J = 1, L
      IF (K(I, J) . NE. 1) GOTO  30
      IF (J1. NE. 0)   GOTO  50
      J1 = J
      GOTO  30
50    J2 = J
30    CONTINUE
      IF (J1. NE. 0. AND. J2. NE. 0) GOTO  60
      GOTO  20
60    IF (K(I, J1) . NE. 1. AND. K(I, J2) . NE. 1) STOP   'ERR1'
      IF (K(J2, J1) , NE. 1) STOP 'ERR2'
      WRITE (10, 70)
70    FORMAT (2X, 36HTHREE  OF  THE  6  MEN  KNOW  EACH  OTHER)
      STOP
20    CONTINUE
      WRITE (10, 80)
80    FORMAT (2X, 35HNOT  FOND  THREE  MEN  KNOW  EACH  OTHER)
      STOP   'ERR3'
      END
```

```
INPUT  ARRAY  K :
  000101
  000000
  000000
  100001
  000000
  100100
```

输入一组数组
设1、4、6三人两两相识

```
     0     0     0     1     0     1
     0     0     0     0     0     0
     0     0     0     0     0     0
     1     0     0     0     0     1
     0     0     0     0     0     0
     1     0     0     1     0     0
```

将输入的数组打印出

```
THREE OF THE 6 MEN KNOW EACH OTHER          六人中有三人两两互相
                                            认识（运行结果）
STOP
```

```
INPUT ARRAY K:
001010
000000
100010
000000
101000
000000
```

又输入一组数组
设1、3、5三人两两相识

```
0  0  1  0  1  0
0  0  0  0  0  0
1  0  0  0  1  0
0  0  0  0  0  0
1  0  1  0  0  0
0  0  0  0  0  0
```

将输入的数组打印出

```
THREE OF THE 6 MEN KNOW EACH OTHER          六人中有三人两两互相
STOP                                        认识（运行结果）
```

（三）思 考 题

1）请你批阅，并用另一种方法进行程序设计。

2）请将该题编写成一个逻辑函数，计算函数的值，若为TRUN，则命题真；否则，命题假。

109 九个不同国际数学家如此对话

九个数学家在一次国际数学会议上相遇，发现他们中的任意三个人中，至少有两个人可用同一种语言对话，如果每个数学家至多可以说三种语言，证明至少有三个数学家可以用同一种语言对话。

（一）笔算步骤和结果

方法1——用反正法

假设至多只有两个数学家会讲同一种语言。数学家 A_1 至多能与三个数学家有共同语言，因而与其他$9-1-3=5$个数学家没有共同语言，设这五个数学家 A_2，A_3，A_4，A_5，A_6，即A_1与A_2不会讲同一种语言，A_1与A_3不会讲同一种语言，…，A_1与A_5不会讲同一种语言。A_2与A_3必会讲同一种语言 L_1，不然的话，A_1，A_2，A_3 三个人中任意两个都不会讲同一种语言，与假设矛盾。同理

A_2与A_4必会讲同一种语言L_2，

A_2与A_5必会讲同一种语言L_3，

A_2与A_6必会讲同一种语言L_4。

L_1，L_2，L_3，L_4中如果有两种相同，A_2 就会讲四种语言，导致矛盾，所以原设不成立，原设不正确。应该是至少有三个数学家能讲同一种语言。

方法2——用V_1，V_2，…V_9这九个点表示九个数学家，在两个人不能用同一种语言对

话时，才用边来连接相应的两个顶点，这样得到一个简单图G，因为每三个人中至少有两个人可以用同一种语言对话，所以G中每三个点之间至少有两个点不相邻，换句话说，在G中没有三角形。

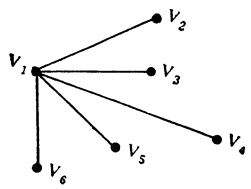


图 109-1

现在我们来证明G中必有一个点V_i的次数不大于4，如果$degV_1 \leqslant 4$，那么V_1就是说的点V_i，如果$degV_i > 4$，那么至少有五个顶点与V_1相邻，不妨设V_2、V_3、V_4、V_5、V_6与它相邻，如图109-1所示。

由于G中没有三角形，所以V_2与V_3、V_4、V_5、V_6均不相邻，从而$degV_2 \leqslant 4$。

上面证明了G中有一个点的次数不大于4，也就是说在这九名数学家中，有一个人至少可以同四个人对话，由于这个人至多会说三种语言，因此至少有两个人与他对话时用的是同一种语言，于是这三个人可以用同一种语言对话。

（二）程 序 设 计

1）设计思路

由算法分析2，利用图论中邻接矩阵来求解，设九个人为九个顶点，二人能对话的不连线；反之，有连线。这样便画成有九个顶点、没有三步循环的任意连通图（如图109-2所示）。图109-2是设1与2、4、6、8有连线，即1和这四个人（2、4、6、8)不能对话，不能对话者有连线。但这四个点（2、4、6、8）中任意三个点间没构成三角连通图，即任意三个人可以有二人互相通话，因为其中有二点没连线。而1、4、6三点中1和4，1和6都有连线，即不能通话，而4和6没连线能通话，即三人中有二人能通话。

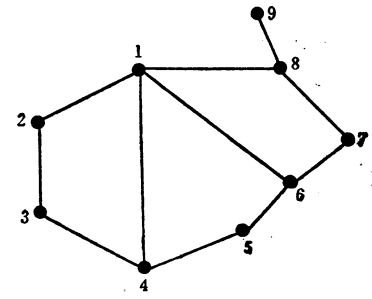


图 109-2

由图109-2所示，二点间有连线用“1”表示，没连线用“0”表示，给出关系矩阵K（9，9）如图109-3所示

0	1	0	1	0	1	0	1	0
1	0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0
1	0	1	0	1	0	0	0	0
0	0	0	1	0	1	0	0	0
1	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	1	0
1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0

图 109-3

编程序来验证题设的正确性。证明任一顶点与其它点连线数目JS≤4便可，即在图109-3矩阵中每行“1”的个数不超过4便对，打印出TRUE（命题真）。否则，打印出FALSE（命题假）。

2）框图 见图109-4

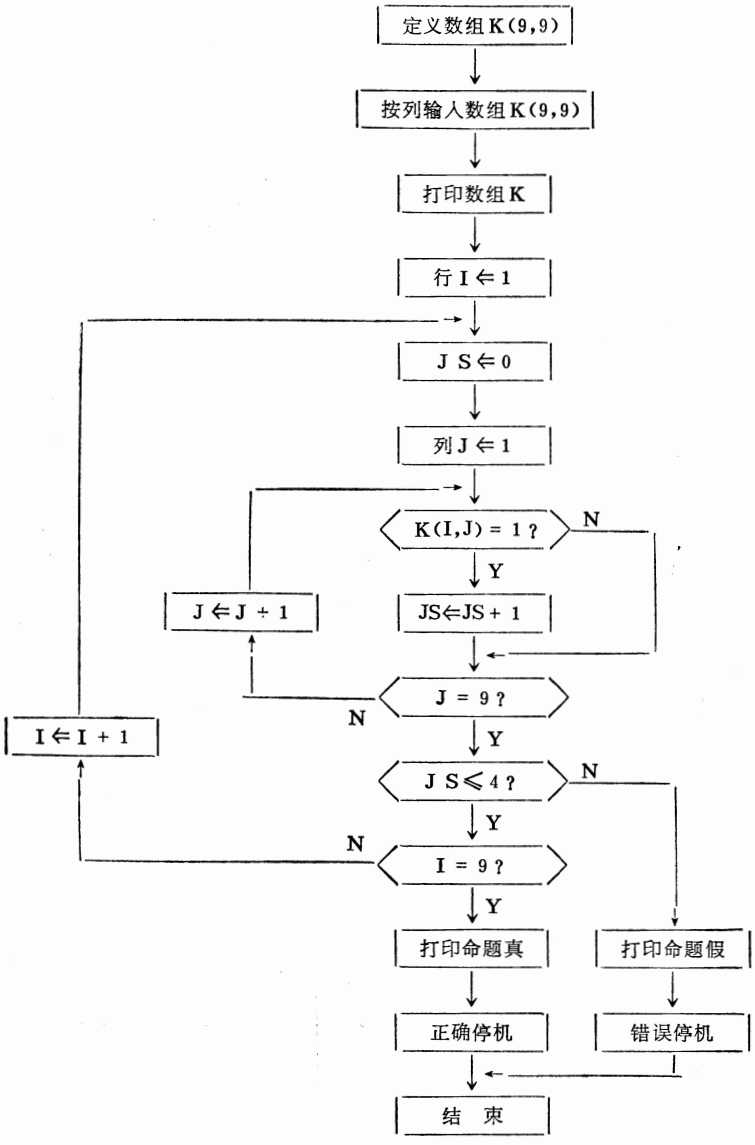


图 109-4

3）源程序及运行结果

```
      DIMENSION  K(9, 9)
      READ (5, 10) K
10    FORMAT (9I1)
      WRITE (6, 20)  ( (K(I, J) , J=1, 9) , I=1, 9)
20    FORMAT (1X, 9I4)
      DO  30  I=1, 9
      JS= 0
      DO  40  J=1, 9
      IF (K(I, J) . NE. 1) GO  TO  40
```

```
      JS=JS+1
40    CONTINUE
      IF (JS. GT. 4) GO TO 60
30    CONTINUE
      WRITE (6, 50)
50    FORMAT (10X, 4HTRUE/)
      STOP'ERR'
60    WRITE (6, 70)
70    FORMAT (10X, 5HFALSE/)
      STOP'ERR'
      END
```

```
0 1 0 1 0 1 0 1 0
1 0 1 0 0 0 0 0 0
0 1 0 1 0 0 0 0 0
1 0 1 0 1 0 0 0 0
0 0 0 1 0 1 0 0 0
1 0 0 0 1 0 1 0 0
0 0 0 0 0 1 0 1 0
1 0 0 0 0 0 1 0 1
0 0 0 0 0 0 0 1 0
TRUE
```

这是输入的图109-3

打印出观察输入对否

(运行后打印结果:TRUE)

(三) 思 考 题

1) 请用另一种方法来证明本题的正确性。

2) 十七位学者，每一位都给其余的人写一封信，信的内容是讨论三个论文题目中的任一个，而且两个人互相通信所讨论的是同一个题目，证明至少有三位学者，他们互相之间通信所讨论的是同一个论文题目。

110 条条大路都能走

如图110-1所示是街区平面图。一个学生从家（A点）走到学校（B点），如果不走远路（不走回头路），即只向东（在图中是向右）或向北（在图中是向上）走的话，有多少种不同的走法？

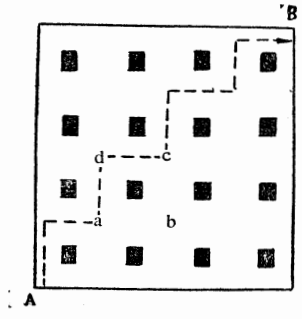


图 110-1

(一) 笔算步骤和结果

先找规律。分析图110-1中a、b、c、d四个点，站在a点看，两条路从家通来：一条自南向北，一条自西向东来的路，所以从家到b点应有3条路线，即东——东——北，东——北——东，北——东——东。d点与b点类似，也有3条路线可通。而c点，可从b点来，也可从d点来，所以从家到c点共有3 + 3 = 6条路线。

由此规律，要找家到街区中任一点的路线数目，只要把

该点西面的那一个点与南面那个点的路线数目相加就行了。用此方法计算，填入图110-2，便迅速找到由家A到学校B的70条不同路线。

有趣的是，如果把街区的图案擦掉。仅留下各街口的数字，便得到一张表格，如图110-3所示。南面第一条路是1、1、1、1、1，南二路是1、2、3、4、5，南三路是1、3、6、10、15。而这些数字还有以下乘法规律：

$$11 = 11$$

$$11 \times 11 = 121$$

$$11 \times 11 \times 11 = 1331$$

$$11 \times 11 \times 11 \times 11 = 14641$$

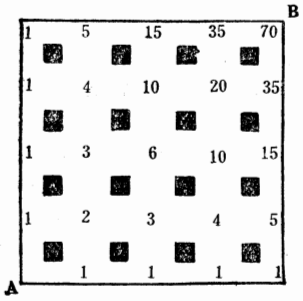


图 110-2

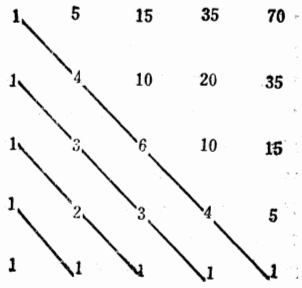


图 110-3

（二）程 序 设 计

1）设计思路

（1）每个交叉点为一站，如图110-4所示1、2、……25个站。

（2）每次只能走一个站，按最近的路线走，不走回头路。故是有向图，如图110-4箭头所示。

例1→2→3→4→5→6→15→16→25，共八站，一种走法。

1→2→9→8→13→14→17→24→25，共八站，另一种走法。

1→10→9→8→13→14→17→24→25，共八站，第三种走法。

……　　……共七十种走法。

定义IA（25，25）据图110-4中方向图赋值，每次只准走一个站，通则用“1”表示，如1→2，2→3。……不通则用“0”表示，如1→3不通，因为跨两站，1→7因跨更多站，所以更不通。

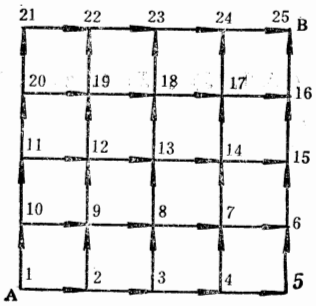


图 110-4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

图 110-5

由此分析得出图110-5，图中写 1 的为通，写 0 表示不通。用 READ 语句由终端按列输入图110-5于数组IA中。

据图论中邻接矩阵的乘积来确定路径的走法，将IA (25, 25)矩阵的乘积置于IB(25, 25) 中，矩阵每乘积一次都放置IB (25, 25) 中。最后数组IB (25, 25) 是数组IA (25, 25) 自乘的 8 次。（因为每条路有八个站）。

2）框图　见图110-6

3）源程序及运行结果

```
      DIMENSION  IA(25, 25) , IB(25, 25)
      READ (11, 10) IA
10    FORMAT (25I1)
      WRITE (10, 20)  ( (IA(I, J) , J = 1, 25) , I = 1, 25)
20    FORMAT (1X, 25I3)
      JS = 0
      DO  30  I = 1, 25
      DO  30  J = 1, 25
      IW = 0
      DO  35  K = 1, 25
35    IW = IW + IA(I, K) *IA(K, J)
```

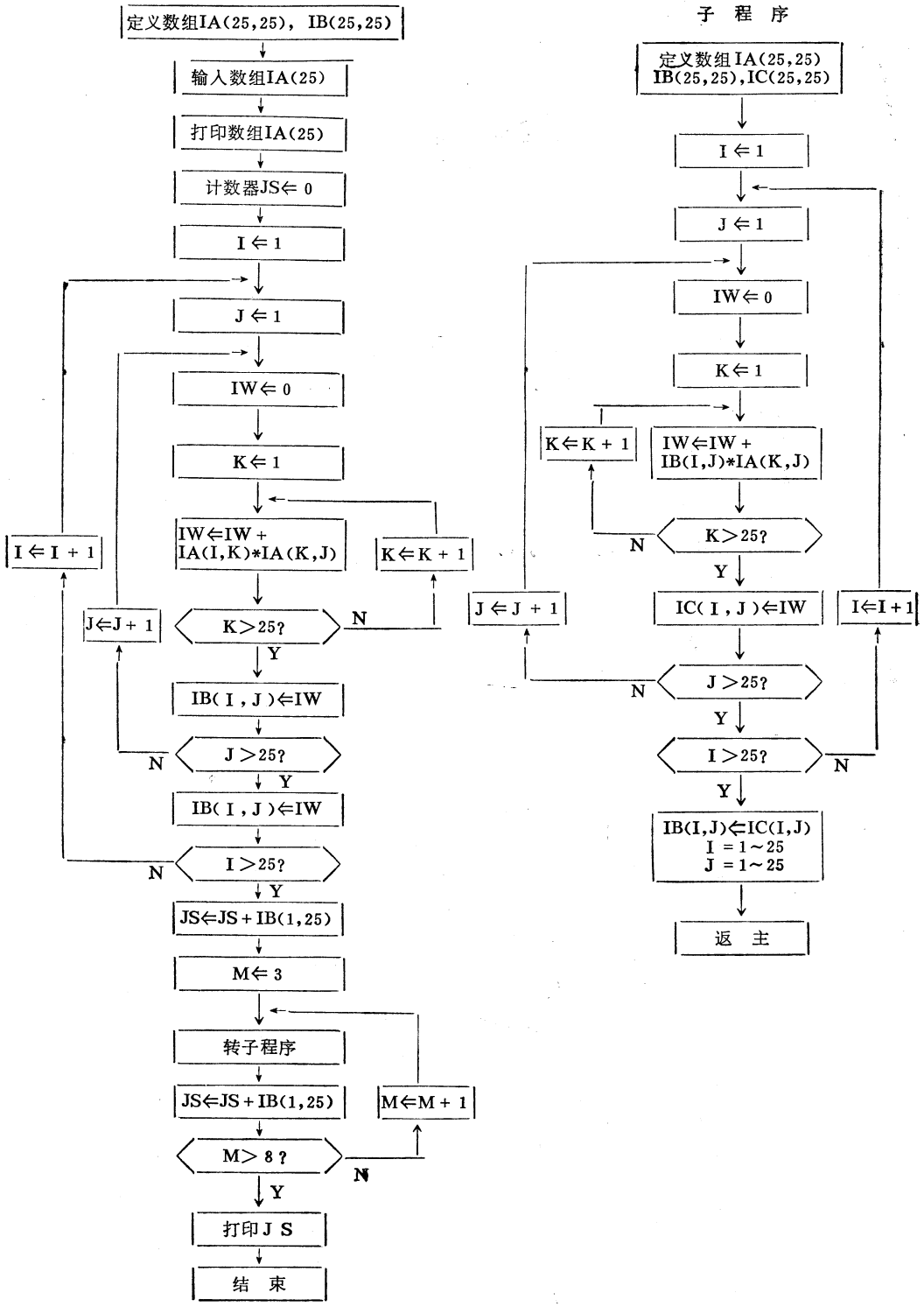

定义数组IA(25,25)，IB(25,25)
输入数组IA(25)
打印数组IA(25)
计数器JS⇐0
I⇐1
J⇐1
IW⇐0
K⇐1
IW⇐IW+IA(I,K)*IA(K,J)
K⇐K+1
K>25?
N
Y
J⇐J+1
IB(I,J)⇐IW
J>25?
I⇐I+1
I>25?
JS⇐JS+IB(1,25)
M⇐3
转子程序
M⇐M+1
M>8?
打印JS
结束
子程序
定义数组IA(25,25)
IB(25,25),IC(25,25)
I⇐1
J⇐1
IW⇐0
K⇐1
K⇐K+1
IW⇐IW+IB(I,J)*IA(K,J)
K>25?
J⇐J+1
IC(I,J)⇐IW
I⇐I+1
J>25?
I>25?
IB(I,J)⇐IC(I,J)
I=1～25
J=1～25
返主

图 110-6

```
30      IB(I, J) =IW          30句IB=IW=IA²
        JS=JS+IB(1, 25)
        DO  39  M=3, 8        控制变量M=3，将得到IA³；M=8将得到IA⁸。
        CALL  CJ (IA, IB)
39      JS=JS+IB (1, 25)      出循环语句39时JS=IA⁸，即8步走到的路径共有JS
                              种走法。
        WRITE (10, 40) JS
40      FORMAT (10X, 3HJS=, I2)
        STOP
        END
        SUBROUTINE  CJ (IA, IB)          子程序用来求矩阵乘积
        DIMENSION  IA(25, 25) , IB(25, 25) , IC(25, 25)
        DO  20  I=1, 25
        DO  20  J=1, 25
        IW=0
        DO  30  K=1, 25
30      IW=IW+IB(I,K)*IA(K, J)
20      IC (I, J) =IW
        DO  50  I=1, 25
        DO  50  J=1, 25
50      IB(I, J) =IC(I, J)
        RETURN
        END          将输入的IA数据打印出，观查输入错否。
```

```
0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

JS = 70（即70种走法，这是运行结果）

（三）思　考　题

请你用另一种算法进行程序设计，将结果打印输出。

十八　茫然大海寻友人住址

111　每行的数是某数的平方

下面图111-1前三行是方格，第四行是已给定的 数 3025，这四行是由 0 ～ 9 这十个不同数字组成，求每行方格的数是几，每行的数是某数的平方数，如第四行$3025=55^2$。

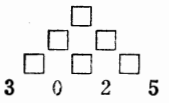


图 111-1

（一）笔算步骤和结果

$$\begin{array}{rcl} 9 & & =3^2 \\ 1\quad 6 & & =4^2 \\ 7\quad 8\quad 4 & & =28^2 \\ 3\quad 0\quad 2\quad 5 & & =55^2 \end{array}$$

（二）程 序 设 计

1）设计思路

由 0 ～ 9 这十个数字堆迭成三角形，最上层是一个数字，最下层的四个数字是已知的 3025，求另外六个数字。定义数组M(10) 置放这十个数字，先将已给的 3025 分别置入M（1）～M（4）。而后利用循环语句由最上层按条件求出M（5），下一层M（6）、M（7），再下一层的M（8）、M（9）、M(10)。每一个数字都要与已知的数字互不相同，即数组M(10) 中的十个数字最后要求是由 0 ～ 9 十个数字充填。将其形象地 打 印 成 三角形的排列。见源程序运行结果。

2）框图　见图111-2

3）源程序及运行结果

```
      INTEGER  M(10)
      M(1) = 3
      M(2) = 0
      M(3) = 2
      M(4) = 5
      DO  10  I1=1, 3
      M(5) =I1*I1
      DO  20  I=1, 4
      IF(M(5) . EQ. M(I) )  GOTO  10
20    CONTINUE
      DO  30  I2=4, 9
      N2=I2*I2
      M(6) =N2/10
      M(7) =N2-M(6) *10
```

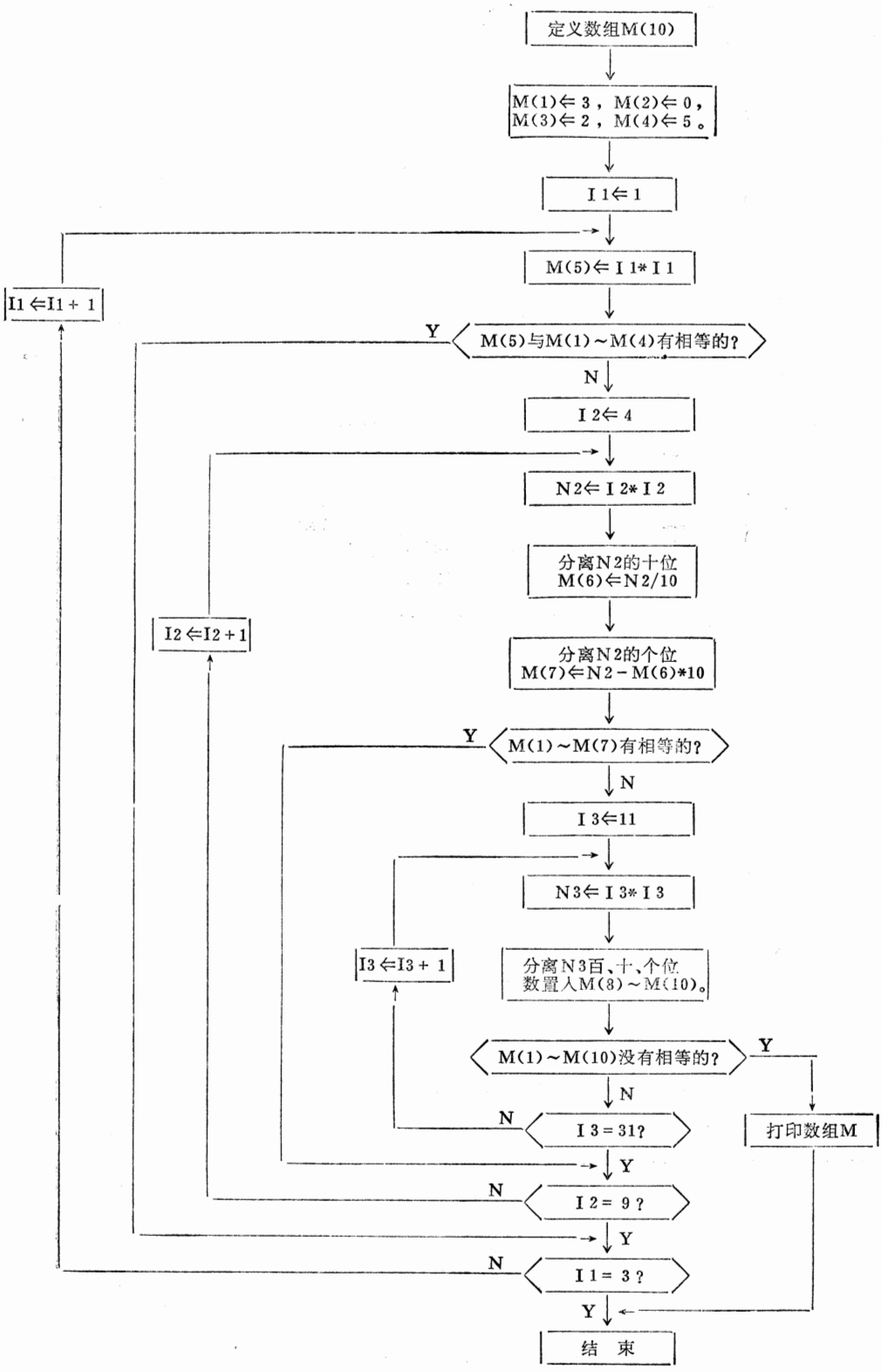

定义数组M(10)
M(1)⇐3，M(2)⇐0，
M(3)⇐2，M(4)⇐5。
I1⇐1
M(5)⇐I1*I1
M(5)与M(1)～M(4)有相等的?
Y
N
I1⇐I1+1
I2⇐4
N2⇐I2*I2
分离N2的十位
M(6)⇐N2/10
I2⇐I2+1
分离N2的个位
M(7)⇐N2-M(6)*10
M(1)～M(7)有相等的?
Y
N
I3⇐11
N3⇐I3*I3
I3⇐I3+1
分离N3百、十、个位
数置入M(8)～M(10)。
M(1)～M(10)没有相等的?
Y
N
I3=31?
N
Y
打印数组M
I2=9?
N
Y
I1=3?
N
Y
结 束

图 111-2

```
      IF (M(6) . EQ. M(7) )  GOTO  30
      DO  40  I=1, 5
      IF ( (M(6) . EQ. M(I) ) . OR.  (M(7) . EQ. M(I) ))  GOTO  30
40    CONTINUE
      DO  50  I3=11, 31
      N3=I3*I3
      M(8) =N3/100
      M(9) =MOD(N3, 100) /10
      M(10) =MOD(N3, 10)
      IF ( (M(8) . EQ. M(9)). OR.  (M(8) . EQ. M(10) ) . OR.  (M(9).
     EQ. M(10)))  GOTO  50
      DO  60  I=1, 7
      IF ( (M(8) . EQ. M(I)) . OR.  (M(9) . EQ.M(I)). OR. (M(10). EQ.
     M(I)))  GOTO  50
60    CONTINUE
      WRITE (5, 70)  (M(I) , I=5, 10) ,  (M(J) , J=1, 4)
      WRITE (2, 70)  (M(I) , I=5, 10) ,  (M(J) , J=1, 4)
70    FORMAT (20X, I6/, 23X, 2I2/, 22X, 3I2/, 21X, 4I2/)
      GOTO  10
50    CONTINUE
30    CONTINUE
10    CONTINUE
      STOP
      END
```

源程序运行后打印结果

```
      9
    1   6
  7   8   4
3   0   2   5
```

（三）思 考 题

仍是主题的由上至下的四行数一共十个数字，由 0～9 这十个不重复的数字组成。每行的数仍是某个数的平方。不过没给出任何一行的具体数字，由读者编程序去寻找，观看共有几种排列的可能，并将其打印出如下三角形式：

□

□ □

□ □ □

□ □ □ □

112 杨 辉 三 角 形

给出如下四行数字，请你继续找出规律逐层往下写至第 n 行。每一数字是上一行的相应位置上的左、右两个数字相加之和，若上行无数字以 0 代之。

```
                1
              1   1
            1   2   1
          1   3   3   1
        ……
```

（一）笔算步骤和结果

这是我国宋朝数学家杨辉在公元1261年著书《详解九章算法》里画的一张图，这便是著名的杨辉三角形。

我们先来考察一下杨辉三角里面数字排列的规律。一般的杨辉三角是如下的图形：

```
                1
              1   1
            1   2   1
          1   3   3   1
        1   4   6   4   1
      1   5   10  10  5   1
    1   6   15  20  15  6   1
    ……      ……      ……
```

第n行　　$1, C_{n-1}^{1}, \cdots, C_{n-1}^{r-1}, \cdots, C_{n-1}^{n-2}, 1$

第n+1行　　$1, C_n^1, C_n^2, \cdots, C_n^r, \cdots, C_n^{n-2}, C_n^{n-1}, 1$

……　　……　　……

这里，记号C_n^r是用来表示下面的数：

$$C_n^r = \frac{n(n-1)\cdots(n-r+1)}{r!} = \frac{n!}{r!(n-r)!},$$

而记号n!(同样r!和 (n−r)!)，我们知道它是代表从1到n的连乘积$n\cdot(n-1)\cdot(n-2)\cdots 3\cdot 2\cdot 1$，称为n的阶乘。$C_n^r$也就是表示从n件东西中取出r件东西的组合数。

和杨辉三角有最直接联系的是二项式定理。学过代数的都知道：

$$(a+b)^1 = a+b,$$

$$(a+b)^2 = a^2+2ab+b^2,$$

$$(a+b)^3 = a^3+3a^2b+3ab^2+b^3,$$

$$(a+b)^4 = a^4+4a^3b+6a^2b^2+4ab^3+b^4,$$

…… ……　　……。

这里，$(a+b)^3$展开后的系数1，3，3，1就是杨辉三角第四行的数字。不难算出$(a+b)^6$的系数是1，6，15，20，15，6，1，即杨辉三角第七行的数字。所以杨辉三角可以看做是二项式的乘方经过分离系数法后列出的表。实际上，我们可以证明这样的事实：一般地说，$(a+b)^n$的展开式的系数就是杨辉三角中第n+1行的数字

$$1, C_n^1, C_n^2, \cdots, C_n^r, \cdots, C_n^{n-1}, 1$$

这便是二项式系数。

（二）程序设计

1）设计思路

由笔算过程已分析出算法，可利用循环控制变量算出。我们除掉最上层仅有一个数字

的1，其下算做第1行，依次可递推至任意N行。可随机输入要算的N行数字置入数组M（15）中，而后赋给数组IM（90）。输出数组IM（90）时，形象地打印出三角形的层次和各行各位的数字，并打印汉语拼音的表头："打印杨辉三角形"。本程序设计最多为12行。

2）框图　见图112-1

3）源程序及运行结果

```
      INTEGER  M(15) , IM(90)
      WRITE (5, 10)
10    FORMAT (10X, 28HDA  YIN  YANG-HUI  SAN  JIAO  XING/5X, 'N=')
      READ (1, 20) N
20    FORMAT (I2)
      JJ = 0
      DO  44  I = 1, N
      DO  30  L = 1, N
      M(L) = 0
30    CONTINUE
      K = 1
      M(1) = 1
      L = I + 1
      DO  50  J = 1, L
      K = K * (I - J + 1) /J
      M(J + 1) = K
50    CONTINUE
      DO  40  IJ = 1, L
      JI = JJ + IJ
      IM(JI) = M(IJ)
40    CONTINUE
      JJ = JJ + L
44    CONTINUE
      WRITE (5, 60) (IM(I) , I = 1, JJ)
60    FORMAT (33X, 2I6/, 30X, 3I6/, 28X, 4I6/, 25X, 5I6/, 22X, 6I6/,
     1  19X, 7I6/, 16X, 8I6/, 13X, 9I6/, 10X, 10I6/, 7X, 11I6/, 4X, 12I6/,
     2  1X, 13I6/)
      STOP
      END
```

```
DA  YIN  YANG-HUI  SAN  JIAO  XING      (打印杨辉三角形)
N = 12
```

```
                    1    1
                  1    2    1
                1    3    3    1
              1    4    6    4    1
            1    5   10   10    5    1
          1    6   15   20   15    6    1
        1    7   21   35   35   21    7    1
      1    8   28   56   70   56   28    8    1
    1    9   36   84  126  126   84   36    9    1
  1   10   45  120  210  252  210  120   45   10    1
1   11   55  165  330  462  462  330  165   55   11    1
1   12   66  220  495  792  924  792  495  220   66   12    1
```

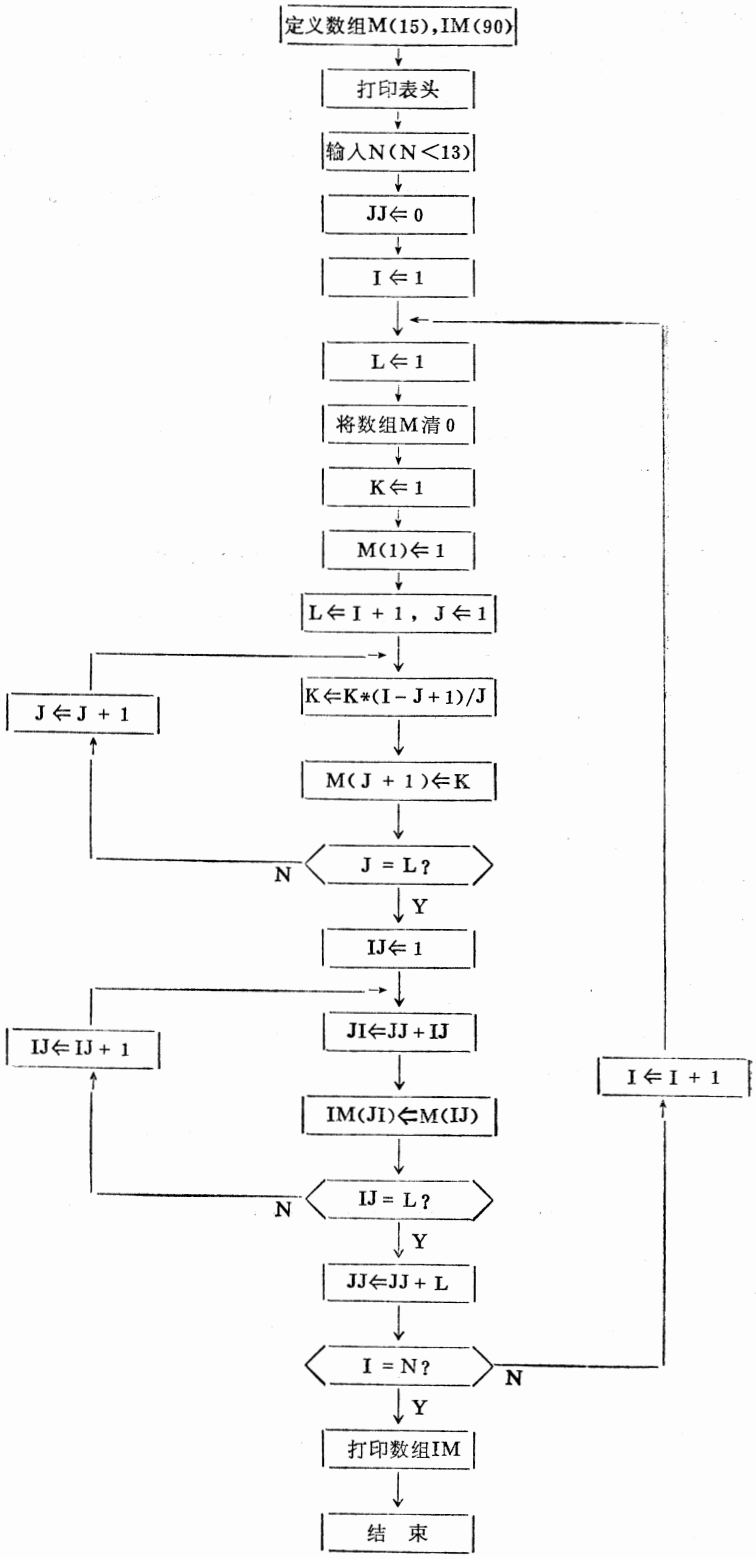

定义数组M(15),IM(90)
打印表头
输入N(N＜13)
JJ⇐0
I⇐1
L⇐1
将数组M清0
K⇐1
M(1)⇐1
L⇐I＋1，J⇐1
K⇐K*(I－J＋1)/J
J⇐J＋1
M(J＋1)⇐K
J＝L？
N
Y
IJ⇐1
JI⇐JJ＋IJ
IJ⇐IJ＋1
IM(JI)⇐M(IJ)
IJ＝L？
N
Y
JJ⇐JJ＋L
I⇐I＋1
I＝N？
N
Y
打印数组IM
结　束

图 112-1

（三）思 考 题

三项式系数——下图112-2中表示由下述两个条件确定的无限三角数阵的一部分。

（1）边界条件，每一条水平线或“底”即每一行都以0，1开始并且以1，0结束，图中省略0，（将第二行1，1，1算做第一个底，则第n个底包括2n+3个数。所以，边界条件没有确定第n个底的其余2n-1个数，n=1，2，3，…）。

```
                1
            1   1   1
        1   2   3   2   1
    1   3   6   7   6   3   1
  1   4  10  16  19  16  10   4   1
1   5  15  30  45  51  45  30  15   5   1
……
```

图 112-2

（2）递归公式，第（n+1）个底中没有被条件（1）确定的任一数都可以作为第n个底上的三个数（其正上和左、右角的邻数）的和计算出来(例如，45=10+16+19)。

要求编写一个程序作到以下三点：

1）打印至第n行。

2）验证每一行各个数字之和是3的该行数的方次，如：

$1+1+1=3=3^1$　　第1行

$1+2+3+2+1=9=3^2$　　第2行

$1+3+6+7+6+3+1=27=3^3$　　第3行

……

3）验证每一行都有如下规律：

$1-1+1=1$

$1-2+3-2+1=1$

$1-3+6-7+6-3+1=1$

……

113 用012三个数字排成序列

要求用0、1、2组成任意长度的非空序列，序列中没有非空的、按元素相等的相邻子序列，并应按字典顺序排列。

（一）算法分析和结论

序列的生成规律。显然

```
0
0 1
0 1 0
```

是符合要求的序列。继续向下做有

```
*0 1 0 1
 0 1 0 2
 0 1 0 2 0
```

0 1 0 2 0 1

0 1 0 2 0 1 0

＊0 1 0 2 0 1 0 1

＊0 1 0 2 0 1 0 2

＊0 1 0 2 0 1 1

0 1 0 2 0 1 2

上述带＊者，显然是不合乎要求的顺序排列。由此可得一个规律，若有一个序列×××…×××，要找出其后面的序列，则先在其后面加上一个0，即×××…×××0，进行比较。若符合要求，则×××…×××0即是。否则把0换成1，重新比较。若0，1，2都不能满足要求，则把其前面的第一位数字由0换成1，由1换成2，并进行比较，若满足要求，即为所求序列，继续向下求。否则，把前面的第二位数字由0换成1，由1换成2，如此求下去，可得所求的任意长度N的序列。

如何比较相邻的子序列？对长度为M的序列，$a_1, a_2, \cdots a_M$，有$\left[\frac{M}{2}\right]$个相邻子序列，要逐一比较。如a_M与a_{M-1}比较，a_{M-1}，a_M与a_{M-3}，a_{M-2}进行比较。最后二个长度为$\left[\frac{M}{2}\right]$的子序列，要根据M是奇数还是偶数，$a_{[\frac{M}{2}]}$，…，a_M要与$a_1, \cdots, a_{[\frac{M}{2}]-1}$或$a_2$，…，$a_{[\frac{M}{2}]-1}$进行比较，便可求得合乎要求的序列。

（二）程序设计

1）设计思路

本程序设计最大的N为100，设一个100×101的数组A存放结果，每一行为一个序列，最后一列存放该行的序列的长度。

由于0，01，010是合乎条件的序列，所以我们由N＝4求起。

每找下一个序列时，首先把前一个序列原封不动地复制下来。然后按上述算法在其后面添一个0，进行比较，长度L⇐L＋1。并进一步比较，若合乎条件，该序列即为所求；否则依次把0换成1，1换成2。若均不符合条件，则L⇐L－1，把其前面的一个数字，由0换成1，或由1换成2，然后再试其后的位数。

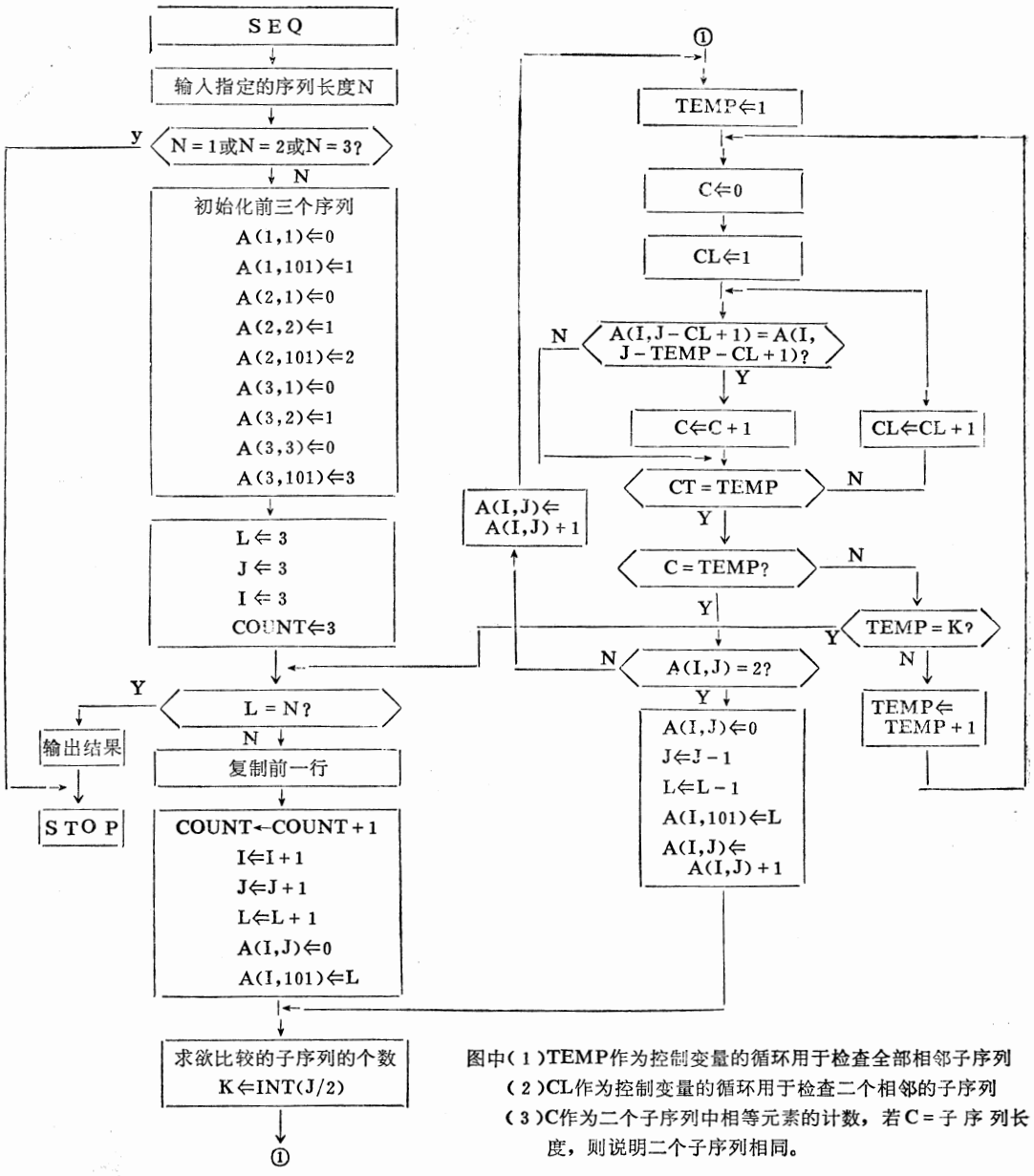
在L＝N时，运行结束，输出结果。

我们在程序中举例指定长度分别为N＝25及N＝60，请见程序运行后打印结果。

2）框图　见图113

3）源程序运行后及其结果

```
C***  PROGRAM  NAME : SEQ
C***  THIS  PROGRAM  CREATS  A  SEQUECE  CONSISTED  OF  0, 1, 2
C***  WHICH  ARE  ORDERDIN  DICTIONARY  SEQUENCE
C***
      INTEGER  N, I, J, TEMP, K, CL, C, COUNT
      INTEGER  A (100, 101)
      DO  5  J=1, 101
      DO  5  I=1, 100
5     A (I, J) =0
```



图中(1)TEMP作为控制变量的循环用于检查全部相邻子序列
(2)CL作为控制变量的循环用于检查二个相邻的子序列
(3)C作为二个子序列中相等元素的计数，若C=子序列长度，则说明二个子序列相同。

图 113

```
      OPEN (4, FILE='SSS', ACCESS='W')
      WRITE (4, 8)
8     FORMAT (5X, 'HOW  LONG  THE  SEQUENCE  IS? ',///)
      READ  (1, 10) N
10    FORMAT (I3)
      IF (N. EQ. 1. OR. N. EQ. 2. OR. N. EQ. 3) GO  TO  150
      A(1, 1) =0
      A(1, 101) =1
      A(2, 1) =0
```

```
      A(2, 2) =1
      A(2, 101) =2
      A(3, 1) =0
      A(3, 2) =1
      A(3, 3) =0
      A(3, 101) =3
      L= 3
      J= 3
      I=3
      COUNT=3
15    IF (L. EQ. N) GO TO 60
      DO 20 TEMP=1, L
20    A(I+1, TEMP) =A(I, TEMP)
      COUNT=COUNT+1
      I=I+1
      J=J+1
      L=L+1
      A(I, J) =0
      A(I, 101) =L
28    K=INT (J/2)
C***  TEST THE ALL SUBSEQUENCES
30    DO 100 TEMP=1, K
      C = 0
C***  TEST A CONJUNCTING SUBSEQUENCE
      DO 50 CL=1, TEMP
      IF (A(I, J-CL+1) . NE. A(I, J-TEMP-CL+1) ) GO TO 50
      C = C + 1
50    CONTINUE
      IF (C. NE. TEMP) GO TO 1000
      IF (A(I, J) . EQ. 2) GO TO 55
C***  CHANG THE VALUE AND TEST AGAIN
      A(I, J) =A(I, J) +1
      GO TO 30
100   CONTINUE
      GO TO 15
C***  RETURN ONE BIT
55    A(I, J) =0
      J=J-1
      L=L-1
      A(I, 101) =L
      A(I, J) =A(I, J) +1
      GO TO 28
60    WRITE (4, 65)
```

```
 65       FORMAT (15X, 'THE RESULTS ARE:', /)
          DO  80  I=1, COUNT
          L=A(I, 101)
 80       WRITE (4, 70)  (A(I, J) , J=1, L)
 70       FORMAT (5X, 100I2, /)
 150      CLOSE (-1)
          STOP
          END
```

HOW LONG THE SEQUENCE IS?

键盘输入：60

THE RESULTS ARE:　　(结果：)

```
0 1
0 1 0
0 1 0 2
0 1 0 2 0
0 1 0 2 0 1
0 1 0 2 0 1 0
0 1 0 2 0 1 2 0
0 1 0 2 0 1 2 0 2
0 1 0 2 0 1 2 0 2 1
0 1 0 2 0 1 2 0 2 1 0
0 1 0 2 0 1 2 0 2 1 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 2 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 2 1 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 2 1 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2
```

```
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 2 0 1 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0 1 2 0 2 1 0
```

这是指定长度为60时的输出

4）附注——若选用一种结果，则应取长度为60的，可以说明问题。

HOW LONG THE SEQUENCE IS?

键盘输入：25

THE RESULTS ARE:　　(结果：)

```
0
0 1
0 1 0
0 1 0 2
0 1 0 2 0
0 1 0 2 0 1
0 1 0 2 0 1 0
0 1 0 2 0 1 2
0 1 0 2 0 1 2 0
0 1 0 2 0 1 2 0 2
0 1 0 2 0 1 2 0 2 1
0 1 0 2 0 1 2 0 2 1 0
0 1 0 2 0 1 2 0 2 1 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0
0 1 0 2 0 1 2 0 2 1 0 1 2 0 1 0 2 0 1 2 0 2 1 0 1
```

这是指定长度为25时的输出

（三）思 考 题

从三个元素（例如1，2，3）组成的字符表中选取字符，生成一个有N个字符组成的序列，使得其中没有两个相邻的子序列相等。例如，长度N = 5的字符序列“12321”是合格的生成序列，而“12323”和“12123”都是不合格的。试编写一个程序，生成所有满足上述条件的N个字符组成的序列。

114 加拿大第七届数学竞赛题

15个席位同等地围绕着圆桌安排，席上有15个客人的名片，客人们没有注意这些名片，直到他们坐下来，才发觉没有一个人坐在他自己的名片前面，证明可以转动圆桌，使得至少有两个客人同时对号入座。

（一）算法分析和结论

对于每个客人，恰好有一种转动圆桌使他对上自己的名片，因为有15个客人，只有14种有效的转动，所以根据“抽屉原则”，必有某种转动至少可容许两个客人对上号。

（二）程 序 设 计

1）设计思路

（1）用数组A（15）表示15个座位号（相当桌上的名片），冲入相应数字，如A（1）⇐1，…，A（15）⇐15。

（2）用数组B（15）表示15人没有一个人按座位坐下，即A（i）≠B（i）其中i = 1～15，数组B（15）随机冲数，但不得有B（i）= A（i）。

（3）人已坐下就不动了，即数组B（15）不动，转动桌子，设按逆时针转，使数组A（15）里的数每转动一次，原数字减1。如原A（2）= 2表示2号位里是2（第2个人名片），转一次后A（2）⇐A（2）- 1⇐2 - 1⇐1，即第1个人的名片转到A(2)位置。当转到A（2）= A（2）- 1 = 0时，便令A（2）⇐15继续再转，直到转到14次时为止。

（4）每转动一次，要看是否有两个及两个以上的人能对号入座。即比较B（i）是否等于A（i），若等于，则记数JS⇐JS + 1，最后判该循环（每次转动一个位置），是否JS = 2（二人对上号），有则打印出，再转动；没有，仍再转动寻找，直到转动14次为止。寻找由1～14次转动，有几次转动使2人及2人以上对上号。

2）框图 见图114

3）源程序及运行结果

```
      DIMENSION  A(15) , B(15)
      INTEGER  A, B
      DO  10  I = 1, 15
10    A(I) = I                    } 名片安放位置
      DO  20  I = 1, 7
      J = 16 - I
      B(J) = I
20    B(I) = J                    } 就座时没有人对上自己名片
      B(8) = 15
      B(1) = 8
```

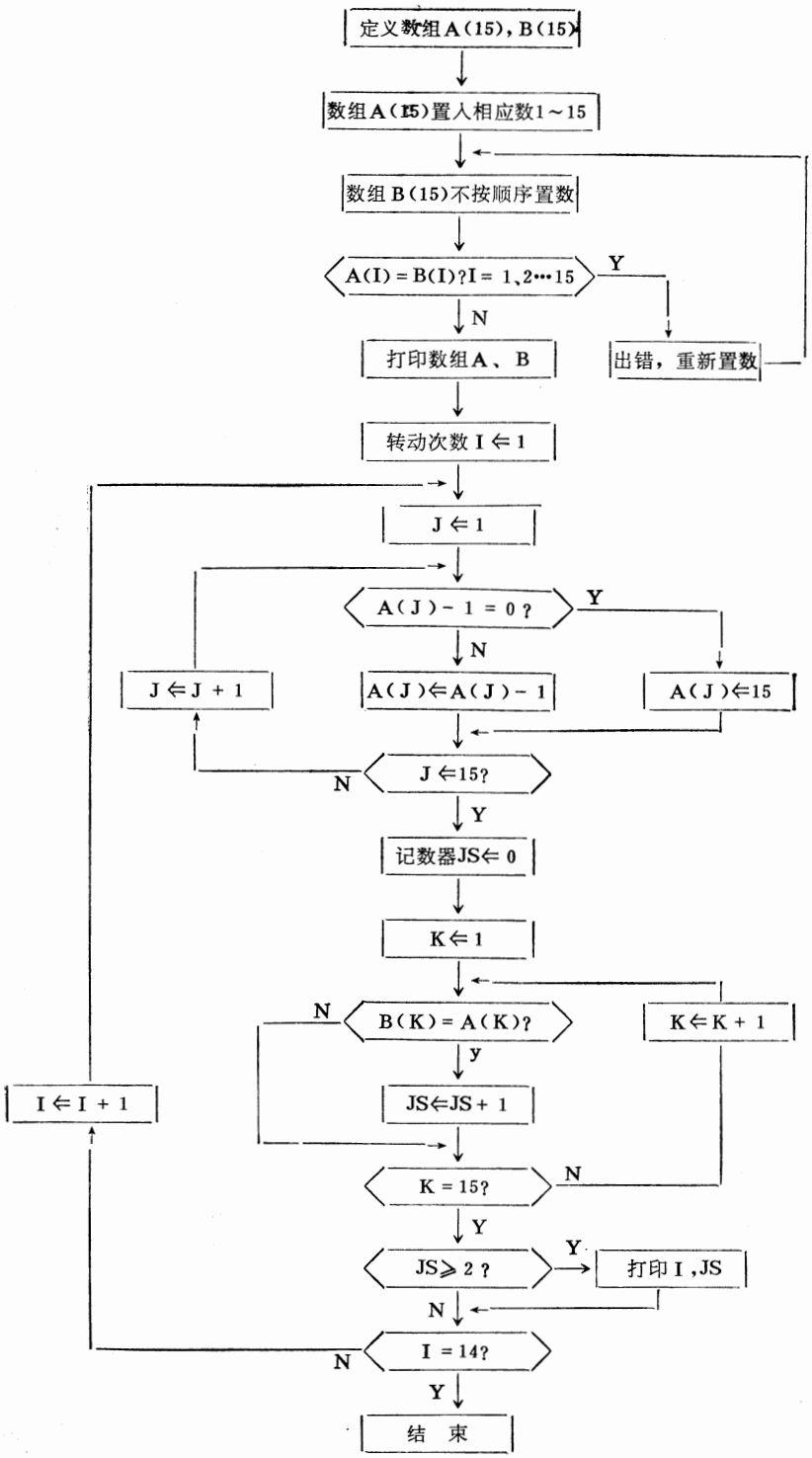

定义数组A(15)，B(15)
数组A(15)置入相应数1～15
数组B(15)不按顺序置数
A(I)=B(I)?I=1,2…15
Y
N
打印数组A、B
出错，重新置数
转动次数I⇐1
J⇐1
A(J)-1=0?
Y
N
J⇐J+1
A(J)⇐A(J)-1
A(J)⇐15
N
J⇐15?
Y
记数器JS⇐0
K⇐1
N
B(K)=A(K)?
K⇐K+1
y
I⇐I+1
JS⇐JS+1
K=15?
N
Y
Y
JS≥2?
打印I,JS
N
N
I=14?
Y
结 束

图 114

```
      DO  30  I=1, 15                                  检查入座情况
      IF (A(I). EQ. B(I) ) STOP'ERR'
30    CONTINUE
      WRITE (6, 35)  (A(MN) , MN=1, 15) ,              打印没对号入座
     (B(NJ) , NJ=1, 15)
35    FORMAT (2X, 6HA (15) =, 15I4/2X, 6HB
     (15) =, 15I4/)
      DO  100  I=1, 14                                 该循环是转动14次
      DO  40  J=1, 15                                  每转动一次，
      IF (A(J) -1. EQ. 0) GO  TO  45                   名片变动位置
      A(J) =A(J) -1
      GO  TO  40
45    A(J) =15
40    CONTINUE
      JS=0
      DO  50  K=1, 15                                  每转动一次检查有几
      IF (B(K) . NE. A(K) ) GO  TO  50                 人对上自己的名片
      JS=JS+1
50    CONTINUE
      IF (JS. GE. 2) GO  TO  60                        有二人对上名片便打印。
      GO  TO  100                                      否则，继续转动。
60    WRITE (6, 70) I
70    FORMAT (2X, 5HTURN , I2, 7H  TIMES:, 13H2  MEN  JOIN  UP/)
100   CONTINUE
      STOP
      END
```

A (15) =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	名片安放的位置入座时没人对上自己的名片转动第8次时，有二人对上名片，打印如下。
B (15) =	8	14	13	12	11	10	9	15	7	6	5	4	3	2	1	

```
TURN  8  TIMES:2  MEN  JOIN  UP
```

（三）思 考 题

举出一种入席顺序的例子，使这15人中恰好有一个客人对号入座。而转动圆桌并不能使更多的客人对上号。编写程序进行验证。

115 找座标上的点

在座标平面的第一象限中，把坐标都是整数的点按以下方法编号：（0，0）点第1号，（1，0）点第2号，（1，1）点第3号，（0，1）点第4号，（0，2）点第5号，（1,2）点第6号，（2,2）点第7号，（2，1）点第8号，（2，0）点第9号…。

按图115-1中箭头顺序，问第2000号的点的坐标是多少？

（一）笔算步骤和结果

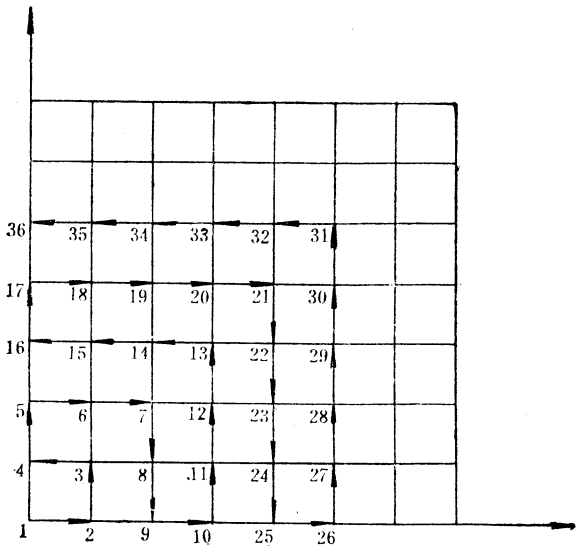


图 115-1

满足条件 $0 \leqslant X \leqslant R$，$0 \leqslant y \leqslant R$ 的座标为整数的点（X，y）一共有 $(R+1)^2$ 个。考虑满足不等式 $(R+1)^2 < 2000$ 的最大整数R，R = 43。所以编号为2000的点的纵座标为44或横座标为44。因44是偶数，所以应该从(0，44)往右数。因 $2000 - 44^2 = 64$，所以第2000号点的横座标是44，并且它的纵座标为 $44 - [64 - 45] = 25$。因此编号2000的点的座标是：(44，25)。

（二）程序设计

1）设计思路

FORTRAN 语言没有零下标的数组，这也无妨，将每点的行列座标值加1，便避开了零。当找出2000号的下标时，将行列座标值各减1，便是题意要求。如点1（0，0）改为（1，1），点2（1，0）改为（2，1），点3（1，1）改为（2，2），…。

从第5点开始找下标值。5点～9点的行列坐标值是1～3，11点～16点的行列座标值是1～4，17点～25点的行列座标值是1～5，…。若将上述每一分段各点的座标列成表（省略），便可发现各列与前后列首尾相接时，坐标变化的规律（在此省略，读者要列表）。如5点～9点的座标由（1，3）到（3，1），称为 I = 3；10点～16点的坐标由（4，1）到（1，4）称为 I = 4；17点～25点的坐标由(1，5)到（1，5）；26点～36点坐标由（6，1）到（1，6），称为 I = 6；…。由此列表发现 I 值变化时，点的座标首尾相接时变化的规律：

I 由奇数变为偶数时，点的行值加1，列值不变，如点9（3，1）到点10（4，1）。当 I 由偶数变为奇数时，点行值不变，列值加1，如点16（1，4）到点17（1，5）。再把上述各段的点，由中间分成上下两部分，观察这两部衔接的变化规律（省略），框图中提到的段和上下部，便指此。

2）框图　见图115-2

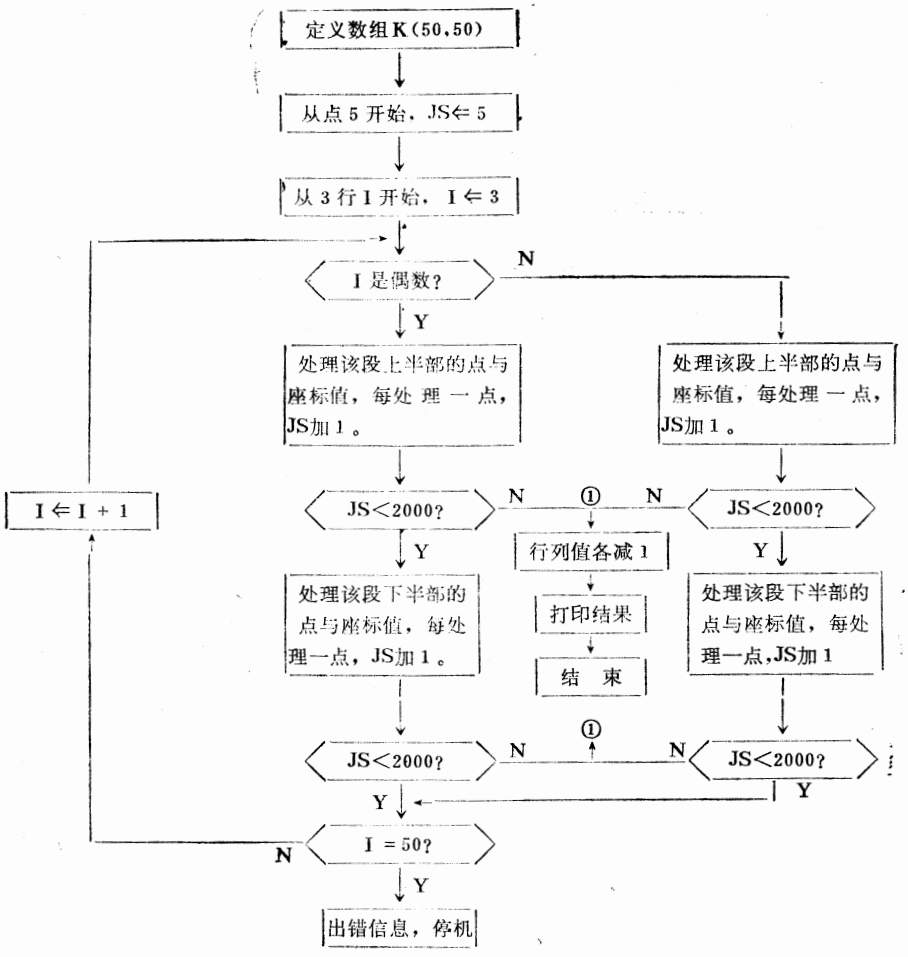


图 115-2

3）源程序及运行结果

```
      DIMENSION  K(50, 50)
      JS=S
      DO  10  I=3, 50
      IF (I. NE. I/2*2) GO  TO  33
      DO  20  J=1, I
      K(I, J) =JS
      IF (JS. LT. 2000) GO  TO  20
      M=I-1
      N=J-1
      GO  TO  100
20    JS=JS+1
      DO  40  L=1, I-1
      I1=I-L
      K(I1, I) =JS
      IF (JS. LT. 2000) GO  TO  40
```

```
          M = I1 - 1
          N = I - 1
          GO  TO  1000
40        JS = JS + 1
          GO  TO  10
33        DO  30  J = 1, I
          K(J, I) = JS
          IF (JS. LT. 2000) GO  TO  30
          M = J - 1
          N = I - 1
          GO  TO  100
30        JS = JS + 1
          DO  50  L = 1, I - 1
          I1 = I - L
          K(I, I1) = JS
          IF(JS. LT. 2000) GO  TO  50
          M = I - 1
          N = I1 - 1
          GO  TO  100
50        JS = JS + 1
10        CONTINUE
          STOP  111
100       WRITE (6, 110) M, N, JS
110       FORMAT (5X, 2HK (, I2, 1H, , I2, 2H) =, I4)
          STOP
          END
```

K (44, 25) = 2000　　程序运行打印结果

（三）思　考　题

79年黑龙江省中学数学竞赛试题

将全部正整数对，按图115-3所示箭头方向依次编号，每一数对的编号写在该数对的前面。例如6（1，3）表示数对（1，3）编号数为6。问数对(100，99)的编号数是多少？

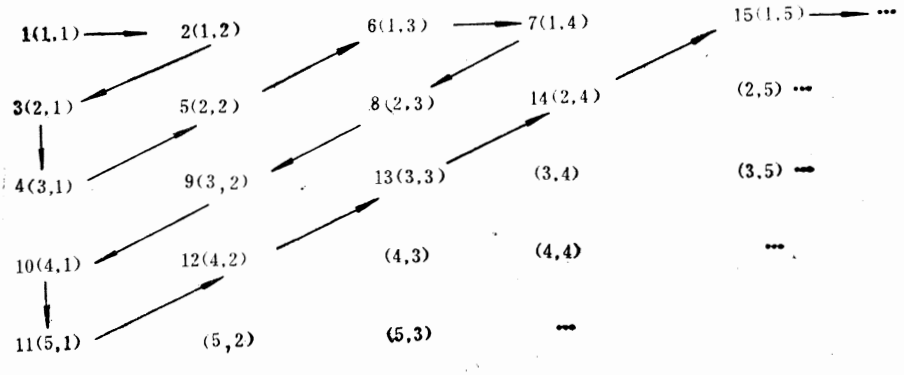


图 115-3

116 茫然大海寻友人住址

一天有位外地陌生人出差北京，公事之余，打听到我们的住址。登门闲谈时，他说："北京近千万人，仅知道你们的名字，到×××等处，才打听到你们的住址。唉，这倒是个趣味题，怎样快速寻找友人…。"

这给了我们以启示，一般姓名用汉语拼音15个字符已够用，问题转化为如何查找字符串。这不仅是道趣味智力题，也是实用题，因为编辑程序中常用到的一种功能是查找字符串，即在已备案的人名中（文本缓冲区里）查找与你指定的字符串相同的字符串的位置。下面选个类似这样的程序来摸拟这种查找过程。

（一）算 法 分 析

1）记住欲查找字符串的长度L；

2）对于M行×N列的文本缓冲区，从第一行的第一列开始顺序查找与字符串中第一个字符相同的字符；

3）若第I行（I＜M）没有则转到第I＋1行的第一列继续查找。

4）若找到I行中J列的字符与字符串的第一个字符相同，则顺序比较其后的L－1个字符，若均与欲找的字符串相同，则记住该位置，此为已找到的字符串。若L－1个字符串中有与指定的字符串不同者，则执行J⇐J＋1，重新查找。

（二）程 序 设 计

1）设计思路

由于是模拟程序，所以与实际的编辑程序有所不同。

首先设文本缓冲区的大小，我们规定为20×70，基本上是显示器一帧的大小。程序中用40×70的二维数组TEXT表示。当找到所需的字符串后，要在该字符串下用光标指示出来，程序中用"∧"表示，故用40行。

读入用户指定的字符串放在STRING中，字符串最大长度为15。输入字符串的结束标志为"∧"。

程序中设有三个循环，一个循环检查I行，一个循环检查I行中各列，另一个循环用于检查字符串。输入原文件是选用国外的，不是针对姓名编写的，而是查找特定的字符串。

2）框图　见图116

3）源程序及运行结果

```
C***  PROGRAM  NAME:  GET  CHAR
C***  THIS  PROGRAM  SEARCH  FOR  THE  STRING  YOU  WANT  TO
      LOOK  FOR
C***  IT  IS  USEFUL  WAY  FOR  EDITER  AND  WRITER
C***
      INTEGER  LENGTH, TEXTI, TEXTJ, I, J, S, II, JJ
      CHARACTER  TEXT (40, 70) , STRING (15) , C
      OPEN (4, FILE=' (P-T:WZH) XEX:                输出文件
      SYMB', ACCESS='W')
      OPEN (5, FILE='TIC:SYMB', ACCESS='R')         源文件
C***  CLEAR  TEXT  BUFFER  NULL
```

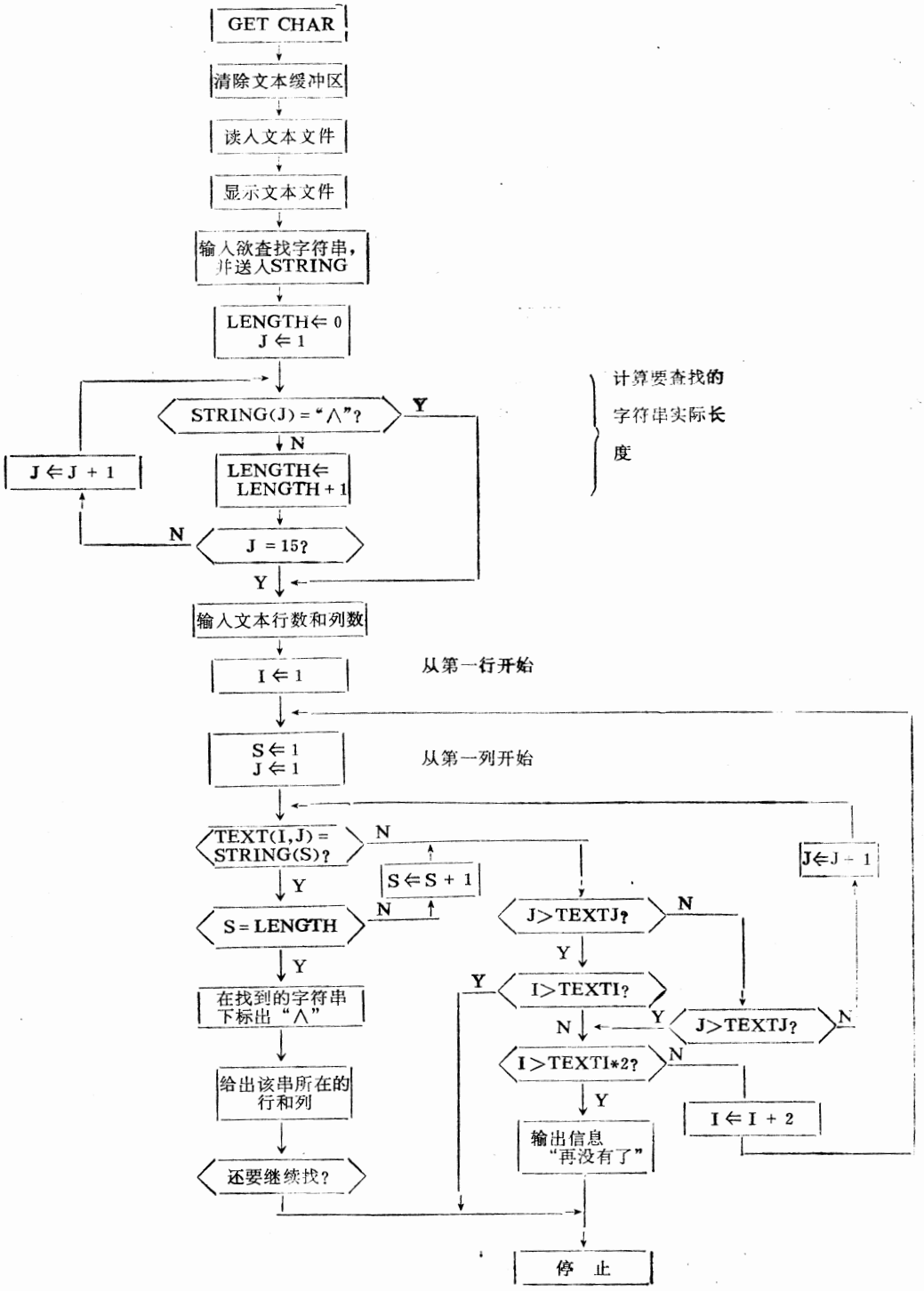

GET CHAR
清除文本缓冲区
读入文本文件
显示文本文件
输入欲查找字符串，并送入STRING
LENGTH⇐ 0
J ⇐ 1
STRING(J) = "∧"?
Y
N
J ⇐ J + 1
LENGTH⇐ LENGTH + 1
N
J = 15?
Y
计算要查找的字符串实际长度
输入文本行数和列数
I ⇐ 1
从第一行开始
S⇐ 1
J ⇐ 1
从第一列开始
TEXT(I,J) = STRING(S)?
N
Y
S⇐ S + 1
N
S = LENGTH
Y
在找到的字符串下标出 "∧"
给出该串所在的行和列
还要继续找?
J>TEXTJ?
N
Y
J⇐J - 1
Y
I>TEXTI?
Y
N
J>TEXTJ?
N
I>TEXTI*2?
N
Y
I ⇐ I + 2
输出信息 "再没有了"
停 止

图 116

```
      DO  10  I=1, 40                                  清除文本
      DO  10  J=1, 70                                  缓冲区
10    TEXT (I, J) =''
      READ (5, 15)  ( (TEXT(I, J), J=1,                读入并显示
        70) , I=1, 40, 2)                              文本缓冲区
      WRITE (4, 20)((TEXT(I,J), J=1, 70),
      I=1, 40)
15    FORMAT (70A1)
20    FORMAT (5X, 70A1)
      WRITE (4, 25)
25    FORMAT (5X, 'TYPING THE STRING                   输入你欲查找的字符串
      YOU  LOOK  FOR')
      READ (1, 27) STRING
27    FORMAT (15A1)
      LENGTH=0
C***  EVALUATE  LENGTH  OF  THE  STRING
      DO  30  J=1, 15                                  计算欲查找字符串长度
      IF (STRING(J) . EQ. '^')GO  TO  40
30    LENGTH=LENGTH+1
40    WRITE (4, 35)
35    FORMAT (5X, 'TYPING  THE                         输入文本缓冲区
        NUMBER OF YOUR TEXT''S LINE')                  的行数和列数
      READ (1, 50) TEXTI
50    FORMAT (I2)
      WRITE (4, 55)
55    FORMAT (5X, 'TYPING  THE
        NUMBER  OF  YOUR  TEXT''S
      COLUME')
      READ (1, 50) TEXTJ
      DO  125  I=1, TEXTI*2, 2                         查找，若找到，则在该字符
      S=1                                              串的始、末二个字符串的下
60    DO  120  J=1, TEXTJ                              面标以“^”。
      IF (TEXT(I, J) . EQ. STRING(S) )
      GO  TO  80
      IF (J. GT. TEXTJ) GO  TO  70
C***  SEARCH  FROM  NEXT  POSITIOS  IN
      CURRENT  LINE
      S=1
      GO  TO  120
70    IF (I. GT. TEXTI) GO  TO  140
      GO  TO  125
80    IF(S. NE. LENGTH)  GO TO 110
      TEXT (I+1, J-LENGTH+1)
      ='^'
      TEXT (I+1, J) ='^'
85    FORMAT (5X, 70A1)
```

```
      WRITE (4, 90)  (I+1)/2, J
90    FORMAT (5X, 'THE BEGINNING OF
      THE STRING FOUND IN TEXT IS
        AT LINE',
     1  2X, I2, 3X, 'COLUME', 2X, I2,/)
      WRITE (4, 20)  ( (TEXT(II, JJ) ,
        JJ=1, 70) , II=1, 40)
      WRITE (4, 95)
95    FORMAT (5X, 'WOULD YOU LIKE
        TO SFARCH THE MORE SAMF
        STRING?')
      READ (1, 100) C
100   FORMAT (1A1)
      IF (C. NE. 'Y') GO TO 160
      S=1
      GO TO 120
110   S=S+1
120   CONTINUE
125   CONTINUE
      WRITE (4, 130)
130   FORMAT (5X, 'NO MORE SUCH
        STRING FOUND', /)
      GO TO 160
140   WRITE (1, 150)
150   FORMAT (5X, 'NO ANY SUCH STRING FOUND IN THE TEXT', /)
160   CLOSE (-1)
      STOP
      END
```

输出找到字符串所在的行和列，及缓冲区内容。

问是否还要继续查找，若回答“Y”，则继续，否则停止

查找结束。

原　文　件

TIC TAC TOE

THIS ANCIENT AND POPULAR GAME MAY SOME DAY BE FOUND ON THE WALLS OF CAVES-WHO KNOWS HOW MANY HOUNDRES OF YEARS PEOPLE HAVE BEEN PLAYING IT! KING TUT, SOCRATES, AND QUEEN ELIZABETH I WOULD ALL PROBABLY RECOGNIZE;

THIS SIMPLE LITTLE GAME REALLY ISN'T SO SIMPLE. CHARLES BABBAGE, ONE OF THE FATHERS OF COMPUTER SCIENCE, DEVELOPED SIX MACHINES SO HE COULD SIMULATE TIC TAC TOE. IT'S BEEN THE SUBJECT OF NUMEROUS OTHER STRATEGIES AND COMPUTER SIMULATION. SOME PROGRAMMERS HAVE EVEN GONE OUT OF THEIR WAY TO DEVELOP VERSIONS OF TIC TAC TOE THAT ONLY THE COMPUTER CAN WIN.

WE HAVEN'T BEEN SO CRUEL. WHO GOES FIRST (YOU OR THE COMPUTER) IS RANDOML Y DECIDED, SO YOU WILL HAVE YOUR CHANCE TO BE THE VICTOR. BEST PLAY FOR BOTH OPPONENTS IS ALWAYS TO PLAY TO A DRAW. HOWEVER IF YOU OR THE COMPUTER GETS LAZY. ONE OF YOU WILL EMERGE THE WINNER.

FOR THE DISCUSSION OF TIC TAC TOE STRATEGY, AND SOME IMAGINATIVE VARIATIONS SUCH AS TOF TAC TIC, SEE'YOUR MOVE'

BY DAVID L. SILVERMAN, NEW YORK, 1971. DONALD D. SPENCER, IN' GAME PLAYING WITH COMPUTERS', NEW JERSEY: HAYDEN, 1975 GIVES YOU THE OPPORTUNITY TO WRITE YOUR OWN TIC TAC TOE PROGRAM WITH PERSONAL TOUCHES.

程序读入原文件后在屏幕上显示的情况如下

TIC TAC TOE

THIS ANCIENT AND POPULAR GAME MAY SOME DAY BE FOUND ON THE WALLS OF CAVES-WHO KNOWS HOW MANY HOUNDRES OF YEARS PEOPLE HAVE BEEN PLAY ING IT! KING TUT, SOCRATES, AND QUEEN ELIZABETH I WOULD ALL PROBABLY RECOGNIZE:

THIS SIMPLE LITTLE GAME REALLY ISN'T SO SIMPLE. CHARLES BABBAGE, ONE OF THE FATHERS OF COMPUTER SCIENCE, DEVELOPED SIX MACHINES SO HE COULD SIMULATE TIC TAC TOE. IT'S BEEN THE SUBJECT OF NUMEROUS OTHER STRATEGIES AND COMPUTER SIMULATION. SOME PROGRAMMERS HAVE EVEN GONE OUT OF THEIR WAY TO DEVELOP VERSIONS OF TIC TAC TOE THAT ONLY THE COMPUTER CAN WIN.

WE HAVEN'T BEEN SO CRUEL. WHO GOES FIRST (YOU OR THE COMPUTER) IS RANDOMLY DECIDED, SO YOU WILL HAVE YOUR CHANCE TO BE THE VICTOR. BEST PLAY FOR BOTH OPPONENTS IS ALWAYS TO PLAY TO A DRAW. HOWEVER IF YOU OR THE COMPUTER GETS LAZY, ONE OF YOU WILL EMERGE THE WINNER.

FOR THE DISCUSSION OF TIC TAC TOE STRATEGY, AND SOME IMAGINATIVE VARIATIONS SUCH AS TOE TAC TIC, SEE'YOUR MOVE' BY DAVID L. SILVERMAN, NEW YORK, 1971. DONALD D. SPENCER, IN' GAME PLAYING WITH COMPUTERS'. NEW JERSEY: HAYDEN, 1975 GIVES YOU THE OPPORTUNITY TO WRITE YOUR OWN TIC TAC TOE PROGRAM WITH PERSONAL TOUCHES.

TYPING THE STRING YOU LOOK FOR　　输入欲查找的字符串，例中输入为"TIC TAC TOE∧"，最后一个"∧"为结束标志。

TYPING THE NUMBER OF YOUR TEXT'S LINE　　"20"

TYPING THE NUMBER OF YOUR TEXT'S COLUME　　"70"

THE BEGINNING OF THE STRING FOUND IN TEXT

IS AT LINE 1 COLUME 34　　告诉你第一个位置在第一行第34列

此为第一次查找 后的结果如下

TIC TAC TOE 找到的第一个字符串

∧　　∧

THIS ANCIENT AND POPULAR GAME MAY SOME DAY BE FOUND ON THE WALLS OF CAVES-WHO KNOMS HOW MANY HOUNDRES OF YEARS PEOPLE HAVE BEEN PLAYING IT! KING TUT, SOCRATES, AND QUEEN ELIZABETH I WOULD ALL PROBABLY RECOGNIZE:

THIS SIMPLE LITTLE GAME REALLY ISN'T SO SIMPLE. CHARLES BABBAGE, ONE OF THE FATHERS OF COMPUTER SCIENCE, DEVELOPED SIX MACHINES SO HE COULD SIMULATE TIC TAC TOE. IT' S BEEN THE SUBJECT OF NUMEROUS OTHER STRATEGIES AND COMPUTER SIMULATION. SOME PROGRAMMERS HAVE EVEN GONE OUT OF THEIR WAY TO DEVELOP VERSIONS OF TIC TAC TOE THAT ONLY THE COMPUTER CAN WIN.

WE HAVEN'T BEEN SO CRUEL. WHO GOES FIRST (YOU OR THE COMPUTER) IS RANDOMLY DECIDED, SO YOU WILL HAVE YOUR CHANCE TO BE THE VICTOR. BEST PLAY FOR BOTH OPPONENTS IS ALWAYS TO PLAY TO A DRAW. HOWEVER IF YOU OR THE COMPUTER GETS LAZY, ONE OF YOU WILL EMERGE THE WINNER.

FOR THE DISCUSSION OF TIC TAC TOE STRATEGY, AND SOME IMAGINATIVE VARIATIONS SUCH AS TOE TAC TIC, SEE'YOUR MOVE' BY DAVID L. SILVERMAN, NEW YORK, 1971. DONALD D. SPENCER, IN' GAME PLAYING WITH COMPUTERS', NEW JERSEY: HAYDEN, 1975 GIVES YOU THE OPPORTUNITY TO WRITE YOUR OWN TIC TAC TOE PROGRAM WITH PERSONAL TOUCHES,

WOULD YOU LIKE TO SEARCH THE MORE SAME STRING? 程序问还要继续查找吗？回答“y”。

THE BEGINNING OF THE STRING FOUND IN TEXT IS AT LINE 8 COLUME 27 告诉你第二个的位置在第8行第27列。

找到第二个位置后显示的结果如下

TIC TAC TOE

^ ^

THIS ANCIENT AND POPULAR GAME MAY SOME DAY BE FOUND ON THE WALLS OF CAVES-WHO KNOWS HOW MANY HOUNDRES OF YEARS PEOPLE HAVE BEEN PLAYING IT! KING TUT, SOCRATES, AND QUEEN ELIZABETH I WOULD ALL PROBABLY RECOGNIZE:

THIS SIMPLE LITTLE GAME REALLY ISN'T SO SIMPLE. CHARLES BABBAGE, ONE OF THE FATHERS OF COMPUTER SCIENCE, DEVELOPED SIX MACHINES SO HE COULD SIMULATE TIC TAC TOE. IT'S BEEN

^ ^

THE SUBJECT OF NUMEROUS OTHER STRATEGIES AND COMPUTER SIMULATION. SOME PROGRAMMERS HAVE EVEN GONE OUT OF THEIR WAY TO DEVELOP VERSIONS OF TIC TAC TOE THAT ONLY THE COMPUTER CAN WIN.

WE HAVEN'T BEEN SO CRUEL. WHO GOES FIRST (YOU OR THE COMPUTER) IS RANDOMLY DECIDED, SO YOU WILL HAVE YOUR CHANCE TO BE THE VICTOR. BEST PLAY FOR BOTH OPPONENTS IS ALWAYS TO PLAY TO A DRAW. HOWEVER IF YOU OR THE COMPUTER GETS LAZY, ONE OF YOU WILL EMERGE THE WINNER.

FOR THE DISCUSSION OF TIC TAC TOE STRATEGY, AND SOME

IMAGINATIVE VARIATIONS SUCH AS TOE TAC TIC, SEE'YOUR MOVE' BY DAVID L. SILVERMAN, NEW YORK, 1971. DONALD D. SPENCER, IN' GAME PLAYING WITH COMPUTERS', NEW JERSEY: HAYDEN, 1975 GIVES YOU THE OPPORTUNITY TO WRITE YOUR OWN TIC TAC TOE PROGRAM WITH PERSONAL TOUCHES.

WOULD YOU LIKE TO SEARCH THE MORE SAME STRING? 回答"y"

THE BEGINNING OF THE STRING FOUND IN TEXT IS AT LINE 10 COLUME 54

⋮

有多次上述情况，因为例子是要找出所有的"TIC TAC TOE"。

⋮

WOULD YOU LIKE TO SEARCH THE MORE SAME STRING? 回答"y"

THE BEGINNING OF THE STRING FOUND IN TEXT IS AT LINE 20 COLUME 11

找到最后一个"TIC TAC TOE"的显示结果

TIC TAC TOE

^ ^

THIS ANCIENT AND POPULAR GAME MAY SOME DAY BE FOUND ON THE WALLS OF CAVES-WHO KNOWS HOW MANY HOUNDRES OF YEARS PEOPLE HAVE BEEN PLAYING IT! KING TUT, SOCRATES, AND QUEEN ELIZABETH I WOULD ALL PROBABLY RECOGNIZE:

THIS SIMPLE LITTLE GAME REALLY ISN'T SO SIMPLE. CHARLES BABBAGE, ONE OF THE FATHERS OF COMPUTER SCIENCE, DEVELOPED SIX MACHINES SO HE COULD SIMULATE TIC TAC TOE. IT'S BEEN THE

^ ^

SUBJECT OF NUMEROUS OTHER STRATEGIES AND COMPUTER SIMULATION. SOME PROGRAMMERS HAVE EVEN GONE OUT OF THEIR WAY TO DEVELOP VERSIONS OF TIC TAC TOE THAT ONLY THE COMPUTER CAN WIN.

WE HAVEN'T BEEN SO CRUEL. WHO GOES FIRST (YOU OR THE COMPUTER) IS RANDOMLY DECIDED, SO YOU WILL HAVE VOUR CHANCE TO BE THE VICTOR. BEST PLAY FOR BOTH OPPONENTS IS ALWAYS TO PLAY TO A DRAW. HOWEVER IF YOUOR THE COMPUTER GETS LAZY, ONE OF YOU WILL EMERGE THE WINNER.

FOR THE DISCUSSION OF TIC TAC TOE STRATEGY, AND SOME

^ ^

IMAGINATIVE VARIATIONS SUCH AS TOE TAC TIC, SEE 'YOUR MOVE' BY DAVID L. SILVERMAN, NEW YORK, 1971. DONALD D. SPENCER, IN 'GAME PLAYING WITH COMPUTERS', NEW JERSEY: HAYDEN, 1975 GIVES YOU THE OPPORTUNITY TO WRITE YOUR OWN TIC TAC TOE PROGRAM WITH PERSONAL TOUCHES.

^ ^

WOULD YOU LIKE TO SEARCH THE MORE SAME STRING? 回答"y"，程序NO MORE SUCH STRING FOUND. 告诉你"没有其它的'TIC TACTOE'"程序停止

（三）思 考 题

1）本题是为查找小串中的字符在同一行而编写的程序，当欲查找的字符分成二部分，每个部分各位于不同行时，本程序不予查找。例如查找“ABC”时，若“AB”在x行，“C”在x＋1行，本程序在编写时不针对这种情况。请你对本程序做适当地改进，也能查出这样的字符串。

2）我们未来得及回答友人所提的问题，请你改动本程序的缓冲区的原文件等，适合查找姓名。

117 数字串排序游戏

任给一个1～9组成的字符串，你能通过若干步的调整使其恢复到自然排列的情况吗？调整时要遵守以下规则：当你指定从第3个位置调整后，只能把1～3二个位置的数字颠倒排过来。例如4 3 2 1 5，若指定调整位置为3，则调整后的结果为2 3 4 1 5，若再次指定调整位置为4，则调整后的结果为1 4 3 2 5。

（一）算 法 分 析

这是一个智力游戏，为了考验你的智力，我们将算法分析省略。若你能把长度为9的随意排列的数字串很快调整过来，则说明你很聪明，已掌握了其中的奥妙，但达到这一水平，你还要很好动脑筋，窥察排序的规律。

（二）程 序 设 计

1）设计思路

思路见框图。在程序里举例是长度为5的数字串4 5 1 2 3，经过三步调整过来的，源程序运行时，打印（或显示）出一步步调整过程。

2）框图 见图117

3）源程序及运行结果

```
C  ***  SEQUENCE  GAME                                  排序游戏
C  ***
        CHARACTER  OUT (5, 77)
        INTEGER  A(9) . B(9) . COUNT. N. L. I. J
        OPEN (4, FILE='EQQ', ACCESS='W')
        DO  10  I=1,5
        DO  10  J=1,77                                  清除缓冲区
  10    OUT (I, J) =' '
C  ***  INPUT  THE  LENTA  AND  CONTENTS  OF  YOUR  STRING
C  ***
        WRITE (1, 15)
  15    FORMAT (5X, 'TYPE  THE  LENGTH  OF              读入你所选择的
           THE  SEQUECE', /)                            数字序列长度L
        READ (1, 20) L
  20    FORMAT (I1)
```

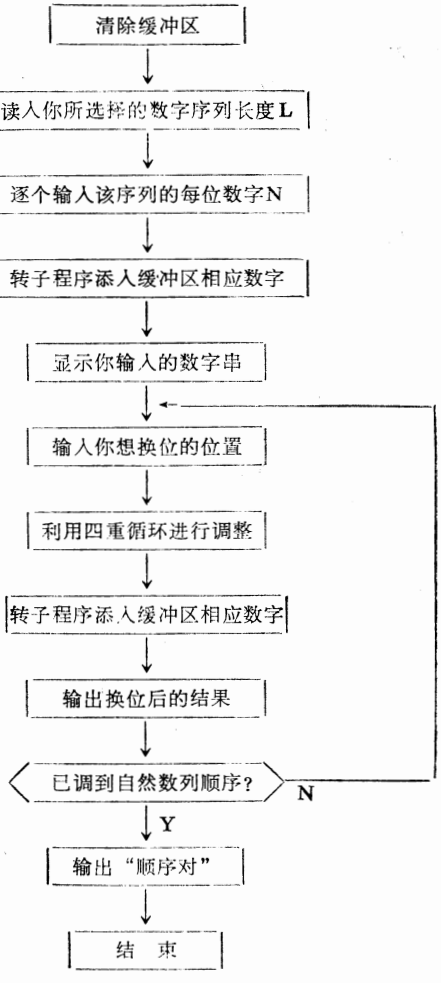


图 117

```
          COUNT = 0
 30       WRITE (1, 25)
 25       FORMAT (3X, 'TYPE A NUMBER IN YOUR
             SEQUENCE MIXED')                          } 逐个输入该序列
          READ (1, 20) N                               } 的每位数字N
          COUNT = COUNT + 1                            }
          CALL FILL (OUT, N, (COUNT-1) * 9)            } 添入缓冲区
          A(COUNT) = N                                 }
          IF (COUNT. NE. L) GO TO 30
C ***     INITIATED STRING
C ***
          WRITE (1, 40) ( (OUT (I, J) , J = 1,77) ,    }
             I = 1,5)                                  } 显示你输入的数字串
 40       FORMAT (2X, 77A1)                            }
          COUNT = 0
C ***     INPUT THE POSITION OF YOU SPECIFIED
```

```
C  ***
  50      WRITE (1, 45)
  45      FORMAT (/, 5X, 'WHAT POSITION YOU
             SPECIFIED? ')                          } 输入你想换位的位置N
          READ (1, 20) N
          IF (N. LE. L) GO TO 60                    }
          WRITE (1, 55)                             } 检查
  55      FORMAT (5X, 'TOO LONG, ONCE AGAIN
             PLEASE')                               }
          GO TO 50
  60      COUNT = COUNT + 1                         }
  70      DO 80 I = 1, N                            }
  80      B(1 + I - 1) = A(N - I + 1)               }
          DO 90 I = 1, N                            } 进行调整
  90      A(I) = B(I)                               }
          DO 95 I = 1, 5                            }
          DO 95 J = 1, N* 9 - 4                     }
  95      OUT(I, J) = ' '
          DO 100 I = 1, N
 100      CALL FILL (OUT, A(I), (I - 1) * 9)
C  ***  OUTPUT A RESULT FOR YOU
C  ***
          WRITE (1, 40) ((OUT(I, J), J = 1, 77),    输出换位后的结果
             I = 1, 5)
          DO 110 I = 1, L
          IF(A(I) . NE. I) GO TO 50                 } 检查是否已调
 110      CONTINUE                                  } 整到自然顺序
          WRITE (1, 120)
 120      FORMAT (5X, 'RIGHT SEQUENCE')             输出“顺序对”
          STOP
          END
C  ***  SUBROUTINE FILL WHICH FILL A NUMBER IN THE ARRAY
          OUT ACCORDING TO
C  ***  THE COLUMU
C  ***
          SUBROUTINE FILL (AOUT, K, P)              子程序，在缓冲区中
          INTEGER K, P, TL, TJ                      添入相应的数字。
          CHARACTER AOUT (5, 77)
          GO TO (210, 220, 230, 240, 250, 260, 270, 280, 290) K
 210      DO 215 TL = 1, 5
 215      AOUT(TL, P + 4) = '1'
          GO TO 300
 220      DO 225 TJ = P + 1, P + 4
```

```
      AOUT(1, TJ) ='2'
225   AOUT(5, TJ) ='2'
      AOUT(2, P+1) ='2'
      AOUT(2, P+4) ='2'
      AOUT(3, P+4) ='2'
      AOUT(4, P+3) ='2'
      GO TO 300
230   DO 235 TJ=P+1, P+4
      AOUT(1, TJ) ='3'
235   AOUT(5, TJ) ='3'
      AOUT(2, P+1) ='3'
      AOUT(2, P+4) ='3'
      AOUT(3, P+3) ='3'
      AOUT(3, P+4) ='3'
      AOUT(4, P+1) ='3'
      AOUT(4, P+4) ='3'
      GO TO 300
240   DO 245 TJ=P+1, P+5
245   AOUT(4, TJ) ='4'
      AOUT(1, P+4) ='4'
      AOUT(2, P+3) ='4'
      AOUT(2, P+4) ='4'
      AOUT(3, P+2) ='4'
      AOUT(3, P+4) ='4'
      AOUT(5, P+4) ='4'
      GO TO 300
250   DO 255 TJ=P+1, P+4
      AOUT(1, TJ) ='5'
      AOUT(3, TJ) ='5'
255   AOUT(5, TJ) ='5'
      AOUT(2, P+1) ='5'
      AOUT(4, P+4) ='5'
      GO TO 300
260   DO 265 TJ=P+1, P+4
      AOUT(1, TJ) ='6'
      AOUT(3, TJ) ='6'
265   AOUT(5, TJ) ='6'
      AOUT(2, P+1) ='6'
      AOUT(4, P+1) ='6'
      AOUT(4, P+4) ='6'
      GO TO 300
270   DO 274 TJ=P+1, P+4
274   AOUT(1, TJ) ='7'
```

```
      DO  277  I=2, 5
277   AOUT(I, P+4) ='7'
      GO  TO  300
280   DO  285  TJ=P+1, P+4
      AOUT(1, TJ) ='8'
      AOUT(3, TJ) ='8'
285   AOUT(5, TJ) ='8'
      AOUT(2, P+1) ='8'
      AOUT(4, P+1) ='8'
      AOUT(4, P+4) ='8'
      AOUT(2, P+4) ='8'
      GO  TO  300
290   DO  295  TJ=P+1, P+4
      AOUT(1, TJ) ='9'
      AOUT(3, TJ) ='9'
295   AOUT(5, TJ) ='9'
      AOUT(2, P+1) ='9'
      AOUT(2, P+4) ='9'
      AOUT(4, P+4) ='9'
300   RETURN
      END
```

依次输入 4，5，1，2，3

```
  4      5555      1      2222      3333
 44      5         1      2  2      3  3      输出你输入的结果
 4 4     5555      1        2        33      4  5  1  2  3
44444      5      1        2      3  3
  4      5555      1      2222      3333
```

输入换位位置为 2

```
5555       4       1      2222      3333
5         44       1      2  2      3  3      输出调整后的数字串
5555     4 4       1        2        33      5  4  1  2  3
  5    44444       1        2      3  3
5555       4       1      2222      3333
```

输入换位位置 5

```
3333     2222      1         4      5555
3  3     2  2      1        44      5         输出调整后的结果
 33        2      1       4 4      5555      3  2  1  4  5
3  3       2      1     44444         5
3333     2222      1         4      5555
```

输入换位位置 3

1	2222	3333	4	5555
1	2 2	3 3	44	5
1	2	33	4 4	5555
1	2	3 3	44444	5
1	2222	3333	4	5555

输出调整后的结果

1 2 3 4 5

（三）思 考 题

请你多做几次长度不同的随意排列的数字串，总结出排序规律。最后，你自己指定长度为 9 的随机数字串，你几次能将其调整过来？

十九　世界名画摩娜丽莎

118　世界名画摩娜丽莎

计算机为我们计算了多种类型的趣味题，在本书结尾时，再请计算机绘画，画一幅达·芬奇的名画——摩娜丽莎。

（一）算法分析

首先拿一幅摩娜丽莎画像，做为模特。据打印机行宽和纸张等宽度，选择适当比例，用数字标写在画像上。要特别注意面部、眼神的细致刻划，细心计算出不同部位的不同数字比例，选择字符或符号勾划出轮廓。

（二）程序设计

1）设计思路

据算法分析写在画像上的比例数字，据画像轮廓、阴影等不同部位，仔细构思，选择浓淡不同字符、符号或空白，进行描绘。程序中仅用两种语句——打印输出语句和格式语句，当然还需有STOP和END等。

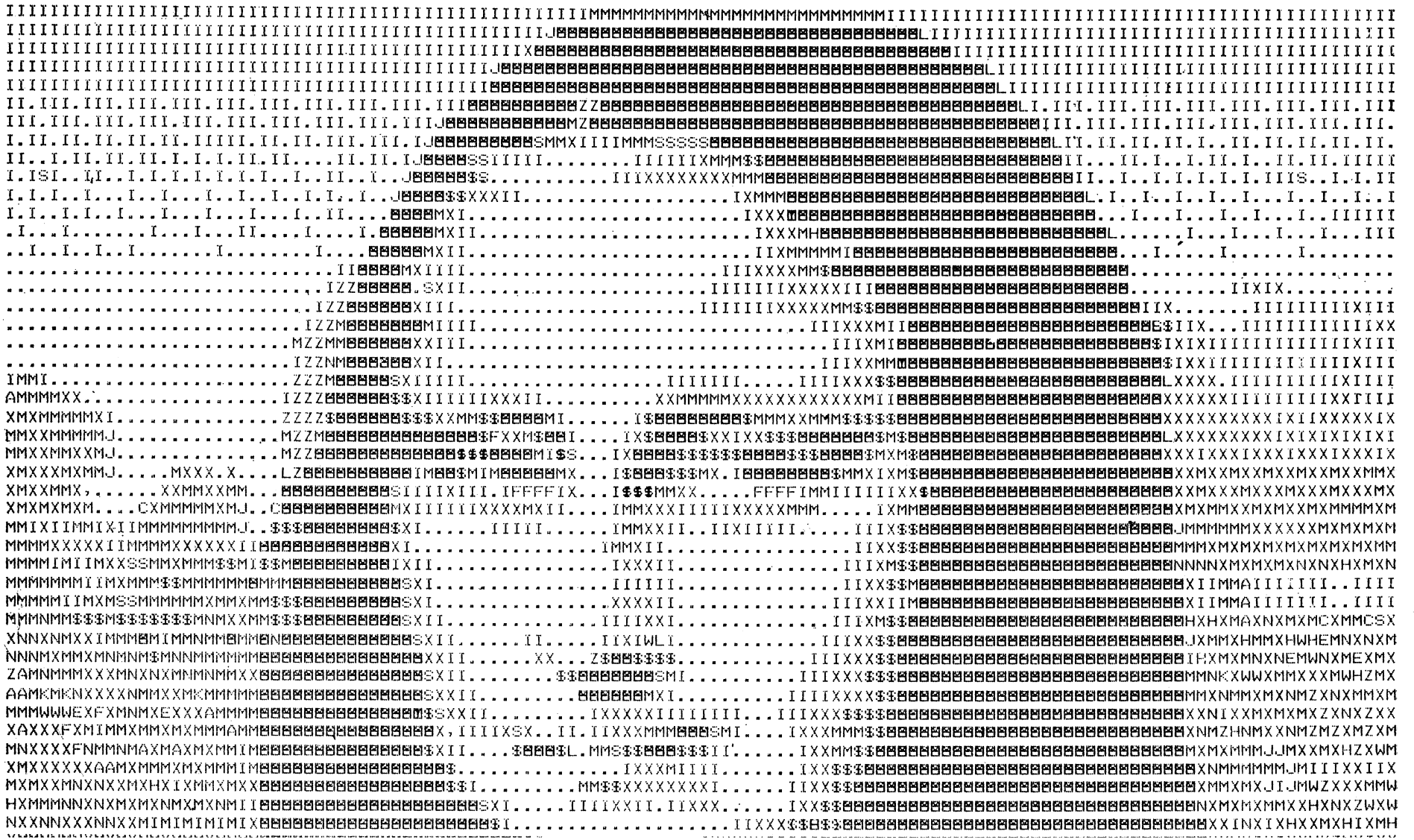
先令程序运行一次，打印出结果，再在打印纸上观察画像的不足之处，进行修正，这时可将修改的比例数字直接标写在打印的画像上。反复多次修正，直到满意为止。

本程序运行时，打印结果是一幅轮廓鲜明、唯妙唯肖的动人画像，假若达·芬奇看见了，也会大吃一惊，他的名画竟能被计算机再现！计算机也捕捉到像中人发自内心的微笑。

2）源程序运行结果打印的画像（图118），是在行宽132个字符打印机上打印的。

3）源程序

```
PROGRAM  MNLS
WRITE (5, 1)
WRITE (5, 2)
WRITE (5, 3)
WRITE (5, 4)
WRITE (5, 5)
WRITE (5, 6)
WRITE (5, 7)
WRITE (5, 8)
WRITE (5, 9)
WRITE (5, 10)
WRITE (5, 11)
WRITE (5, 12)
WRITE (5, 13)
WRITE (5, 14)
WRITE (5, 15)
```



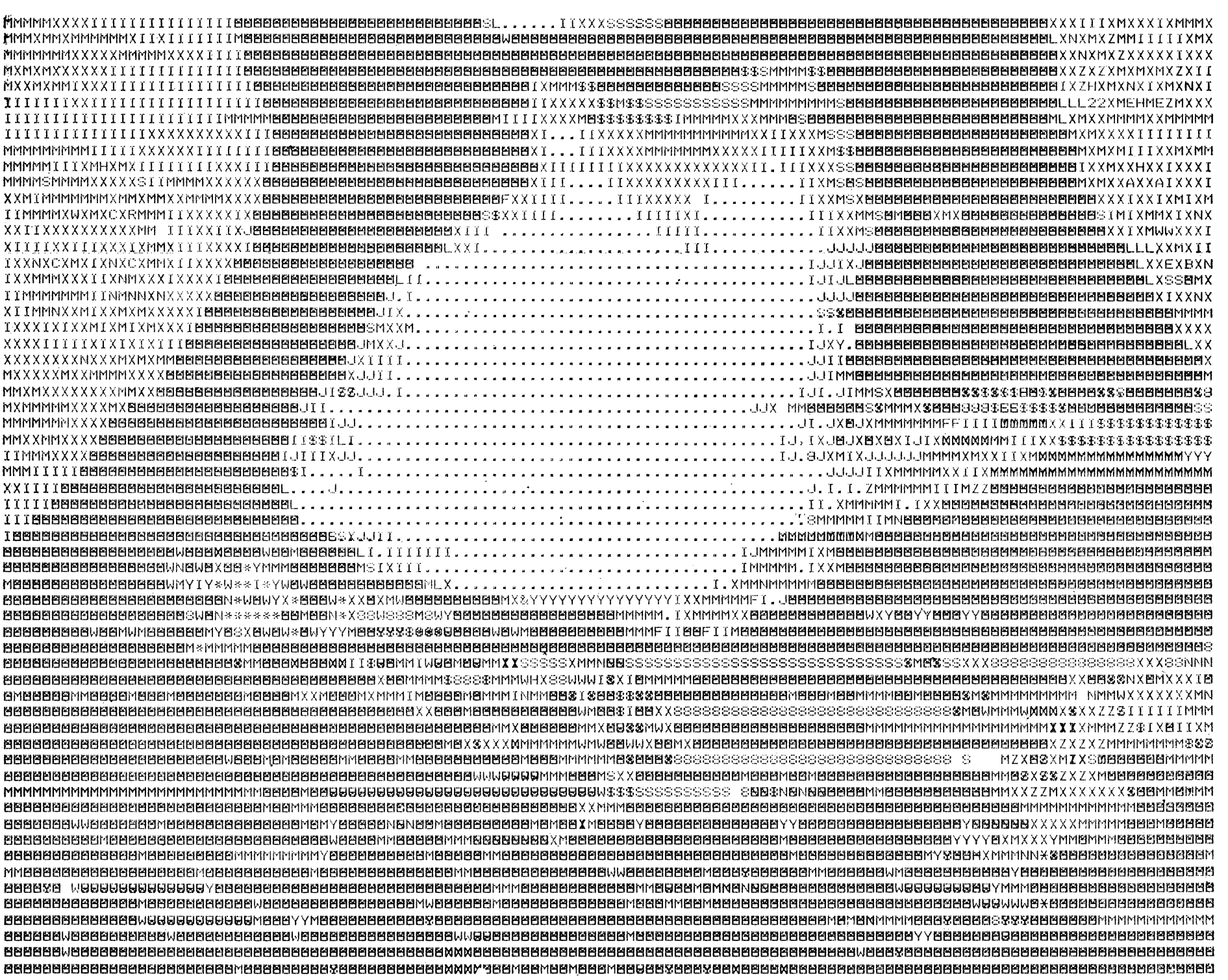


图 118

```
WRITE (5, 16)
WRITE (5, 17)
WRITE (5, 18)
WRITE (5, 19)
WRITE (5, 20)
WRITE (5, 21)
WRITE (5, 22)
WRITE (5, 23)
WRITE (5, 24)
WRITE (5, 25)
WRITE (5, 26)
WRITE (5, 27)
WRITE (5, 28)
WRITE (5, 29)
WRITE (5, 30)
WRITE (5, 31)
WRITE (5, 32)
WRITE (5, 33)
WRITE (5, 34)
WRITE (5, 35)
WRITE (5, 36)
WRITE (5, 37)
WRITE (5, 38)
WRITE (5, 39)
WRITE (5, 40)
WRITE (5, 41)
WRITE (5, 42)
WRITE (5, 43)
WRITE (5, 44)
WRITE (5, 45)
WRITE (5, 46)
WRITE (5, 47)
WRITE (5, 48)
WRITE (5, 49)
WRITE (5, 50)
WRITE (5, 51)
WRITE (5, 52)
WRITE (5, 53)
WRITE (5, 54)
WRITE (5, 55)
WRITE (5, 56)
WRITE (5, 57)
WRITE (5, 58)
```

```
      WRITE (5, 59)
      WRITE (5, 60)
      WRITE (5, 61)
      WRITE (5, 62)
      WRITE (5, 63)
      WRITE (5, 64)
      WRITE (5, 65)
      WRITE (5, 66)
      WRITE (5, 67)
      WRITE (5, 68)
      WRITE (5, 69)
      WRITE (5, 70)
      WRITE (5, 71)
      WRITE (5, 72)
      WRITE (5, 73)
      WRITE (5, 74)
      WRITE (5, 75)
      WRITE (5, 76)
      WRITE (5, 77)
      WRITE (5, 78)
      WRITE (5, 79)
      WRITE (5, 80)
      WRITE (5, 81)
      WRITE (5, 82)
      WRITE (5, 83)
      WRITE (5, 84)
      WRITE (5, 85)
      WRITE (5, 86)
      WRITE (5, 87)
      WRITE (5, 88)
      WRITE (5, 89)
      WRITE (5, 90)
      WRITE (5, 91)
      WRITE (5, 92)
      WRITE (5, 93)
      WRITE (5, 94)
      WRITE (5, 95)
1     FORMAT (4X, 53 (1HI) , 27 (1HM) , 46 (1HI))
2     FORMAT (4X, 49 (1HI) , 1HJ, 33 (1HM) , /1H+, 53X, 33 (1HS) , 1HL,
        42 (1HI))
3     FORMAT (4X, 47 (1HI) , 1HX, 38 (1HM) , 40 (1HI) , /1H+, 51X, 38 ( 1
        HS) )
4     FORMAT (4X, 44 (1HI) , 1HJ, 44 (1HM) , 1HL, 36 (1HI ),/1H+, 48X, 44
```

```
     (1HS) )
5       FORMAT (4X, 44 (1HI) , 46 (1HM) , 1HL, 35 (1HI) /1H+, 47X, 46(1HS))
6       FORMAT (4X, 2HII, 10 (1H. , 3HIII) , 10 (1HM), 2HZZ, 38 (1HM), 2HLI,
     1  8 (1H. , 3HIII) /1H+, 45X, 10 (1HS) , 2X, 38 (1HS) )
7       FORMAT (4X, 3HIII, 9 (1H. , 3HIII) , 1HJ, 12 (1HM) , 1HZ, 41 (1HM),
          8 (3HIII,
     1  1H. ) /1H+, 43X, 11 (1HS) , 2X, 41 (1HS) )
8       FORMAT (4X, 1HI, 5 (1H. , 2HII), 1H. , 3HIII, 3 (2HII, 1H. ), 2(3HIII,
     1  1H. ), 2HIJ, 9 (1HM), 1X, 2HMM, 1HX, 4 (1HI), 3 (1HM), 5X, 31(1HM),
     1  1HL, 4(2HII, 1H.), 1HI, 2 (1H. ) , 5 (2HII, 1H. )/1H+, 42X, 10 (1HS),
          10X,
     1  36 (1HS) )
9       FORMAT (4X, 6HII..I. , 3 (3HII. ), 3 (2HI. ), 2 (3HII. ) , 3 ( 1H. ), 2
          (3HII.) ,
     1  28H...II.I..I.II.I..II.II.IIIII/1H+, 41X, 6 (1HS) , 23X, 29 (1HS) )
10      EORMAT (4X, 11HI. 1SI..II...7 (2HI. ), 12H. I..II..I..J, 5 (1HM) ,1HI,
     1  12 (1H. ), 3 (1HI), 8 (1HX), 31 (1HM) , 7HII..I... 5 (2HI. ), 6HIIIS..,
     1  2 (2HI. ), 2 (1HI) /1H+, 40X, 7 (1HS) , 25X, 28 (1HS) )
11      FORMAT (4X, 3 (2HI. ) , 2H. I, 3 (1H. ) , 2 (3HI..) . 1H.. 3 (3HI..),
          2HI.. 2 (3
     1  5 (3HI..). 2H. I, 3 (3H..I) /1H+, 39X, 6 (1HS) . 29X, 27 (1HS) )
12      FORMAT (4X, 1HI, 2 (2H. I, 3H..I) , 4 (4H...I) , 4H..II, 4 (1H.),
     1  5 (1HM) , 2HXI, 25 (1H. ), 1HI, 3 (1HX) , 28 (1HM), 3 (4H...I), 2H..,
     1  4HI..., 3HI.., 6 (1HI) /1H+, 38 X, 4 (1HS) , 32X, 1HI, 27 (1HS) )
13      FORMAT (4X, 1H. , 2 (4HI...) , 1H. , 3 (4H...I) , 2 (5HI ....),2HI. ,
     1  6 (1HM) , 3HXII, 25 (1H. ) , 1HI, 3HXXX, 2HMH, 26 ( 1HM ) , 4HL...,
     1  5 (4H...I) . 2HII/1H+, 37X, 5 (1HS) , 35X, 26 (1HS) )
14      FORMAT (4X, 4 (3H..I) , 7 (1H. ) , 1HI, 8 (1H. ) , 1HI, 4 ( 1H. ) , 6
          (1HM) ,
     1  6 (1H. ). 1HI, 8 (1H. ) /1H+, 36X, 5 (1HS) , 39X, 24 (1HS) )
15      FORMAT (4X, 30 (1H. ) , 2HII, 5 (1HM) , 1HX, 4 (1HI) , 23 (1H. ) ,
          3 HIII, 4 (1HX) ,
     1  2HMM. 1HI, 27 (1HM) , 24 (1H. ) /1H+, 35X, 4 (1HS) , 38X, 28 (1HS))
16      FORMAT (4X, 29 (1H. ) , 8HIZZMMMMM, 5H. SXII, 22 (1H. ) , 7(1HI),
          8HXXXXXIII,
     1  23 (1HM) , 9 (1H. ) , 5HIIXIX, 10 (1H. ) /1H+. 35X, 5 (1HS) , 42X, 23
          (1HS) )
17      FORMAT (4X, 28 (1H. ) , 3HIZZ, 6 (1HM) . 4HXIII, 22 (1H. ) ,7 (1HI),
          5 (1HX) ,
     1  4HMMII, 24 (1HM) , 3HIIX, 7 (1H. ) , 9 (1HI) . 1HX, 3HIII/1H+, 34X,
          6 (1HS) ,
     1  40X, 26 (1HS) )
18      FORMAT (4X, 27 (1H. ) , 3HIZZ, 9 (1HM) , 4 (1HI) , 30 (1H. ) , 3HIII,
```

```
     3HXXX, 3HMII,
    1  22 (1HM) , 1HL, 3HIII, 1HX, 3H...,12 (1HI), 2HXX/1H+, 34X, 7 ( 1HS),
       44X,
    1  24 (1HS) )
19      FORMAT (4X, 26 (1H. ) , 3HMZZ. 8 (1HM) , 2HXX, 3HIII. 33 ( 1H. ) ,
       3HIII, 3HXMI,
    1  23 (1HM) , 5HIIXIX, 13 (1HI) . 1HX, 3HIII/1H+, 34X, 6 (1HS) , 44X, 24
       (1HS) )
20      FORMAT (4X, 26 (1H. ) , 4HIZZN, 7 (1HM), 3HXII, 34 (1H. ),5HIIIXX,
       26 (1HM) ,
    1  2HII, 2HXX, 13 (1HI) , 1HX, 3HIII/1H+, 34X, 6 ( 1HS ) , 44X. 1HI, 24
       (1H S) )
21      FORMAT (4X, 4HIMMI, 22 (1H. ) , 3HZZZ, 6 ( 1HM). 1X, 1HX, 5(1HI),
       18 (1H. ),
    1  7 (1HI) , 5 (1H. ) , 4 (1HI) . 3HXXX, 2HII, 24 (1HM) , 1HL, 4 (1HX) ,
       1H. , 10 (1HI) ,
    1  1HX, 4 (1HI) , /1H+, 33X, 6 (1HS) , 43X, 26 (1HS) )
22      FORMAT (4X, 1HA, 4 (1HM) , 2HXX, 18 (1H. ) , 1HI, 3HZZZ, 6 (1HM),
       2HII,
    1  6 (1HX ) , 9 ( 1HI ) , 2HXX, 4 ( 1HI ) . /1H+, 32X, 8 ( 1HS ) , 44X, 24
       (1HS) )
23      FORMAT (4X, 3HXMX, 5 (1HM ) , 2HXI, 15 ( 1H. ) , 4 ( 1HZ ) , 1HI, 6
       (1HM) , 3HIII, 2HXX,
    1  2HMM, 2HII, 5 (1HM) , 1HI, 6 (1H. ) , 2HII, 8 (1HM) , 1HI, 3 ( 1HM),
       2HXX, 3HMMM,
    1  5 (1HI) , 24 (1HM) , 10 (1HX) , 4HIXII, 5 ( 1HX) , 2HIX/1H+, 32X, 10
       (1HS) , 4X,
    1  6 (1HS) , 9X, 10 (1HS) , 8X, 29 (1HS) )
24      FORMAT (4X, 4HMMXX, 5 (1HM) , 1HJ, 15 (1H. ) , 3HMZZ, 15 (1HM),
       4HIFXX, 5HMIMMI,
    1  4 (1H. ) , 3HIXI, 4 (1HM ) , 4HIXXI, 5HXXIII, 7 ( 1HM ) , 3HIMI . 23
       (1HM), 1HL,
    1  8 (1HX) , 6 (2HXI) , /1H+, 32X, 15 (1HS) , 4X, 3HSSS, 7X, 6 (1HS) ,
       5 X,
    1  3HSSS, 7 (1H8) , 1HS, 1X, 24 (1HS) )
25      FORMAT (4X, 2 (4HMMXX) , 2HMJ, 15 (1H. ) , 3HMZZ, 13 ( 1HM ) ,
       3 HIII, 5 (1HM) , 3HI$I,
    1  3H..., 2HIX, 4 (1HM) , 6 (1HI) , 4 (1HM) , 3HIII, 4 (1HM) , 5HIMXMI,
       22 (1HM) ,
    1  5 (4HXXXI) , 1HX/1H+, 31X, 13 (1HS) , 3H$$$, 4 (1HS) , 2X, 2HSS,
       5X, 14 (1HS) ,
    1  8 (1HS) , 3X, 23 (1H8) )
26      FORMAT (4X, 5HXMXXX, 5HMXMMJ, 5 (1H. ) , 6HMXXX. X, 4 (1H.),
```

```
     2HLZ, 10 (1HM),
   1 7HIMMMIMI, 7 (1HM) , 4HX..., 2HII, 3HMMM,3HIII, 4HMX. I, 8 (1HM),
   1 8HIMMXIXMI, 23 (1HM) . 6 (3HXXM) , 2HMX, /1H+, 30X, 10 ( 1HS ) ,
     2X, 3HSSS, 3X,
   1 5 (1HS) , 6X, 7 (1HS) , 4X, 9 (1HS) , 6X, 1HS, 23 (1H8) )
27   FORMAT (4X, 8HXMXXMMX, 6 (1H. ) , 2 (4HXXMM) , 3H..., 10(1HM),
     1HS, 4 (1HI) ,
   1 6HXIII. I, 4 (1HF) , 2HIX, 3H..., 4 (1HI) , 4HMMXX, 5 (1H. ), 4(1HF),
     3HIMM,
   1 1H$, 22 (1H8) )
28   FORMAT (4X, 4 (2HXM) , 4 (1H. ) , 2HCX, 5 ( 1HM ) , 6HXMJ .. C, 11
     (1HM) , 1HX, 6 (1HI) ,
   1 4 (1HX) , 4HMXII, 4 (1H.) , 6HIMMXXX, 5 (1HI) . 5 (1HX) , 3HMMM,
     5 (1H.) , 2HIX,
   1 25 (1HM) , 3HXMX, 2HMM, 2HXX, 3HMXM, 2HXX, 2HMX, 4 (1HM) , 2
     HXM/1H+, 28X,
   1 10 (1HS) , 48X, 23 (1HS) )
29   FORMAT (4X, 2 (6HMMIXII) . 9 (1HM) , 6HJ..III, 8 ( 1HM ) , 3HIXI, 6
     (1H. ),
   1 5 (1HI) , 6 (1H. ), 5HIMMXX, 7HII .. IIX, 5 ( 1HI ) , 5 ( 1H. ), 3HIII,
     3HXII,
   1 23 (1HM) , 1HJ, 6 (1HM) , 5 (1HX) , 4 (2HXM), /1H+, 27X, 12 ( 1HS),
     45X, 25 (1HS) )
30   FORMAT (4X, 4 (1HM) , 5 (1HX) , 2HII, 4 (1HM) . 6 ( 1HX ) , 2HII, 12
     (1HS) , 2HXI,
   1 17 (1H.), 6HIMMXII, 17 (1H.), 6HIIXXII, 26 (1HM ), 8 ( 2HXM), 1HM, /
     1H+,
   1 26X, 12 (1HM) , 46X, 25 (1HS) )
31   FORMAT (4X, 4 (1HM) , 2HIM, 2HII, 1HM, 2HXX, 2HSS, 2HMM, 1HX,
     3HMMM, 2HII,
   1 1HM, 3HIII, 10 (1HM) , 4HIXII, 16 (1H.) , 1HI, 3HXXX, 2HII, 15 ( 1H.),
     3 (1HI) ,
   1 2HXM, 2HII, 23 (IHM) , 4 (IHN) , 3 (2HXM) , 2 (2HXN) , 2HXH, 2HXM,
     2HXN/1H+,
   1 22X, 2HSS, 2X, 2HSS, 1X, 9 (1HS) , 46X, 25 (1HS) )
32   FORMAT (4X, 7 (1HM) , 2HII, 2HMX, 3HMMM, 2HII, 20 (1HM), 3HSXI,
     16 (1H. ),
   1 6 (1HI) , 15 (1H. ), 2HII, 2HXX, 2HII, 25 (1HM) . 6HXIIMMA, 7 (1HI),
     2H...
   1 4 (1HI) /1H+, 17X, 2HSS, 6X, 1HS, 3X, 10 (1HS ) , 44X, 2HSS, 1X, 24
     (1HS) )
33   FORMAY (4X, 5 (1HM) , 7HIIMXMSS, 6 (1HM) , 2 (1HX, 2HMM), 3HIII,
     9 (1HM) ,
```

```
     1  1X, 2HXI, 16 (1H. ) , 4 (1HX) , 2HII, 14 ( 1H. ) , 3HIII, 2HXX, 2HII,
          25 (1HM) ,
     1  6HXIIMMA, 7 (1HI) , 2H.., 4 (1HI) /1H+, 27X, 13 (1HS) , 46X,24(1HS))
34        FORMAT (4X, 3HMMM, 1HN, 2HMM, 3HIII, 1HM, 7 ( 1HI ) , 3HMNM,
          2HXX, 2HMM, 3HIII,
     1  2HII, 25 (1HM) , 2 (2HHX) , 4HMAXN, 2 (2HXM) , 2HCX, 2HMM,3HCSX
          /1H+, 9X,
     1  3HSSS, 1X, 7 (1HS) , 7X, 13 (1HS) , 43X, 27 (1HS) )
35        FORMAT (4X, 1HX, 2HNN, 3HXNM, 2HXX, 1HI, 5 (1HM) , 1HI, 2HMM,
          1HN, 6 (1HM) ,
     1  1HN, 12 (1HM) , 2H  X, 2HII, 6 (1H. ) , 2HII, 5 (1H. ), 2HII,5HXIWLI,
          13 (1H. ),
     1  3HIII, 2HXX, 2HII, 26 (1HM) , 2HJX, 2HMM, 2HXH, 2HMM, 11HXHWHE
          MNXNXM/1H+,
     1  15X, 1HS, 7X, 1HS, 2X, 1HS, 1X, 13 (1HS) , 41X, 28 (1HS) )
36        FORMAT (4X, 3HNNN, 2HMX, 2HMM, 2HXM, 2 (2HNM) , 2HIM, 2HNN,
          21 (1HM) , 2HXX,
     1  2HII, 6 (1H. ) , 2HXX, 3H..., 2HZI, 2HMM, 4 ( 1HI ) , 12 ( 1H. ) , 3
          (1HI) , 3HXXX,
     1  2HII, 26 (1HM) , 2HIH, 2 (2HXM) , 13HNXNEMWNXMEXMX/1H+, 16X,
          1HS, 9X,
     1  15 (1HS) , 16X, 7 (1HS) , 18X, 28 (1HS) )
37        FORMAT (4X, 4HZAMN, 3 (1HM) , 3HXXX, 5HMNXNX, 2 ( 2HMN ) , 2
          HMM, 2HXX,
     1  15 (1HM) , 2H  X, 2HII, 8 (1H. ) , 2HII, 7 (1HM) , 3H  MI. 10 (1H.) ,
          3HIII,
     1  3HXXX, 3HIII, 28 (1HM) , 3HNKX, 2HWW, 1HX, 2HMM, 3HXXX, 6HMW
          HZMX/1H+,
     1  26X, 16 (1HS) , 11X, 10 (1HS) , 18X, 29 (1HS) )
38        FORMAT (4X, 2HAA, 2 (2HMK) , 1HN, 4 (1HX) , 1HN, 2HMM, 2HXX,
          2HMK, 20 (1HM) ,
     1  1H, 2HXX, 2HII, 9 (1H. ), 7 ( 1HM ) , 2HXI, 10 ( 1H. ) , 4 ( 1HI ) , 4
          (1HX) , 2HII,
     1  28 (1HM) , 2HXN, 2HMM, 9HXMXNMZXNX, 2HMM, 2HXM/1H+, 26X,
          16 (1HS) , 13X,
     1  6 (1HS) , 21X, 28 (1HS) )
39        FORMAT (4X, 3HMMM, 3HWWW, 9HEXFXMNMXE, 3HXXX, 1HA, 19 ( 1
          HM) , 2HI, 2HXX,
     1  2HII, 9 (1H.) , 1HI, 5 (1HX) , 8 (1HI) , 3 ( 1H. ), 3HIII, 3HXXX, 4 ( 1
          HI) , 27 (1HM) ,
     1  2HXX, 2HNI, 2HXX, 3 (2HMX) , 5HZXNXZ, 2HXX/1H+, 26X, 14(1HS),
          1HI, 2HSS,
     1  36X, 31 (1HS) )
```

```
40      FORMAT (4X, 2HXA, 3HXXX, 4HFXMI, 2 (2HMM, 1HX),2HMX, 3HMMM,
          1HA, 18 (1HM) ,
     1  2HX, 4 (1HI) , 3HXSX, 2H... 2HII, 1H., 2HII, 3HXXX, 6(1HM), 3H MI,
     1  2 (3HZXM) /1H+, 26X, 16 (1HS) , 22X, 4 (1HS) , 13X, 29 (1HS) )
41      FORMAT (4X, 2HMN, 4 (1HX), 2HFN, 2HMM, 9HNMAXMAXMX,2HMM,
          1HI, 16 (1HM),
     1  2HIX, 2HII, 4 (1H, ), 1HI, 3HMMM, 3HIL, , 2HMM, 1X, 2HII, 3HMMM,
          5 (1HI),
     1  6 (1H, ) , 1HI, 2HXX, 2HMM, 2HII, 29 (1HM) , 3HXMX, 3HMMM,2HJJ,
          1HM, 2HXX,
     1  7HMXHZXWM/1H+, 26X, 16 (1HS) , 7X, 5 (1HS) , 4X, 9 ( 1HS ) , 13X,
          30 (1HS) )
42      FORMAT (4X, 2HXM, 6 (1HX), 2HAA, 2HMX, 3HMMM,3HXMX, 3HMMM,
          1HI, 18 (1HM) ,
     1  1HI, 15 ( 1H, ) , 1HI, 3HXXX, 1HM, 4HIIII, 7 ( 1H, ) , 1HI, 2HXX, 3
          HIII, 30 (1HM) ,
     1  2HXN, 6 (1HM) , 2HJM, 3HIII, 5HXXIIX/1H+, 26X, 18 ( 1HS ) , 34X, 33
          (1HS) )
43      FORMAT (4X, 2 ( 2HMX ) , 5HXMNXN, 2HXX, 6HMXHX1X, 2HMM, 2
          HXM, 2HXX, 17 (1HM) ,
     1  3HIII, 9 (1H, ) , 4HMMII, 8 (1HX) , 1HI, 6 (1H, ) , 6HIIXXII,31(1HM),
          3HXMM,
     1  9HXMXJIJMWZ, 3HXXX, 3HMMW/1H+, 26X, 19 ( 1HS ) , 12X, 2HSS, 19
          X, 33 (1HS) )
44      FORMAT (4X, 3HHX, 2HMMM, 4HNNXN, 2 ( 2HXM ) , 6HXNMXMX, 4
          HNMII, 20 (1HM) ,
     1  3H XI, 5 (1H, ) , 4 (1HI ) , 5HXXII, , 5HIIXXX, 6 ( 1H, ),5HIXXII, 32
          (1HM) ,
     1  2HNX, 2 (2HMX) , 4HMMXX, 8HHXNXZWXW/1H+, 26X, 21 ( 1HS ) , 39
          X, 34 (1HS) )
45      FORMAT (4X, 5HNXXNN, 3HXXX, 4HNNXX, 5 ( 2HMI ) , 1HX, 21 ( 1
          HM) , 2HII,
     1  20 (1H, ) , 5HIIXXX, 5H (1HI) , 33 (1HM) , 8HXXINXIXH, 9HXXMXHIX
          MH/1H+,
     1  26X, 22 (1HS) , 26X, 38 (1HS) )
46      FORMAT (4X, 5HXMMNN, 2HMN, 2 ( 2HXM ) , 4HXNMX, 3HMMM, 1
          HX, 28 (1HM) , 2HII,
     1  14 (1H, ) , 1HI, 5 (1HX) , 5 (1HI),35 (1HM) , 4HXXNX, 2 ( 2HIX ) , 7
          HXHXZNXI,
     1  2HXX/1H+, 26X, 25 (1HS) , 21X, 40 (1HS) )
47      FORMAT (4X, 4HXMXX, 2HMX, 4 (1HM) , 6HXMXXMM, 3 ( 2HXM ) , 26
          (1HM) , 2HSI,
     1  11 (1H, ) , 3HIII, 5 (1HX) , 3HIII, 37 (1HM) , 4 (1HX) , 3HNNN, 4HMX
```

```
     XX,
    1  3HHXX, 3HIXX/1H+, 26X, 24 (1HS) , 22X, 40 (1HS) )
48      FORMAT (4X, 5 (1HM) , 4 (1HX) , 15 (1HI) , 26 (1HM) , 2HSL, 6(1H.),
       5HIIXXX,
    1  26 (1HS) , 13X, 46 (1HS) )
49      FORMAT (4X, 3HMMM, 3HXMM, 1HX, 6 (1HM) , 4HXIIX, 7 ( 1HI ) , 28
       (1HM) , 1HW,
    1  56 (1HM) 7HLXNXMXZ, 2HMM, 5 (1HI) , 3HXMX/1H+, 28X, 27 ( 1HS),
       1X, 56 (1HS) )
50      FORMAT (4X, 7 (1HM) , 5 (1HX) , 5 (1HM) , 4 (1HX) , 4 (1HI) , 85 ( 1
       HM) , 2HXX,
    1  5HNXMXZ, 5 (1HX) , 1HI, 3HXXX/1H+, 28X, 85 (1HS) )
51      FORMAT (4X, 3 (2HMX) , 5 (1HX) , 14 (1HI) , 52 (1HM) , 2HII, 1X, 4
       (1HM) , 2HII,
    1  24 (1HM) , 2HXX, 2 (2HZX) , 3 (2HMX) , 2HZX, 2HII/1H+, 28X, 55 ( 1
       HS) , 4X,
    1  26 (1HS) )
52      FORMAT (4X, 3HMXX, 5HMXMMI, 3HXXX, 15 (1HI) , 29 ( 1HM ) , 5HIX
       MMM, 2HII,
    1  13 (1HM) , 4X, 5 (1HM) , 1X, 25 ( 1HM ) , 16HIXZHXMXNXIXMXNXI/1H
       +, 29X,
    1  29 (1HS) , 5X, 19 (1HS) , 5X, 26 (1HS) )
53      FORMAT (4X, 7 (1HI) , 2HXX, 18 (1HI) , 28 (1HM) , 2HII, 5 (1HX) , 5
       HIIMII, 11X,
    1  9 (1HM) , 1X, 22 (1HM) , 5HLLL22, 8HXMEHMEZM, 3HXXX/1H+, 30X,
       28 (1HS) , 7X,
    1  2HSS, 1X, 13 (1SH) , 9X, 23 (1HS) )
54      FORMAT (4X, 23 (1HI) , 29 (1HM) , 4 (1HI) , 4 (1HX) , 2HMM, 9(1HI),
       5 (1HM) ,
    1  31X, 23 (1HS) , 10X, 9 (1HS) , 12X, 27 (1HS) )
55      FORMAT (4X, 15 (1HI) , 10 ( 1HX ) , 3HIII, 27 ( 1HM ) , 2HXI, 3H...,
       2HII,
    1  5 (1HX) , 11 (1HM) , 4HXXII, 3HXXX, 1HM, 3X, 23 (1HM),2HXM, 4 ( 1
       HX) , 8 (1HI)
    1  /1H+, 31X, 27 (1HS) , 31X, 25 (1HS) )
56      FORMAT (4X, 9 (1HM) , 5 (1HI) , 6 (1HX) , 8 (1HI) , 27 (1HM), 2HXI,
       3H...,
    1  3HIII, 4 (1HX) , 7 (1HM) , 5 (1HX) , 5 (1HI) , 5HXXMII, 24 (1HM) ,
    1  2 (2HXM) , 5HIIIXX, 4HMXMM/1H+, 31X, 27 (1HS) , 32X, 25 (1HS))
57      FORMAT (4X, 5 (1HM) , 3HIII, 6HXMHXMX, 8 (1HI) , 6HXXXIII, 28 ( 1
       HM) , 1HX,
    1  8 (1HI) , 13 (1HX) , 3HII, , 3HIII, 3HXXX, 2X, 23 (1HM) , 4HIXXM , 5
       HXXHXX,
```

```
     1  5HIXXXI/1H+, 31X, 28 (1HS) , 31X, 25 (1HS) )
58       FORMAT (4X, 4 (1HM) . 5HSMMMM, 5 (1HX) , 3HSII, 4 ( 1HM ) , 6 ( 1
           HX) , 28 (1HM) ,
     1  4HXIII, 4 (1H. ), 2HII, 9 ( 1HX ) , 3HIII, 6 ( 1H. ) , 7HIIXM  M  ,23( 1
           HM) ,
     1  2HXM, 2 (2HXX, 1HA) , 5HIXXXI/1H+, 30X, 28 (1HS) , 32X, 25 (1HS))
59       FORMAT (4X, 4HXXMI, 7 (1HM) , 2 (1HX, 2HMM) , 2HXX, 4 (1HM ) ,
           4 (1HX) ,
     1  25 (1HM) , 3HFXX, 4 ( 1HI ) , 5 ( 1H. ), 3HIII, 5 ( 1HX ) , 1X, 1HI , 9
           (1H. ) ,
     1  4HIIXX, 3HMSX, 24 (1HM) , 6HXXXIXX, 2 (3HIXM ) /1H+, 3X, 1HX,
           26X,
     1  25 (1HS) , 38X, 24 (1HS) )
60       FORMAT (4X, 6HIIMMMM, 8HXWXMXCXR, 5HMMMII, 5 (1HX) , 2HIX,
           24 (1HM),
     1  2H, I, 6HXXIIII, 8 (1H. ) , 5 (1HI) , 2HXI, 11 (1H. ), 3HIII, 4HXXMM
           1X,
     1  5 ( 1HM ) , 3HXMX, 14 ( 1HM ) , 5HSIMIX, 7HMMXIXNX/1H +, 29X, 26
           (1HS) , 39X,
     1  2HSS. 1X, 3HSSS, 3X, 14 (1HS) )
61       FORMAT (4X, 4HXXII, 10 (1HX) , 2HMM, 1X, 3HIII, 6HXXIIXJ, 21 (1
           HM) , 1HX,
     1  3HIII, 1X, 16 (1H. ) , 5 ( 1HI ) , 12 ( 1H. ) , 6HIIXXM, 24 ( 1HM ) , 5
           HXXIXM,
     1  6HWWXXXI/1H+, 29X, 21 (1HS) , 43X, 25 (1HS) )
62       FORMAT (4X, 1HX, 4 (1HI) , 5HXXIII, 5HXXXIX, 3HMMX, 3HIII, 4 (1
           HX) , 1HI,
     1  20 (1HM) , 4HLXXI, 21 (1H. ) , 3HIII, 12 (1H. ) , 5 (1HJ) , 26 (1HM),
           5HLLLXX,
     1  4HMXII/1H+, 29X, 20 (1HS) , 45X, 26 (1HS) )
63       FORMAT (4X, 3HIXX, 12HNXCXMXIXNXCX, 5HMMXII, 4 (1HX ) , 19 ( 1
           HM) , 1X,
     1  40 (1H. ) , 6HIJJIXJ, 28 (1HM) , 3HLXX, 5HEXBXN/1H+, 27X, 19(1HS),
           47X,
     1  28 (1HS) )
64       FORMAT (4X, 6HIXXMMM, 5HXXXII, 6HXNMXXX, 6HIXXXXI,18(1HM ),
           3HLII,
     1  40 (1H. ) , 5HIJIJL, 30 (1HM),2HLX,2X,3HMMX/1H+,26X,18(1HS),48X,
     1  30 (1HS) , 2X, 3HSSS)
65       FORMAT (4X, 3HIIM, 5 ( 1HM ) , 3HMII, 6HNMNNXN, 5 ( 1HX ) , 18 ( 1
           HM) , 3HJ, I,
     1  42 (1H. ) , 4 (1HJ) , 31 (1HM) , 6HXIXXNX/1H+, 25X, 18 (1HS ) , 49X,
           31 (1HS) )
```

```
66      FORMAT ( 4X, 5HXIIMM, 5HNXXMI, 5HXXMXM, 5 ( 1HX ) , 1HI, 18 ( 1
          HM) , 3HJIX,
     1  45 (1H. ) , 1HX, 38 (1HM) /1H+, 24X, 18 (1HS) , 46X, 37 (1HS) )
67      FORMAT (4X, 4HIXXX, 6HIXIXXM, 2 (3HIXM ) , 4HXXXI, 18 ( 1HM ) ,
          1X, 4HMXXM,
     1  42 (1H. ) , 3HI. I, 1X, 33 (1HM) , 4 (1HX) /1H+, 23X, 19 (1HS) , 50X,
          33 (1HS) )
68      FORMAT (4X, 4 (1HX) , 5 (1HI) , 4 (2HXI) , 2HII, 18(1HM), 5HJMXXJ,
          42 (1H. ) ,
     1  5HIJXY. , 34 (1HM) , 3HLXX/1H+, 22X, 18 (1HS) , 52X, 34 (1HS))
69      FORMAT (4X, 8 (1HX) , 4HNXXX, 2(2HMX), 20 (1HM) , 2HJX, 4(1HI),
          42 (1H. ) ,
     1  4HJJII, 37 (1HM) , 1HX/1H+, 21X, 18 (1HS) , 52X, 37 (1HS) )
70      FORMAT (4X, 1HM, 5 (1HX) , 3HMXX, 4 (1HM) , 4 (1HX) , 19 (1HM),
          5HXJJII,
     1  43 (1H. ) , 3HJJI, 39 (1HM) /1H+, 20X, 19 (1HS) , 53X, 36 (1HS) )
71      FORMAT (4X, 4HMMXM, 8 (1HX) , 4HMMXX, 17 (1HM) , 7HJIZZJJJ, 2
          H. I, 41 (1H. ),
     1  2HXJ/1H+, 19X, 17 (1HS) , 2X, 2HSS, 54X, 1HS, 1X, 33 (1HS) )
72      FORMAT (4X, 2HMX, 5 (1HM) , 4 (1HX) , 2HMX, 18 ( 1HM ) , 3HJII, 44
          (1H. ) , 3HJJX,
     1  18 (1HS) , 53X, 8 (1HS) , 4X, 30 (1HS))
73      FORMAT (4X, 7 (1HM) , 4 (1HX) , 23 (1HM) , 3HIJJ, 46 ( 1H. ), 8HJI.
          JXMJX,
     1  7 (1HM) , 2HFF, 9 (1HI) , 2HXX, 15 (1HI) /1H+, 14X, 23 (1HS) , 54X,
          1HS, 15X,
     1  5 (1HM) , 5X, 12 (1HS) )
74      FORMAT (4X, 6HMMXXMM, 4 (1HX) , 20 (1HM) , 5 (1HI) , 1HL, 45 ( 1
          H. ) , 4HIJ, I,
     1  1OHXJMJXMXMXI, 2HJI, 6 (1HX) , 2X, 5HIIIXX, 16 ( 1HI ) /1H+, 13X,
          20 (1HS) ,
     1  2X, 2HSS, 2X, 1HI, 50X, 1HS, 2X, 1HS, 1X, 1HS, 5X, 7 (1HM) , 5X, 16
          (1HS) )
75      FORMAT (4X, 5HIIMMM, 4 (1HX) , 20 (1HM) , 5HIJIII, 3HXJJ, 44(1H.),
          2HIJ,
     1  3H. JJ, 4HXMIX, 6 (1HJ) , 4 (1HM) , 4HXMXX, 4HIIXM, 3HXXX, 15( 1
          HY) /1H+,
     1  12X, 20 (1HS) , 55X, 1HS, 23X, 15 (1HM) )
76      FORMAT (4X, 3 (1HM) , 5 (1HI) , 22 ( 1HM ) , 2HII, 5 ( 1H. ) , 1HI, 48
          (1H. ) , 4 (1HJ) ,
     1  3HIIX, 5 (1HM) , 5HXXIIX, 23 (1HY) /1H+, 11X, 23 (1HS) , 72X, 23 ( 1
          HM) )
77      FORMAT (4X, 2HXX, 4 (1HI) , 23 (1HM) , 1HL, 4 ( 1H. ) , 1HJ, 49 (1
```

```
     H. ), 6HJ. I. I. ,
   1 1HZ, 6 (1HM) , 4HIIIM, 2HZZ, 23 (1HM) /1H+, 9X, 23 (1HS) , 74X, 23
     (1HS) )
78   FORMAT (4X, 5 (1HI) , 25 (1HM) , 1HL, 53 (1H. ) , 4HII. X, 5 (1HM),
     5HI. IXX,
   1 28 (1HM) /1H+, 8X, 25 (1HS) , 68X, 28 (1HS) )
79   FORMAT (4X, 3HIII, 28 (1HM) , 53 (1H. ) , 1H8, 5 (1HM) , 4HIIMN, 32
     (1HM) /1H+,
   1 6X, 28 (1HS) , 63X, 32 (1HS) )
80   FORMAT (4X, 1HI. 33 (1HM) , 1HL, 49 (1H. ) , 5HJJIII, 2HX, 35(1HS)
     ' /1H+, 4X,
   1 35 (1HS) , 5HXJJII, 40X, 45 (1HM) )
81   FORMAT (4X, 18 (1HM) , 1HW, 8 (1HM) , 1HW, 9 (1HM) , 3HLI. , 7
     (1HI), 30 (1H. ),
   1 2HSS, 1X,6(1HS) , 50X, 39 (1HS) )
82   FORMAT (4X, 17 (1HM) , 6HWNMWMX, 4HMM*Y, 11 (1HM), 6HSIXIII,
     33 (1H. ) , 1HI,
   1 7 (1HS) , 51X, 38 (1HS) )
83   FORMAT (4X. 17 (1HM) , 7HWMYIY*W, 2H**,6HI*YWMW,13 (1HM),
     2HLX, 27 (1H. ) ,
   1 6HI. XMMN, 46 (1HM) /1H+, 4X, 16 (1HS) , 13X, 1HS, 1X, 12 (1HS) ,
     41X, 41 (1HS) )
84   FORMAT (4X, 23 (1HM) , 8HN*WMWYX*, 3HMMM, 4HW*XX, 4HMX-
     MW, 11 (1HM) ,
   1 2HX&, 15 (1HY) , 3HIXX, 5 (1HM) , 4HFI. J, 44 (1HM) /1H+, 3X, 23
     (1HS) , 3X, 1HS,
   1 4X, 3HSSS, 4X, 1HS, 3X, 10 (1HS) , 30X, 44 (1HS) )
85   FORMAT (4X, 19 (1HM) , 4H8W8N, 6 (1H*) , 5 (1HM) , 3HN * X, 6H88
     W888, 4HM8WY,
   1 22 (1HM) , 3H. IX, 4 (1HM) , 2HXX, 12 (1HM) , 3HWXY, 4HMMYY, 5
     HMMMYY, 24 (1HM)
   1 /1H+, 3X, 19 (1HS) , 2X, 1HM, 7X, 2HSS, 1X, 2HSS, 13X, 17 (1HS) ,
     14X,
   1 12 (1H8) , 3X, 2H88, 2X, 3H888, 2X, 24 (1H8) )
86   FORMAT (4X, 9 (1HM) , 5HWMMMW, 8 (1HM) , 11HYM8XMWMW*MW,
     6HYYYMMM, 4HYYYI,
   1 1X, 2HSS, 3X, 6 (1HS) , 2X, 1HS, 2X, 1H8, 1X, 1H8, 2X, 1H8, 5X, 14
     (1H8) , 1X,
   1 1H8, 2X, 10 (1H8) , 6X, 2H88, 4X, 49 (1HS) )
87   FORMAT (4X, 20 (1HM) , 5H*MMMM, 100 (1HM) /1H+, 3X, 19 (1H8),
     7X, 100 (1H8) )
88   FORMAT (4X, 24 (1HM) , 1HX, 11 (1HM) , 7HIIIWMMM, 7HIWWMMMW,
     4HMMII, 5 (1HS) ,
```

```
     1  6HXMMNNN, 11X, 17 (1HS) , 1X, 4HXMMX, 5HSSXXX, 15 ( 1H8) , 8HX-
          XX88NNN/1H+,
     1  3X, 25 ( 1H8 ) , 2X, 4H888X, 5H888XX, 2X, 3H888, 4X, 2H88, 1X, 2H88,
          2X, 2HXX,
     1  9X, 32 (1HS) , 1X, 2HSS)
89        FORMAT (4X, 39 (1HM) , 1HX, 6 (1HM) , 1HI, 3X, 4HIMMM, 5HWHX88,
          7HWWWIXXI,
     1  45 (1HM) , 6HXXMMXX, 4HNXMM, 5HXXXIM/1H+, 3X, 39 (1H8 ) , 1X,
          2H88, 4X, 1HS,
     1  4 (1H8) , 12X, 1H8, 2X, 1H8, 5X, 39 (1H8) , 2X, 4 (1H8) , 2X, 1HS, 5X,
          1HS)
90        FORMAT (4X, 31 (1HM) , 2HXX, 5 ( 1HM ) , 5HXMMMI, 10 ( 1HM ) ,
          6 HINMMMM, 5HXIXMM,
     1  4HIIXX, 22 (1HM) , 10 (1HM) , 3HXMX, 9 (1HM) , 5H  NMMW, 7(1HX),
          2HMN/1H+, 3X,
     1  1H8, 1X, 5 (1H8) , 2X, 4 (1H8) , 1X, 3H888, 1X, 7 (1H8) , 1X, 4 (1H8),
          4X, 3H888,
     1  7X, 3H888, 1HS, 1X, 1HS, 7X, 3H888, 1X, 21 (1H8 ) , 1X, 3HS88, 1X,
          2 H88, 4X,
     1  2H88, 1X, 5 (1HS) , 1X, 1H8)
91        FORMAT (4X, 43 (1HM) , 5HXXMMM, 12 (1HM) , 6HWMMMII,4HMMXX,
          29X, 4HXMMW,
     1  4HMMMW, 7 (1HX) , 3HZZZ, 5 (1HI) , 4HIMMM/1H+, 3X, 43 (1H8) ,
          2 X, 3 (1H8) , 1X,
     1  11 (1H8) , 2X, 3 (1H8) , 1X, 2H88, 2X, 30 (1H8) , 1X, 1H8,
     1  5X, 3 (1HM) , 1X, 1HS, 4X, 1HS)
92        FORMAT (4X, 53 (1HM) , 1HX, 8 (1HM) , 2 (4HXMWX) , 39 (1HM) ,
          4 (1HX) , 5HMMMZZ,
     1  2 (4HIIXM) /1H+, 3X, 51 (1H8) , 3X, 6 (1H8) , 3X, 4 (1H8) , 3X, 20 ( 1
          HS) , 19X,
     1  3HIII, 6X, 1H8, 2X, 1H8)
93        FORMAT (4X, 48 (1HM) , 6 (1HX) , 5 (1HM) , 4HMWMW, 4HMMWW,
          5HXMMMX, 37 (1HM) ,
     1  3 (2HXZ) , 8 (1HM) , 3HIXZ/1H+, 3X, 46 (1H8) , 1X, 1H8, 1X,1HS,3X,
          1HM, 9X,
     1  2H88, 3X, 2H88, 2X, 37 (1H8) , 14X, 3H888)
94        FORMAT (4X, 23 (1HM) , 1HW, 41 (1HM),5HXMMMX, 34X, 3 (3HMZX),
          1X, 12 (1HM) /
     1  1H+,3X,23 (1H8) , 1X, 3HSSS, 1X, 1HS, 1X, 5 (1HS ) , 2X, 4 ( 1H8 ) ,
          2 X, 11 (1H8) ,
     1  1X, 3HSSS, 6X, 35 (1H8) , 1X, 1HS, 6X, 2H88, 2X, 4HIXSI, 6 (1HS ))
95        FORMAT (4X, 49 (1HM) , 7 (1HW) , 7 (1HM) , 1X, 2HXX, 40 ( 1HM ) ,
          4 (2HZX) , 12 (1HM)
```

```
1  /1H+, 3X, 49 (1H8) , 3X, 4 (1H9) , 3X, 3H888, 1X, 1HS, 2X, 11 (1H8),
     1X, 3H888,
1  1X, 2HSS, 1X, 18 (1H8) , 2X, 2H88, 1X, 2HSS, 5X, 11 (1H8) )
   STOP
   END
```

（三）思　考　题

1）请按此程序亲自打印这幅名画，挂在墙上观查效果，你会感到栩栩如生，富有立体感，逼真到令人惊讶、赞叹的境界，赠给友人，留作纪念。

2）编写打印人物的程序，需要仔细构思，要较高技巧。开始练习时，建议你拿一幅熊猫像，按尺寸比例勾出大致轮廓，若有些差距，也不会太失真，再修改一次，便成了可爱的熊猫。