

BASIC 趣味程序 实例与分析

张宝玺 主编



电子工业出版社

16209 · 206

2.80 元

BASIC 趣味程序

实例与分析

张宝玺 主编

电子工业出版社

一九八五年

内容提要

内容提要

本书包括七十一个我国程序设计人员用BASIC语言编写的游戏程序和智力锻炼程序，内容新颖，题材多样，生动有趣，引人入胜。每一程序均附有使用说明，大部分程序给出了设计思路。因此，不仅可以供娱乐之用，而且可以帮助广大BASIC语言初学者巩固和提高BASIC编程能力，从中学到一些编程技巧。对于专门从事程序设计的计算机软件人员，本书亦有一定参考价值。

BASIC 趣味程序实例与分析

张宝玺 主编

电子工业出版社出版（北京万寿路）

新华书店北京发行所发行 全国各地新华书店经售

电子外文印刷厂印刷

开本：787×1092毫米 1/16印张：13.5 插页：340千字

1985年10月第一版 1985年10月第一次印刷

印数：1—20,000 册 定价：2.80元

统一书号：15290.206

前 言

BASIC语言是目前国际上通用的计算机算法语言。由于它结构简单灵活，好懂易学，具有人机对话的特点，因此特别适合初学计算机的各种人员和广大青少年学习使用。随着新技术革命的深入发展和计算机的普及应用，在我国各行各业已经掀起了一股学习计算机，学习BASIC语言的热潮，学校中也把BASIC语言列为计算机学习课程。初步统计，到1984年底止，全国学习过BASIC语言的人员已经达到三百万人左右，而且人数还在迅速增加。目前，BASIC程序设计人员，初学者以及广大青少年迫切希望有更多内容丰富、生动活泼的BASIC语言普及读物，用于激发学习兴趣，传授编程技巧，锻炼智力，巩固和提高程序设计能力。本书就是为了满足这一需要而编写的。

本书共采集BASIC趣味程序70个，其中包括模拟游戏程序、智力锻炼程序以及一些涉及基本算法的程序。每一程序都编写了使用说明，部分程序附有运行结果，对一些难于理解的程序还给出了设计思想，语句说明和程序框图。这样，即使未学过BASIC语言的人，也可按照说明运行程序，进行游戏；学过BASIC语言的人，则可通过对程序设计思想和基本算法的介绍，提高程序设计的能力和编程技巧。这对于其他方面的程序设计也将有所补益。

本书给出的趣味程序，绝大部分都在TRS-80微型机上做过调试和验证。书中给出的源程序和运行结果就是程序在机器上运行后直接打印出来的。所以，读者在输入程序时，只要不出现错误，应该都能运行。书中给出的程序已录入磁带和磁盘，如果读者需要，可直接向计算机世界编辑部邮购。书中对不能在TRS-80微型机上运行的个别程序均做了说明。大部分程序可以直接或稍做改动就能在IBM-PC和Apple等广泛流行的一些微型机上运行。书末附有BASIC II程序语句、函数说明以及其它一些资料，可供读者调试和修改程序时参考。

本书由计算机世界编辑部组稿。计算机世界编辑部的刘雅英同志具体指导了本书的编写工作，国营辽声无线电厂何迎月等同志在此书的编写中付出了大量的劳动，谨致衷心的感谢。

编 者

一九八四年十二月

目 录

1	神州大地	1
2	巧打金字塔	3
3	打印三角形四边通用程序	4
4	两个两位数	6
5	巧取糖果	7
6	求重复数的平方	10
7	数字电子钟	13
8	炮击来敌	15
9	封锁游戏	18
10	打印体力情绪智力曲线图	21
11	电脑医生	23
12	抢数游戏	26
13	猜字母游戏	27
14	猜数	30
15	猜猜看	32
16	编制不等值的随机数表	34
17	初一英语选择填空练习	37
18	简易汉字库	38
19	十五子棋	41
20	数字游戏—谁先说出100	44
21	奇怪的数学题	46
22	巧填加减号	48
23	约瑟夫	50
24	字母数字谜	51

25	超级智力游戏	53
26	一字棋	57
27	史密斯魔阵	60
28	遥控汽车	64
29	公路赛车游戏	67
30	电子雪花	71
31	跷跷板游戏	73
32	打保垒	75
33	打飞机	76
34	射击	77
35	电脑识谱	78
36	美妙动听的音乐	79
37	模拟钢琴程序	82
38	孙子点兵	84
39	解九连环	86
40	巧分牛奶	89
41	周游世界	91
42	传教士与吃人生番	95
43	乒乓球比赛	97
44	打印万年历	100
45	飞机投弹	104
46	打魔星	105
47	巧分工种	109
48	军官问题	112
49	梵塔问题	115
50	地图染色	120
51	键盘控制行印机作图	123

52	恭贺新禧	125
53	用功能键作图	127
54	模拟光笔	130
55	探索迷宫	133
56	哥尼斯堡桥问题	135
57	填牌游戏	139
58	移子游戏	143
59	解奇数阶魔方	146
60	解奇偶阶魔方	147
61	魔方的判别	150
62	任意函数的图象	151
63	JOT	155
64	单人击球游戏	162
65	“S先生和P先生”谜题	164
66	快速掷骰子游戏	167
67	Wari游戏	171
68	打靶游戏	177
69	射击游戏	181
70	弹球游戏	186
附录一	BASIC II 程序语句一览表	195
附录二	BASIC II 字符串函数一览表	197
附录三	BASIC II 数学函数一览表	198
附录四	BASIC II 特殊函数一览表	199
附录五	导出函数表	199
附录六	控制、ASC II、图形符代码对照表	200
附录七	BASIC II 错误信息表	201
附录八	磁盘BASIC错误信息表	203
附录九	BASIC II 内存贮器分配图	204

— 神州大地

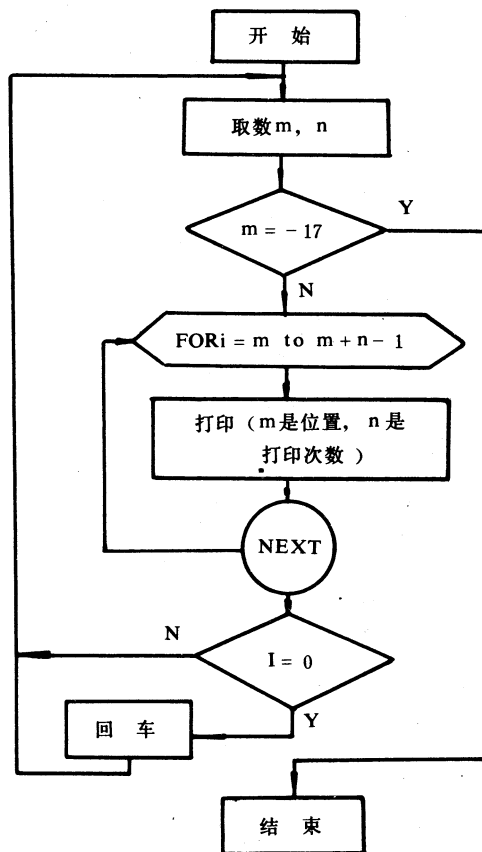
袁希文

用计算机打印一张生我、养我的母亲——神州大地的图形和字样是一件很有意义的事情。下面是打印神州大地图形的程序。

程序设计思想与说明

本图形程序非常简单，语句也不多。M的取数是程序中要求打印图形的起始位置，N的取数则是程序要求打印符号（M）的个数。一排排打印，直到M、N的取数是-1，则说明图形已全部打完。当M、N取数为0时，程序安排机器回车，表示一行打印完毕，另从下行开始打印，反复循环。计算机打印出的图形与本文给出的图形相差90°。

框图

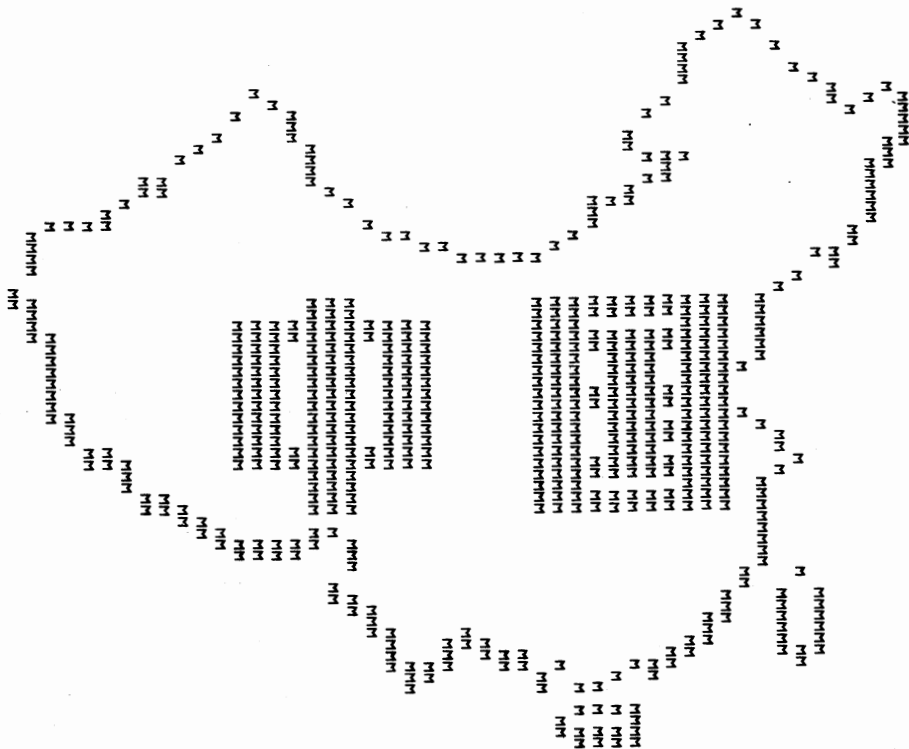


源程序及运行结果

```
10 READ M,N
20 IF M=-1 GOTO 90
30 FOR I=M TO M+N-1
40 LPRINTTAB(I);"M";
50 NEXT I
60 IF M<>0 GOTO10
70 LPRINT
80 GOTO 10
90 END
```

100 DATA 10,4,0,0,9,1,14,2,0,0,10,1,16,5,0,0
110 DATA 11,1,22,1,0,0,9,2,24,1,0,0
120 DATA 8,1,24,1,54,5,0,0,7,1,26,1,42,1
130 DATA 52,1,59,1,0,0
140 DATA 5,1,27,1,40,2,43,1,54,5,0,0
150 DATA 3,1,28,6,39,1,44,7,0,0
160 DATA 2,1,34,1,38,1,52,1,0,0
170 DATA 3,1,28,19,54,2,0,0
180 DATA 4,1,28,19,56,2,0,0
190 DATA 5,4,15,1,28,19,58,1,0,0
200 DATA 10,1,15,3,28,2,31,2,36,2
210 DATA 39,2,42,2,45,2,59,1,0,0
220 DATA 11,1,15,1,17,1,28,2
230 DATA 31,13,45,2,60,1,0,0
240 DATA 13,2,18,2,28,2,31,13,45,2,60,1
250 DATA 64,3,0,0
260 DATA 19,1,28,2,31,13,45,2,61,1,64,1,66,1,0,0
270 DATA 19,3,28,2,31,2,36,2,42,2,45,2
280 DATA 62,1,64,1,66,1,0,0
290 DATA 22,1,28,19,62,1,64,1,66,1,0,0
300 DATA 23,1,28,19,60,1,65,1,0,0
310 DATA 24,1,28,19,61,1,0,0,24,1,59,1,0,0
320 DATA 24,1,59,1,0,0,24,1,58,1,0,0,24,1,57,1,0,0
330 DATA 23,1,58,2,0,0
340 DATA 23,1,30,13,60,1,0,0
350 DATA 22,1,30,13,60,2,0,0
360 DATA 22,1,30,13,57,3,0,0
370 DATA 21,1,30,2,41,2,55,2,0,0
380 DATA 19,1,28,19,49,3,54,1,0,0
390 DATA 18,1,28,19,48,1,53,1,0,0
400 DATA 14,4,28,19,48,1,0,0
410 DATA 11,3,30,2,41,2,49,1,0,0
420 DATA 10,1,30,13,49,1,0,0
430 DATA 9,1,30,13,49,1,0,0
440 DATA 11,1,30,13,49,1,0,0
450 DATA 13,1,48,1,0,0,14,1,47,1,0,0,15,1,46,1,0,0
460 DATA 17,2,45,1,0,0
470 DATA 17,2,45,1,0,0
480 DATA 19,1,42,2,0,0,20,2,41,1,0,0
490 DATA 21,1,41,1,0,0,21,1,38,2,0,0
500 DATA 21,1,31,7,0,0,22,4,28,3,0,0,27,1,0,0,-1,-1

打印图形



二、巧打金字塔

王玮中

本程序语句不多，比较简单，打出的图案却很美观。它利用四重循环达到可控大小、层数的金字塔图案。其中输入X为打印层数，Y为每小“△”中打印行数。

关键是60语句中的TAB函数表达式，I为每一层最高点左边位置；J为小“△”每行起点位置；K为每层中打印“△”个数；2*Y为第二层开始每两“△”顶点相隔的位置数；A为每个小“△”中每行打印的个数。

源程序

```
10 INPUT X,Y
20 FOR I=1 TO X
30 FOR J=1 TO Y
40 FOR K=1 TO I
50 FOR A=1 TO 2*J-1
60 LPRINTTAB(40-I*Y-J+K*2*Y+A);"*";
70 NEXTA,K
80 LPRINT
90 NEXT J,I
100 END
```

运行结果

```
RUN  
? 4,4_
```

```
      *  
     ***  
    *****  
   *********  
  *         *  
 ***       ***  
*****   *****  
*****  
 *         *         *  
***       ***       ***  
*****   *****   *****  
*****  
 *         *         *         *  
***       ***       ***       ***  
*****   *****   *****   *****  
*****
```

三、打印三角形、四边形的通用程序

张冠三

打印图形是在程序设计中经常遇到的问题。下面介绍打印各种各样实体三角形、平行四边形和梯形等的通用程序。

程序设计思想与语句说明

①设计思想 对这些图案的分析可以看出，尽管这些图案千差万别，但归纳起来只有三点不同之处，即打印的总行数可以不同，每行的起始位置不同，以及每行打印的字符数目不同；同时这些图案还有相同之处，即各行打印字符数成等差数列，各行起始位置也成等差数列。所以，每行打印的数目是行数 N 的一次函数，每行起始位置也是 N 的一次函数。它的通式分别为 $(K - G) * (N - 1) + G$ 和 $(C - S) * (N - 1) + S$ ，其中 S 为第一行起始位置， G 为第一行打印数目， C 为第二行起始位置， K 为第二行打印数目。有了每行打印的起始位置和打印的数目，图形就不难打印了。程序1为有出格报警功能的通用程序。其中 M 为打印的总行数， V 为最末一行第一个字符的位置($V > 0$)， $V + (K + G) * (M - 1) + G$ 为末行的打印数目。

②语句说明

10~20语句，输入 S 、 G 、 C 、 K 、 M 。

40~100语句，出格报警；删除它们程序也能运行。

110~180语句，打印图形。其中130语句为确定打印起始位置，140语句为每行打印数目。

程序①及运行结果

```
10 PRINT "S,G,C,K,M,=";  
20 INPUT S,G,C,K,M  
40 V=(C-S)*(M-1)+S  
50 IF V>=0 THEN 80
```

```

60 PRINT"TAB(X);X<0"
70 GOTO20
80 IF V+(K-G)*(M-1)+G<=79 THEN 110
90 PRINT" TAB(X);X>80 "
100 GOTO 20
110 FOR N=1 TO M
130 PRINT TAB((C-S)*(N-1)+S);
140 FOR I=1 TO (K-G)*(N-1)+G
150 PRINT"*";
160 NEXT I
170 PRINT
180 NEXT N

```

RUN
 S,G,C,K,M,=? 20,1,19,3,7
 *

READY
 >_

程序②为同时打印两个图案的程序，其中U为左图的末行字符数，U₁为右图末行字符个数。程序请读者自行分析。

程序②及运行结果

```

10 PRINT"S,G,C,K,S1,G1,C1,K1,M,=";
20 INPUT S,G,C,K,S1,G1,C1,K1,M
40 V=(C-S)*(M-1)+S
50 V1=(C1-S1)*(M-1)+S1
60 U=(K-G)*(M-1)+G
70 U1=(K1-G1)*(M-1)+G1
80 IF V>=0 GOTO 110
90 PRINT"TAB(X);X<0"
100 GOTO 10
110 IF V1+U1<=79 GOTO 140
120 PRINT"TAB(X); X>79"
130 GOTO 10
140 IF V+U<V1 GOTO 170
150 PRINT "TAB(X);X1>X2"
160 GOTO 10
170 FOR N=1 TO M
180 PRINT TAB((C-S)*(N-1)+S);
190 FOR I=1 TO (K-G)*(N-1)+G
200 PRINT "*";
210 NEXT I
220 PRINT TAB((C1-S1)*(N-1)+S1);
230 FOR I=1 TO (K1-G1)*(N-1)+G1
240 PRINT "E";
250 NEXT I
260 PRINT
270 NEXT N

```

```

RUN
S,G,C,K,S1,G1,C1,K1,M,=? 10,1,9,3,30,1,29,3,5
*
***
*****
*****
*****
*****
E
EEE
EEEE
EEEEEE
EEEEEEEE
EEEEEEEE

```

READY
>_

四、两个两位数

郑德义

一个两位的正整数，如果它的个位与十位数字对调，则产生另一个两位正整数，我们称后者为前者的对调数。现给定一个两位的正整数，你能否找到另一个两位正整数，使得这两个两位正整数之和等于它们各自对调数之和？例如： $12 + (32) = (23) + 21$ 。你能不能把具有这一特征的一对对两位数都找出来？

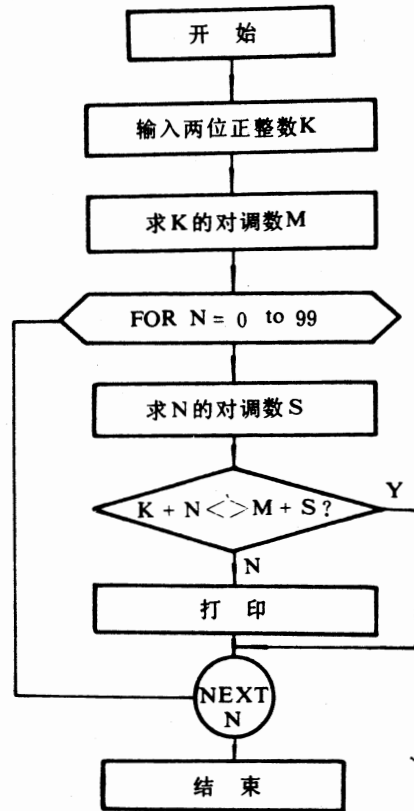
此题可以用代数方法求解，不过很难找到所有的解。若用计算机求解，它不但速度快，而且能找出所有满足条件的解。

程序设计思想与语句说明

首先要说明：对于00、01、02……09这10个数也是两位的正整数，它们也是在求解的范围之内。为了程序设计简单，在输出这10个数时用0、1、2……9来表示。

我们用试探法求解。在输入一个两位数K后，通过等式 $G = \text{INT}(K/10)$ 和 $H = K - G * 10$ 可以求出它的十位数和个

程序框图



程序及运行结果

```
10 INPUT K
20 G=INT(K/10)
30 H=K-G*10
35 M=H*10+G
40 FOR N=0 TO 99
50 I=INT(N/10)
60 J=N-I*10
65 T=I*10+J:S=J*10+I
70 IF K+N<>M+S THEN90
80 PRINT K;"+";T;"=";"S;"+";M
```

位数。把这两个数对调，即可得到K的对调数。

从0开始，用上面办法求出对调数，然后把它的对调数和K的对调数相加，再把K和0相加，比较二者是否相等，如相等，0即为K的解，不相等，则不是解；再用1、2、3……直到99去试探，直到求出所有的解。如果在试探中那个数是K的解，则把该数打印出来。

程序中，第10语句是输入任意两位正整数K。20~35语句是计算K的对调数。40~90语句是求K的解，其中70语句完成判断功能，80语句打印出K和它的解以及它们的对调数和的等式。

上面介绍了给出一个两位正整数，求另外的两位正整数，使它们的和等于它们对调数之和；如果给出一个两位的正整数，能否找到另一个两位的正整数，使两个正整数之积等于它们对调数之积呢？一共能找到多少个这样的正整数呢？请读者自己编一个程序解决上面的问题。

```
90 NEXT N
100 END
```

```
RUN
? 13
13 +( 20 )=( 2 )+ 31
13 +( 31 )=( 13 )+ 31
13 +( 42 )=( 24 )+ 31
13 +( 53 )=( 35 )+ 31
13 +( 64 )=( 46 )+ 31
13 +( 75 )=( 57 )+ 31
13 +( 86 )=( 68 )+ 31
13 +( 97 )=( 79 )+ 31
```

READY

>

```
RUN
? 19
19 +( 80 )=( 8 )+ 91
19 +( 91 )=( 19 )+ 91
```

READY

>

```
RUN
? 56
56 +( 10 )=( 1 )+ 65
56 +( 21 )=( 12 )+ 65
56 +( 32 )=( 23 )+ 65
56 +( 43 )=( 34 )+ 65
56 +( 54 )=( 45 )+ 65
56 +( 65 )=( 56 )+ 65
56 +( 76 )=( 67 )+ 65
56 +( 87 )=( 78 )+ 65
56 +( 98 )=( 89 )+ 65
```

READY

>

五、巧取糖果

吕元长

今有36颗糖，分成8堆，围成一个正方形，使其每边的糖果总数为10，如图1所示（用具体数字表示每堆实际的糖数）。你能从中取出一些糖果而又使其每边的糖果总数不变（仍为10）吗？或者，你能从中取出两颗，而保持其每边总数为10吗？你还能在留下的方形中再取出两颗，仍保持每边的糖果总数不变吗？你能如此反复进行，每次取走两颗后，均保持正方形每边的糖果总数一直不变吗？一共可以进行几次呢？

在家庭晚会上，你不妨设下这样一个糖果方阵，谁能按题意取出糖果，就可以吃到糖果，这无疑会给你的晚会增添欢乐的气氛。

当然，你可以设下每边总数为5、8或者13等等的任意数目的任一糖果方阵，进行上述游戏。

1	8	1
8	8	8
1	8	1

图1

利用计算机可以打印出这样的方阵来，并打印每次取走两颗后，仍保持原方阵每边总数不变的一个方阵。在计算机打印出“？”时，你只要键入你想设下的糖果方阵每边的总数，计算机就会立即输出一系列方阵，每一方阵的糖果总数依次少2，这些方阵均能满足每边的糖果总数为要求的值。若是谁对这个游戏不太熟悉，没把握取对糖果，可以请教计算机，保证让他吃上糖果。例如，在问号后打入“5”，表示希望得到每边总数为5的糖果方阵，计算机立即输出这样的方阵3个（见图2）。你可以按第一种形式（糖果总数最多）布下糖果方阵，那么最终将有两人可以巧取并吃上糖果。同样可以打印出每边总数为10、11或者其它任意数目的任一系列糖果方阵来。

程序设计思想及说明

从图2可以看出，任一系列糖果方阵的排列有以下规律：

①第一个方阵每边中间一堆的数目为每边总数减2，四个角上的数目都为1。

②第二个方阵每边中间一堆的数目比第一个方阵中的少1，左上角和右下角的数目比第一个方阵中的多1，左下角和右上角的数目不变。

③第三个方阵每边中间一堆的数目比第二个方阵中的少1，左上角和右下角的数目不变，左下角和右上角的数目比第二个方阵中的多1。

依此类推，以下方阵依次按每边中间一堆的数目比上一相邻方阵中的少1，左上角和右下角或者左下角和右上角则交替增1组成。直到输出一个每边中间一堆的数目为1时所组成的方阵为止。本程序即是根据以上规律（当然还有其它取法，请读者自己分析）编制的巧取糖果的程序。此程序可以打印每边总数为3（每边三堆，每堆至少放一颗）以上的任一系列的方阵。

程序中变量A表示方阵每边中间一堆的糖果颗数，B表示左上角及右下角的糖果颗数，C表示左下角和右上角的糖果颗数。打印输出第一个方阵后，A依次减1，B和C则交替增1（当H为偶数时B增1，H为奇数时C增1）。程序中270语句即起这样的作用。

8堆糖果可看作排列成一个3行3列的二维数组。用其下标可以排列成图3。可见，行和列下标之差的绝对值等于0时，正好可以表示左上角和右下角；等于1时，可以表示方阵每边的中间位置；等于2时，可以表示左下角和右上角。程序中90、100语句就是根据这一道理来判断出方阵中三种不同位置的。当输出打印方阵正中（2行2列位置）时，只需输出5个空格即可。

源程序及运行结果

```

10 DIMM(3,3)
20 INPUT X
25 PRINT
30 A=X-2
40 B=1
45 C=1
50 H=1
60 FOR I=1 TO 3
70 FOR J=1 TO 3
80 IF I=2 THEN 180
90 IF ABS(I-J)=1 THEN 150
100 IF ABS(I-J)=2 THEN 130
110 M(I,J)=B
120 GOTO 160
130 M(I,J)=C
140 GOTO 160
150 M(I,J)=A
160 PRINT M(I,J); " ";

```

RUN

? 5

1 3 1

3 3

1 3 1

2 2 1

2 2

1 2 2

2 1 2

1 1

2 1 2

图2

11 12 13

21 22 23

31 32 33

图3


```

170 GOTO 210
180 IF J=2 THEN 200
190 GOTO 90
200 PRINT"      ";
210 NEXT J
220 PRINT
230 NEXT I
232 PRINT
235 PRINT
240 A=A-1
250 IF A=0 THEN 320
260 H=H+1
270 IF INT(H/2)=H/2 THEN 300
280 C=C+1
290 GOTO 60
300 B=B+1
310 GOTO 60
320 END

```

RUN

? 5_

1	3	1
3		3
1	3	1
2	2	1
2		2
1	2	2
2	1	2
1		1
2	1	2

RUN

? 8_

1	6	1	3	3	2
6		6	3		3
1	6	1	2	3	3
2	5	1	3	2	3
5		5	2		2
1	5	2	3	2	3
2	4	2	4	1	3
4		4	1		1
2	4	2	3	1	4

六、求重复数的平方

邓 丽

数具有许多奇特的性质。有些特性能加深人们的记忆，而有的却能帮助人们完成那些难以求解的计算。

我们常见到这样的数字，如8888，999999……9等等，我们称这样的数字为重复数，即每位数字均由整数N组成的M位数字，称为N的M位重复数。这里N为1~9的任一整数，其中 $M \leq 100$

这里介绍利用重复数的特性，在计算机上求重复数的平方的方法。用这种方法，求任一指定的N的M位重复数的平方既快又准确，读者不妨试试。

图1、图2分别为求 3333^2 ($N=3, M=4$) 和 777^2 ($N=7, M=3$) 的示意图。

$$\begin{array}{r}
 9 \\
 999 \\
 99999 \\
 9999999 \\
 \hline
 11108889
 \end{array}$$

图 1

$$\begin{array}{r}
 49 \\
 4949 \\
 494949 \\
 4949 \\
 49 \\
 \hline
 603729
 \end{array}$$

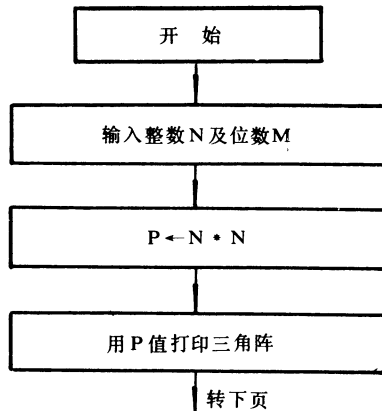
图 2

你看出其中的奥妙了吗？如果 $N \leq 3$ ，则用 N^2 的值排成三角形阵，否则用 N^2 的值排成菱形阵。然后按加法竖式进行加法运算，得到的和就是所求的结果。

程序及变量说明

100~140语句，输入N和M；打印题目
赋值；
150~290语句，打印图案。其中，150~220语句打印三角形阵，240~290语句是当 $N > 3$ 时，打印倒三角形阵，从而形成菱形阵；

程序框图



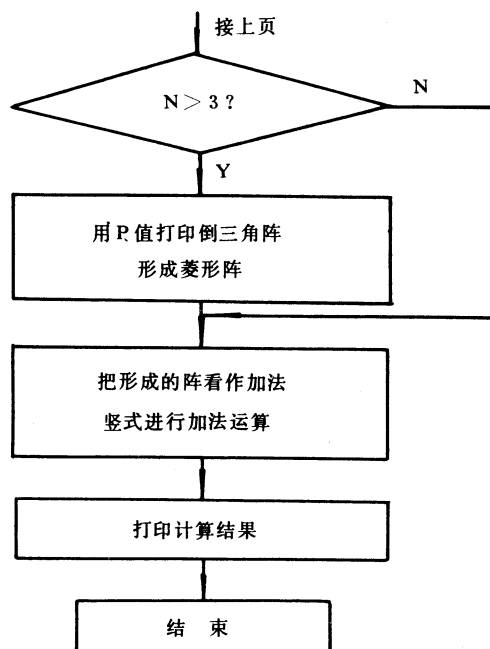
350~490语句, 计算重复数平方的值;

500~540语句, 输出结果;

560~590语句, 打印横断线的子程序。

A数组存阵中各元素; B数组表示各位的和; C为进位值; T表示阵中最多的一行数的个数。计算时, 如果 $N > 3$, 则P、Q分别表示 N^2 的个位和十位。

注: 本程序在APPLE-II微型机上通过, 用其它机器输出格式要相应地改变。



源程序及运行结果

```
100 INPUT N,M
110 PRINTTAB( INT((M+1)/2)+1);
120 FOR I=1 TO M :PRINT N;:NEXT I:PRINT"[2":GOSUB560
130 LETP=N*N
140 LET T=2*M-1
150 FOR I=1 TO M
160 LET S=I
170 IF N<=3 THEN S=2*I-1
180 PRINTTAB(M-I+2);
190 FOR J=1 TO S
200 PRINT P;
210 NEXT J:PRINT
220 NEXT I
230 IF N<=3 THEN 330
235 IF M=1 THEN 300
240 FOR I=M-1 TO 1 STEP -1
250 PRINTTAB(M-I+2);
260 FOR J=1 TO I
270 PRINTF;
280 NEXT J:PRINT
290 NEXT I
300 LET Q=INT(P/10)
310 LET P=P-10*Q
320 LET T=T+1
330 GOSUB 560
340 DIM A(M,T),B(T):C=0
350 FOR J=T TO 1 STEP -1
360 LET S=M-J+1
370 IF J<=M THEN400
380 LET S=J-M
390 IF N<=3 THEN S=S+1
400 FOR I=S TO M
```



```

-----
308641975308641358024691358025

JRUN
?2
??10
      222222222^2
-----
          4
         444
        44444
       4444444
      444444444
     44444444444
    4444444444444
   444444444444444
  44444444444444444
 4444444444444444444
44444444444444444444
-----
4938271603950617284

```

七、数字电子钟

田松林

这是一个模拟电子钟的程序，它不但能以数字形式显示时(H = 0 ~ 23)分(M = 0 ~ 59)秒(S = 0 ~ 59)，而且还能响铃报时。

程序设计思想与语句说明

① 设计思想

电子钟程序的关键是建立一个比较准确的秒节拍和使不断变化的显示内容出现在CRT的某一固定位置上。由于电子计算机的时钟频率是相当稳定的，因而它执行某一功能的时间也相对不变，这就给我们提供了获得相对准确的时间节拍的可能。用空循环语句

```

150 FOR I = 1 TO 280
155 NEXT

```

来调整走时精度，只要适当选取循环变量的终值（这里取280），总可以使秒循环部份的程序运行一次之后，恰好是一秒钟（或者误差最小）。

有了秒发生器，通过循环，得到分、小时的时间就不困难了。

② 语句说明

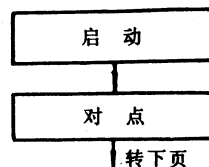
20~70语句是显示提示信息。

80~110语句是校准时钟。键入某一时刻的时间时，不妨先拨快一分，待时间一到，键入“0”，则启动时钟程序运行。

140~144语句是显示H、M、S三个大字。

150~360语句是计时部份程序。其中175~182是在大字H、

程序框图



M、S 上方显示时、分、秒的程序。此处所以用POKE 语句是因为防止字符向上滚动（如果用@的打印格式，每显示一次时、分、秒的值，整个屏幕都向上滚动一次）。把时分秒的数值用POKE 语句送到显示单元中去，必须事先把H、M、S 的值的个位、十位值分别取出，再转换成ASCII 码，送到内存视频 RAM 的相应单元中去，才能显示正确无误。175~182 语句即是完成此功能的。

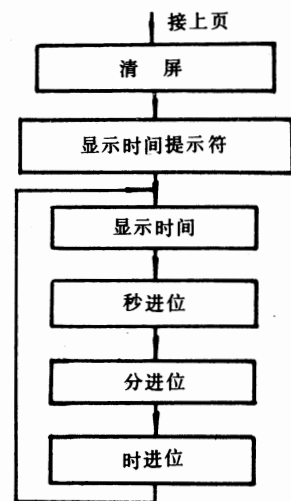
280~290 语句是秒时钟定时程序。

源程序及运行结果

```

20 PRINT"                ELECTRONIC    CLOCK"
30 PRINT"                -----"
40 PRINT:PRINT:PRINT
50 PRINT"                HOW    TO    USE"
60 PRINT
70 PRINT"                FIRST SET THE CLOCK(TIME: H=0-23 M=0-59)
,THEN RUN IT"
80 PRINT:INPUT"                THIS TIME IS  H=";B
90 INPUT"                M=";C
100 PRINT"                S=0"
110 PRINT:INPUT"                READY?    HIT A AND RETURN WHEN
    READY----";A$
120 IF A$<>"0" THEN 80
130 CLS
140 PRINT@456,"                II    II                III III                IIIIII
I"
141 PRINT@520,"                II    II                IIIIIII                II"
142 PRINT@584,"                IIIIIII                II I II                IIIIII
I"
143 PRINT@648,"                II    II                II    II                I
I"
144 PRINT@712,"                II    II                II    II                IIIIII
I"
150 FOR H=B TO 23
160 FOR M=C TO 59
170 FOR S=0 TO 59
175 H$=RIGHT$(STR$(H),2):H1$=LEFT$(H$,1):H2$=MID$(H$,2
,1):M$=RIGHT$(STR$(M),2):M1$=LEFT$(M$,1):M2$=MID$(M$,2
,1):S$=RIGHT$(STR$(S),2):S1$=LEFT$(S$,1):S2$=MID$(S$,2
,1)
180 POKE 15760,72:POKE15761,61:U=VAL(H1$)+48:POKE 1576
2,U:Y=VAL(H2$)+48:POKE 15763,Y:POKE 15773,77:POKE 1577
4,61:Y=VAL(M1$)+48:POKE15775,Y:U=VAL(M2$)+48:POKE15776

```



```

,U
182 POKE 15786,83:POKE 15787,61:U=VAL(S1$)+48:POKE1578
8,U:Y=VAL(S2$)+48:POKE 15789,Y
280 FOR I=1 TO 280
290 NEXT
310 NEXT S
320 NEXT M
330 C=0
340 NEXT H
350 B=0
360 GOTO 150
370 END

```

READY
>RUN

ELECTRONIC CLOCK

HOW TO USE

FIRST SET THE CLOCK(TIME: H=0-23 M=0-59), THEN RUN IT

THIS TIME IS H=? 23
M=? 59
S=0

READY? HIT A AND RETURN WHEN READY----? 0_

H=23		M=59		S=05	
II	II	III	III	IIIIIIII	
II	II	IIIIIIII		II	
IIIIIIII		II I II		IIIIIIII	
II	II	II	II		II
II	II	II	II	IIIIIIII	

八、炮击来敌

陈 炜

程序运行后，先输入入侵者个数及进攻速度。入侵者越多，速度越快，游戏难度越大。按“ENTER”键游戏开始。

游戏者用“←”和“→”键控制大炮左右移动，空格键开炮。

游戏中，游戏者必须在来敌到达炮台前将他们全部消灭，否则就作“失败”。游戏者每次只能开一炮，不能连续开炮。当敌人被全部消灭或敌人到达炮台后，游戏就结束。计算机打印出战斗结果。若重新开始，按“Y”键即可。

程序语句说明

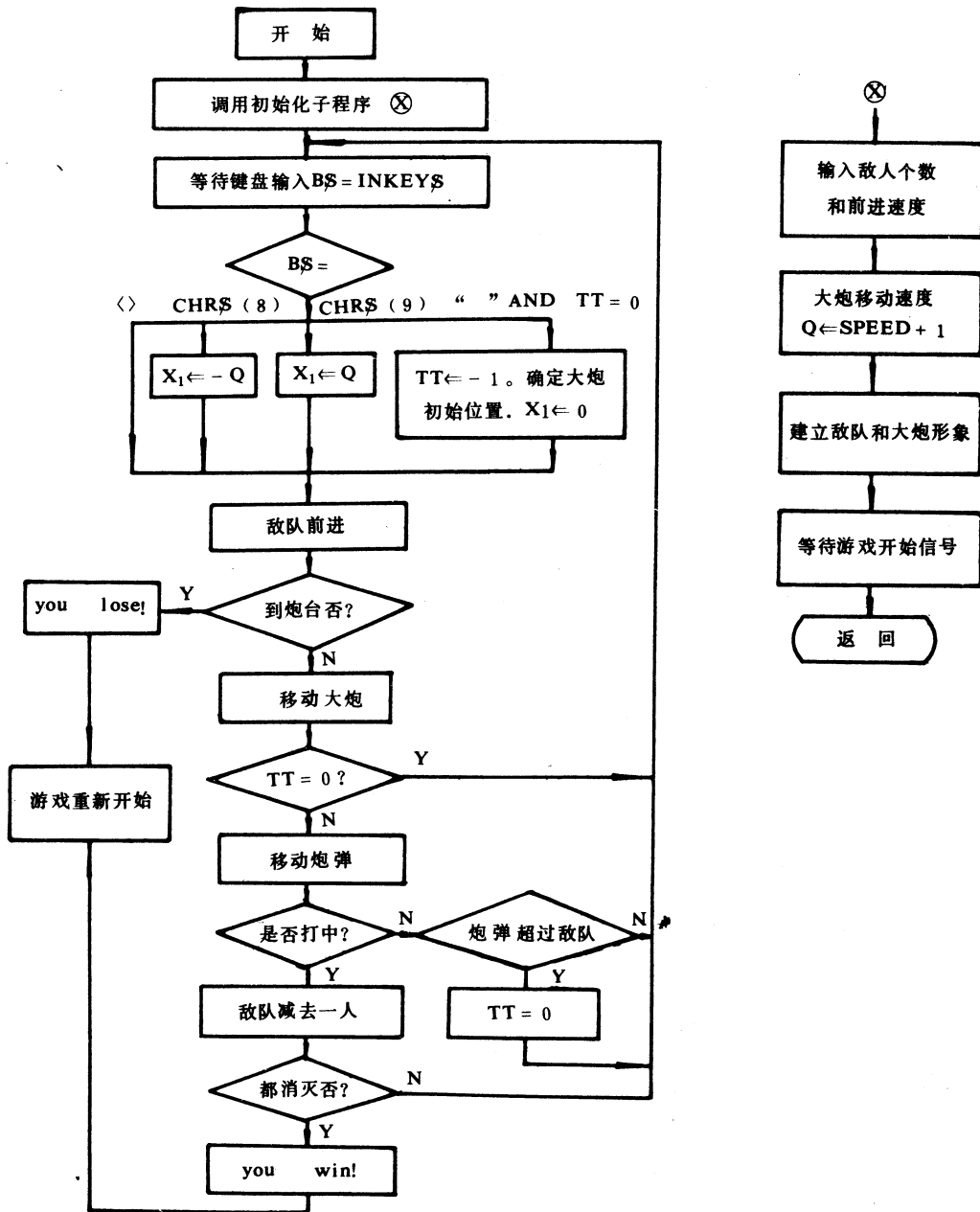
20~100行, 完成敌队移动和大炮移动。

110~170行, 处理炮弹移动和判断是否打中。

200~270行, 程序初始化子程序。

300~330行, 打印战斗结果。

程序框图



源程序

```
10 CLEAR 500 :GOSUB 200
20 B$=INKEY$
30 IF B$=CHR$(8) THEN X1=-0
40 IF B$=CHR$(9) THEN X1=0
50 IF B$=" " AND TT=0 THEN TT=-1 :Y=X+1:X1=0:FOR L=1 TO
5: OUT 255,4:OUT 255,3:NEXT
60 PRINT@T,S$;:T=T+SPEED: IF T<959 THEN PRINT@T,A$;ELS
E300
70 PRINT@X," ";X=X+X1
80 IF X<962 THEN X=X+58
90 IF X>1020 THEN X=X-58
100 PRINT@X,D$;
110 IF TT=0 THEN 20
120 PRINT@Y," ";Y=Y-64:IF PEEK(15360+Y)=42 THEN 150
130 IF Y<T THEN TT=0 : GOTO 20
140 PRINT@Y,". ";:GOTO 20
150 U=U+1:IF U=NN THEN 310
160 A$=LEFT$(A$,Y-T)+" "+RIGHT$(A$,NN-Y+T-1)
170 TT=0:GOTO20
200 CLS:PRINT"PLEASE INPUT THE NUMBER OF THE ENEMIES (
10--30). "
210 INPUT NN :IF NN<10 OR NN>30 THEN 200
220 CLS:PRINT"PLEASE INPUT THE SPEED OF THE ENEMIES (1
--3). "
230 INPUT SPEED : IF SPEED<1 OR SPEED>3 THEN 220
240 D=SPEED+1 :FOR I=1 TO NN:A$=A$+"*":NEXT :FOR I=1 T
O SPEED:S$=S$+" ":NEXT
250 PRINT"PLEASE HIT 'ENTER' TO START PLAY."
260 X=1000:D$=CHR$(190)+CHR$(143)+CHR$(189)
270 B$=INKEY$:IFB$<>CHR$(13) THEN270 ELSE CLS:RETURN
300 CLS:PRINT"YOU LOSE!":GOTO320
310 CLS:PRINT"YOU WIN!"
320 PRINT"DO YOU WANT TO PLAY AGAIN?(Y OR N)"
330 R$=INKEY$:IF R$="Y" THEN RUN ELSE IF R$<>"N" THEN3
30
```

运行结果

RUN_

PLEASE INPUT THE NUMBER OF THE ENEMIES (10--30).

? 15_

PLEASE INPUT THE SPEED OF THE ENEMIES (1--3).

? 2

PLEASE HIT 'ENTER' TO START PLAY.

九、封锁游戏

杨宁海

这是个趣味浓厚的游戏程序。游戏的双方可利用各自的四个方位键控制封锁线建立方向，使其不被堵截而得分。游戏中可提高人们的智力和判断能力。

游戏方法

程序开始执行时先将游戏者姓名输入。然后选择封锁线建立的速度。这个速度由数字控制，数字越小速度越快。开始可输入10，待熟练以后，可逐渐减小数字，提高游戏速度。速度确定后，游戏就可以开始了。用空格键作为游戏开始的命令键，如不按空格键，则大约半分钟后自动开始。

右边的游戏者使用“O”、“.”、“K”、“;”四个键，分别表示向上、下、左、右移动；左边的游戏者使用“W”、“X”、“A”、“D”四个键，分别表示向上、下、左、右移动。通过对这些键的控制来确定各自的封锁线建立方向。

比赛中，谁的封锁线前进道路被堵截，谁就输一局，胜者得一分。若在同一线上，想返回是不行的，因为倒退的路已被自己堵住，所以倒退则判失败。前进路线只能是空白区。当任何一方的分数累计达到10分时，就宣布比赛结束，并再次公布比赛结果。如果要继续玩，则回答Y（或Yes），即可重新开始，否则结束。

程序设计思想与语句说明

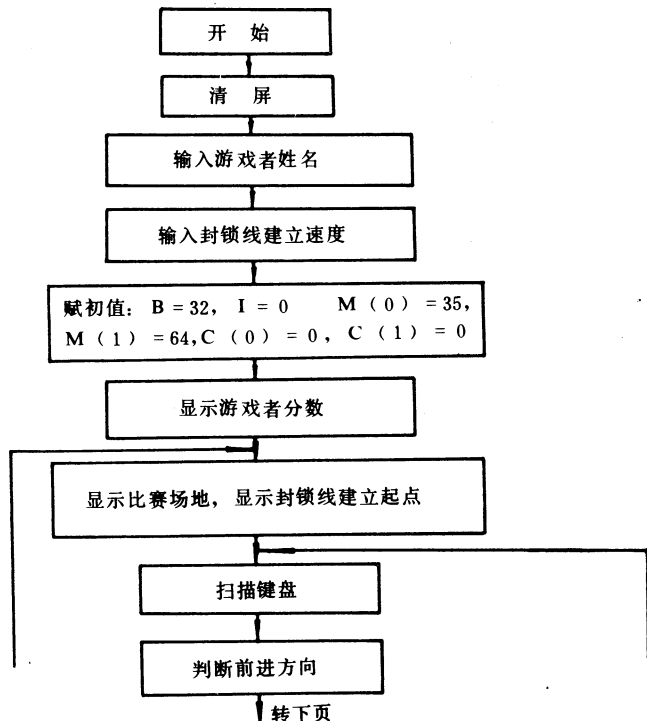
① 设计思想

游戏双方均设置4个方向键，控制前进方向。每前进一步，通过POKE语句向显示RAM相应单元送一个#（或@）代码，显示器上显示出井（或@）。用PEEK函数探测前进方向的内存显示单元是否为空。如为空，则向该单元送代码；不为空说明被封锁，转去显示封锁点。

显示封锁点采用方法基本同上。组成一个循环，当本次循环该点为空时，则送#（或@）代码以显示#（或@）。如不为空则送空格代码。下次循环和本次正好相反，因此可得到不断闪动的效果。

② 语句说明

程序框图



20~70行是请求输入姓名和速度，如姓名字母超过五个，则只取前五个字母；如果速度数字大于10或小于1，则需重新输入，其值应在1~10之间。

100~170行是显示记分牌和比赛场地。

180~190行是显示游戏者建立封锁线的起点。

200~220行是等待发布开始命令。

240~300行是判断行走方向。

310~320行是判断是否被封锁。

330~380行是封锁后的处理。

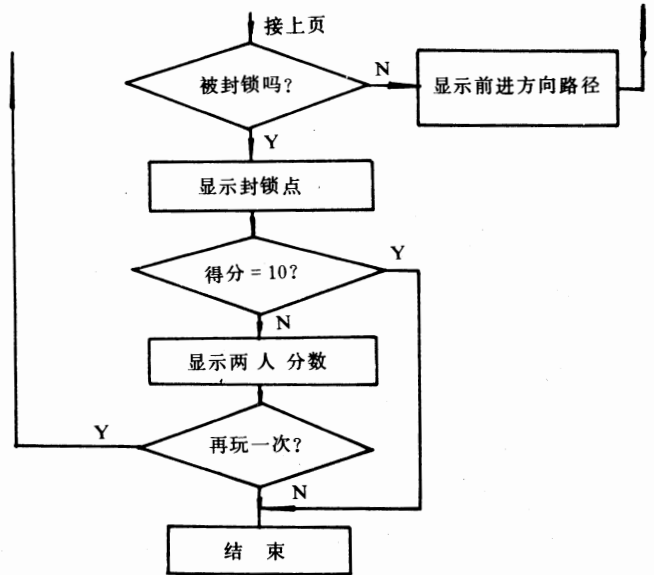
390~440行是游戏结束后的处理。

源程序及运行结果

```

10 REM THIS IS A PLAYER PROGRAM
20 CLS
30 INPUT "YOUR NAME : ";A$,B$
40 IF LEN(A$)>4 THEN A$=LEFT$(A$,5)
50 IF LEN(B$)>4 THEN B$=LEFT$(B$,5)
60 INPUT "SPEED FROM 1 TO 10";S
70 IF S>10 OR S<1 THEN 60
80 B=32 : I=0
90 M(0)=35 : M(1)=64 : C(0)=0 : C(1)=0
100 CLS
110 PRINTA$:" ";C(0);TAB(55)B$:" ";C(1);
120 FOR X=3 TO 125
130 SET(X,4):SET(X,45)
140 NEXT X
150 FOR Y=4 TO 45
160 SET(2,Y):SET(3,Y):SET(124,Y):SET(125,Y)
170 NEXT Y
180 X(0)=15628:X(1)=16176:D(0)=64:D(1)=-64
190 FOR J=0 TO 1:POKE X(J),M(J):POKE X(J)+1,M(J):NEXT
J
200 FOR J=1 TO 500:Y$=INKEY$
210 IF Y$=" " THEN 230
220 NEXT J
230 I=1-I
240 FOR J=1 TO S
250 D$=INKEY$
260 IF D$="D" THEN D(0)=2 ELSE IF D$="W" THEN D(0)=-64

```



```

270 IF D$=":" THEN D(1)=2 ELSE IF D$="0" THEN D(1)=-64
280 IF D$="A" THEN D(0)=-2 ELSE IF D$="X" THEN D(0)=64
290 IF D$="K" THEN D(1)=-2 ELSE IF D$="." THEN D(1)=64
300 NEXT J
310 X(I)=X(I)+D(I)
320 IF PEEK(X(I))= B THEN POKE X(I),M(I):POKE X(I)+1,M
(I):GOTO 230
330 C(1-I)=C(1-I)+1:T=X(I)-D(I)
340 FOR K=1 TO 20
350 FOR KK=1 TO 5
360 IF PEEK(T)=B THEN POKE T,M(I):POKE T+1,M(I) ELSE P
OKE T,B:POKE T+1,B
370 NEXT KK,K
380 IF C(1-I)= 10 THEN 440
390 CLS
400 PRINT @142,A$":":C(0),B$":":C(1):IF C(0)=10 OR C(1)
=10 THEN 440
410 PRINT @465,"SHALL I CONTINUE (Y/N)":
420 INPUT W$
430 IF W$="Y" OR W$="YES" THEN 100
440 CLS :PRINT@660,"G O O D B Y !":END

```

RUN_

YOUR NAME :? AA

?? BB

SPEED FROM 1 TO 10? 8_

AA: 0

BB: 0

```

.....
*      @@      *
*      @@      *
*      @@      *
*      @@      *
*      @@      *
*
*                                     ##
*                                     ##
*                                     ##
*                                     ##
*
.....

```

十、《打印体力、情绪和智力曲线图》

邓高湖

体力、情绪和智力曲线图又叫生物节奏曲线图。关于这个问题的详细讨论，请参阅《科学画报》81年第2期上陈庚同志的“揭开生物节奏的秘密—体力、情绪和智力盛衰的规律”一文。这里仅对这个问题作一简单的介绍。

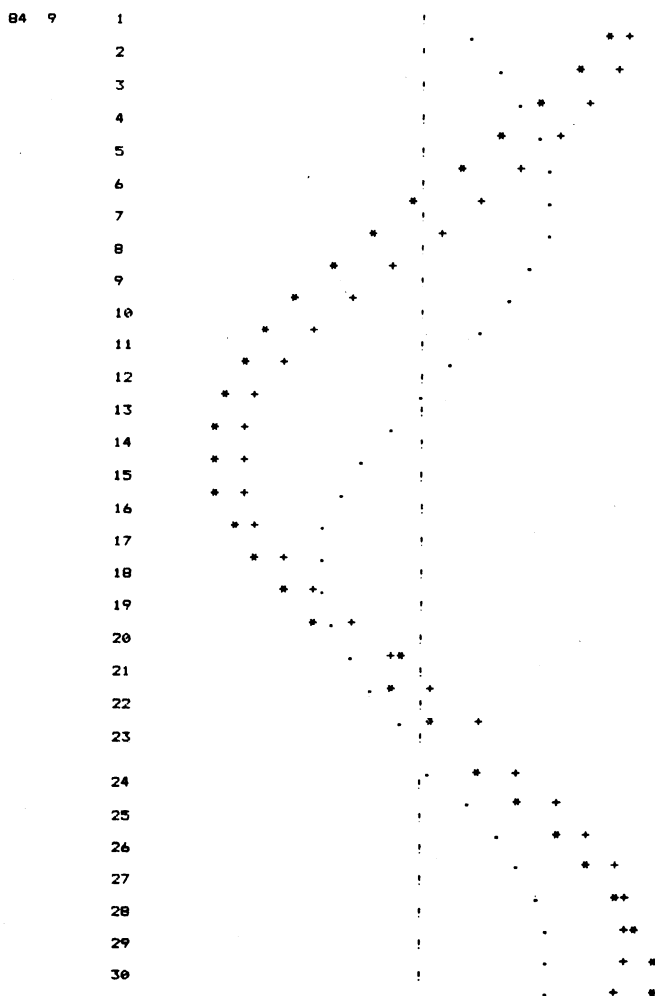
为了揭示生物节奏的奥秘，西方一些学者经过长期观察与研究，发现每个人从他诞生之日起，直到生命的终结，都存在着周期分别为23天、28天和33天的体力、情绪和智力曲线图的变化规律，把这三种变化形象地描绘在同一坐标系中，就形成了一幅直观的体力、情绪和智力周期变化曲线图。

下图就是用计算机打印的曲线图，它反映了一个出生于60年6月3日的人，在84年9月份的生物节奏变化情况。图中的点线为体力曲线，“十”线为情绪曲线，“*”线为智力曲线。左边的数字代表日期。当某条曲线处于中线的右边时，对应的日子称为该曲线的“高潮期”。在这样的日子里，此人在体力上感到体力充沛，生机勃勃，在情绪上觉得心情愉快；在智力上，表现出思维敏捷。当某条曲线处于中线左边时，对应的日子称为该曲线的“低潮期”。在这样的日子里，体力上易疲劳，怠劲；情绪上烦躁不安，意志沮丧；在智力方面出现注意力不易集中，反应较慢等现象。跨越中线的日子称为“临界期”。在临界期中，极易出现差错，粗枝大叶，容易染病，机体各方面的协调功能较差，容易发

源程序

```
RUN
? 60,6,3
? 84,9
? 1
```

FUHAO: (.)--TILI: (+)--QINGXU: (*)--ZHILI.



生事故等等。

人体的生物节奏图是这样的有规律，但是它并不能用来预卜吉凶，而只是提示人们，在某段日子里可能出现的体力、情绪和智力的趋向而已。借助它，你可以及时掌握自己的生物节奏变化情况，以便更好安排工作、学习和生活，即使处于“临界期”或“低潮期”，也完全可以用坚强的意志和毅力加以克服。

由于用手工绘制一张这样的曲线图，既麻烦又费工，且不易画准确，使用计算机却很容易办到，运行下面这个程序，你只需输入出生年月日，打印起始年月，以及打印多少个月份，机器即可为你打印一张满意的生物节奏曲线图。

源程序

```
10 DIMM(12),T(12),A$(49)
20 FOR I=1 TO 12
30 READ T(I),M(I)
40 NEXT I
50 INPUT NO,YO,RO
60 INPUT N1,Y1
70 INPUT N
80 LET SO=T(YO)+RO
90 LET S1=T(Y1)
100 LET S=(N1-NO)*365+S1-SO
110 LET L=NO
120 LET K=N1
130 IF SO<=60 LET L=NO-1
140 IF S1<59 LET K=N1-1
150 LET S=S+INT(K/4)-INT(L/4)
155 LPRINT"FUHAD:(.)--TILI:(+)--QINGXU:(*)--ZHILI."
157 LPRINT:LPRINT
160 IF N1/4=INT(N1/4) LET M(2)=29
170 FOR J=1 TO N
180 LET D=M(Y1)
190 FOR I=1 TO D
200 LET A$=""
"
202 IF I=1 LPRINT N1;Y1;
204 LPRINT TAB(11);I;TAB(15);A$
210 LET X=(S+I)*2*3.14159
220 LET A=INT(45+12*SIN(X/23))
230 LET B=INT(45+20*SIN(X/28))
240 LET C=INT(45+23*SIN(X/33))
250 IF A<B AND B<C GOTO290
260 IFA<C AND C<B GOTO291
270 IF B<A AND A<C GOTO292
271 IF B<C AND C<A GOTO293
272 IF C<A AND A<B GOTO294
273 IF C<B AND B<A GOTO 295
280 IF I=1 THEN PRINTN1;Y1;:GOTO300
290 LPRINT TAB(A)". ";TAB(B)"+";TAB(C)"*":GOTO300
291 LPRINT TAB(A)". ";TAB(C)"*";TAB(B)"+":GOTO300
292 LPRINT TAB(B)"+";TAB(A)". ";TAB(C)"*":GOTO300
293 LPRINT TAB(B)"+";TAB(C)"*";TAB(A)".":GOTO300
294 LPRINT TAB(C)"*";TAB(A)". ";TAB(B)"+":GOTO300
```

```

295 LPRINT TAB(C)"*"; TAB(B)"+";TAB(A)".
300 NEXT I
305 LET S=S+D
310 LET Y1=Y1+1
320 IF Y<13 THEN GOTO 380
330 LET Y1=1
340 LET N1=N1+1
350 LET N=N-J
360 LET M(2)=28
370 GOTO 160
380 NEXT J
390 DATA 0, 31, 31, 28, 59, 31, 90, 30, 120, 31, 151, 30, 181
395 DATA 31, 212, 31, 243, 30, 273, 31, 304, 30, 334, 31
400 END

```

该程序的实现思想是这样的：在程序的一开始，先将平年中各月的天数和截止月之前的年内累计天数读入数组M(12)和T(12)。接着自键盘上接收出生年、月、日，打印起始年月和需打印月数。在计算从出生那天起，到欲打印的第一天为止的总天数时，先按平年算出总天数，再加上所经历的2月29日的个数。然后将总天数逐天加1，分别乘以 $2\pi/23$ 、 $2\pi/28$ 和 $2\pi/33$ 作为SIN函数的自变量，求出对应的体力、情绪和智力曲线在这一天应处的位置，由于打印机只能单向打印，程序中250~273行用了一系列判断语句用来判断A、B、C的值、找出三者的大小顺序，以便在290~295行打印相对应的符号。

十一、电脑医生

刘路政

计算机诊断属于人工智能中的专家系统领域，目前国内外已经出现了比较成功的系统，比如中医诊断程序等。本文介绍一个简单的“电脑医生”程序，它可以根据不同的“症状”，诊断出16种疾病，并开出“处方”。

程序运行方法

在打入RUN后，“患者”即可根据电脑医生的要求输入姓名和体温。然后电脑医生列出一张“症状清单”，请“患者”从中找出自己所患的症状并输入。若症状不止一个，可同时输入几个症状，例如输入“1245”，表示同时有头痛、咳嗽、发冷、恶心等症状。电脑医生根据输入的症状和体温即可诊断出“患者”得了什么病，并开出处方。有时“患者”输入的症状组合太奇怪，电脑医生可能诊断不出他得了什么病，此时它便显示出“很抱歉，这病我看不了，请另请医生”。在输入症状时，不一定要按顺序输入，例如“1245”和“5214”表示同样的症状组合，电脑医生会得出同样的诊断结果。但是，同一个症状只能输入一次，例如，若输入“11245”电脑医生就无法诊断了。另外，如果只是有点低烧，则可以只输入体温，用输入“0”表示没有其它症状，此时电脑医生会诊断出“患者”得了低烧，开点解热止痛片。

程序说明

语句120将症状组合送入变量C中。语句140—170将变量C中的症状组合拆开，分别送入A(0)~A(4)中去。例如输入症状号码为“5412”，经过执行语句140~170后A(0)~A(4)中分别是“0”，“5”，“4”，“1”，“2”。语句180~240将数组元素A(0)~A(4)中的症状代码按其数值大小进行排队。语句250将A(0)~A(4)再重新拼成变量C。按上

例，此时C中内容为“1245”。语句123是为了在以上拆、拼过程中产生舍入误差而设置的。语句270~380用来进行诊断。在数组B、T中分别存放有症状组合和每种症状组合所对应的体温下限和上限。在诊断时，首先用C中的症状组合与B中的每种症状组合进行比较，若C与B中的某元素相等，则再用体温T1与T中对应的体温上、下限比较，若T1在此上、下限内，则找出对应的诊断结果（病名）及处方，并显示出来。若C与所有B中的症状组合均不符合，或虽符合而体温不在其对应的上下限内，则认为该病电脑医生无法诊断。

此程序是在汉字机上实现的，若读者使用不带汉字的机器，可把程序中的所有汉语换成相应的英语，或者用汉语拼音，不过使用起来就不方便了。

源程序

```

READ:
LIST
10 DIM A(4), B(15), T(15, 2)
20 PRINT "您好！我是电脑医生。您叫什么名字？";
30 INPUT NY
40 PRINT NY; "同志，您的体温多少？";
50 INPUT T1
60 PRINT "请问您有哪些症状。您可选择下列症状前的号码，例如 1 2
45 ) ? "
70 PRINT " 1. 头痛           2. 咳嗽 "
80 PRINT " 3. 腹痛           4. 发冷 "
90 PRINT " 5. 恶心           6. 拉肚子 "
100 PRINT " 7. 厌食          8. 呕吐 "
110 PRINT " 9. 疲乏无力 "
120 INPUT C
123 C=C+0.01
130 FOR I=0 TO 4
140 I1=4-I
150 A(I)=INT(C/10(I1))
160 C=C-A(I)*10(I1)
170 NEXT I
180 FOR I=0 TO 3
190 FOR J=I+1 TO 4
200 IF A(I)=A(J) THEN 230
220 X=A(I); A(I)=A(J); A(J)=X
230 NEXT J

```



```

240 NEXT I
250 C=A(0)*10000+A(1)*1000+A(2)*100+A(3)+10+A(4)
260 RESTORE
270 FOR I=0 TO 15
280 READ B(I),T(I,1),T(I,2)
290 NEXT I
300 FOR I=0 TO 15
310 IF ABS(C-B(I))>0.1 THEN 380
320 IF T1>T(I,1) THEN 380
330 IF T2>T(I,2) THEN 380
340 FOR J=0 TO I:READ IX,DY:NEXT J
350 PRINT NY;" 同志,您得了 "IX
360 PRINT"          处          方"
370 PRINT DY:END
380 NEXT I
390 PRINT" 很抱歉,这病找着不了,请另请医生看"
500 DATA 12,37,38,14,38,43,145,36,43,124,38,43,1245,38,43,24,37,
43,36,36,5,37,5789,36,5,40,358,36,5,40,189,38,43,369,38,40,1,36,
5,38,7,36,5,37,2,36,5,37,6,36,5,37,0,37,39
510 DATA 感冒,桑菊感冒片(4-6片/次 2次/日)
520 DATA 流感,银翘解毒片(4-6片/次 2次/日)
530 DATA 流感,银翘解毒片(4-6片/次 2次/日)
540 DATA 流感,银翘解毒片(4-6片/次 2次/日)
550 DATA 流感,银翘解毒片(4-6片/次 2次/日)
560 DATA 肺炎,青霉素(肌肉注射)
570 DATA 肠炎,四环素(1片/次 4次/日)
580 DATA 肝炎,休息,注意营养
590 DATA 阑尾炎,住院开刀
600 DATA 乙型脑炎,立即住院治疗
610 DATA 痢疾,痢特灵(1片/次 3次/日)
620 DATA 头痛,去痛片(2片/次 3次/日)
630 DATA 消化不良,酵母片(4片/次 3次/日)
640 DATA 咳嗽,止咳糖浆(10ML3次/日)
650 DATA 腹泻,活性炭片(4片/次)
660 DATA 低烧,解热止痛片(2片/次 3次/日)
READY

```

十二、抢数游戏

段亿清

这是一个数学游戏程序，小朋友和计算机从1开始交替说数，谁先说到所要抢的数谁就胜了。此游戏程序能提高娃娃们的智力。

游戏方法

程序开始执行时，由小朋友确定要抢的数，例如30，然后确定小朋友和计算机谁先说数，例如计算机先说1，小朋友接着说2或2、3，计算机又接着说等等。计算机和小朋友交替地说，谁先说到所要抢的数，谁就胜了。

做此游戏时要注意：

1. 不可连续说三个数。
2. 每次说一个数或两个数均可以。如果只想一次给出一个数，则当第二次提问时给回车信号即可。如果给回车时出现“小朋友回答错”，则在第二次提问时给出“0”和回车。
3. 从1开始，按照自然数的顺序说数。

源程序及运行结果

本游戏程序是在汉字机上实现的，提示信息 and 输出结果均以汉字给出。如果无汉字机则相应的汉字信息可用拼音代替。

```
10 KZ=0
20 PRINT"      抢数游戏"
30 INPUT"抢哪个数?":M%
40 NZ=M%-3*INT(M%/3):IX=NZ+1-3*INT((NZ+1)/3)
50 INPUT"谁先抢:计算机(1),小朋友(2)":A$
60 IF A$="2" GOTO 150
70 LZ=KZ-3*INT(KZ/3)
80 KZ=KZ+1
90 PRINT"计算机:":KZ
100 IF KZ=M% GOTO 280
110 IF LZ<>IX GOTO 150
120 KZ=KZ+1
130 PRINT"计算机:":KZ
140 IF KZ=M% GOTO 280
150 INPUT"小朋友:":A1$
160 INPUT"小朋友:":A2$
170 IF VAL(A1$)<>KZ+1 GOTO 260
180 IF KZ+1=M% GOTO 300
190 IF A2$="0" GOTO 240
200 IF VAL(A2$)<>KZ+2 GOTO 260
210 IF KZ+2=M% GOTO 300
220 KZ=KZ+2
230 GOTO 70
240 KZ=KZ+1
250 GOTO 70
260 PRINT"小朋友回答错"
270 GOTO 150
```

```

280 PRINT"计算机胜!"
290 GOTO 10
300 PRINT"小朋友胜!"
310 GOTO 10
320 END

```

RUN

抢数游戏

```

抢哪个数 ?? 20
谁先抢? 计算机 (1), 小朋友 (2)? 2
小朋友 :? 1
小朋友 :? 2
计算机 : 3
小朋友 :? 4
小朋友 :? 0
计算机 : 5
小朋友 :? 6
小朋友 :? 7
计算机 : 8
小朋友 :? 9

```

十三、猜字母游戏

徐爱菊 王震权

运行本程序，计算机可以猜中你所默记的任何一个字母。

游戏开始时，计算机首先显示 8 个不同的字母。停一段时间，叫你认准那一个字母，并且默记住它。然后，计算机显示出其中的四个字母，同时问你有没有你所默记的那个字母？如果有，你就回答“1”；如果没有，你就回答“0”。这样连续回答 3 次。如果你输入的不是“1”或“0”，计算机就会告诉你“错了，再试一次”；如果你按要求输入“1”或“0”，计算机就立即显示出你所默记的那个字母，并问你再玩一次吗？如果想再玩一次，就输入“Y”，游戏重新开始。否则输入“N”，计算机就向你告别：“Good-bye Comrade!”。

程序设计思想和语句说明

① 设计思想 程序一开始利用随机数发生器产生 65~90 之间的一个数字。通过字符串函数 CHR\$ 把数字变成字母。由于要求 8 个字母都不能相同，所以程序中用了一个二重循环。判别如果本次产生的字母和以前的字母相同，就重新产生。

猜字母部分程序主要使用了计算机的逻辑判断功能。我们以下面的例子来说明它是怎样实现的。

假如产生的 8 个字母是 ABCDEFGH。我默记“F”字母叫计算机猜。计算机第一次显示 C、D、G、H 4 个字母如图 1，我输入“0”，说明没有 C、D、G、H 4 个字母，第 2 次计

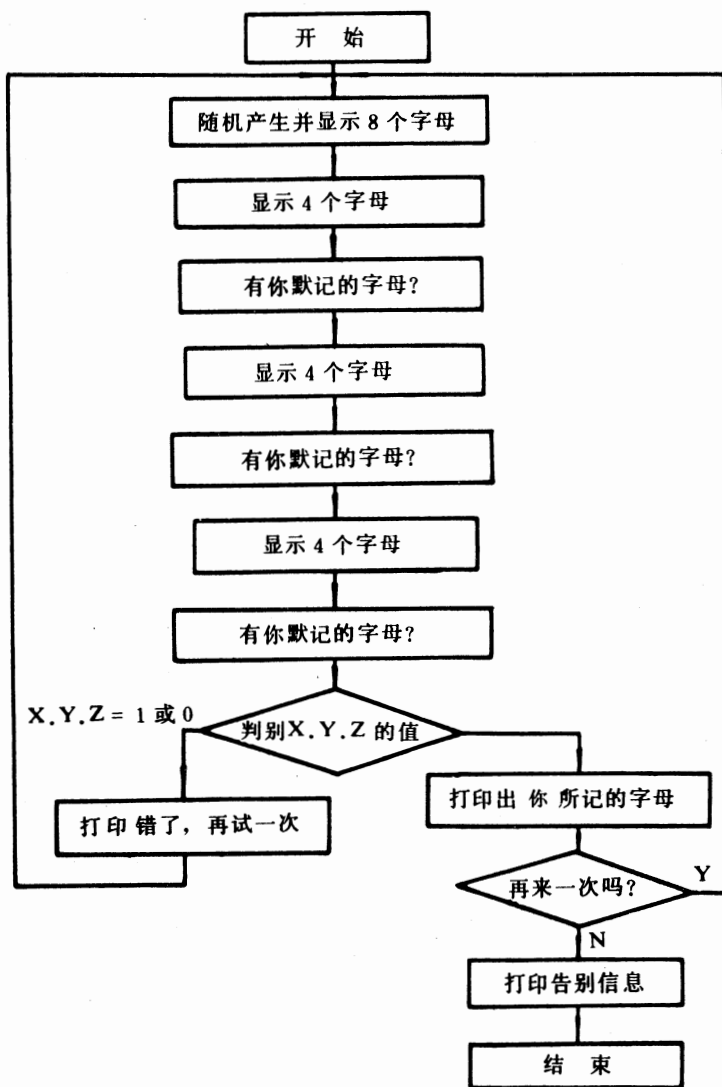
计算机显示 A、C、E、G 4 个字母，我又输入 0，说明 A、C、D、E、G 5 个字母都不是我默记的字母；第 3 次计算机显示 A、B、C、D 4 个字母，我又键入 0 说明 A、B、C、D、E、G、H 都不是我默记的字母，显然，只有“F”是我默记的字母。计算机是通过逻辑与的关系（即 X 与 Y 与 Z 皆为 0）来判断你默记的字符的。

A	B	C	D	E	F	G	H	
—	—	—	—	—	—	—	—	X = 0
△	△	△	△					Y = 0
□	□	□	□					Z = 0

图 1

②语句说明 本程序分四个部分，10~57 语句是产生并显示 8 个不同的字母；60~110 语句，计算机提问三次，每次显示 4 个字母；120~240 语句，进行逻辑判断；250~450 语句，显示你所默记的字母并询问你还要玩一次吗？

程序框图



源程序及运行结果

```
10 DIM A$(8):RANDOM
20 CLS:X=645
30 FOR I=1 TO 8
40 A=RND(90):IF A<65THEN 40
50 A$(I)=CHR$(A):PRINT@X,A$(I)::X=X+5:NEXT I
55 FOR I=0 TO 3000 :NEXT
60 CLS:PRINT@645,A$(4),A$(8),A$(3),A$(7)
70 INPUT"IS THERE THE WORD HERE";X
80 CLS:PRINT@645,A$(7),A$(5),A$(3),A$(1)
90 INPUT"IS THERE THE WORD HERE";Y
100 CLS:PRINT@645,A$(1),A$(2),A$(3),A(4)
110 INPUT"IS THERE THE WORD HERE";Z
120 IF X=1 THEN 180
130 IF X<>0 THEN 220
140 IF Y=1 AND Z=1 THEN 250
150 IF Y=0 AND Z=1 THEN 270
160 IF Y=1 AND Z=0 THEN 290
170 IF Y=0 AND Z=0 THEN 310
180 IF Y=1 AND Z=1 THEN 330
190 IF Y=0 AND Z=1 THEN 350
200 IF Y=1 AND Z=0 THEN 370
210 IF Y=0 AND Z=0 THEN 390
220 INPUT "YOU GOOFED .TRY AGAIN.",K$
230 IF K$="Y" THEN 20
240 GOTO430
250 CLS:PRINT@645,A$(1)
260 GOTO400
270 CLS: PRINT@645,A$(2)
280 GOTO400
290 CLS:PRINT@645,A$(5)
300 GOTO400
310 CLS:PRINT@645,A$(6)
320 GOTO400
330 CLS:PRINT@645,A$(3)
340 GOTO400
350 CLS:PRINT@645,A$(4)
360 GOTO400
370 CLS:PRINT@645,A$(7)
380 GOTO400
390 CLS:PRINT@645,A$(8)
400 PRINT@901,
410 INPUT"ONCE AGAIN";K$
420 IF K$="Y" THEN 20
430 CLS
440 PRINT@763,"GOOD-BYE, COMRADE!"
450 EDN
```

RUN_

E O K R G J X C

R C K X
IS THERE THE WORD HERE? 0_

X G K E
IS THERE THE WORD HERE? 0_

E O K R
IS THERE THE WORD HERE? 0_

J
ONCE AGAIN? _

十四、猜数

刘科成 孙力

对分法是计算机上经常使用的一种方法,人们也常用这种方法进行所谓的“二十四猜”;即让某人事先想好世界上任意一件东西,你应用对分法提问(如问“有生命的还是无生命的?”;“是动物还是植物?”),一般在提问二十四次内肯定猜出。

“三分法”比对分法更为有效,下面是利用“三分法”编制的一个游戏程序。

请游戏者事先想好0~80中的一个数,然后计算机显示四张表格,你只要回答每张表格中该数出现的次数,计算机可立即“猜”出你所想的数。

程序设计思想与说明

① 设计思想

对于一个0~80的数,可以用三进制表示,即 $X = a_13^0 + a_23^1 + a_33^2 + a_43^3$

其中, $a_i (i = 1, 2, 3, 4)$ 的取值为0, 1, 2(对于0~80的数是这样,大于80的数,也可以用三进制表示只是取值要增加)。建立数表时,0~80中的数X是否在表中出现或出现几次,根据 a_i 而定。例如数1,可分解为 $1 \times 3^0 + 0 \times 3^1 + 0 \times 3^2 + 0 \times 3^3$,即 $a_1 = 1, a_2 = 0, a_3 = 0, a_4 = 0$ 。所以数1只在第一张表格中出现1次;又如数34可用三进制表示为: $34 = 1 \times 3^0 + 2 \times 3^1 + 0 \times 3^2 + 1 \times 3^3$,即 $a_1 = 1; a_2 = 2; a_3 = 0, a_4 = 1$,故34这个数在第1张表格中出现1次,在第2张表格中出现2次,第3张表格中不出现,第4张表格中出现1次;对于0~80的数,就是根据这样的分解,把它们分别归併到四张表格内。当你回答了所想的数在每张表格中出现的次数后(0、1或2),计算机只要将每张表格中出现的次数乘上该表格所对应的3的乘方数,然后再求和即可得到你所想的数,

例如，在先后4张表格出现时你回答1，0，2，1四个数，计算机自动进行下面的计算

$$\text{SUM} = 1 \times 3^0 + 0 \times 3^1 + 2 \times 3^2 + 1 \times 3^3 = 46$$

即你所想的数为46

②语句说明

30~180语句是根据上述原理制成4张表格。

200语句是提示游戏者想好一个数字，想好后按任意键都会立即显示表格。

210~320语句是根据游戏者的回答计算出该数。

源程序和运行结果

```
10 REM A PUZZLE PROGRAM
15 CLS:PRINT TAB(15);"IN RUNNING"
20 DIM A(4,81)
30 K1=0:K2=0:K3=0:K4=0
40 N=-1
50 FOR I4=0 TO 2:FOR I3=0 TO 2:FOR I2=0 TO 2:FOR I1=0
TO 2
55 N=N+1
60 IF I1=0 THEN 90
70 K1=K1+1:A(1,K1)=N
80 IF I1<>2 THEN 90
81 K1=K1+1:A(1,K1)=N
90 IF I2=0 THEN 120
100 K2=K2+1:A(2,K2)=N
110 IF I2<>2 THEN 120
111 K2=K2+1:A(2,K2)=N
120 IF I3=0 THEN 150
130 K3=K3+1:A(3,K3)=N
140 IF I3<>2 THEN 150
141 K3=K3+1:A(3,K3)=N
150 IF I4=0 THEN 180
160 K4=K4+1:A(4,K4)=N
170 IF I4<>2 THEN 180
171 K4=K4+1:A(4,K4)=N
180 NEXT I1,I2,I3,I4
199 CLS
200 PRINT"PLEASE CHOOSE A NUMBER FROM 0 TO 80 IN MIND.
":PRINT"ARE YOU READY?(PRESS ANY KEY)"
205 X$=INKEY$:IF X$="" GOTO 205
210 SUM=0
220 FOR I=1 TO 4 :CLS
230 FOR J=1 TO 9
240 FORK=1 TO 9
245 B=A(I,(J-1)*9+K)
250 PRINT TAB(4*K-LEN(STR$(B))):B;
260 NEXT K:PRINT
270 NEXT J:PRINT:PRINT
280 INPUT"HOW MANY TIMES DOES IT EMERGE?(0 OR 1 OR 2)"
:T
290 IF T<>0 AND T<>1 AND T<>2 THEN 280
300 SUM=SUM+T*3[(I-1)
310 NEXT I
320 PRINT "THE NUBRER YOUR CHOSEN IS ";SUM:PRINT:PRINT
" IS IT CORECT ?":PRINT:PRINT
330 PRINT"DO YOU WANT TO TRY ONCE MORE ?(PRESS Y OR
ANY KEY)"
335 X$=INKEY$:IF X$="" GOTO 335
340 IF X$="Y" THEN 199
350 END
```

```

RUN_
      IN RUNNING
PLEASE CHOOSE A NUMBER FROM 0 TO 80 IN MIND.
ARE YOU READY?(PRESS ANY KEY)

  1  2  2  4  5  5  7  8  8
10 11 11 13 14 14 16 17 17
19 20 20 22 23 23 25 26 26
28 29 29 31 32 32 34 35 35
37 38 38 40 41 41 43 44 44
46 47 47 49 50 50 52 53 53
55 56 56 58 59 59 61 62 62
64 65 65 67 68 68 70 71 71
73 74 74 76 77 77 79 80 80

HOW MANY TIMES DOES IT EMERGE?(0 OR 1 OR 2)? 1_
  3  4  5  6  6  7  7  8  8
12 13 14 15 15 16 16 17 17
21 22 23 24 24 25 25 26 26
30 31 32 33 33 34 34 35 35
39 40 41 42 42 43 43 44 44
48 49 50 51 51 52 52 53 53
57 58 59 60 60 61 61 62 62
66 67 68 69 69 70 70 71 71
75 76 77 78 78 79 79 80 80

```

```

HOW MANY TIMES DOES IT EMERGE?(0 OR 1 OR 2)? 2_
  9 10 11 12 13 14 15 16 17
18 18 19 19 20 20 21 21 22
22 23 23 24 24 25 25 26 26
36 37 38 39 40 41 42 43 44
45 45 46 46 47 47 48 48 49
49 50 50 51 51 52 52 53 53
63 64 65 66 67 68 69 70 71
72 72 73 73 74 74 75 75 76
76 77 77 78 78 79 79 80 80

HOW MANY TIMES DOES IT EMERGE?(0 OR 1 OR 2)? 0_
27 28 29 30 31 32 33 34 35
36 37 38 39 40 41 42 43 44
45 46 47 48 49 50 51 52 53
54 54 55 55 56 56 57 57 58
58 59 59 60 60 61 61 62 62
63 63 64 64 65 65 66 66 67
67 68 68 69 69 70 70 71 71
72 72 73 73 74 74 75 75 76
76 77 77 78 78 79 79 80 80

HOW MANY TIMES DOES IT EMERGE?(0 OR 1 OR 2)? 0
THE NUBRER YOUR CHOSEN IS 7

IS IT CORECT ?

DO YOU WANT TO TRY ONCE MORE ?(PRESS Y OR ANY,KEY)
READY
>_

```

十五、猜猜看

郑德义

生活中存在许多数据，比如你的年龄；生日；家庭人口数；班级里学生数等等。这些数据你是知道的，要让人们来猜，只能靠经验、分析和碰运气了，但是要让计算机猜，保险百分之百猜中。请你试一试，就来猜一猜你的年龄和生日吧。

程序设计思想和语句说明

①设计思想 计算机是如何猜到这些数的呢？

下面以 5×5 的 A 矩阵为例来说明猜数的方法。

设被猜的数为: $X = 16$, 5×5 的 A 矩阵:

S (1) 25 2 23 4 21

S (2) 6 19 8 17 10

S (3) 15 12 13 14 11

S (4) 16 9 18 7 20

S (5) 5 22 3 24 1

X 所在的组 (行) 号是 4, 然后求 A 矩阵的转置矩阵 B:

S (1) 25 6 15 16 5

S (2) 2 19 12 9 22

S (3) 23 8 13 18 3

S (4) 4 17 14 7 24

S (5) 21 10 11 20 1

X 在 B 矩阵的组 (行) 号为 1, 所谓矩阵转置是将原来的矩阵的行变成列, 列变成行。所以第二次回答 X 在 B 矩阵的组 (行) 号时, 实际上是回答了 A 矩阵的列号。对于任何一个矩阵, 只要知道了某元素的行、列号, 该元素就被唯一确定了。数就被猜中了。

② 语句说明 程序中 100 语句由程序人员事先赋值, 其中 G 为年号 (使用该程序的时间), K 为年龄表的组号, K^2 是年龄数的范围。130~220 语句产生 K 行 K 列 A 矩阵一年龄数初表。这里不打印是为了节省语句而有意安排在循环语句中。

240~280 语句是求 A 矩阵的转置 B 矩阵。290~360 语句是以行号来打印矩阵。370 语句是被猜者应回答的年龄数, 出生月、日期所在年龄表中的组号, 分别赋给 $Y(L)$ 、 $M(L)$ 、 $D(L)$ 。400~420 语句是打印“你今年是 $\times \times$ 岁, 你出生在 $\times \times$ 年 $\times \times$ 月 $\times \times$ 日”的汉语拼音。

源程序及运行结果

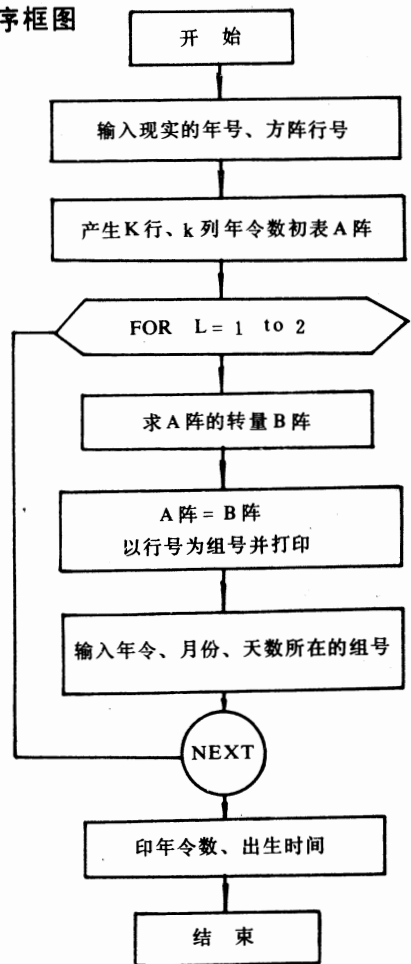
① 源程序

```

100 INPUT G,K
110 PRINT
120 DIM A(K,K),B(K,K)
130 V=0
140 FOR J=1 TO K
150 FOR I=1 TO K
160 V=V+1
200 A(J,I)=V
210 NEXT I
220 NEXT J
230 FOR L=1 TO 2
240 FOR J=1 TO K

```

程序框图



```

250 FOR I= 1 TO K
260 B(I,J)=A(J,I)
270 NEXT I
280 NEXT J
290 FOR J=1 TO K
300 PRINTTAB(0);"S(";J;")";
310 FOR I=1 TO K
320 A(J,I)=B(J,I)
330 PRINT TAB(2+5*I);A(J,I);
340 NEXT I
350 PRINT
360 NEXT J
370 INPUT Y(L),M(L),D(L)
380 PRINT
390 NEXT L
400 PRINT"NI JIN NIAN SHI";B(Y(2),Y(1))
410 PRINT"NI CHU SHENG ZAI";G-B(Y(2),Y(1));
420 PRINT B(M(2),M(1));B(D(2),D(1))
430 END

```

② 运行结果

RUN

? 1984,6

S(1)	1	7	13	19	25	31
S(2)	2	8	14	20	26	32
S(3)	3	9	15	21	27	33
S(4)	4	10	16	22	28	34
S(5)	5	11	17	23	29	35
S(6)	6	12	18	24	30	36

? 2,6,1_

S(1)	1	2	3	4	5	6
S(2)	7	8	9	10	11	12
S(3)	13	14	15	16	17	18
S(4)	19	20	21	22	23	24
S(5)	25	26	27	28	29	30
S(6)	31	32	33	34	35	36

? 5,1,1

NI JIN NIAN SHI 26
NI CHU SHENG ZAI 1958 6 1
READY
>

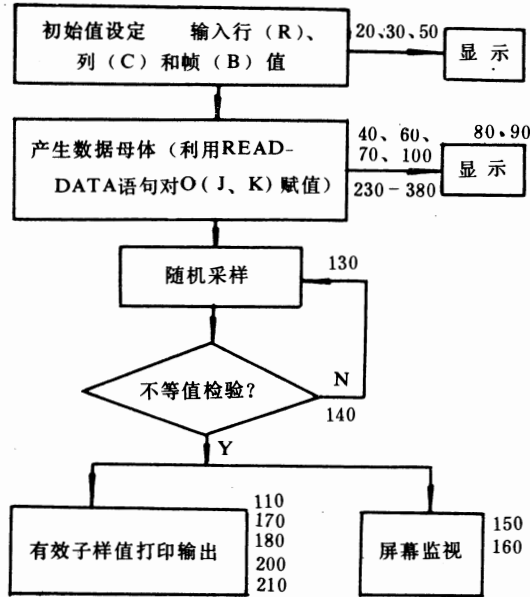
程序启动后，首先问你现在的年号和K值（由被猜年龄范围而定）。如果执行程序的时间是在1984年，你的年龄又在36以下，则可给出G = 1984，K = 6。随后计算机自动生成6 × 6矩阵。如果你的年龄是26岁，生日是6月1日，那么你就告诉计算机，这三个数所在矩阵的行数（2，6，1）；然后，计算机再自动生成转置矩阵，你再告诉你的年龄，生日三个数所在的行数（5，1，1），计算机就可打印或显示出你的年龄、生日了。

十六、编制不等值的随机数表

姚燕平

在工程上或出于系统模拟的需要，有时要取少量的样本，但要求它们的数值彼此不相等，且是均匀分布的随机数。现有的随机数表或随机函数产生语句如RND(X)都满足不了要求。本文介绍的程序能产生不等值的随机数。

程序框图



程序设计思想和语句说明

① 设计思想 程序先利用二维数组和READ—DATA语句制造出所要求的彼此不等值的母体。该母体可视为矩阵。随后对该母体随机采样。由于随机采样仍是用RND(X)语句进行的，所以会有重复的子样值被采出，因此还要作“不等值检验”，即将本次采样值与先前所采的已输出的有效子样值相比，只有通过“不等值检验”的子样才准许其输出；否则，重新采样，重新作“不等值检验”。这样“不等值检验”是利用POINT和SET语句完成的。只要POINT(X, Y) = -1，即当前所采的子样值与先前已输出的子样值（已经用SET语句点

亮）相等，就判为无效数据，并使程序返回采样语句；反之，就输出子样。它的优点是：①其X、Y坐标与数组的维数或矩阵的列数相对应；②屏幕上出现的由随机子样组成的发亮区也与X、Y坐标及数值相对应。只要有亮点就有子样输出，直到X*Y区全部点亮和数据子样全部取完为止。因此，便于监视。

程序中设有三个循环语句保证全套正常运行。一是赋值循环，使数据母体在随机采样之前产生；二是随机采样及显示、打印循环；最后利用帧循环将两者合成一整体，使其协同工作。

语句50的A循环以及语句70和190的G循环起延时作用，仅为便于观察而设，不影响整个工作。语句230~280只给出16×16的DATA值，作为实例，取1~256顺序排列。如改为随意排列，则输出伪随机子样的规律更难探索。若所需随机数大于16×16，应加大母体容积，即增加DATA语句，使二者匹配。亦可改变DATA数据中的数值，送出其它样本值。语句220中的RESTORE保证数据母体的赋值与帧长无关，每次能从头开始。“不等值检验”语句140置于SET(X, Y)之前，简化了对采样初始值的处理。

② 语句说明 程序中120、190、200语句分别用于行、列、帧的循环。其它语句说明见程序框图。

源程序及运行结果

① 源程序

```

10 'RANDOM DATAS WHICH DO'NT EQUAL EACH OTHER
20 INPUT"ROW=";R:INPUT"COL=";C
30 INPUT"B=";B:PRINT"DATA BLOCKS IS";B
40 DIM D(R,C)
50 FOR A=1 TO 500:NEXT A
60 FOR L=1 TO B:CLS:FOR J=1 TO R:FOR K=1 TO C
70 READD(J,K):FOR B=0 TO 50:NEXT B
  
```

```

80 T=T+1:PRINT@32,"T=";T
90 PRINT TAB(32)"D=";D(J,K);:PRINT:T1=C*R:PRINT TAB(32)
)"THE EACH BLOCK OF DATAS IS";T1;:PRINT :PRINTTAB(32)"
TOTAL BLOCKS IS";B;
100 NEXT K,J:T=0
110 LPRINT "DATA BLOCK";L;:LPRINT"R=";R;" C=";C;"RANDO
M DATAS=";R*C:PRINT
120 FOR K=1 TO R:FOR J=1 TO C
130 X=RND(R):Y=RND(C)
140 IF POINT(X,Y)THEN130
150 SET(X,Y):T2=T2+1
160 PRINT @512,"RANDOM DATA=";D(X,Y);:PRINT@576,"TUTAL
DATAS=";T2;:PRINT@640,"CURRENT BLOCK IS NO.";L;
170 LPRINT USING" ###";D(X,Y);
180 IF T2=C*R THEN 200
190 FOR G=0 TO 100:NEXT G:NEXT J:PRINT:LPRINT:NEXT K
200 PRINT:LPRINT
210 PRINT :LPRINT
220 RESTORE:NEXT L
230 DATA 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,
19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,
37,38,39,40,41,42,43,44,45,46,47,48
240 DATA 49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,
82,83,84,85,86,87,88,89,90,91,92,93,94,95,96
250 DATA 97,98,99,100,101,102,103,104,105,106,107,108,
109,110,111,112,113,114,115,116,117,118,119,120,121,12
2,123,124,125,126,127,128,129,130,131,132,133,134,135,
136,137,138,139,140,141,142,143,144
260 DATA 145,146,147,148,149,150,151,152,153,154,155,1
56,157,158,159,160,161,162,163,164,165,166,167,168,169
,170,171,172,173,174,175,176,177,178,179,180,181,182,1
83,184,185,186,187,188,189,190,191,192
270 DATA 193,194,195,196,197,198,199,200,201,202,203,2
04,205,206,207,208,209,210,211,212,213,214,215,216,217
,218,219,220,221,222,223,224,225,226,227,228,229,230,2
31,232,233,234,235,236,237,238,239,240
280 DATA 241,242,243,244,245,246,247,248,249,250,251,2
52,253,254,255,256

```

② 运行结果

RUN

程序工作时，先在屏幕上显示行数R，列数C和帧数B，以判别初始值的设置是否满足要求。

ROW=? 4

清屏后用四行字符(赋值次数T、数值D、帧长T1、帧数B)显示

COL=? 5

数据母体的建立过程。随后屏幕左上角由X=R和Y=C规定的

B=? 2_

DATA BLOCK 1 R=4 C=5 RANDOM DATAS=20

范围内不断出现亮点，在其下方同时出现数据，如采样值D，总采样次数T₂，现行帧数L等三行字符，打印机不断打出随机数。

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

```
DATA BLOCK 2 R= 4 C= 5 RANDOM DATAS= 20
16 14 8 18 17
20 7 13 9 19
2 4 1 3 6
5 10 11 12 15
```

十七、初一英语选择填空练习

李忠蒲

用微型计算机作为辅助教育的手段应该是微型计算机应用的一个方面，它不仅能引起学生对学习的兴趣，同时也会对计算机推广应用起一定作用。

下面是一个比较简单的初一英语选择填空练习程序，为了使程序篇幅尽可能紧缩仅选列15个练习题供作参考。

几点说明

1、由于一般TRS—80微型机还没有小写字母输入和显示功能，练习效果受一定影响。配置或改用有小写字母的微型机效果更好。

2、一般TRS—80微型机还没有汉字处理功能，英译汉和汉译英练习还不能进行。

3、此程序只列15个例题，题多时只要修改第10、20、40语句再加DATA语句即可。

4、此程序是随机练习，如需按顺序练习只要修改第40语句即可。

5、只要做一些改进，计算机辅助代数、几何练习的程序也是可以得到的。

此程序设计起来比较简单，它把练习题和答案做为字符串数据放到DATA语句中。程序启动后，首先把这些数据读入字符串数组M\$(I)和N\$(I)中。然后随机地选择M\$中的练习题，并且显示在显示器上。你输入答案后，程序自动和事先给出的答案比较，一致则判对，否则，显示错误。

源程序及运行结果

```
10 DIM M$(15),N$(15)
20 FOR I=1 TO 15:READ M$(I),N$(I):NEXT I
30 RANDOM
40 X=INT(15*RND(0))+1
50 PRINTX;" ";M$(X)
60 INPUT A$
70 IF A$<>N$(X) THEN PRINT"ERROR":GOTO 50
80 PRINT"VERY GOOD":GOTO 40
90 DATA "I ( ) A STUDENT. (AM,ARE,IS)", "AM"
100 DATA "THIS IS A ( ). (APPLE,ORANGE ,BANANA)", "
BANANA"
110 DATA "( ) A JEEP?(THIS IS,IS THIS,IT IS)", "IS
THIS"
120 DATA "THAT IS( ) CUP. (I,ME,MY)", "MY"
130 DATA "IS IT( ) BROTHER?(YOU,HE,YOUR)", "YOUR"
```

140 DATA "THE WALLS ARE(). (WHITE, RED, BLACK)", "WHI
TE"
150 DATA "HE IS AN ()BOY. (CHINA, CHINESE, ENGLISH)
", "ENGLISH"
160 DATA "WE'RE(). (A STUDENT, STUDENT, ALL STUDENT
S)", "ALL STUDENTS"
170 DATA "WHAT IS ON THE()DESK? (STUDENT, TEACHER'
S, TEACHER)", "TEACHER'S"
180 DATA "THERE ARE()IN A YEAR. (TEN MONTHS, TWELVE
MONTH, TWELVE MONTHS)", "TWELVE MONTHS"
190 DATA "()IS MY SISTER (HE, SHE, WE)", "SHE"
200 DATA "()ROOM IS THIS? (WHOSE, WHO, WHERE)", "WHOS
E"
210 DATA "WHAT'S()THE ROOM? (IN, ON, UNDER)", "IN"
220 DATA "WE LIKE(). (SHE, HE, HER)", "HER"
230 DATA "THERE ARE()DAYS IN A WEEK. (FIVE, SEVEN, N
INE)", "SEVEN"
240 END

RUN

7)HE IS AN ()BOY. (CHINA, CHINESE, ENGLISH)

? CHINA

ERROR

7)HE IS AN ()BOY. (CHINA, CHINESE, ENGLISH)

? ENGLISH

VERY GOOD

10)THERE ARE()IN A YEAR. (TEN MONTHS, TWELVE MONTH, T
WELVE
MONTHS)

? _

十八、简易汉字库

汤观全

调用汉字并打印出来是一件饶有趣味的事情,下面介绍一个建立字库和打印汉字的程序。

程序设计思想与语句说明

①设计思想 首先建立一个字库,即在方格纸上设计好若干汉字,把各汉字图阵表达为八进制数据并存储起来;启动程序后逐次打入欲打印字数及各汉字的代码。程序通过识别代码,选出汉字并在荧光屏上(或打印机上)显示(打印)出来。

假如我们已经建立如下表的字库。此字库中仅有七个汉字,其代码见表1。代码中字母为汉字拼音的首字母,数字为该汉字的起笔号。如果汉字较多,还可加码细分。

汉 字	初	等	高	数	中	习	学
代 码	C ₃	D ₄	G ₃	S ₃	Z ₂	X ₁	X ₃

笔 划	横	竖	点	撇
起 笔 号	1	2	3	4

(表 1)

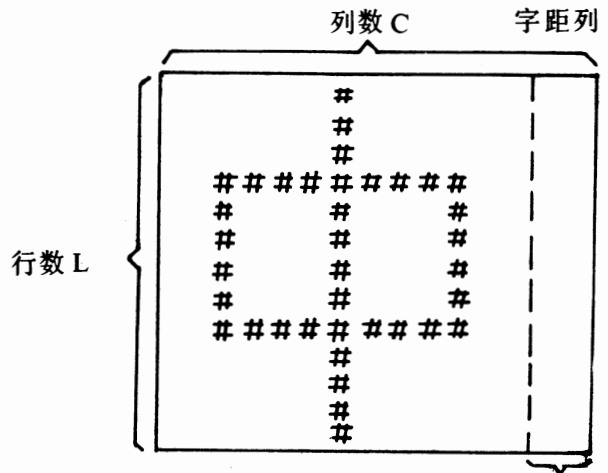


图 1

图阵中每个字的格式见图 1。

这里设计的汉字 $L = 13$, $C = 10$ 。其八进制数据写法以“中”字为例，第一个数据应是 0040 (详见程序第 350 行数据)，字距列的二进制位已计算在内。

选中某汉字后，首先要将该汉字的数据八翻十，以便通过逻辑运算功能找出某一位为 1，如为 1 则打印一个汉字组符号。

语句说明

从框图可以看出，程序共分三部分：准备；选中汉字并将数据八翻十；打印处理。现简要说明如下：

40 行，根据列数 C 算出一个数据的八进制的个数， T 是减去 1 后的值。

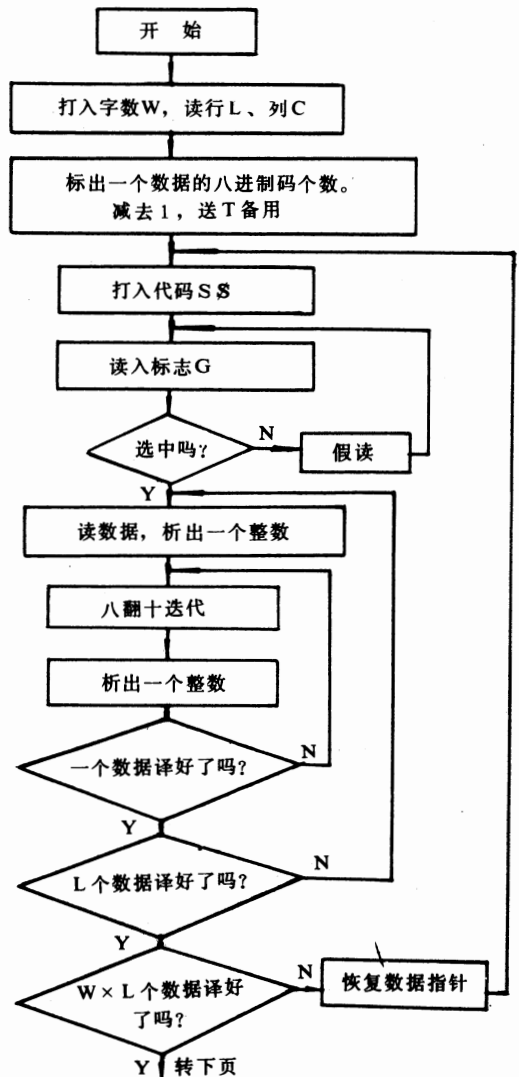
60~80 行是选择汉字代码，并读字库中的代码，如果读入的代码与选择的代码不符，则假读一批数据，否则，读入该代码后的数据。

100~130 行，八翻十，其中 R 依次为数据由高位到低位的每一位值，通过乘加迭代得到十进制数。120 行的 $(U - R) * 10$ 在单精度运算中有时会使近似值不足，取整后数据就偏离要求，100 行中加上 0.1 便避免了这一弊病。

190 行，利用逻辑运算功能对数据从高位开始逐位截取逐位判是否为 1；截取得某位为 1 则按紧凑格式打印。

我们如果要改变用什么字符来形成汉字，只要对程序做如下修改，就具有选择打印字符的能力，增改如下：

程序框图



5 INPUT "CHARACTER USED AS PRINTING:35/#; 69/E; 191/B-LOCK";CH

190 IF D(I, J) AND (2↑K) PRINT CHR\$(CH) ELSE PRINT""; 图符代码实际上可任意挑选。

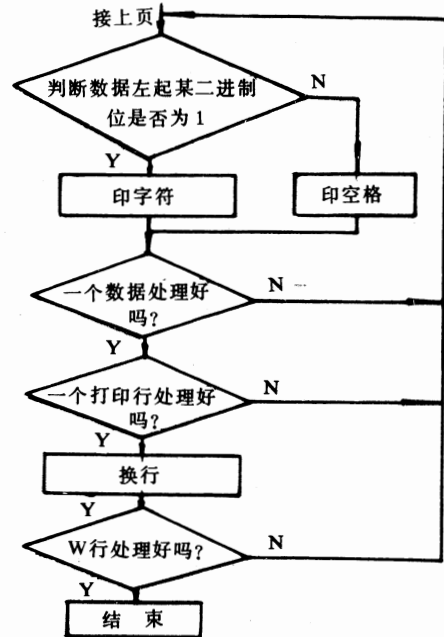
此程序很容易改为实用的子程序, 库中字的增删极为方便, 300行以后任何空程序行可放置任何汉字。在此基础上还可进一步搞一个图例库, 试行计算机绘画。

源程序及运行结果

```

10 INPUT "WARD NUMBER";W
20 READ L,C
30 DIM D(W,L)
40 B=C/3: IF INT(B)=B THEN T=B-1 ELSE T=INT(B)
50 FOR I=1 TO W
60 INPUT "SELECTING WORD";S$
70 READ G$
80 IF S$<>G$ FOR K=1 TO L:READ F:NEXT K:GOTO70 ELSE PR
INT"WAIT A MINUTE,PLEASE,"
90 FOR J=1 TO L
100 READ D(I,J):U=(D(I,J)+.1)/10[T:S=0
110 FOR K=T TO 0 STEP -1
120 R=INT(U):S=S*8+R:U=(U-R)*10
130 NEXT K:D(I,J)=S
140 NEXT J:RESTORE:READ F,F
150 NEXT I
160 FOR J=1 TO L
170 FOR I=1 TO W
180 FOR K=C-1 TO 0 STEP -1
190 IF D(I,J)AND(2↑K) LPRINT"#";ELSE LPRINT" ";
200 NEXT K,I:LPRINT" "
210 NEXT J
220 END
300 DATA 13,10
310 DATA C3,400,476,1722,122,222,422,1522,622,522,422,
422,422,516
320 DATA D4,410,716,1224,40,370,40,774,10,1776,10,210,
10,170
330 DATA G3,40,1776,0,774,404,774,0,1776,1002,1372,121
2,1372,1008
340 DATA S3,220,720,236,1764,224,724,224,424,1764,524,
510,224,1542
350 DATA Z2,40,40,1776,1042,1042,1042,1042,1042,1776,4
0,40,40,40
360 DATA X1,0,1776,2,402,202,112,22,42,102,202,402,100
2,16
370 DATA X3,504,504,1776,1002,1362,20,40,1776,40,40,40
,40,340

```




```

RUN
WARD NUMBER? 5
SELECTING WORD? C3
WAIT A MINUTE, PLEASE,
SELECTING WORD? D4
WAIT A MINUTE, PLEASE,
SELECTING WORD? G3
WAIT A MINUTE, PLEASE,
SELECTING WORD? S3
WAIT A MINUTE, PLEASE,
SELECTING WORD? Z2
WAIT A MINUTE, PLEASE,

```

```

          #      # #      #
        # ##### ### ##### ***** ### #
      ##### # # # # # #
        # # # # # ##### # # # # #
        # # # # ##### # # # # # # # #
      ## # # # #####
      ## # # # # ##### # # # # # #
        # # # # ##### # ##### # #####
        # # # # # # ##### # # # # #
        # # # # # # # # # # # # # #
        # # # # # # # # # # # # #
        # # # # # # # # # # # #
        # # ##### ##### # # # # # #

```

上面介绍的程序采用字符作为点阵而构成汉字，所显示的汉字字形稀疏，不太美观，显示图案效果更差一些。如果对程序做如下修改和补充，在TRS-80机上（或其它有作图功能的计算机）使用作图语句，会得到更加满意的效果。

增加三行语句

```
155CLS
```

```
165JJ = J : II = 0
```

```
185II = II + 1 : IF II = 120 II = 0 : JJ = JJ + 15
```

修改三行语句

```
190 IF D(I, J) AND (2 ↑ K) SET (II, JJ + 2)
```

```
200 NEXT K, I
```

```
220 GOTO 220
```

经上述改动后，汉字字形紧凑。如欲输出图案，修改185行的 $JJ = JJ + 15$ 为 $JJ + 13$ ，则有 120×39 点阵。退出本程序使用BREAK键。

十九、十五子棋

焦艺

这是一个古老的游戏。有一个可放十六枚棋子的棋盘，棋子共15枚，编号从1~15。对于计算机给出的某种布局，你可以通过有限次移动，使棋子按编号从小到大排列起来，就算获胜。而完成这种排列所用的时间越少，你的技巧就越高。

这个问题看起来很简单，做起来可不太容易。因为棋子在棋盘上一共有 $16! = 20,922,789,888,000$ 种排列形式(也称布局)。可以用数学方法证明，其中一半的布局是不能取胜的布局。即使是取胜的布局，有些也需要移动上百步棋才能获胜。

运行本程序时，计算机随机地产生一个布局，检查这个布局是否可取胜，要通过下面的算法来实现。我们用字母A~P标记棋盘的十六个位置。如附图所示。

A	B	C	D
F	F	G	H
	J	K	L
M	N	O	P

对任一种布局：1、设在位置A的棋子编号为n（如果这个位置没有棋子，则 $n = 16$ ），计算这个位置后编号大于n的棋子个数之和。2、对位置B到P，重复第一步，然后把所有的和加起来。

3、如果B、D、E、G、I、L、M、O中任一位置没有棋子，在总上加1。4、如果

总和为偶数；该布局有解；如果总和为奇数，该布局无解。有解布局是可取胜布局。经计算机检查确认为有解布局后，屏幕上显示：“YOUR MOVE”，然后，你可键入要走的棋子号，一直到棋子从小到大排列好。注意移子时只能往空子处移动。

程序说明

100~999: 主程序

4000~4050: 把输入的棋子编号转换成棋子所在的位置编号；

5000~5130: 检查该布局是否有解；

6000~6070: 显示棋局；

7000~7060: 检查输入是否合法；

8000~8090: 检查是否可取胜；

源程序及运行结果

```
10 REM***THE FITHEEN PUZZLE
20 REM***W.L. COLSHER
30 DIM A(16)
40 CLS:M=0:PRINT"GENERATING THE PUZZLE TAKES AWHILE. P
LEASE WAIT."
50 FOR I=1 TO 16 :A(I)=0 :NEXT I
60 FOR I=1 TO 16
70 R=RND(16)
80 IF A(R)<>0 THEN 70
90 A(R)=I
100 NEXT I
110 GOSUB 5000
120 IF F=1 THEN 50
130 GOSUB 6000
140 PRINT" ":INPUT"YOUR MOVE ";X
145 GOSUB 4000
150 GOSUB 7000
160 IF F<>0 THEN 180
170 PRINT"ILLEGAL MOVE,RE-ENTER":FOR I=1 TO 500 :NEXT
I:GOTO 130
180 A(X+F)=A(X):A(X)=16
190 GOTO 8000
200 M=M+1:GOTO 130
999 END
4000 REM***CONVERT NUMBER TO LOCATION IN ARRAY
4010 FOR I=1 TO 16
4020 IF A(I)=X THEN 4040
4030 NEXT I
4040 X=I
4050 RETURN
5000 REM***VERIFY SOLUTION POSSIBLE
5005 F=1
5010 S=0
5020 FOR I=1 TO 15
5030 FOR J=I+1 TO 16
5040 IF A(I)>A(J) THEN S=S+1
5050 NEXT J:NEXT I
5060 FOR I=1 TO 8
5070 READ X
```

```

5080 IFA(X)=0 THEN S=S+1
5090 NEXT I
5095 RESTORE
5100 A=INT(S/2)
5110 IF A*2=S THEN F=0
5120 RETURN
5130 DATA 2,4,5,7,10,12,13,15
6000 REM***DISPLAY GAME BOARD
6005 CLS:L=339:PRINT @217,"MOVE";M
6010 FOR I=1 TO 4
6015 PRINT @L,," ";
6020 FOR J=1 TO 4
6025 N=A((I-1)*4+J)
6028 IF N=16 THEN N=0
6030 IF N<10 THEN PRINT" ";N;
6038 IF (N>=10)AND (N<16) THEN PRINTN;
6040 NEXT J
6050 L=L+64
6060 NEXT I
6070 RETURN
7000 REM***CHECK FOR LEGAL MOVE
7010 F=0
7015 IF X+1>16 THEN 7025
7020 IF A(X+1)=16 THEN F=1
7025 IF X-1<=0 THEN 7035
7030 IF A(X-1)=16 THEN F=-1
7035 IF X+4>16 THEN 7045
7040 IF A(X+4)=16 THEN F=4
7045 IF X-4<=0 THEN 7060
7050 IF A(X-4)=16 THEN F=-4
7060 RETURN
8000 REM***CHECK FOR A WIN
8010 FOR I=1 TO 16
8020 IF A(I)=I THEN200
8030 NEXT I
8040 GOSUB6000
8050 PRINT" ":PRINT" "
8060 PRINT"CONGRATULATIONS !!! YOU DID IT IN ONLY ";M;
" MOVES!!"
8070 PRINT" ":INPUT"TO PLAY AGAIN,HIT ENTER.";A$
8090 GOTO 10

```

RUN_

GENERATING THE PUZZLE TAKES AWHILE. PLEASE WAIT.

MOVE 0

4	1	10	14
3	9	6	15
7	13	12	2
11	8	0	5

YOUR MOVE ? 5_

MOVE 6

4	1	10	14
3	9	6	15
7	13	12	2
11	8	5	0

YOUR MOVE ? 2_

二十、数字游戏—谁先说出“100”?

吕元长

甲乙两人说数游戏，每人想说的数不得小于1（即不能不说或说负数）和大于10，否则算犯规。谁先说都可以。假定甲说一个1至10的任一整数，乙在甲说的数上加上自己想加的一个整数（该数也为1至10的任一整数），然后说出它们的和。接着，甲再在乙的和上加上一个1至10的任一整数，并说出新的和。就这样一个个轮流接下去说，一直到最后的和是100为止。谁先说出100，谁就获胜。

用什么办法可以取胜呢？当你找到了致胜的关键（即需抢先说出一些关键数字），只要你在游戏开始时先说出这些关键数字，你一定获胜。如果对方不知道这个关键，不用说，他只会把临时想到的数加上去。因此，在和他反复游戏时，你完全可以放心地“冒冒险”，让他先说，并且在50范围以内不用关键性的数，来迷惑对方，以免露马脚。

例如：甲说2，乙说9，甲说15，乙说20，甲说26……等等。

下面的程序就是根据上面的要求编制的。它涉及几个关键的数，在这几个关键数之前，计算机说的数是随机产生的，一旦进入关键数，计算机就用几个公式，进行逻辑思维，从而产生它应说的数。关键数是什么？计算机怎样产生逻辑判断的？请读者自行分析程序。

程序运行时机器首先问：WHO SAYS IT FIRST?

你回答（在键盘上按键）0，表示让机器先说（用第一人称表示）；你回答0以外的任何数字，均表示人（你）先说。

在你找到了致胜的“秘诀”以后，不妨可以举一反三，考虑在另一种情况下进行游戏的办法。例如，最大的加数不是10，而是另一个数；最终的和不是100，而是另一个事先规定好的数。而你还可以把上述程序稍加改进，当人（你）连续犯规三次（证明你是有意捣乱），计算机就拒绝和你游戏。

源程序及运行结果

10 N=100	RUN
20 K=10	WHO SAYS IT FIRST? 0
30 Q=INT(N/(2*K))	
40 H=0	I SAY: 8
50 PRINT"WHO SAYS IT FIRST";	HOW MANY DO YOU SAY? 15
60 INPUT V	
70 PRINT	I SAY: 20
120 IF V<>0 THEN 300	HOW MANY DO YOU SAY? 30
130 B=0	
140 A=INT(K*RND(0))+1	I SAY: 36
150 PRINT"I SAY:";A+B	HOW MANY DO YOU SAY? 45
160 PRINT"HOW MANY DO YOU SAY";	
170 INPUT X	I SAY: 48
180 PRINT	HOW MANY DO YOU SAY? 56
190 Y=X-A-B	
200 IF Y<1 THEN 330	
210 IF Y>10 THEN 330	
220 IF X=N THEN 490	I SAY: 57
230 IF X>N-(K+1) THEN 460	HOW MANY DO YOU SAY? 67
240 IF X>Q*(K+1)+1 THEN 350	
250 IF X=Q*(K+1)+1 THEN 410	I SAY: 68
260 B=X	HOW MANY DO YOU SAY? 76
270 IF H=0 THEN 140	
280 A=Q*(K+1)+1-B	I SAY: 78
290 GOTO150	HOW MANY DO YOU SAY? 79
300 A=0	
310 B=0	I SAY: 89
320 GOTO160	HOW MANY DO YOU SAY? 90
330 PRINT"OBEY THE RULES:"	
340 GOTO160	I SAY: 100
350 Q=Q+1	I WIN:
360 IF X<=Q*(K+1)+1 THEN 380	READY
370 Q=Q+1	>_
380 B=X	
390 A=Q*(K+1)+1-B	
400 GOTO150	
410 H=1	
420 Q=Q+1	
430 B=X	
440 A=1	
450 GOTO150	
460 PRINT"I SAY:";N	
470 PRINT" ", "I WIN:"	
480 GOTO500	
490 PRINT" ", "YOU WIN."	
500 END	

二十一、 奇怪的数学题

郑德义

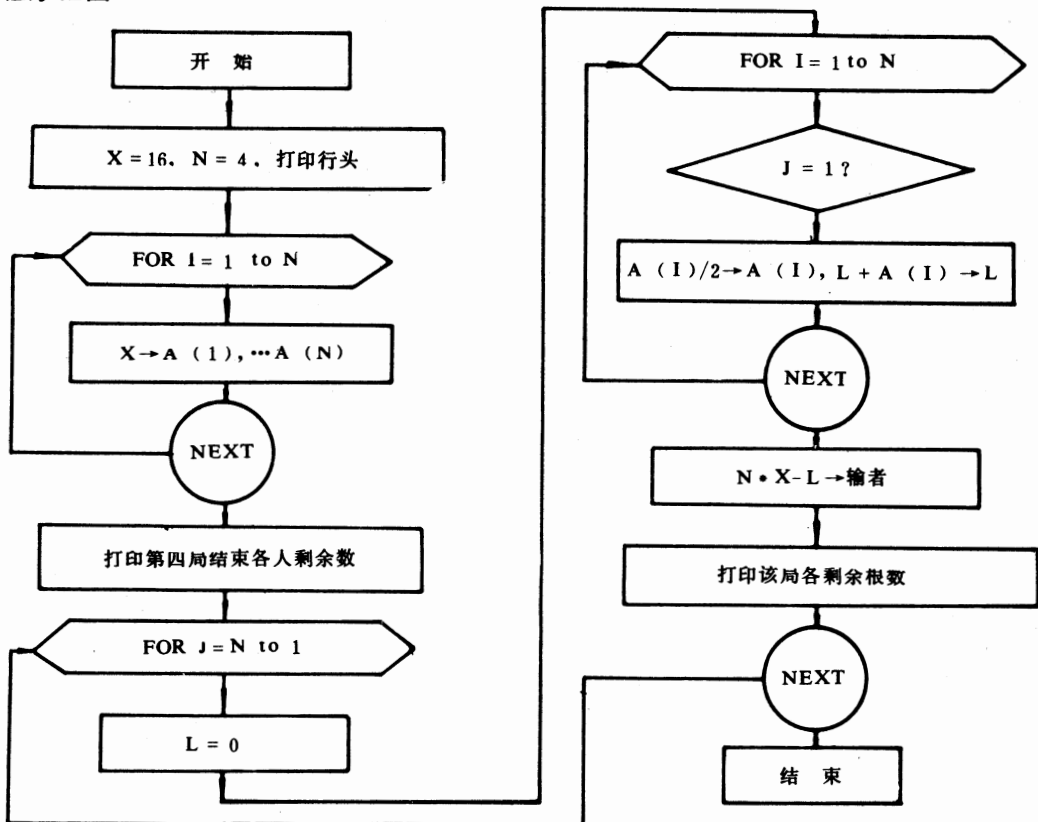
“奇怪的数学题”记叙着这样一件事：四个人做游戏，每玩一次三个人赢、一个人输。输的人要按赢者手中的火柴根数赔偿，即赢者手中有多少根火柴，输者就赔他多少根火柴。他们商定只玩四次。现在知道玩过四次之后，每人恰好输过一次，而且每人手中都正好有16根火柴。这四个人做游戏前各有多少根火柴？

根据题意给出的条件，似乎无法求解。但是我们可以发现该题满足如下两个条件：一是此题有唯一的一个结果，即每人都有16根火柴；二是该题的运算是1对1的（有逆运算），即把三个人的火柴根减半，减下来的火柴根数全给第四个人。基于上述两点，可以用倒推法求解。现在让我们把最先输的人叫做A，把第二局输的人叫B，把第三局输的人叫C，把最后一局输的人叫D。我们知道四人最后都剩下16根火柴，而且A、B和C赢了最后一局，所以在玩最后一局之前，A、B和C各有8根，D有40根。这些数目很容易从赔偿规则推算出来，如果我们接下去继续往前推算，就可以导出他们开始做游戏前各人的火柴根数。

这道《奇怪的数学题》是倒推法妙用的经典例子，它是加倍博弈问题之一。下面我们利用计算机来模拟人解题，做一个初步的尝试。

程序中10号语句的X为最后每人剩下的火柴根数，15号语句的N为参加游戏的人数。

程序框图



源程序及运行结果

```
10 LET X=16
15 LET N=4
20 PRINT"ORDER";
23 PRINTTAB(11);"A";TAB(21);"B";TAB(31);"C";TAB(41);"D
"
25 FOR I=1 TO N
30 LET A(I)=X
35 NEXT I
40 PRINT(";N;");
43 PRINTTAB(10);A(1);TAB(20);A(2);TAB(30);A(3);TAB(40)
;A(4)
45 FOR J=N TO 1 STEP-1
50 L=0
55 FOR I=1 TO N
60 IF J=I THEN75
65 A(I)=A(I)/2
70 L=L+A(I)
75 NEXT I
80 A(J)=N*X-L
85 PRINT(";J-1;");
87 PRINTTAB(10);A(1);TAB(20);A(2);TAB(30);A(3);TAB(40)
;A(4)
90 NEXT J
95 END
```

RUN

ORDER	A	B	C	D
(4)	16	16	16	16
(3)	8	8	8	40
(2)	4	4	36	20
(1)	2	34	18	10
(0)	33	17	9	5

READY

>_

二十二、巧填加减号

郑德义

“巧填加减号”出自波兰著名学者P·斯琴高兹《数学100题》。它涉及到计算机科学的许多基础知识，而且具有相当的趣味性。当数字N较小时，老少都可以做，当数字N较大时，人们就望尘莫及了，只能靠计算机来完成。

题意

图A是由14个“+”号和14个“-”号组成的。两个同号下面都是“+”号，两个异号下面都是“-”号。

假定第一行有N个符号，那么在这类图形中就共有 $\frac{1}{2} N(N+1)$ 个符号（图A中N=7）。因N=3, 4, 7, 8, 11, 12, ……时， $\frac{1}{2} N(N+1)$ 都是偶数，所以可以问，当第一行有N（N=3, 4, 7, 8, ……）个符号时，是否也能构成类似图A的图形。

程序设计思想

从图A中可以看出，图形成三角矩阵。将“+、-”换成“1、0”就成为二进制元素的三角阵，下一行的元素是根据上一行相邻符号决定的，如果元素相同填“1”，相异填“0”，就正好对应二进制半加和。我们使用试探法，即从 $2^N - 1$ 这个数开始，逐步递减，分别试探，直到构成的三角阵满足“1”与“0”的个数相等为止。

```

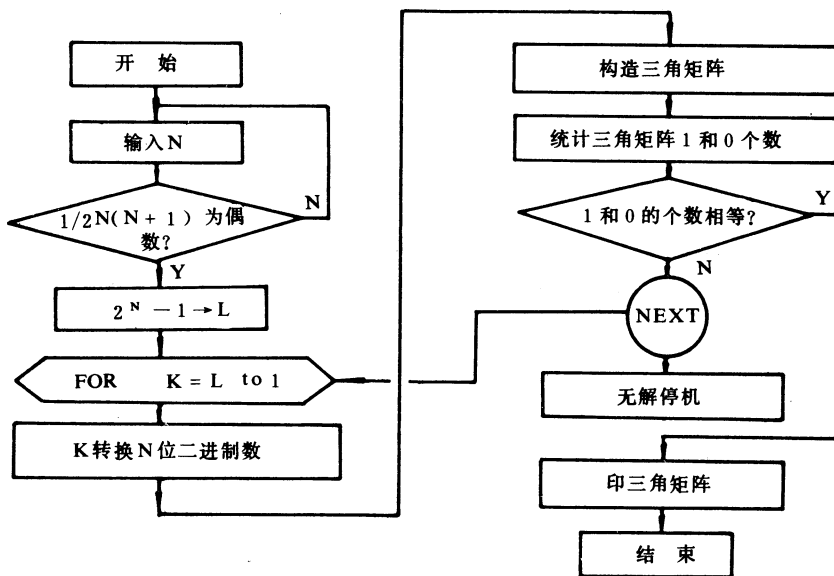
+ - + - + - -
- - - - - +
+ + + + -
+ + + -
+ + -
+ -
-
    
```

程序说明

《巧填加减号》至今没有从理论上解决通解。如想了解该题分布的规律，只需将390语句改成GOTO 280，计算机就会打印出N个符号组成的所有三角阵图形。如果只是作为趣味游戏，则可将试探的起始值改为 2^{N-1} （即L值），把语句8改为 8 FOR Y=L TO $2^N - 1$ ，则试探的次数要少得多。

图 A

程序框图



源程序及运行结果

```

1 INPUT N
2 M=1/2*(N*(N+1))/2
3 IF M-INT(M)<>0 THEN 1
4 PRINT
5 DIMA(N,N)
7 L=INT(2/N)-1
8 FOR Y=L TO 1 STEP -1
9 K=Y
10 FOR I=N TO 1 STEP -1
20 A(I,I)=K-INT(K/2)*2
30 K=INT(K/2)
60 NEXT I
100 FOR I=1 TO N-1
110 FOR J=1 TO N-I
120 A(I+1,J)=0
130 IF A(I,J)<>A(I,J+1) THEN 150
140 A(I+1,J)=1
150 NEXT J
160 NEXT I
200 C=0
210 FOR I=1 TO N
220 FOR J=1 TO N-I+1
230 IF A(I,J)=0 THEN 250
240 C=C+1
250 NEXT J
260 NEXT I
270 IF C=M THEN 300
280 NEXT Y
290 STOP
300 FOR I=1 TO N
310 FOR J=1 TO N-I+1
320 IF A(I,J)=0 THEN 350
330 PRINTTAB(I+J*2);"+";
340 GOTO 360
350 PRINT TAB(I+J*2);"-";
360 NEXT J
370 PRINT
380 NEXT I
390 END

```

RUN

? 7

```

+ + + - + + -
+ + - - + -
+ - + - -
- - - +
+ + -
+ -
-

```

READY

>_

二十三、约瑟夫问题

丛远华 刘建东

N个小孩围成一圈，并给他们依次编号，老师指定从第S个小孩开始报数，报到第M个的小孩出列。然后依次重复下去，直到所有小孩出列。试求小孩的出列顺序，这就是约瑟夫（Josephus）问题。

解题方法

我们注意到这个问题本身的数据构成一个闭合的环，这就启发我们用环形链来解。

右图中每个节点都是由数据与指针两个字段组成。因此，我们可以定义两个数组D和R，D(I)存放第I个结点的数据值，R(I)存放指针值。由于下标I而将此统一成一个节点。通过对R作用来处理指针值，通过R的值来访问D。类似的遍历、插入及删除同于链表。但对本题，我们又可简化为只用数组R即可。因为D值正好与对应下标相等，因此指针的值即为对应的D值。

程序说明

程序1中，15~30行置指针的值，40~50求指向被删除节点，及其前节点的指针值。60行删除节点，并将P指向下一循环开始点。

程序1中求一个节点打印一个节点，不能保留成一队列。程序2做了改进。利用R(0) (T=0)指向第一个被删除点。当求出一个删除点，即将此与其它删除点链接起来。例如删除4，则将R(0)指向4，删除8，则将R(4)指向8（图2）。这一更动很有意思：我们可以用链将数组中空余单元链接起来，以备使用，而只需一个空余单元指向始址。

程序2基本同于程序1。55行实现上述任务。70~85行则用于打印最后形成的链。

```
源程序1及运行结果      50 NEXT D
5 INPUT N,M,S           55 LPRINT TAB(C*5);P;
10 DIM R(N)             60 P=R(P):R(Q)=P
15 FOR C=1 TO N-1      65 NEXT C
20 R(C)=C+1            70 END
25 NEXT C
30 R(N)=1: P=S         RUN
35 FOR C=1 TO N        ? 8
40 FOR D=1 TO M-1     ?? 4
45 Q=P:P=R(P)         ?? 1_
```

4 8 5 2 1 3 7 6

图1

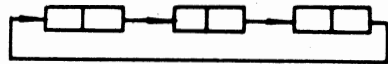
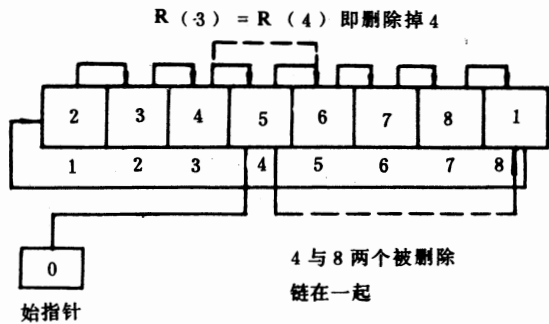


图2



源程序 2 及运行结果

```

5 INPUT N,M,S
10 DIM R(N)
15 FOR C=1 TO N-1
20 R(C)=C+1
25 NEXT C
30 R(N)=1:P=S:T=0
35 FOR C=1 TO N
40 FOR D=1 TO M-1
45 Q=P:P=R(P)
50 NEXT D
55 R(T)=P:T=P
60 P=R(P):R(Q)=P
65 NEXT C
70 T=0
75 FOR C=1 TO N
80 LPRINT R(T);:T=R(T)
85 NEXT C
90 END

RUN
? 8,4,2_

5 1 6 3 2 4 8 7

```

二十四、字母数字谜

戴定凯

字母数字谜是一种用英文字母代替数字的数学等式。式中每个英文字母代表一个从 0 ~ 9 之间的数字，不同的英文字母代表不同的数字。而且在用数字代替了相应的字母后，还必须使这个等式成立。例如：

$$\begin{array}{r}
 \text{SEND} \\
 + \text{MORE} \\
 \hline
 \text{MONEY}
 \end{array}
 \quad \text{可写成} \quad
 \begin{array}{r}
 9567 \\
 +1085 \\
 \hline
 10652
 \end{array}$$

这类字谜初看上去似乎很容易找到答案，但事实上往往使一些精于计算的人也感到伤脑筋。字母数字谜作为一种游戏，它可以锻炼我们的思考能力。作为一种解题手段，据报道，它与人们正探索的一些重大科研领域都有关系。

如何找出字母数字谜的答案？在上述例子中，共出现八个字母，即 D、E、Y、N、R、O、S、M。如果将每种可能的数字组合都去试探一下来求解这个谜，它将超过 81 次。不过你可以让计算机来解决这个谜。

为了缩短计算时间，不使程序过于繁杂，我们可以先找一下字谜中数字的相互关系。在 SEND + MORE = MONEY 的例子中，假定 D 为 3，E 为 4，那么 Y 一定为 7，也就是说，Y 的值是由 D 和 E 的值决定的。同样道理，R 的值是取决于 N 和 E 的值。只要对这个字母数字谜略加分析，不难得出如下结论：我们如果为字母确定数字的顺序是与平时做加法的次序一致，Y、R、O 的值是由其它字母的取值来决定的，而且很明显，M 应为 1。因此，对八个字母的数字组合的试探可简化为对 D、E、N 和 S 四个字母的数字组合进行试探便可以了。

源程序

```

10 DIM A(8)
50 LET A(1)=0
60 LET A(2)=0
70 LET T=2
80 GOSUB 700
90 IF J=1 GOTO 640
100 LET C1=0
110 LET A(3)=A(1)+A(2)
120 IF A(3)<10 GOTO 150
130 LET A(3)=A(3)-10
140 LET C1=1
150 LET T=3
160 GOSUB 700
170 IF J=1 GOTO 640
180 LET A(4)=0
190 LET T=4
200 GOSUB 700
210 IF J=1 GOTO 620
220 LET C2=0
230 LET A(5)=A(2)-(A(4)+C1)
240 IF A(5)=0 GOTO 270
250 LET A(5)=A(5)+10
260 LET C2=1
270 LET T=5
280 GOSUB 700
290 IF J=1 GOTO 620
300 LET C3=0
310 LET A(6)=A(4)-(A(2)+C2)
320 IF A(6)>=0 GOTO 350
330 LET A(6)=A(6)+10
340 LET C3=1
350 LET T=6
360 GOSUB 700
370 IF J=1 GOTO 620
380 LET A(7)=1
390 LET T=7
400 GOSUB 700
410 IF J=1 GOTO 600
420 LET A(8)=A(6)-A(7)
430 IF A(8)=0 GOTO 600
440 LET A(8)=A(8)+10
450 IF A(8)<>1 GOTO 600
460 LET T=8
470 GOSUB 700

```

```

480 IF J=1 GOTO 600
500 PRINT " D "; "E "; "Y "; "N "; "R "; "O "; "S "; "M
"
510 PRINT A(1);A(2);A(3);A(4);A(5);A(6);A(7);A(8)
550 STOP
560 END
600 LET A(7)=A(7)+1
610 IF A(7)<10 GOTO 390
620 LET A(4)=A(4)+1
630 IF A(4)<10 GOTO 190
640 LET A(2)=A(2)+1
650 IF A(2)<10 GOTO 70
660 LET A(1)=A(1)+1
670 IF A(1)<10 GOTO 60
680 STOP
690 END
700 FOR I=1 TO T-1
710 IF A(I)<>A(T) GOTO 750
730 LET J=1
740 RETURN
750 NEXT I
760 LET J=0
770 RETURN

```

本文给出的程序就是解这个字母数字谜的。在程序中, A(1), A(2), …… , A(8) 分别存放 D、E、Y、N、R、O、S 和 M 的值。程序的核心是要对 A(1)、A(2), …… , A(7) 的值进行试探。因而程序设计了四层嵌套循环, 循环的开始语句分别是第 60、70、190、390 语句。700~770 的程序段为第一个子程序, 它的作用是逐个判别赋予某个字母的数字是否与前面已赋值过的字母的数字相同。如果这两个数字相同, 则令 J = 1, 如果不相同, 令 J = 0。在主程序中再根据 J 的情况决定是否要更换赋予该字母的数字。如果 0~9 这十个数字逐一更换仍不能满足 J = 0 的条件, 就得对前面已赋

过值的值进行更改。这样一层层嵌套循环, 直到每个字母的值各不相同, 而且满足等式为止。程序中 C₁、C₂、C₃ 分别代表个位、十位、百位的进位值。

在你了解了这个字母数字谜的程序编制方法后, 你可以自己编制一个通用程序来求解所有的加法形式的文字数字谜, 不过在这样一个程序中, 要用到指针变量, 程序编制起来也复杂得多。

运行结果

RUN

D E Y N R O S M

7 5 2 6 8 0 9 1

BREAK IN 550

READY

>_

二十五、超级智力游戏

石志钢

“超级智力游戏”即是人与计算机下智力棋。在计算机上进行这种游戏时, 计算机为设码者, 人为破译者。有两种游戏方式(0 与 1 方式)。0 方式中当给出一组破码字符时, 计算机不仅指出其中有多少字符是对的, 还提示其位置也正确。1 方式中仅给出有多少字符正确。游戏时, 你可以选择 7、8 或 9 个基本字符, 字符越多难度越大。选定后, 屏幕上出现这些字符, 你选出其中字符作为破译输入。计算机将你的输入与它设置的字符相比较后, 显示出“*”, 表示你的字符与位置均正确(在 1 方式中字符与位置不对应), “·”表示有此字符但位置不对。空格表示密码中无此字符。S 键用于擦去输入字符, V 键保持上次破译中对应字符。

程序设计思想与语句说明

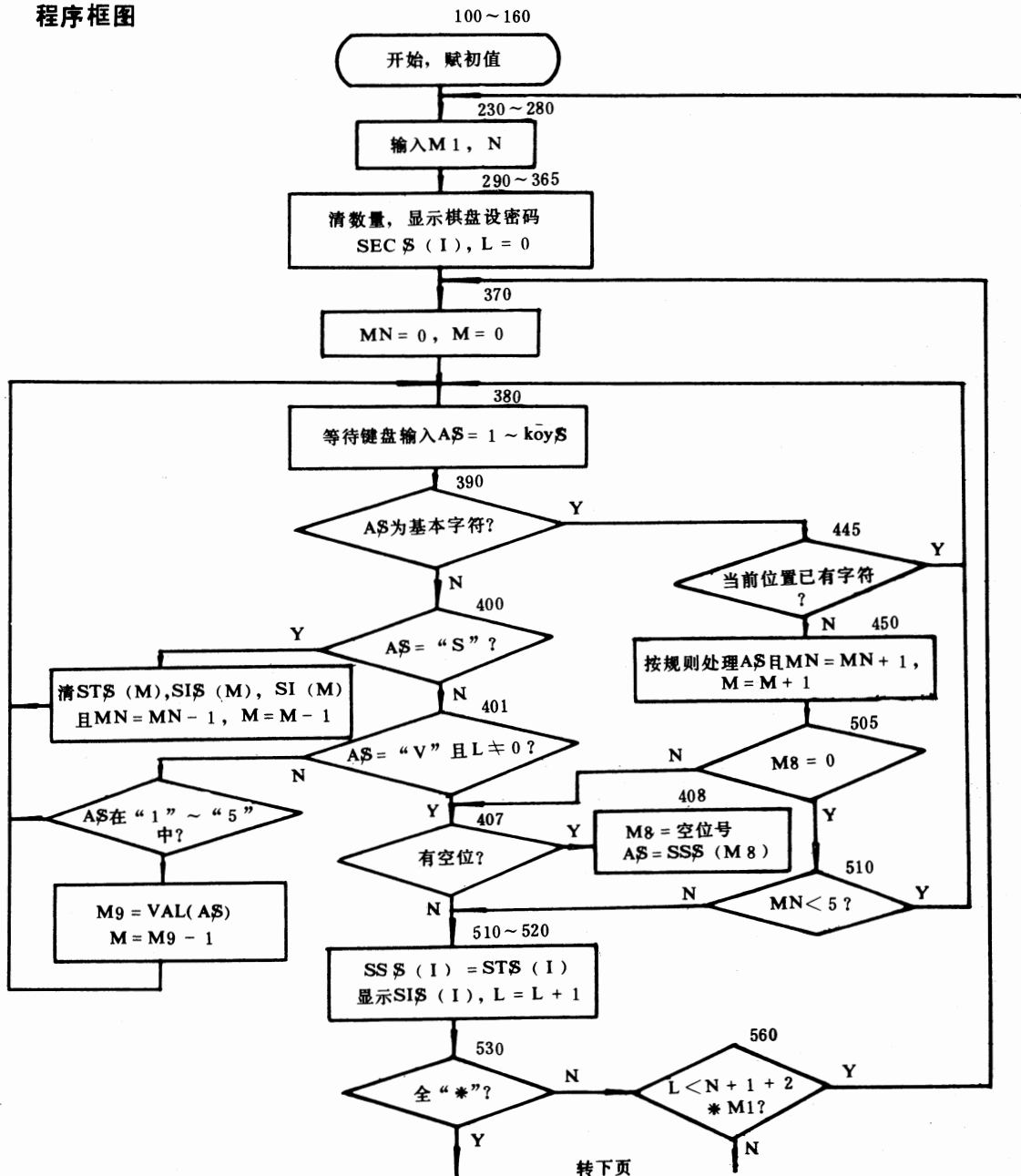
①设计思想 此程序的设计思想很简单。计算机随机地在 N 个字符中产生 5 个字符做为密码。破译者从键盘输入破译码(5 个字符), 然后把键入的字符和设置的密码进行比较, 输出相应的状态信息。

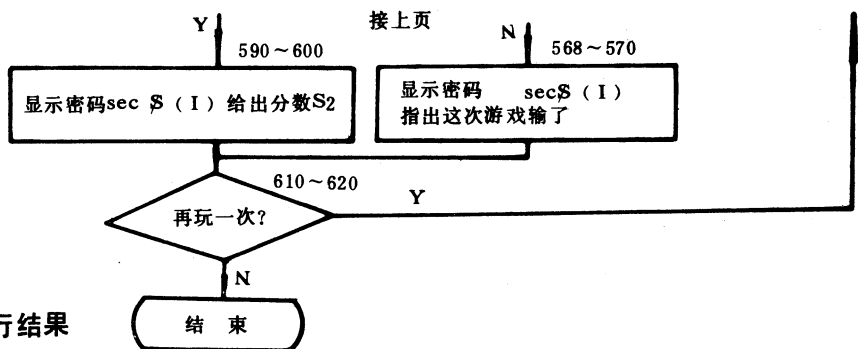
②变量说明 程序中以 M 表示方式; N 存放基本字符; L 指示破码次数; MN 指示本次

猜中字符个数; M 则控制当前字符的位置; SEC\$(I) 中是密码字符; 对一次破译字符的指示信息码放在 SIS(I) 中, ST\$(J) 存放本次破译的字符, 而 SI(J) 是它们的指针, 指示 SIS(J) 中第 J 个字符的指示码存放在 SIS(I) 中的哪个位置, 即 $I = SI(J)$, 在擦去字符时, 也要用到此指针, 同时 M9 也帮助完成 S 命令功能。SS\$(I) 存放上次破译字符, 同 M8 一起完成 V 命令。

④ 语句说明 300 行是从基本字符中随机地取 5 个作为密码。455~459 行处理特殊字符组合。512~516 行专为处理 1 方式。595~600 行计算并显示分数。其它程序行见框图。

程序框图





3. 源程序及运行结果

```

100 DIM SEC$(5),SI$(5),SS$(5),ST$(6),T$(5),SI(5)
150 FOR I=1 TO 5 :SEC$(I)=" ":SI$(I)=" ":SS$(I)=" "
160 ST$(I)=" ":SI(I)=0 :NEXT I :CLS
230 PRINT@325,"WHICH MODE(0/1) DO YOU LIKE TO PLAY":IN
PUT M1
250 IF M1=0 OR M1=1 THEN 260 ELSE 230
260 PRINT @462,"WHAT DO YOU PLAY WITH 7,8,9 ";;INPUT N
280 IF N>=7 AND N<=9 THEN 290 ELSE 260
290 CHS$="ABCDEFGHJKLMN":ES$=LEFT$(CHS$,N)
300 FOR I=1 TO 5 :SEC$(I)=MID$(ES$,RND(N),1):NEXT I :CL
S
320 PRINT"THE ALLOWABLE CHAR AND INFOR-CODE ARE <";
330 PRINT USING"%          %";ES$;"AND * .<"
340 FOR I=1 TO 30 :PRINT TAB(4)"==";;NEXT I
350 PRINT TAB(16)"CHARACTER FIELD",,"INFORMATION CODE"
;
360 PRINTTAB(19)"1 2 3 4 5",TAB(51)"1 2 3 4 5":L=0
365 PRINT@897,"<S> IS USED TO DELETE A CHAR AND <V> TO
KEEP IN LAST CHAR."
370 PRINT@(264+64*L),L+1;;M=0:M8=0:M9=0:MN=0
380 A$=INKEY$:IF A$=""THEN 380
390 FOR I=1 TO N :IF A$=MID$(ES$,I,1) THEN 445 ELSE NE
XT I
400 IF A$="S" AND (M+SGN(M9))>=1 THEN 410
401 IF A$="V" AND L<>0 THEN 407
402 IF A$>="1" AND A$<="5" THEN 404 ELSE 380
404 M9=VAL(A$):M=M9-1:GOTO 380
407 FOR K=1 TO 5 : IF ST$(K)=" " THEN 408 ELSE 409
408 M8=K :M=M8-1:A$=SS$(M8): GOTO450
409 NEXT K:MN=5 : GOTO510
410 IF ST$(M+SGN(M9))=" "THEN 380 ELSE M=M+SGN(M9)
415 SET(34+4*M,13+3*L):RESET(34+4*M,13+3*L)
420 PRINT@(272+64*L+2*M)," ";
430 IF SI(M)=0 THEN 440 ELSE ELSE SI(SI(M))="":SI(M)=
0
440 ST$(M)=" ":M=M-1 :MN=MN-1 :GOTO380
445 IF ST$(M+1)=" " AND M<5 THEN 450 ELSE 380
450 IF A$=SEC$(M+1) THEN 454 ELSE 460
  
```

```

454 IF SI$(M+1)=" " OR M>=4 THEN SI$(M+1)="*": SI(M+1
)=M+1:GOTO500
455 FOR I=M+2 TO 5:IF A$=SEC$(I) THEN 457
456 NEXT I:SI$(M+1)="*":SI(M+1)=M+1:GOTO500
457 IF SI$(I)<>" "THEN NEXT I:GOTO459:ELSE SI$(I)="."
458 FOR J=1 TO 5 :IF SI(J)=M+1 THEN SI(J)=I ELSE NEXT
J
459 SI$(M+1)="*":SI(M+1)=M+1:GOTO500
460 FOR I=1 TO 5 : IF A$=SEC$(I) THEN 470 ELSE 490
470 IF SI$(I)="." OR SI$(I)="*" THEN 490
480 SI$(I)="." : SI(M+1)=I : GOTO 500
490 NEXT I
500 ST$(M+1) =A$:PRINT@(275+64*L+2*M),A$;
505 IF MB<>0 THEN 409 ELSE M=M+1 : M9=0 : MN=MN+1
510 IF MN<5 THEN 380 ELSE J=0 :FOR I=1 TO 5 :SS$(I)=ST
$(I):ST$(I)=" ":NEXT I
512 IF M1=0 THEN 520 ELSE L1=RND(5)
514 FOR K=1 TO 5 : Q=L1-5*SGN(FIX((L1-1)/5))
515 T$(K)=SI$(Q):L1=L1+1 :NEXT K
516 FOR K=1 TO 5 :SI$(K)=T$(K):NEXT K
520 FOR I=1 TO 5 : PRINT@(305+64*L+2*I),SI$(I);
530 IF SI$(I)="*" THEN J=J+1 :NEXT I :ELSE NEXT I
540 PRINT:L=L+1 :FOR I=1 TO 5 :SI$(I)=" ":SI(I)=0:NEXT
I
550 IF J=5 THEN 580
560 IF L<N+1+2*M1 THEN 370
565 PRINT@291, "SECRET CODES";
568 FOR I=1 TO 5 :PRINT@(357+I),SEC$(I);:NEXT I
570 PRINT@64,"YOU HAVE BEEN DEFEATED!";:GOTO610
580 PRINT@64,"THE SECRET CODES ARE <";
590 FOR I=1 TO 5 :PRINTSEC$(I);:NEXT I:PRINT">. ";
595 S1=80+10*(N-7)+20*M1:S2=S1-10*(L-M1):IF L<=6 THEN
S2=S2+5*(6-L)
600 PRINT"YOUR SCORE=";S2
610 INPUT "WOULD YOU LIKE TO GO ON?PLEASE(Y/N)";B$
620 IF B$="Y" THEN 150 ELSE END

```

RUN_

WHICH MODE(0/1) DO YOU LIKE TO PLAY

? 0

WHAT DO YOU PLAY WITH 7,8,9 ? 8_

THE ALLOWABLE CHAR AND INFOR-CODE ARE <ABCDEFGH AND * .<

```

=====
          CHARACTER FIELD                INFORMATION CODE
1          A B C D E                      .
2          E F G H B                      . .
3

```

<S> IS USED TO DELETE A CHAR AND <V> TO KEEP IN LAST CHAR.

二十六、一字棋

钱文瀚

博弈程序是人工智能的研究内容之一。作为初步尝试，下面介绍一个一字棋程序。

程序运行时，屏幕上首先显示图 1 (10—100行)。游戏者 (客方) 想在哪一格下 × (390 400, 460—480行), 就打那一数字键 (120行)。计算机“思考”(170—370行)后作出反应, 在它认为适当的空格内下 ○ (390—450行)。这样交替进行, 任一方有三枚棋子排成一字 (行、列或对角线) 即赢。若让计算机先下, 游戏者先打数字键 0。万一打错了键 (该格内已有棋子), 错误信息被 130行挡住, 可重打。

计算机“思考”的第一步是按紧要程度给各空格评分。设:

1. 主方二缺一 (图 2) 给 20分;
2. 客方二缺一 (图 3) 给 10分;
3. 主方一缺二 (图 4) 给 3分;
4. 客方一缺二 (图 5、6) 给 2分。

迭加后得分最高的各格似对主方最有利。几个空格并列第一时, 按中央 (5)、四角 (1, 9, 3, 7) 和偶格 (2, 8, 4, 6) 的顺序选取, 因为它们分别是 4, 3, 2 排一字的交点 (图 4—6)。

这样做还不完善。如图 7, 按上述原则主方下 3, 给客方以可乘之格 (下 7, 构成两个二缺一)。实际上应下偶格, 打乱客方的阵势。另一失策是图 8, 主方下 5 成平局, 若下 9 必胜。为了弥补以上缺陷, 程序中增添了 220—250行。这样一来, 计算机就象一个老练的棋手, 百战不败。一旦计算机赢了或九格全满, 就自动清屏再现图 1 (490行)。

T 表示轮到谁下棋, 它在 2 (客方) 与 -1 (主方) 之间转换 (500行)。这次下的是第 N 格, 记在数组 S (N) 上 (380行); 本来是空格, 其值为 0, 下 × 记 2, 下 ○ 记 -1, 所以 S (N) 又描述当前的棋局。按理 N = 1 到 9, 140行使 S (0) 只许游戏者在一开始打 0, 否则算打错。8 排一字通过 170, 200行逐排赋给 A, B, C。S (A) + S (B) + S (C) + 3 之值 (> 1) 取决于该排棋子的种类与数目, 故可使选择转向 (210行) 区别对待, 包括选取不同的 E、F 值 (260—290行), 给空格加 E 分而满格不加分 (300行)。320—370行是寻找最高分的空格, 并将记分牌 E (N) 清零。为了在相同的高分中体现上述优先次序, A 值仍从 170 行依次读出。这套数据在安排上兼顾两种需要。经过几个回合之后, 可能各格均得零分, 为防止 N = 5 而不在空格上, 350行加条件 AND S (A) = 0。

7	8	9						(20)				(3)	(3)	(3)
4	5	6		○		×	×	(10)	(3)	○	(3)			
1	2	3	○						(3)	(3)	(3)			

图 1

图 2

图 3

图 4

(2)		(2)
(2)	(2)	
×	(2)	(2)

图 5
程序框图

(2)	×	(2)
	(2)	
	(2)	

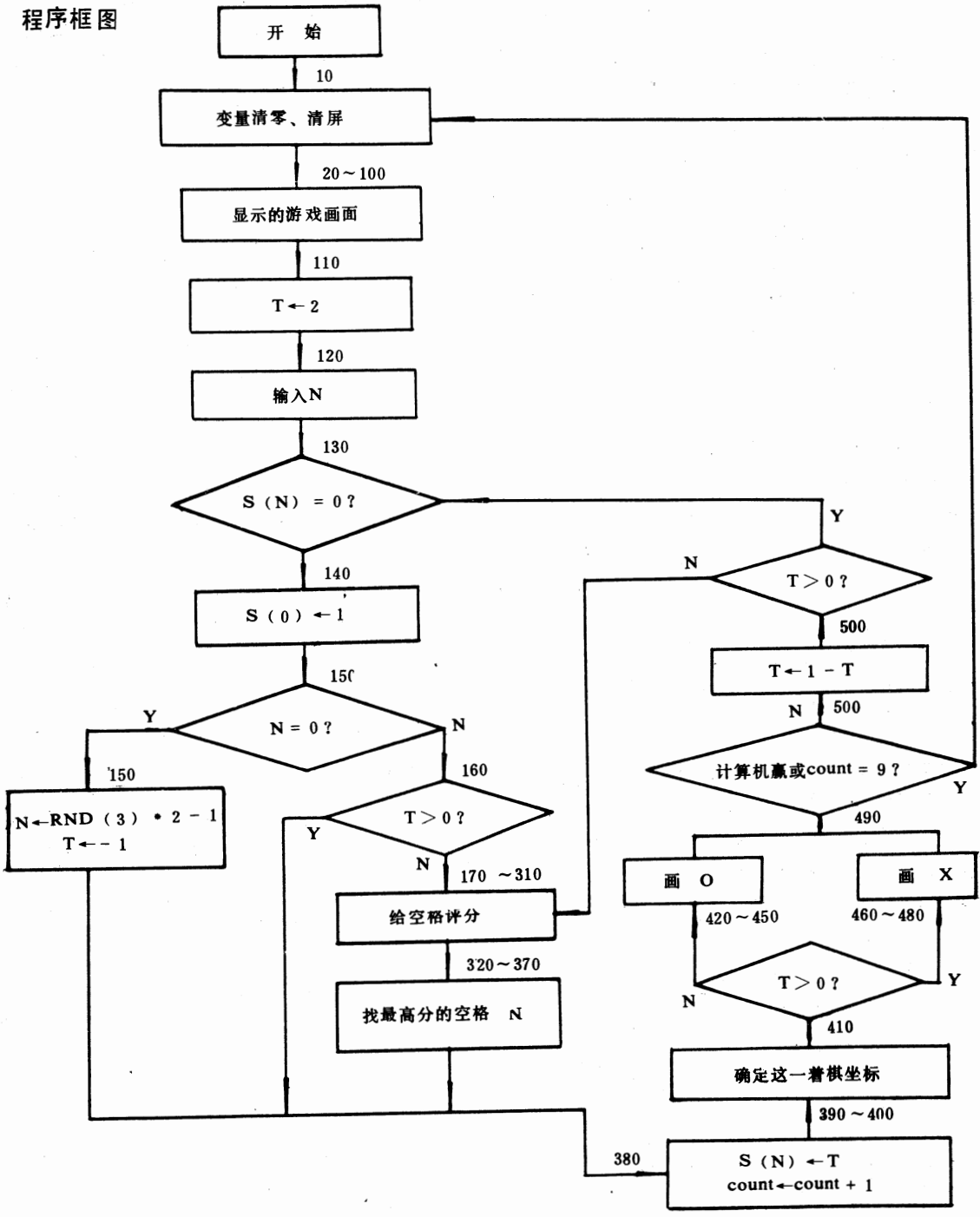
图 6

7 (7)	8 (5)	×
4 (5)	○	6 (5)
×	2 (5)	3 (7)

图 7

7 (5)	8 (0)	9 (5)
4 (3)	5 (5)	6 (2)
○	2 (0)	×

图 8



源程序

```

10 CLEAR:CLS
20 FOR I=1 TO 9
30 PRINT @B30+I*16-INT((I-1)/3)*432, I
40 NEXT I
50 FOR I=0 TO 47
60 SET(47, I):SET(48, I):SET(80, I):SET(81, I)
70 NEXT I
80 FOR I=17 TO 111
90 SET(I, 15):SET(I, 32)
100 NEXT I
110 T=2
120 N$=INKEY$: IF N$="" THEN 120 ELSE N=VAL(N$)
130 IF S(N)<>0 THEN 120
140 S(0)=1
150 IF N=0 THEN N=RND(3)*2-1:T=-1:GOTO 380
160 IF T>0 THEN 380
170 DATA 5, 1, 9, 3, 7, 5, 2, 8, 5, 4, 6, 5, 1, 2, 3, 7, 8, 9, 1, 4, 7, 3, 6
, 9
180 RESTORE
190 FOR I=1 TO 8
200 READ A, B, C
210 ON S(A)+S(B)+S(C)+3 GOTO 260, 280, 310, 220, 290, 240, 2
70
220 IF COUNT<>2 OR I<5 OR S(B)<>0 THEN 310
230 N=30-A*(2-S(A))-C*(2-S(C)):E(N/3)=6: THEN 320
240 IF COUNT<>3 OR I>2 OR S(5)>0 THEN 310
250 E(RND(4)*2)=6: GOTO 320
260 E=20:F=20:GOTO 300
270 E=10:F=-5:GOTO 300
280 E=3:F=3:GOTO 300
290 E=2:F=-1
300 E(A)=E(A)+E+F*S(A):E(B)=E(B)+E+F*S(B):E(C)=E(C)+E+
F*S(C)
310 NEXT I
320 E=-1:RESTORE
330 FOR I=1 TO 11
340 READ A
350 IF E<E(A) AND S(A)=0 THEN E=E(A):N=A
360 E(A)=0
370 NEXT I
380 S(N)=T:COUNT=COUNT+1
390 I=INT((N-1)/3)
400 X0=(N-3*I)*33-2:Y0=41-17*I
410 IF T>0 THEN 460
420 FOR I=1 TO 16
430 X=11.5*COS(0.105*I):Y=5.25*SIN(0.105*I)
440 SET(X0+X, Y0+Y):SET(X0+X, Y0-Y):SET(X0-X, Y0+Y):SET(X
0-X, Y0-Y)
450 NEXT I:GOTO 490
460 FOR X=-11 TO 11
470 SET(X0+X, Y0+X/2):SET(X0+X, Y0-X/2)
480 NEXT X
490 IF E>19 OR COUNT=9 THEN 10
500 T=1-T: IF T>0 THEN 120 ELSE 180

```

二十七、史密斯魔阵

蔡琪君

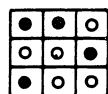


图 1

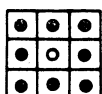


图 2

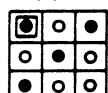


图 3

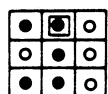
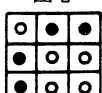


图 4

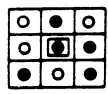
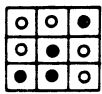


图 5

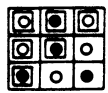
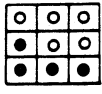


图 6

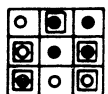


图 7

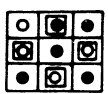


图 8

注 灯亮 灯暗 按钮

六面体魔方诞生以后，曾经风靡一时，同时又引来了许多人设计新的智力游戏，其中流行较广的一种申子游戏，称为“史密斯魔阵”。这种游戏的装置很简单，由 3×3 个灯和按钮组成，如图 1 所示。在随机地给出一种初始状态后，按动按钮，灯就按一定的规律变化，直到达到目标状态，如图 2 所示。当然谁用的步数最少，完成得最快，谁就取胜了。

按动按钮以后，史密斯魔阵的灯变化规则如下：

1) 按动四个角上按钮中的任意一个，则引起包括它及其周围共四个灯改变状态，如图 3 所示。

2) 按动四条边上中间的按钮中的任意一个，则引起它所在边的三个灯改变状态，如图 4 所示。

3) 按动中心的那个按钮，则引起包括它及其上下左右共 5 个灯改变状态，如图 5 所示。

程序启动后，屏幕随机地给出一个魔阵，其中“ \circ ”表示灯暗，“ $*$ ”表示灯亮。在你输入按键的行列号以后，屏幕给出按该键后魔阵状态的变化情况，并再次请你输入；直到达到目标状态。这时计算机告诉你所用的步数共是多少，并问你是否想知道刚才这个魔阵的最优解，如果你打入“1”，表示要，计算机会打出最优解来。

程序设计思想与语句说明

设计思想

首先，我们把魔阵中 9 个灯编上号，从 1 - 9。相应的按钮号为 1 - 9。根据魔阵中灯的变化规则，我们可以列出按动 1 - 9 号按钮，相应的灯改变状态的表，如表 1。

例如，按下 3 号按钮，可使 2，3，5，6 号灯改变状态（亮变暗；暗变亮）。

显然，按下某个按钮，最多可使 5 个灯同时发生变化。为了记录这个规则，我们把上表记入数组 Q (9, 5) 中，以便于在程序运行时，当“按下”某一个按钮的同时，使相应的灯发生变化。

其次，我们还发现，为求得史密斯魔阵的最优解，可以得到以下的几个规律：

表 1

按下按钮	变化的灯
1	1 2 4 5 0
2	1 2 3 0 0
3	2 3 5 6 0
4	1 4 7 0 0
5	2 4 5 6 8
6	3 6 9 0 0
7	4 5 7 8 0
8	7 8 9 0 0
9	5 6 8 9 0

a. 按动左上角阵上的按钮, 仅引起左上角的灯改变状态, 如图 6 所示。魔阵旋转后的情况类似。

b. 按图 7 所示的过程按键, 仅引起第 1 行第 2 列的灯改变状态, 魔阵旋转后的情况类似。

c. 按图 8 所示的过程按键, 仅引起中心那个灯改变状态。

d. 对于任意一个按键, 仅当按动的次数为奇数时, 才能改变相应灯的状态。

根据以上规律, 不难找出要使某个灯改变状态, 要按哪些键, 具体对应关系见表 2。例如要使 4 号灯改变状态, 可连续按 2, 3, 4, 8, 9 五个按钮。为了在程序中应用此规律, 我们定义一个数组 P (9,6) 存放上表。

表 2

改变状态的灯	要按的键
1	1 2 3 4 5 7
2	2 4 6 7 9 0
3	1 2 3 5 6 9
4	2 3 4 8 9 0
5	2 4 5 6 8 0
6	1 2 6 7 8 0
7	1 4 5 7 8 9
8	1 3 4 6 8 0
9	3 5 6 7 8 9

上述 2 个数组和灯 (或按钮) 的对应关系如图 9。我们可以通过下标值 (A, B) 求得按钮号。如输入 (2, 1), 可找到按钮号为 4。再依表 1 可查到: 按 4 号按钮可使 1, 4, 7 号灯改变状态。程序中是通过算式:

$K = Q((A - 1) * 3 + B, I)$ 来计算哪个灯号应该改变状态的。其中 A、B 为下标值, 由键盘输入, I 为循环变量, 取值 1—5。找到了状态改变的灯号, 再通过下式把该灯的状态改变。

$$N(K) = (N(K) - 1) * (-1)$$

其中 N (K) 是工作数组, 存 9 个灯的状态。如果原来 N (K) = 1 (灯亮), 经上式后 N (K) = 0。否则 N (K) = 1。

这样, 我们就解决了按下魔阵中某个按钮, 魔阵中哪些灯状态改变的问题。

那么怎样求得史密斯魔阵的最优解呢? 有了表 2, 这个问题变得很简单。

我们的目的是使史密斯魔阵中只有中间的一个灯亮 (为 “*” 号), 其余灯全暗 (为 “○” 号)。那么, 我们就从 1 到 9 号灯逐个地查, 如果该灯亮着, 就按表 2 的规律, 按动相应的按钮把它变暗。

1	2	3
4	5	6
7	8	9

(1,1)	(1,2)	(1,3)
(2,1)	(2,2)	(2,3)
(3,1)	(3,2)	(3,3)

图 9

源程序

```

5 DIM M(9),N(9),P(9,6),C(9),Q(9,5),Y$(2,2)
10 FOR I=1 TO 9
15 FOR J=1 TO 6
20 READ P(I,J)
25 NEXT J
30 LET C(I)=0
35 NEXT I
40 FOR I=1 TO 9
45 FOR J=1 TO 5
50 READ Q(I,J)

```

```

55 NEXT J
60 NEXT I
65 PRINT
70 FOR I=1 TO 9
75 A=INT(RND(0)+.5)
80 M(I)=A
85 N(I)=A
90 IF A<>0 THEN GOTO105
95 P(I,0)=0
100 GOTO110
105 P(I,0)=1
110 NEXT I
115 P(5,0)=0
120 IF M(5)=1 THEN 130
125 P(5,0)=1
130 Y$(1,1) ="0"
133 Y$(2,2) ="*"
135 T=0
140 R=0
142 CLS
143 PRINT@250," "
145 FOR I=1 TO 3
150 FOR J=1 TO 3
155 A=N(I*3-3+J)+1

160 PRINTTAB(22);Y$(A,A);" ";
165 IF A=2 THEN R=R+1
170 NEXT J
175 PRINT
180 PRINT
185 NEXT I
190 IF R<>1 THEN 220
195 IF N(5)=0 THEN 220
200 PRINT"OK! YOU SUCCEED. YOU USED";T;"STEPS"
205 INPUT"DO YOU WANT TO KNOW THE OPTIMUM SOLUTION? 0-
1";A
210 IF A<>0 THEN 260
215 STOP
220 PRINT@950," ": INPUT"PRESS WHICH BOTTUM(I,J)";A,B
225 PRINT
230 FOR I=1 TO 5
235 K=Q((A-1)*3+B,I)
240 N(K)=(N(K)-1)*(-1)
245 NEXT I
250 T=T+1
255 GOTO140
260 FOR I=1 TO 9
265 IF P(I,0)=0 THEN 295
270 FOR J=1 TO 6
275 A=P(I,J)
280 M(A)=(M(A)-1)*(-1)
285 C(A)=C(A)+1
290 NEXT J
295 NEXT I
300 PRINT
305 PRINT"THE OPTIMUM SOLUTION IS:"
310 FOR I=1 TO 3
315 FOR J=1 TO 3
320 IF C(I*3-3+J)-INT(C(I*3-3+J)/2)*2=1 THEN PRINT("(";
I;",";J;")"

```

```

325 NEXT J
330 NEXT I
335 PRINT
340 INPUT"DO YOU WANT PLAY AGAIN? 0-1";A
345 IF A<>0 THEN 70
350 DATA 1,2,3,4,5,7,2,4,6,7,9,0,1,2,3,5,6,9,2,3,4,8,9
,0,2,4,5
355 DATA 6,8,0,1,2,6,7,8,0,1,4,5,7,8,9,1,3,4,6,8,0,3,5
,6,7,8,9
360 DATA 1,2,4,5,0,1,2,3,0,0,2,3,5,6,0,1,4,7,0,0,2,4,5
,6,8
365 DATA 3,6,9,0,0,4,5,7,8,0,7,8,9,0,0,5,6,8,9,0
370 END

```

语句说明

5~60语句，对 P，Q数组赋初值，即把表 1，表 2 分别读入 Q，P 两数组。
70~125语句，随机地设置一个史密斯魔阵，并与目标状态比较，所有与目标状态不同的灯号都记录在 P(i, 0) 中。
145~185语句，输出魔阵。
190~215语句，判别游戏是否结束。
220~250语句，输入按键号码，并根据 Q 数组改变相应灯的状态。
255~290语句，求史密斯魔阵的最优解。所有与目标状态不同的灯，按照 P 数组给出的指示改变状态，并对各灯在改变状态的过程中，相应按键的次数进行计数。
300~325语句，输出最优解。仅当按动次数为奇数的键才输出，这就是最优解了。

运行结果

```

RUN_
* * 0
0 * *
* * *
PRESS WHICH BOTTUM(I,J)? 1,1_
0 0 0
* 0 *
* * *
PRESS WHICH BOTTUM(I,J)? 3,3_
0 0 0
0 * 0
0 0 0

```

```

OK! YOU SUCCEED. YOU USED 19 STEPS
DO YOU WANT TO KNOW THE OPTIMUM SOLUTION? 0-1? 1_

```

```

THE OPTIMUM SOLUTION IS:
( 1 , 2 )
( 1 , 3 )
( 2 , 2 )
( 3 , 1 )
( 3 , 3 )

```

```

DO YOU WANT PLAY AGAIN? 0-1?

```

二十八、遥控汽车

周惠源 方伟

计算机模拟是计算机的一项重要应用。下面是在TRS-80计算机上设计的模拟活动图案一遥控汽车的游戏程序。此程序能使汽车向右、向左，停车、计算里程和交通信号灯等多项控制与显示。

程序的设计思想与说明

① 设计思想

TRS-80型机具有图示功能。用该机器的BASIC II语言的图示语句，可以把汽车的外形完整地显示在荧光屏上。然后用遥控汽车的功能键去控制汽车的向右、向左和停车。当欲使汽车向右行驶时，只给出x方向的增量，y方向不变，同时熄灭汽车原图形的图示块；欲使汽车向左行驶则给出x反方向的增量（x值变小），道理是一样的。

(2) 程序语句说明

2000~2050行，用汉语拼音打印游戏的提示符。

2100行，设置里程初值与交通信号灯。

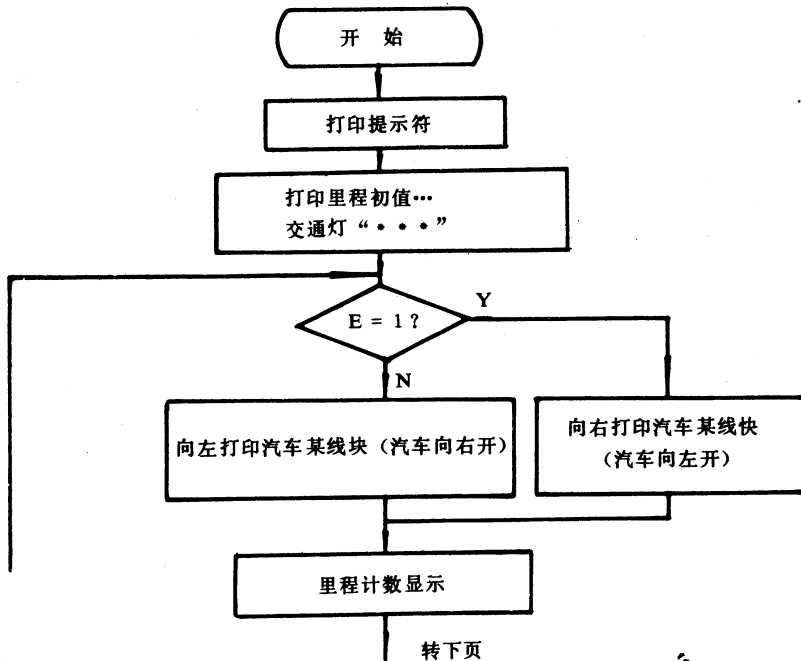
2115~2265行，打印汽车线条图块。

2267~2270行，显示里程计数。

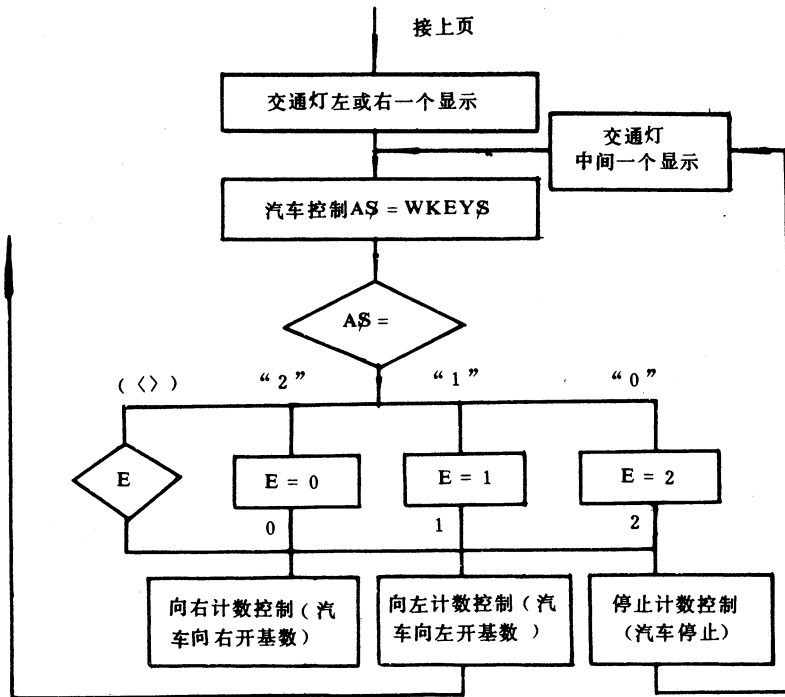
2272~2280行，显示交通信号灯。

2285~2352行，控制汽车的动作。

程序框图



源程序 1



```

10 CLEAR 100
2000 CLS:PRINT TAB(24)"YAO KONG QI CHE"
2010 PRINT :PRINT STRING$(64,45)
2020 PRINT"XIANG ZUO: /"
2030 PRINT "XING YOU : Z"
2040 PRINT"TING CHE : Q"
2050 PRINT:PRINTSTRING$(64,45)
2100 G=1 :POKE16002,48:POKE16003,48:POKE16004,48:POKE
16057,42:POKE16059,42:POKE16061,42
2115 SET(I,36):SET(I,37):SET(I,38)
2120 IF E=1 THEN Q=I+4 :IF Q>127 THEN Q=Q-128:GOTO2125
ELSE2125 ELSE Q=I-4
2125 IF Q<0 THEN J=128+Q:SET(J,39) ELSE SET(Q,39)
2130 IF E=1 THEN Q=I+5 :IF Q>127 THEN Q=Q-128 :GOTO 21
35 ELSE 2135ELSE Q=I-5
2135 IF Q<0 THEN J=128 +Q : SET(J,40) ELSE SET(Q,40)
2140 IF E=1 THEN Q=I+7 :IF Q>127 THEN Q=Q-128:GOTO 214
5 ELSE 2145 ELSE Q=I-7
2145 IF Q<0 THEN J=128+Q:SET(J,35)ELSE SET(Q,35)
2150 IF E=1 THEN Q=I+8 :IF Q>127 THEN Q=Q-128:GOTO 215
5 ELSE 2155 ELSE Q=I-8
2155 IF Q<0 THEN J=128 +Q:SET(J,34)ELSE SET(Q,34)
2160 IF E=1 THEN Q=I+9 :IF Q>127 THEN Q=Q-128: GOTO 21
65 ELSE 2165 ELSE Q=I-9
2165 IF Q<0 THEN J=128 +Q :SET(J,33):RESET(J,35):RESET
(J,40):ELSE SET(Q,33):RESET(Q,35):RESET(Q,40)
2170 IF E=1 THEN Q=I+10:IF Q>127 THEN Q=Q-128:GOTO2175
ELSE 2175 ELSE Q=I-10

```

```

2175 IF Q<0 THEN J=128+Q : RESET(J,34):RESET(J,39) ELSE
RESET(Q,34):RESET(Q,39)
2180 IF E=1 THEN Q=I+18 : IF Q>127 THEN Q=Q-128 : GOTO 2
185 ELSE 2185 ELSE Q=I-18
2185 IF Q<0 THEN J=128+Q:SET(J,34):SET(J,35) ELSE SET(Q
,34):SET(Q,35)
2190 IF E=1 THEN Q=I+20 : IF Q>127 THEN Q=Q-128: GOTO 2
195 ELSE 2195 ELSE Q=I-20
2195 IF Q<0 THEN J=128+Q : RESET(J,34):RESET(J,35) ELSE
RESET(Q,34):RESET(Q,35)
2200 IF E=1 THEN Q=I+128 : IF Q>127 THEN Q=Q-128 : GOTO
2205 ELSE 2205 ELSE Q=I-28
2205 IF Q<0 THEN J=128+Q: SET(J,34):SET(J,39) ELSE SET
(Q,34):SET(Q,39)
2210 IF E=1 THEN Q=I+29 : IF Q>127 THEN Q=Q-128:GOTO 22
15 ELSE 2215 ELSE Q=I-29
2215 IF Q<0 THEN J=128 +Q : SET(J,35):SET(J,40):RESET(J
,33) ELSE SET(Q,35):SET(Q,40):RESET(Q,33)
2220 IF E=1 THEN Q=I+30 : IF Q>127 THEN Q=Q-128:GOTO222
5 ELSE 2225 ELSE Q=I-30
2225 IF Q<0 THEN J=128+Q:RESET(J,34) ELSE RESET(Q,34)
2230 IF E=1 THEN Q=I+31:IF Q>127 THEN Q=Q-128: GOTO 22
35 ELSE 2235 ELSE Q=I-31
2235 IF Q<0 THEN J=128+Q :RESET(J,35) ELSE RESET(Q,35)
2240 IF E=1 THEN Q=I+33:IF Q>127 THEN Q=Q-128 : GOTO 2
245 ELSE 2245 ELSE Q=I-33
2245 IF Q<0 THEN J=128+Q:RESET(J,40) ELSE SET(Q,40)
2250 IF E=1 THEN Q=I+34:IF Q>127 THEN Q=Q-128 : GOTO 22
55 ELSE 2255 ELSE Q=I-34
2255 IF Q<0 THEN J=128 +Q :RESET(J,39) ELSE RESET(Q,39
)
2260 IF E=1 THEN Q=I+38 : IF Q>127 THEN Q=Q-128 : GOTO22
65 ELSE 2265 ELSE Q=I-38
2265 IF Q<0 THEN J=128+Q:RESET(J,36):RESET(J,37):RESET
(J,38) ELSE RESET(Q,36):RESET(Q,37):RESET(Q,38)
2267 IF W/2=INT(W/2) THEN 2272 ELSE U=U+1
2268 U$=STR$(U):L=LEN(U$):IF L<4 THEN C$=STRING$(4-L,"
")+U$ ELSE IF FU=1000 THEN U=0 : GOTO 2268
2269 FOR T=2 TO 4 : Y$=MID$(C$,T,1):IF Y$=" " THEN POK
E 16000+T,48 ELSE V=ASC(Y$):POKE 16000+T,V
2270 NEXT T
2272 IF G=1 THEN IF I/2=INT(I/2) THEN W=W+1:POKE 16057
,42 ELSE POKE 16057,128
2274 IF H=1 THEN IF I/2=INT(I/2) THEN W=W+1 : POKE 160
61,42 ELSE POKE 16061,128
2280 A$=INKEY$ :IF A$="" THEN 2330
2285 IF A$="Q" THEN E=2: F=1: POKE 16057,42:POKE 16061
,42: GOTO2280
2290 IF A$="Z" THEN POKE 16059,42:POKE 16061,42:IF H=1
THEN 2340 ELSE 2342
2292 IF E=1 THEN 2310
2295 IF A$="/" THEN POKE 16057,42:POKE 16059,42: IF G=1
THEN 2350 ELSE 2352
2300 I=I+1 : IF I=128 THEN I=0
2305 GOTO2115
2310 I=I-1 : IF I=-1 THEN I=127
2315 GOTO 2115

```

```

2330 ON E GOTO 2310,2336
2335 GOTO 2300
2336 IF Y/2=INT(Y/2) THEN POKE 16059,42:ELSE POKE 1605
9,128
2337 FOR K=1 TO 50 : NEXT :Y=Y+1 :GOTO 2280
2340 IF I>91 THEN I=I+36-128 ELSE I=I+36
2342 E=0 :F=0: G=1 :H=0: GOTO2300
2350 IF I<35 THEN I=I-36 +128 ELSE I=I-36
2352 E=1 :F=0: G=0 : H=1:GOTO 2310
2400 END

```

下面介绍的是另一个遥控汽车的程序。此程序设计的思想是：用 POKE 语句在显示器上作出一个汽车图形。汽车的左右行进控制也是通过 POKE 语句完成的，因为 POKE 语句作图比用 SET 语句快，所以此遥控汽车的速度比上一个要快。

因为程序较短，请读者自己分析。

源程序2

```

10 CLEAR 1000:DEFINT A-M:W=5:C=16059:Y=707:X=.5
20 READ A:IF A=-2 THEN CLS:GOTO80
30 IF A=-1 THEN A$=B$:B$="":GOTO20
40 IF A<50 AND A<>32 THEN READ B:C$=STRING$(A,B) ELSE C$=CHR$(A)
50 B$=B$+C$:GOTO20
60 DATA 32,160,158,4,131,4,131,173,144,48,32,19,191,47,32,13,9,143,135,9,32,139,143,135,32,-1
70 DATA 32,184,135,3,131,171,151,3,131,139,180,48,32,170,18,191,149,46,32,130,2,143,129,8,32,130,2,143,129,32,-2
80 U=Y:IF Y=U THEN C$=B$ ELSE C$=A$
90 Z=Z+ABS(X/2):PRINT@640,INT(Z);@Y,C$;
100 W=-W:POKE C+4*X,37+W
110 D$=INKEY$:IF D$<>" "THEN X=SGN(VAL(D$)-2)/2
120 Y=Y+X:IF Y<707 OR Y>879 THENY=T-X:Z=Z-ABS(X/2)
130 GOTO80

```

二十九、公路赛车游戏

刘炳文

当一辆高速赛车沿着弯曲得无法预测的公路行驶，会是怎样的一种情景呢？为了不让赛车驶出路面，司机必须准确地操纵赛车，否则就要冒撞车的危险。这里提供的程序所展示的就是高速赛车在公路上急驰的情景。由操作员充当司机，由程序控制公路的弯曲状态。

公路赛车游戏的难易程度完全由游戏者本人掌握。通过调整路面宽度和能见度参数，即可以很容易，也可以很困难。

游戏的方法

打入 RUN 后，程序首先显示初始图形，该图形为一段弯曲的公路，路面上标有“ROAD RACE”（公路赛车），然后要求操作员选择并输入两个参数：路面宽度和能见度。路面宽度可以在 4~15 之间任意选择，而能见度可在 1~4 之间选择。1~4 分别表示极坏、坏、尚好、

好四种情况。当能见度选为四时，赛车出现在屏幕的高处，这样，在沿着弯曲的公路行驶时视野就会开阔一些；当能见度选为1时，赛车出现在屏幕的低处，因此在驱车前进时只有短暂的观察时间。

设置好路面宽度和能见度之后，比赛就准备开始。此时赛车停在公路的起跑线上，公路的左侧有一个“起跑信号计数器”，从上到下有5个光块陆续闪亮，当最后一个光块闪亮时，赛车就开始起步。公路在屏幕上向前延伸，它的弯曲和转向是由程序随机控制的，游戏者必须准确地驾驶赛车，才能使它保持在跑道上。

赛车的控制通过键盘上的两个键来实现。压下“，”键将使赛车左移；而压下“.”键将使赛车右移。如果不按这两个键，赛车将向前直行。

赛车不停地前进，直到出了跑道才停下来。把每一次这样的“撞车”看作一天赛车的结束。在每个比赛日后，显示当前行驶的英里数，以及连续比赛数天后所行驶的累计里程。

每次撞车后，程序暂停执行，等待操作员按键，可以通过按“C”，“R”，“Q”键使比赛向下进行。选择“C”键将是在同样的路面条件下在新的一天里继续比赛，原来的累计英里数将被保留下来。按“R”键将是要重新进行比赛，这时可以改变路面条件（路面宽度和能见度），并重新计算比赛日。选择“Q”键是退出比赛，并让计算机直接返回到BASIC状态。后两种操作都将显示比赛中每天行驶的英里数。

程序中的几点说明

1. 要改变控制赛车左右移动的键是很容易的。游戏者可根据自己的习惯和方便，重新定义150语句中的两个字符串变量L\$和R\$。

2. 为了调整公路的弯曲度，可以修改130语句中LC的值。但LC必须选在0~0.5之间，否则跑道将会偏到一侧，并且不向前延伸。LC的值越大，公路的弯曲度越大，LC=0，公路变成直的。

3. 为了增加游戏的准确度和趣味性，还可以在路面上设置一些障碍物，这可以通过在语句600~650各个RETURN之前插入适当的PRINT或POKE语句来实现。但是一定要把地址变量POKE插入15360~16383之间，多余的操作可能会使系统产生中断。

程序流程与所用变量

120~180语句，预置变量。

200~290语句，得到由操作员输入的路面条件。

300~330语句，设置公路（赛车跑道）。

350~370语句，确定下一步的跑道状态。

400~420语句，实时修正赛车的位置。

430~550语句，一天赛事结束后的处理。

600~650语句，显示下一步的路段。

700~780语句，比赛开始的图形显示。

800~950语句，初始图形显示。

W：公路宽；V：能见度；M：当前比赛日的驱车里程；N：比赛天数；T：整个比赛驱车总里程；H：比赛所用的时间；L\$，R\$：控制赛车左、右移的字符串；L，R：公路左边、右边的位置；LC，RC：公路左右移动（即弯曲）的随机值；EL，ER：公路所允许的最左、最右位置；Q\$：操作员回答的信息；C：赛车位置，Z，ZC，ZP：为赛车地址POKE的自变量，RS\$，LS\$，LT\$，RT\$：显示公路路段字符串；B：空格字符；P：打

印位置；D：为显示信号POKE的自变量数组；I，K，Q：循环变量和工作变量；C₁，C₂：显示赛车的图示符号代码。

源程序和运行结果

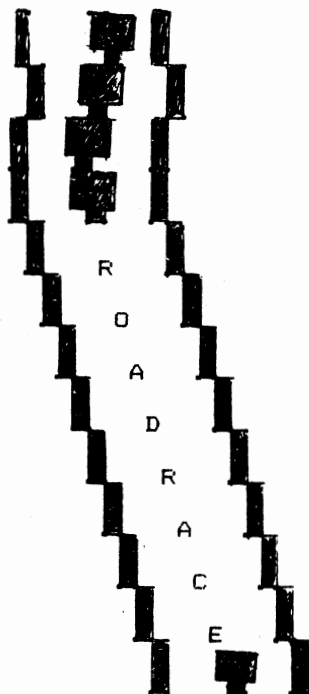
```
100 REM: ROADRACE
110 REM: COPYRIGHT 1979 BY PHIT FELDMAN AND TOM RUGG
120 CLEAR 200:RANDOM:Q=0:Q$=""
130 LC=.45:RC=1-LC
140 LS$=CHR$(191)+ CHR$(32):RS$=CHR$(32)+CHR$(191)
150 L$=",";R$="."
160 LT$=CHR$(184)+CHR$(135):RT$=CHR$(139)+CHR$(180)
170 B=32:EL=2:ER=58:C1=175:C2=159:DEFINT V, W
180 GOSUB 800
200 PRINT:T=0:N=0
210 INPUT"ROAD WIDTH (4-15)";W
220 IF W<4 OR W>15 THEN 200
230 PRINT:PRINT"VISIBILITY CONDITIONS"
240 PRINT" 1 - TERRIBLE"
250 PRINT" 2 - BAD"
260 PRINT" 3 - FAIR"
270 PRINT" 4 - GOOD":PRINT
280 INPUT"VISIBILITY (1-4)";V
290 IF V<1 OR V>4 THEN 280
300 N=N+1:L=24:R=L+W+2:Z=16384-128*V
310 C=INT((L+R)/2):H=0
320 FOR J=1 TO 16:Q$=INKEY$:GOSUB 600:NEXT
330 GOSUB 700
350 H=H+1:Q=RND(0):IF Q>RC AND R<ER THEN GOSUB 640:GOTO 400
360 IF Q<LC AND L>EL THEN GOSUB 620:GOTO 400
370 GOSUB 600
400 ZC=Z+C:ZP=ZC+1:Q$=INKEY$:IF Q$=L$ THEN C=C-1
410 IF Q$=R$ THEN C=C+1
420 IF PEEK(ZC)=B AND PEEK(ZP)=B THEN POKE ZC,C1:POKE ZP,C2:GOTO 350
430 FOR J=1 TO 8:Q$=INKEY$:POKE ZC,B:POKE ZP,B:FOR K=1 TO 50
440 NEXT:POKE ZC,C1:POKE ZP,C2:FOR K=1 TO 50:NEXT:NEXT
450 M=H*5:T=T+M:PRINT:PRINT"YOU WENT";M;"MILES FOR A"
460 PRINT"TOTAL OF";T;"MILES IN";N;"DAY(S)":PRINT
470 PRINT"HIT 'C' - CONTINUE RACE"
480 PRINT"      'R' - RESTART RACE"
490 PRINT"      'Q' - QUIT"
500 Q$=INKEY$:IF Q$="C" THEN 300
510 IF Q$<>"R" AND Q$<>"Q" THEN 500
520 PRINT:PRINT"AVERAGE MILES PER DAY=";T/N
540 IF Q$="R" THEN 200
550 END
600 PRINT TAB(L);RS$:TAB(R);LS$
610 RETURN
620 PRINT TAB(L);LT$:TAB(R-1);LT$
630 L=L-1:R=R-1:RETURN
```

```

640 PRINT TAB(L+1);RT$;TAB(R);RT$
650 L=L+1:R=R+1:RETURN
700 P=Z+C:GOSUB 950:P=Z+C-15360-20
710 Q$=CHR$(191)+CHR$(B)+"0"+CHR$(B)+CHR$(191)
720 P=P-320:PRINT @ P,STRING$(5,176);
730 FOR J=1 TO 5:P=P+64:PRINT @ P,Q$;:NEXT
740 PRINT @ P+64,STRING$(5,131);
750 FOR J=1 TO 600:NEXT
760 FOR J=4 TO 0 STEP -1:FOR K=1 TO 300:NEXT
770 PRINT @ P+2-64*J,CHR$(143);:NEXT:PRINT @ 896,CHR$(
B)
780 RETURN
800 DIM D(9):L=18:R=28:CLS
810 FOR J=1 TO 9:READ D(J):NEXT
820 DATA 82,79,65,68,82,65,67,69,32,
830 GOSUB 600
840 GOSUB 640
850 FOR J=1 TO 2:GOSUB 620:NEXT
860 FOR J=1 TO 10:GOSUB 640:NEXT
870 P=15383:POKE P,C1:POKE P+1,C2
880 FOR J=1 TO 500 :NEXT
890 P=P+65:GOSUB 950
900 FOR J=1 TO 2:P=P+63:GOSUB 950:NEXT
910 P=P+65:GOSUB 950
920 FOR J=1 TO 9:P=P+65:GOSUB 950
930 POKE P,D(J):POKE P+1,B:NEXT
950 POKE P,C1:POKE P+1,C2:FOR K=1 TO 50:NEXT:RETURN

```

RUN_



ROAD WIDTH (4-15)? 10_

VISIBILITY CONDITIONS

- 1 - TERRIBLE
- 2 - BAD
- 3 - FAIR
- 4 - GOOD

VISIBILITY (1-4)? 3

. .

. .

. .

三十、电子雪花

范幸义

雪花的形状千姿百态，图案十分优美，更有趣的是，在千万个雪花图案中，竟很难找到一样的。这和人类指纹一样，世界上很难找到有两人的指纹完全一样，这是由人的遗传编码来决定的。而雪花图案是由自然结构编码来决定的。虽然我们目前还不了解这些编码，造不出自然界中那样优美的雪花，但我们可以利用电子计算机来模拟雪花图案。这对于了解遗传编码，设想自然结构编码都是有益的，对于从事程序设计的人也是一件趣事。

本文介绍一个雪花模拟程序，运行这个程序，可以在显示器上，随机得到数以万计的不同形状的雪花图案。

程序的设计思想和语句说明

雪花的种类繁多，但它们都是六角对称图形，并可分成四种基本类型：1. 六角梅花型，2. 正六边形型，3. 六角蛛网型，4. 六角树枝型。本程序利用随机函数RND(x)，每次产生1~63之间的两个不同的整数，由这两个整数来选择TRS-80机的图示符号，用来构成图案。再把图案设计成四种基本类型，并由RND(x)函数产生一个1到4之间的整数，作为图案的类型选择，因此，一共可产生 $A_2^3 \times 4 = 15624$ 种不同的雪花图案。

语句说明

20~35 接收键盘空格键，程序开始运行，清屏。

40~230 在屏幕上打印一个*型线，把屏幕分成六部份。

240~260 产生1-63之间的两个不等值的伪随机整数。

270~290 产生1-4之间的随机整数，按该整数值判断转向，产生某一类型的雪花。

300~426 打印出六边型雪花程序。

430-520 打印六角芯部分图案。

530-630 打印六角蛛网型雪花图案。

635-680 打印六角梅花型雪花图案。

700-860 打印六角树枝型雪花图案。

源程序

```
20 X$=INKEY$
30 IF X$<>" " THEN 20
35 CLS
40 FOR X=20 TO 110:SET(X,25):NEXT X=42:X1=83
50 FOR Y=4 TO 44
60 X=X+1:X1=X1-1:GOSUB 200
70 NEXT
110 GOTO 240
200 FOR I=1 TO 2
210 SET(X+I,Y):SET(X1+I,Y)
220 NEXT
230 RETURN
240 S1=RND(63):S=S1+128
250 T1=RND(63):T=T1+128
260 IF S=T THEN 250
270 K1=RND(4)
280 ON K1 GOSUB 300,530,635,700
```

```

290 GOTO 20
300 FOR I=85 TO 108
310 POKE 15360+I,S:POKE 16192+I,S
320 NEXT :B=2
330 FOR I=148 TO 522 STEP 64
340 B=B-2
350 POKE 15360+I+B,S:POKE 15359+I+B,S:POKE 15841+I+B,S
:POKE 15840+I+B,S
360 NEXT
370 B=-2
380 FOR I=172 TO 501STEP 64
390 B=B+2
400 POKE 15360+I+B,S:POKE 15361+I+B,S:POKE 15777+I+B,S
:POKE 15776+I+B,S
410 NEXT I
420 POKE 15882,S:POKE 15927,S:
425 GOSUB 430
430 FOR I=411 TO 421
440 POKE 15360+I,T:POKE 15616+I,T
450 NEXT
460 FOR I=473 TO 487
470 POKE 15360+I,T:POKE 15488+I,T
480 NEXT
490 FOR I=535 TO 553
500 POKE 15360+I,T
510 NEXT
520 RETURN
530 A=INT(T/5):B=INT(S/10)
540 FOR Q=0 TO 6.28138 STEP 0.05
550 SET(A*COS(Q)+64,25-B*SIN(Q)):SET((A-B)*COS(Q)+64,2
5-(B-4)*SIN(Q)):SET((A-16)*COS(Q)+64,25-(B-8)*SIN(Q))
560 NEXT
570 FOR I=83 TO 91
580 POKE 15360+I,S:POKE 16192+I,S:POKE 15378+I,S:POKE
16210+I,S
590 NEXT
600 FOR I=455 TO 457
610 POKE 15360+I,S:POKE 15424+I,S:POKE 15488+I,S:POKE
15471+I,S:POKE 15535+I,S:POKE 15407+I,S
620 NEXT
630 RETURN
635 GOSUB 430
640 FOR Q=0 TO 6.28138 STEP 0.05
650 A=INT(S/10):B=INT(S/20)
660 SET(A*COS(Q)+48,11-B*SIN(Q)):SET(A*COS(Q)+80,11-B*
SIN(Q)):SET(A*COS(Q)+48,39-B*SIN(Q)):SET(A*COS(Q)+80,3
9-B*SIN(Q)):SET(A*COS(Q)+29,25-B*SIN(Q)):SET(A*COS(Q)+
100,25-B*SIN(Q))
670 NEXT
680 RETURN
700 FOR I=84 TO 86
710 POKE 15360+I,S:POKE 15382+I,S:POKE 15429+I,S:POKE
15441+I,S:POKE 15490+I,S:POKE 15508+I,S:POKE 16192+I,S
:POKE 16214+I,S:POKE 16133+I,S:POKE 16145+I,S
720 POKE 15360+I+705,S:POKE 15360+I+725,S
730 NEXT
740 FOR I=284 TO 287

```



```

750 POKE 15360+I,S:POKE 15366+I,S:POKE 15419+I,S:POKE
15433+I,S:POKE 15807+I,S:POKE 15813+I,S:POKE 15739+I,S
:POKE 15754+I,S
760 NEXT
770 FOR I=460 TO 462
780 POKE 15360+I,S:POKE 15360+I+5,S:POKE 15360+I+32,S:
POKE 15360+I+37,S:POKE 15360+I+128,S:POKE 15360+I+133,
S:POKE 15360+I+128+32,S:POKE 15360 +I+128+37,S:POKE 15
360+I+59,S:POKE15360+I+105,S
790 NEXT
800 FOR I=538 TO 549
810 POKE 15360+I,T
820 NEXT
830 FOR I=476 TO 483
840 POKE 15360+I,T:POKE 15488+I,T
850 NEXT
860 RETURN

```

程序采用键盘扫描，监视“空格”键。打入RUN命令后，按一下“空格”键，计算机则在屏幕上作出一个雪花图案，并把雪花保持在屏幕上，直至你再按“空格”键，又在屏幕上作出一个和原来不同的雪花图案。直到你中断程序运行为止。

三十一、跷跷板游戏

刘可

微型机有很强的图示功能。我们不仅可以用它来画表格，描绘函数图象及各种各样的图形，而且还能用它来进行动画表演。以微型机的显示屏幕为舞台，可以“演出栩栩如生的节目”给人们带来极大乐趣。

下面的程序是模仿儿童喜欢玩的跷跷板游戏编写的。它的使用很简单：将程序输入计算机后，只要打入“RUN”命令，“演出”就开始了。

程序运行后，首先显示一块标牌，宣布“演出”开始，接着是演员出场。然后进行跷跷板的“特技表演”。此时屏幕上显示出带有支架的“跷跷板”，一个“演员”先站在左端，另一个演员出现在右侧较高的位置上，正准备跳向板的右端。这位“演员”似乎有些“胆怯”，因而第一次没有敢跳，后来终于下定决心，跳了下来，并把左端的“演员”抛向空中；当左端的“演员”落下时，又把右端的“演员”跷上去，这样周而复始，跷跷板的动画表演便不停地进行下去。

一般来说，活动的图形显示程序分为两部分，第一部份为如何组织图形，第二部份为如何使图形动起来。在这个程序中，语句200至450为组织图形程序。它使用了一系列的字符串，根据预先设计好的“底图”，在屏幕上画出“跷跷板”和“演员”；而语句460至800是使图形动起来的程序，其中语句460调用语句1300和1400开始的子程序，显示标牌，预告“演员”；语句500至540调用语句900至970的子程序，显示表演器材（跷跷板）及“演员”出场，然后从语句590开始表演。

程序中使用的变量大部分为字符串变量。除字符串变量外，变量D、E为显示图形时的位置增量，L为支架（跷跷板）左端的位置，H为“演员”头部的位置。

“演员”在表演中“跳”的次数，由语句590中的循环终值决定。当前的程序为5次，如想跳20次可以把语句590改为：

```
590 FOR J= 1 TO 20
```

当然，也可以把它改为3，50，100，……。

语句1800为时间延迟，它使得程序在显示表演器材和“演员”出场时有一定节奏。为了改变延迟的时间，可以把语句中的1400换成不同的值，值越大，延迟时间越长。

修改语句1310中的字符串“TRS-80THEATRE”，可以改变游戏开始时的字幕。不过要注意，所用字符串不得超过16个字符，否则显示结果将超出字幕的标牌之外。例如，为了用“SIMON FENSTER”代替原来的字幕，可以把语句1310改为：

```
1310 PRINT @H, "SIMON FENSTER"; PRINT @H+E, "┌┌┌PROUDLY"
```

源程序

```
100 REM ***** PLAY ON A SEESAW *****
120 CLEAR 310:D=64:E=128
200 N$=CHR$(32):K$=CHR$(140):K1$=CHR$(131):K2$=CHR$(19
1)
210 B$=K1$+CHR$(171)+K2$+CHR$(151)+K1$:T$=K$+CHR$(188)
+K$
220 C$=CHR$(184)+CHR$(135)+N$+CHR$(139)+CHR$(180)
230 L$=STRING$(3,176):M$=STRING$(3,140)
240 U$=STRING$(3,131):S$=N$+N$:A$=S$+CHR$(188):W$=L$+M
$+U$
250 D$=S$+CHR$(188)+STRING$(15,N$)+W$
260 E$=B$+S$+S$+L$+T$+U$:Y$=S$+S$+N$:Y1$=STRING$(8,32)
270 Z$=CHR$(160)+CHR$(190)+K2$+CHR$(189)+CHR$(144)
280 F$=CHR$(184)+CHR$(183)+CHR$(176)+CHR$(143)+K$
290 F$=F$+K$+U$+S$+Z$:G$=STRING$(18,N$)+W$
320 J$=W$+S$+CHR$(160)+CHR$(190)+K2$+CHR$(189)
330 J$=J$+CHR$(144):H$=STRING$(9,N$)+L$+T$+U$
360 D1$=U$+M$+L$+STRING$(15,N$)+CHR$(188)
370 E1$=STRING$(9,N$)+U$+T$+L$+S$+S$+B$
380 F1$=STRING$(11,N$)+Z$+S$+U$+K$+K$
390 F1$=F1$+CHR$(143)+CHR$(176)+CHR$(187)+CHR$(180)
400 V1$=CHR$(131)+K$+CHR$(176)+CHR$(160)+CHR$(176)
410 V1$=V1$+CHR$(188):V2$=CHR$(176)+K$+CHR$(139)
420 V2$=V2$+CHR$(143)+K2$+K1$+K1$
430 V3$=CHR$(172)+CHR$(144)+N$+CHR$(160)+CHR$(156)
440 V4$=N$+CHR$(171)+CHR$(188)+CHR$(151)
450 V5$=K1$+K1$+K2$+K1$+K1$
460 GOSUB1300:GOSUB1400:GOSUB1800
500 CLS:L=783:GOSUB900:GOSUB1800:FOR H=770 TO 782:CLS
510 GOSUB900:GOSUB910:NEXT:CLS:GOSUB920:GOSUB1800
530 FOR H=298 TO 319:PRINT@H,K1$:NEXT:H=123:GOSUB910
540 GOSUB1800:GOSUB1200:H=101:GOSUB910
590 FOR J=1 TO 5
600 H=101:FOR K=1 TO 8:GOSUB940:H=H+D:GOSUB910
610 NEXT:H=591:CLS:GOSUB930:GOSUB910:FOR K=1 TO 8
620 GOSUB940:H=H-D:GOSUB910:NEXT:FOR K=1 TO 8:GOSUB940
630 H=H+D:GOSUB910:NEXT:H=613:CLS:GOSUB920:GOSUB910
640 FOR K=1 TO 8:GOSUB940:H=H-D:GOSUB910:NEXT:NEXT
```

```

700 FOR K=1 TO 3:GOSUB940:H=H+1:GOSUB910:NEXT
710 FOR K=1 TO 8:GOSUB940:H=H+D:GOSUB910:NEXT
720 GOSUB940:H=L-37:GOSUB950:GOSUB970:H=H+66:GOSUB960
800 GOSUB1800:CLS:PRINTCHR$(23)
810 PRINT@468,"F I N I S":GOSUB1800:CLS:END
900 PRINT@L,G$::PRINT@L+D,H$::PRINT@L+E,J$::RETURN
910 PRINT@H,A$::PRINT@H+D,D$::PRINT@H+E,C$::RETURN
920 PRINT@L,D$::PRINT@L+D,E$::PRINT@L+E,F$::RETURN
930 PRINT@L,D1$::PRINT@L+D,E1$::PRINT@L+E,F1$::RETURN
940 PRINT@H,Y$::PRINT@H+D,Y$::PRINT@H+E,Y$::RETURN
950 PRINT@H,V1$::PRINT@H+D,V2$::PRINT@H+E,K1$::RETURN
960 PRINT@H,V3$::PRINT@H+D,V4$::PRINT@H+D,V5$::RETURN
970 PRINT@H,Y1$::PRINT@H+D,Y1$::PRINT@H+E,Y1$::RETURN
1200 FOR H=123 TO 107 STEP -1:GOSUB910:GOSUB940:NEXT
1210 GOSUB910:GOSUB1800:FOR J=1 TO 3:GOSUB940:H=H-1
1220 GOSUB910:NEXT:GOSUB1800:FOR J=1 TO 6:GOSUB940:H=H
+1
1230 GOSUB910:NEXT:FOR K=1 TO 500:NEXT
1240 FOR J=1 TO 8:PRINT@H,S$+N$::FOR K=1 TO 50:NEXT
1250 PRINT@H,A$::FOR K=1 TO 50:NEXT:NEXT:GOSUB1800
1260 FOR J=1 TO 8:GOSUB940:H=H-1:GOSUB910:NEXT:RETURN
1300 CLS:PRINTCHR$(23):H=336:U$=CHR$(143)
1310 PRINT@H,"TRS-80 THEAIRE":PRINT@H+E," PROUDLY"
1330 PRINT@H+E+E," PRESENTS"
1340 FOR J=200 TO 242 STEP2:PRINT@J,U$:NEXT
1350 FOR J=200 TO 712 STEP2:PRINT@J,K2$::NEXT
1360 FOR J=242 TO 754 STEP2:PRINT@J,K2$::NEXT
1370 FOR J=712 TO 754 STEP2:PRINT@J,U$::NEXT
1380 GOSUB1800:FOR J=0 TO 26:PRINT@H+E+J,"-";
1390 FOR K=1 TO 90:NEXT:NEXT:GOSUB1800:RETURN
1400 M$="D N A":U$=" S S A W":W$=STRING$(16,32)
1410 L$="P L A Y":CLS:PRINTCHR$(23):FOR J=308 TO 260 S
TEP-2
1420 PRINT@J,W$:GOSUB1700:PRINT@J,L$:NEXT
1430 FOR J=552 TO 524 STEP-2:PRINT@J,W$:GOSUB1700:PRIN
T@J,M$
1440 NEXT:FOR J=802 TO 788 STEP-2:PRINT@J,W$:GOSUB1700
1450 PRINT@J,U$:NEXT:GOSUB1800:FOR J=36 TO 740 STEP2
1460 PRINT@J,"E":GOSUB1700:PRINT@J,N$::NEXT:PRINT@798
,"E";
1470 FOR J=40 TO 744 STEP2:PRINT@J,"E":GOSUB1700
1480 PRINT@J,N$::NEXT:PRINT@802,"E":RETURN
1700 FOR K=1 TO 12:NEXT:RETURN
1800 FOR I=1 TO 1400:NEXT:RETURN

```

三十二、打堡垒

张刚

这是在PC—1500机上实现的游戏程序。此程序可用RUN指令开工，也可用 **DEF** | **S** 标号开工。

程序运行后，显示屏上由左至右显示一条黑带，而后，一声长鸣，游戏开始。显示屏下部出现四个“凸”形符号，代表四个堡垒，上部有一自左向右移动的小黑点，代表载炮体，当“炮体”(小黑点)移至“堡垒”(凸形符号)的上方时，按下空格键(SPACE)，“炮体”

即向堡垒发射一发炮弹，若命中堡垒的要害部位（中间突出部分），则给计25分，并发出一串象征轰炸的音响，显示出所得分数，继续向右移动……，直至移到显示屏的右端为止。若未击中则扣掉10分。若炮体四次经过堡垒上方时均能击中，则得满分100分。若想结束程序的运行，可在显示屏上出现“Y、N?”时，打入“Y”以外的任何字符，否则可打入“Y”。

本程序可几个人轮流控制操作，以得分多者为胜。

源程序

```
100 : "S" CLEAR : CLS : WAIT 0 : FOR I=0 TO 155 : GCURSOR I : GPRINT  
INT 127- POINT I : NEXT I : BEEP 1, 200, 300  
105 : FOR I=10 TO 140  
110 : CLS : WAIT 1 : GCURSOR I : GPRINT 1  
113 : BEEP 1  
115 : GCURSOR 24 : GPRINT 64; 96; 64 : GCURSOR 59 : GPRINT 64;  
96; 64 : GCURSOR 94 : GPRINT 64; 96; 64 : GCURSOR 129 : GPRINT 64  
; 96; 64  
120 : A$ = INKEY$  
130 : IF A$ = " " THEN 150  
135 : NEXT I  
140 : PRINT (M+N) * 25 : FOR J=50 TO 150 : BEEP 1, J, 1000/J : NEX  
T J  
145 : INPUT "YN?", B$  
147 : IF B$ = "Y" THEN 100  
149 : END  
150 : GCURSOR I : WAIT 32 : GPRINT 127  
153 : IF I=25 OR I=60 OR I=95 OR I=130 BEEP 3 : N=N+1 : CURS  
OR I/6+1 : PRINT (M+N) * 25 : GOTO 160  
155 : M=M-0.4 : CURSOR I/6+1 : PRINT (M+N) * 25  
160 : GOTO 135
```

三十三、打飞机

张刚

本程序也是在PC-1500机上实现的。程序可用RUN命令开工，也可用DEF B标号开工。

程序运行后，一架飞机出现在显示屏上方，并不断地变换位置，显示屏下方有两门大炮（凸形符号），左侧的那门炮用空格键控制；右侧的那门用等号键（=）控制。击中飞机便计上10分，飞机消失，显示屏右端显示累计得分数，同时响起飞机坠毁的音响，以后新的一架飞机又出现在显示屏上。若按下SPACE或☐键时，未击中飞机，则扣去1分。

由于程序中飞机的位置采用随机函数来控制，所以，飞机的位置变幻莫测，难以捉摸，若想每次都击中飞机而又不错过任何一次机会，也并不是件容易的事。

当显示屏上出现“Y、N?”时，打入“Y”则程序重新开始执行，若打入“Y”以外的任一字符，程序都结束运行。

源程序

```
300 : "B" CLEAR
305 : FOR I=1 TO 3:FOR A=1 TO 152
310 : B=RND(A+50):WAIT 10
315 : IF B<A-20 OR B>A+50 THEN 310
317 : IF B>150 THEN 310
320 : CLS : BEEP 1:GDCURSOR B: GPRINT 2;2;7;2;2;3
325 : GDCURSOR A-1: GPRINT 64;96;64
326 : IF A>123 THEN 330
327 : GDCURSOR A+29:GPRINT 64;96;64
330 : A$=INKEY$
340 : IF A$=" " THEN 500
345 : IF A$="=" THEN 600
350 : NEXT A
360 : P=P+(M+N):PRINT(M+N)*10:BEEP 5,100,100:M=0,N=0
370 : NEXT I:PRINT P*10:FOR I=1 TO 20:BEEP 1,50,20:NEXT
  I:INPUT"YN?",P$
380 : IF P$="Y" THEN 300
390 : END
500 : GDCURSOR A:GPRINT 127
510 : IF A>=B AND A<=B+6 LET N=N+1:PRINT(M+N)*10:FOR J=3
  0 TO 70:BEEP 1,J,2000/J:NEXT J:GOTO530
520 : M=M-0.1
530 : GOTO350
600 : IF A>125 THEN 350
605 : GDCURSOR A+30:GPRINT 127
610 : IF A+30>=B AND A+24<=B LET N=N+1:PRINT (M+N)*10:
  FOR J=30 TO 70:BEEP 1,J,2000/J:NEXT J:GOTO630
620 : M=M-0.1
630 : GOTO350
```

三十四、射击

张刚

本程序也是在PC-1500机上实现的。程序可用RUN命令开工,也可用DEF F标号开工。

程序运行后,显示屏左侧出现一支枪,枪的右方有一个动荡的不断的靶子(小黑点),当枪口与靶子处于同一直线时,按下SPACE键,从枪口射出一发子弹,击中目标鸣笛四声,加记10分;若未击中目标,扣去一分。其他项与打堡垒、打飞机程序大致相同。

源程序

```
1000 : "F" CLEAR:WAIT 0:DIM A(150),B(7)
1100 : FOR K=1 TO 3:FOR I=50 TO 150STEP 5
1130 : A(I)=RND(I)
1140 : IF A(I)<40GOTO1130
1150 : FOR J=0 TO 7
1200 : B(J)=RND(J)
1300 : CLS :GDCURSOR A(I):BEEP 1:GPRINT 2[B(J)
1330 : WAIT 10: GDCURSOR 25:GPRINT 56;24;8;8;8
1350 : A$=INKEY$
1360 : IF A$=" "GOSUB 2000
1400 : NEXT J:NEXT I:FOR O=50 TO 100:BEEP 1,I,10:NEXT O
```

```

1500 :WAIT 64: PRINT(N+M)*10
1600 :NEXT K
1610 :FOR X=1 TO 16
1615 :RESTORE
1620 :READ B,C
1630 :BEEP 1,B,(60*C)
1640 :NEXT X
1650 :DATA 245,0.3,220,0.3,200,0.5,180,0.5,160,1.5,118
,1,145,1.5,118,1.5,160,3
1660 :DATA 180,1,180,1,193,1,193,1,220,1,220,1,245,2
1700 :INPUT"Y,N,?",P$
1800 :IF P$="Y" THEN 1000
1900 :END
2000 :WAIT 0:GDCURSOR 25:GPRINT 56;24;8;8;8:FOR L=31 TO
A(I):GDCURSOR L:GPRINT 8:NEXT L
2100 :IF 2[IB(J)=8 BEEP 3:N=N+1:GOTO2300
2200 :M=M-0.1
2300 :RETURN

```

三十五、电脑识谱

钱翔

当你拿到一段简谱时，你能完整地把它唱出来吗？如果不行，可以请计算机来帮忙。

运行下面这段程序，计算机首先问你总的音符数有多少个？如果你准备输入的曲子有100个音符，你就键入100。随后你就可以根据计算机的提问，不断地把乐谱中的音符与相应的节拍顺序输入，100个音符输完后，计算机就开始为你演奏你刚输入的曲子了。如果你还想再听几遍，可在计算机显示：“AGAIN Y/N？”时键入“Y”。

例如要让电脑演奏下面这段曲子 | 6 6565 3 5332……可进行如下操作：

RUN

ZYFS (总音符数) ? 100

YF (音符) ? 6

pi (拍) ? 1

YF? 6

pi? 1/2

YF? 5

pi? 1/2

YF? 6

pi? 1/2

⋮

⋮

⋮

⋮

键盘与音符的对应关系如下：

7	1	2	⋯	7	1	2
---	---	---	---	---	---	---

程序中1010语句输入总音符数；1030~1060语句输入音符与拍数；1080~1170语句判断所输入的音符；1500语句是发声语句，它的格式如下：BEEP a_1 , a_2 , a_3 。其中 a_1 是发音的次数， a_2 是音调， a_3 是每声的长短。

本程序是在PC-1500袖珍计算机上实现的,如果将发声语句与数据作相应的改动,也可在其它微机上实现。

源程序

```
1000 CLEAR
1010 INPUT "ZYFS?";A
1020 DIM B(200),C(200)
1030 FOR I=0 TO A-1
1040 INPUT "YF?";B(I)
1050 INPUT "PI?";C(I)
1060 NEXT I
1070 D=60:PRINT "          START"
1080 FOR I=0 TO A
1085 IF B(I)=0 LET N=255:GOTO1500
1090 IF B(I)=1 LET N=245 :GOTO 1500
1100 IF B(I)=2 LET N=220:GOTO1500
1110 IF B(I)=3 LET N=195:GOTO1500
1120 IF B(I)=4 LET N=180 : GOTO1500
1130 IF B(I)=5 LET N=160:GOTO1500
1140 IF B(I)=6 LET N=143 :GOTO1500
1150 IF B(I)=7 LET N=130 :GOTO1500
1160 IF B(I)=8 LET N=100:GOTO1500
1170 IF B(I)=9 LET N=90:GOTO1500
1500 BEEP 1,N,C(I)*D
1510 NEXT I
1520 INPUT "AGAIN Y/N?";A$
1530 IF A$="Y" GOTO1080
1540 END
```

三十六、美妙动听的音乐

徐玉英

WICTOR-9000微型计算机具有独特的音频分辨能力。利用它的这一特点,可以让它奏出美妙动听的音乐,其音色与电子琴非常相似。而且编程简单,使用方便。

使用GW-BASIC语言,主要利用它的音响SOUND语句。其格式为:SOUND<frequency>,<duration>,SOUND功能是按给定的频率和持续时间从扬声器发出音响。<frequency>是以赫兹为单位的频率,为有效数值表达式,其值域为无符号整数37至32767;<duration>是扬声器发声的持续时间,为有效值表达式,其值域是无符号整数0至65535。在编程时,可根据每支歌曲的共同特点来简化程序。其思路是利用每支歌曲都是由音阶1、2、3、4、5、6、7和它们的高低音组成,把频率赋给相应的音阶,然后再把音阶编成一段小程序(见程序1)。当你要输入歌谱时,先把程序1调出来,再加上DATA数据语句。这样,一首歌曲只需十分钟左右就可输入完毕。

程序说明:

20语句,定义音阶数组和音频数组;

30~50语句,求音频数值;

60语句,定义音阶1,2,3,4,5,6,7;

70~75语句, 定义音阶 $\dot{1}, \dot{2}, \dot{3}, \dot{4}, \dot{5}, \dot{6}, \dot{7}, ;$;

80~85语句, 定义音阶 $1, 2, 3, 4, 5, 6, 7, ;$;

90 语句, 定义休止符 0。定义休止符时, 可以在休止符处让它模拟鼓点声, 也可让它不发声;

100~130语句, 读数据语句;

120 语句, 音响语句, 利用READ语句读入的数据发声;

110~140语句, 不断读入数据并发音响, 直到读到小于等于-1的数据结束。

程序2可使VICTOR9000演奏出大家喜欢的歌曲“我的中国心”。

程序1

```
10 REM
20 DIM D(32), F(36)
30 FOR I=0 TO 36
40 F(I)=220*2[(I/12)]
50 NEXT I
60 D(1)=F(0):D(2)=F(2):D(3)=F(4):D(4)=F(5):D(5)=F(7):
D(6)=F(9):D(7)=F(11)
70 D(11)=F(12):D(12)=2*D(2):D(13)=2*D(3):D(14)=2*D(4):
D(15)=2*D(5)
75 D(16)=2*D(6):D(17)=2*D(7)
80 D(26)=.5*D(6):D(27)=.5*D(7):D(25)=.5*D(5):D(24)=.5*
D(4):D(23)=.5*D(3)
85 D(22)=.5*D(2):D(21)=.5*D(1)
90 D(9)=.2*D(1)
100 READ A,B
110 WHILE A>-1
120 SOUND D(A),B
130 READ A,B
140 WEND
```

程序2

```
10 REM MY CHINA HEART
20 DIM D(32), F(36)
30 FOR I=0 TO 36
40 F(I)=20*2[(I/12)]
50 NEXT I
60 D(1)=F(0):D(2)=F(2):D(3)=F(4):D(4)=F(5):D(5)=F(7):D
(6)=F(9):D(7)=F(11)
70 D(11)=F(12):D(12)=2*D(2):D(13)=2*D(3):D(14)=2*D(4):
D(15)=2*D(5):D(16)=2*D(6)
80 D(26)=.5*D(6):D(27)=.5*D(7):D(25)=.5*D(5):D(24)=.5*
D(4):D(23)=.5*D(3)
85 D(22)=.5*D(2):D(21)=.5*D(1):D(17)=2*D(7):D(9)=.2*D(
1)
100 READ A,B
110 WHILE A>-1
120 SOUND D(A),B
130 READ A,B
140 WEND
200 DATA 3,18,6,18,11,9,7,4,5,6,4.5,13,9,11,9,6,72,3,1
8,13,18
```


210 DATA 12, 9, 13, 4.5, 12, 4.5, 11, 9, 7, 9, 6, 63, 9, 9
220 DATA 26, 13.5, 3, 4.5, 2, 4.5, 3, 4.5, 1, 4.5, 27, 4.5, 26, 27,
9, 9
230 DATA 3, 4.5, 6, 9, 5, 4.5, 3, 4.5, 2, 4.5, 1, 4.5, 2, 4.5, 3, 27,
3, 4.5, 5, 4.5
240 DATA 6, 13.5, 7, 4.5, 6, 4.5, 5, 4.5, 3, 4.5, 2, 4.5, 1, 4.5, 1,
9, 2, 4.5, 3, 18
250 DATA 2, 13.5, 3, 4.5, 27, 4.5, 26, 4.5, 25, 9, 26, 27, 9, 9
260 DATA 26, 13.5, 3, 4.5, 2, 4.5, 3, 4.5, 1, 4.5, 27, 4.5, 26, 27,
9, 9
270 DATA 3, 4.5, 6, 9, 5, 4.5, 3, 4.5, 2, 4.5, 1, 4.5, 2, 4.5, 3, 27,
3, 4.5, 5, 4.5
280 DATA 6, 13.5, 7, 4.5, 6, 4.5, 5, 4.5, 3, 4.5, 2, 4.5, 1, 4.5, 1,
9, 2, 4.5, 3, 18
290 DATA 2, 13.5, 3, 4.5, 27, 4.5, 26, 4.5, 25, 9, 26, 27, 9, 4.5, 3
, 4.5
300 DATA 5, 13.5, 3, 4.5, 3, 9, 9, 4.5, 3, 4.5, 11, 13.5, 6, 4.5, 6,
9, 6, 4.5, 11, 4.5
310 DATA 6, 9, 5, 9, 1, 9, 2, 4.5, 1, 2.25, 2, 2.25, 3, 27, 9, 4.5, 3,
4.5
320 DATA 11, 13.5, 6, 4.5, 6, 13.5, 6, 4.5, 11, 13.5, 12, 4.5, 13,
27
330 DATA 13, 4.5, 12, 4.5, 7, 13.5, 5, 4.5, 6, 27, 9, 9
340 DATA 26, 13.5, 3, 4.5, 2, 4.5, 3, 4.5, 1, 4.5, 27, 4.5, 26, 27,
9, 9
350 DATA 3, 4.5, 6, 9, 5, 4.5, 3, 4.5, 2, 4.5, 1, 4.5, 2, 4.5, 3, 27,
3, 4.5, 5, 4.5
360 DATA 6, 13.5, 7, 4.5, 6, 4.5, 5, 4.5, 3, 4.5, 2, 4.5, 1, 4.5, 1,
9, 2, 4.5, 3, 18
370 DATA 2, 13.5, 3, 4.5, 7, 4.5, 6, 4.5, 5, 9, 6, 27, 9, 4.5, 3, 1.5
, 5, 1.5, 6, 1.5
380 DATA 11, 13.5, 7, 4.5, 7, 13.5, 6, 4.5, 6, 13.5, 3, 4.5, 3, 18
390 DATA 26, 13.4, 4, 4.5, 4, 9, 3, 9, 2, 27, 9, 9
400 DATA 7, 13.5, 6, 4.5, 6, 13.5, 5, 4.5, 5, 13.5, 2, 4.5, 2, 18
410 DATA 9, 4.5, 3, 4.5, 1, 4.5, 2, 4.5, 3, 13.5, , 7, 4.5, 6, 27, 9,
4.5, 3, 4.5
420 DATA 5, 13.5, 3, 4.5, 3, 9, 9, 4.5, 3, 4.5, 11, 13.5, 6, 4.5, 6,
9, 6, 4.5, 11, 4.5
430 DATA 6, 9, 5, 9, 1, 9, 2, 4.5, 21, 2.25, 2, 2.25, 3, 27, 9, 4.5, 3
, 4.5
440 DATA 11, 13.5, 6, 4.5, 6, 13.5, 6, 4.5, 11, 13.5, 12, 4.5, 13,
27
450 DATA 13, 4.5, 12, 4.5, 7, 13.5, 5, 4.5, 6, 27, 9, 9
460 DATA 26, 13.5, 3, 4.5, 2, 4.5, 3, 4.5, 1, 4.5, 27, 4.5, 26, 27,
9, 9
470 DATA 3, 4.5, 6, 9, 5, 4.5, 3, 4.5, 2, 4.5, 1, 4.5, 2, 4.5, 3, 27,
3, 4.5, 5, 4.5
480 DATA 6, 13.5, 7, 4.5, 6, 4.5, 5, 4.5, 3, 4.5, 2, 4.5, 1, 4.5, 1,
9, 2, 4.5, 3, 18
490 DATA 2, 13.5, 3, 4.5, 7, 4.5, 6, 4.5, 5, 9, 6, 63, -1, -1

三十七、模拟钢琴程序

范幸义

在计算机上模拟钢琴是一个很有趣的事情。下面介绍的程序是利用键盘的键作琴键，在 TRS-80 机上模拟钢琴的程序。

程序的设计思想和语句说明

●设计思想

a. 发声原理 每台计算机都有一个时钟主振频率，这是计算机设计指令系统所特有的指令时间。如果用计算机迅速启动某种外设，然后又迅速关闭某种外设，这一开一关的速度用指令进行控制，则计算机可以产生一个振荡频率。控制这个频率，使它成为某一音符的频率，计算机就会发出声音了。

TRS-80 微机指令系统中的 OUT 指令，是把寄存器的内容输出到某个通道口。地址 OFFH 是盒式磁带机启动地址。利用指令 OUT，可以启动和关闭磁带机。例如，指令

```
LD A, 4
```

```
OUT (OFFH), A
```

用来启动磁带机马达。当迅速开/关马达时，可以得到一个矩形波音调。利用 B、C 寄存器和 A 累加器，能编写单音符发音子程序：

```
LD IX, INT          取音符信息地址
LD C, (IX)          取音符时间长度 (节拍)
LOP LD B, (IX+01H)  取音符频率
LD A, 05H
OUT (0FFH), A      打开半周期
LOP1 DJNZ LOP1     频率延时
LD B, (IX+01H)     取频率
LD A, 02H
OUT (0FFH), A      打开另一半周期
LOP2 DTNZ LOP2     频率延时
DEC C              长度减一
JP NZ, LOP         长度不够时继续
RET                返回
```

从某个地址开始，相邻两个地址存入两个字节的数值，第一个字节是音符的长度，第二个字节是音符的频率。然后调用子程序，就可以发出相应的声音。若把音符排成一个音符表，按音符表调用子程序，就可以演奏曲子了。

b. 模拟钢琴 用键盘上的键来选择音符表中某一个音符，调用发音子程序发音，计算机就具有模拟钢琴的功能了。用不同的键来表示不同的音符，当按下相应的键，计算机就能发出相应的声音。因此程序分两部分，一部分是键盘扫描，一部份是发音子程序和音符表，音符表中的音符按钢琴的低音、中音、高音排成。由按键来选择某一音符。如一按键就能发音，若按住键不放就能连续发音（弹曲时，音符的发音长短各不相同）。因要求计算机执行程序的速度快，故程序用机器语言写成。为了使用方便，采用 BASIC 语言置入内存的方法，故

本程序也是一个BASIC程序。前部份把机器语言置入内存，並定义这个机器语言子程序。后一部份是在屏幕上显示“电子琴”三个大字，然后调用机器语言子程序，就能在键盘上弹出优美的曲子。

TRS-80机键盘共有53个键，用其中21个键作为音符键，每个键代表的音符如图所示。



音符 1 2 3 4 5 6 7



音符 1 2 3 4 5 6 7 i 2 3 4 5 6 7

为了使声音响亮，可以用录音机发音。运行本程序时，把TRS-80机随机配备的CTR-80盒式磁带机上的黑色插头拔出不用，把灰色大、小插头仍插在磁带上，取出磁带，並按下PLAY键，使之处于放音状态，並开大音量电位器，就可以运行本程序。利用TRS-80机本身的磁带机，音量不够理想。如采用具有扩音功能的录音机，在运行时，键盘上弹出的乐曲，其声音将更优美、宏亮。对一些比较低档的录音机，处于放音状态时，计算机输出的信号会在磁头处对地短路，故需把录音机打到录音状态。

● 语句说明

本程序比较简单，但它涉及到机器语言程序，对于只会BASIC语言的人来说就显得困难了。由于机器语言已经超出本书的范围，故不介绍。

源程序

```

10 FOR I=0 TO 196
20 READ X:POKE &H7A00+I,X
30 NEXT
40 DATA 14,0,58,1,56,183,194,54,122,14,8,58,2,56,183,1
94,54,122,14,16
50 DATA 58,4,56,183,194,54,122,14,24,58,8,56,183,194,5
4,122,14,27
60 DATA 58,16,56,183,194,54,122,58,64,56,183,202,0,122
,14,35,6
70 DATA 255,4,203,63,194,56,122,120,129,79,6,0,221,33,
78,122
80 DATA 203,33,221,9,195,164,122,1,34,0,0,0,0,0,0,0,0,
1,185,0
90 DATA 0,0,0,0,1,49,0,0,0,0,0,0,0,0,0,0,1,36,1,35,1,2
29,1,165
100 DATA 0,0,1,147,1,52,0,0,1,198,0,0,1,129,0,0,0,0,1,
117,1,104,1,91
110 DATA 1,79,1,70,1,60,1,52,0,0,0,0,0,0,1,255,0,0,1,1
9,1,15,0,0,221,78,0,121
120 DATA 183,202,0,122,221,70,1,62,5,211,255,16,254,22
1,70,1,62,2,211,255
130 DATA 16,254,13,194,72,122,195,0,122,32,32
140 CLS

```

源程序

```
150 FOR K=15627 TO 15920
160 READ B:POKE K,B
170 NEXT K
180 DATA 160,32,32,32,32,32,32,32,32,32,32,32,32,176,1
76,176,176,176,176,32,32,32,32,32,32,32,32,160,1
76,176,176,160,176,176
190 DATA176,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
32,32,32,32,32,32,32,32,32,32,156,140,174,140,140,1
72,32,32,32,32,32,32,32,32,32,32,32,32,176,158,129
200 DATA 32,32,32,32,32,32,32,32,32,32,136,140,157,140
,136,140,157,140,32,32,32,32,32,32,32,32,32,32,32,3
2,32,32,32,32,32,32,32,32,32,32,32,157,140,174,140,
140,174
210 DATA 32,32,32,32,32,32,32,32,32,140,140,140,141,188,1
40,140,32,32,32,32,32,32,32,32,32,32,130,131,131,179,1
66,147,131,131,32,32,32,32,32,32,32,32,32,32,32,32,
32,32,32,32,32,32,32,32,32,32,32
220 DATA 141,140,174,140,140,142,32,32,32,32,32,32,32,
32,32,32,32,32,32,130,173,32,32,32,32,32,32,32,32,3
2,168,142,131,129,132,131,139,172
230 DATA 32,32,32,32,32,32,32,32,32,32,32,32,32,32,32
,32,32,32,32,32,32,32,32,32,32,32,32,170,176,176,184,3
2,32,32,32,32,32,32,32,32,32,170,176,176,184,135
240 DATA 32,32,32,32,32,32,32,32,32,32,130,130,131,131
,131,187,135,130
250 DEFUSR=%H7A00
260 Y=USR(1)
```

改进意见

上述程序在运行时，启动录音机的继电器始终在动作，这样会影响继电器的寿命。

TRS-80机的0FFH口除启停录音机电机外，还可以控制屏幕显示格式；利用D₁和D₀位，还可以在0FFH口得到不同波形。而启动录音机用的是D₂位，因此，只要将原发音子程序中的：

```
LDA, 05H
OUT (0FFH), A
```

改成

```
LDA, 01H
OUT (0FFH), A
```

就可以使继电器不动作，而达到录音机扬声器放音的同样效果。

具体改法：只要将原程序120行DATA语句中的第9个数据5改成1即可。

三十八 孙子点兵

陈涤平 金洪明

魏晋南北朝的数学典籍《孙子算经》中有这样一道数学古题：“今有物不知其数，三三数之剩二；五五数之剩三；七七数之剩二。问物几何？答曰：二十三。”

到了明朝，数学家程大位编写的《算法统宗》里将此题算法编成一首诗，名叫“孙子点兵歌”，国外也有称之谓“中国余数定理”的。

```
三人同行七十稀，
五树梅花廿一枝，
```

七子团圆正半月，
除百零五便得知。

意思是：以 3 为模的余数去乘 70，以 5 为模的余数去乘 21，以 7 为模的余数去乘 15，然后将这三个乘积相加，再加上或减去若干个 105，答数就出来了。

相传，著名军事家孙臆在点将台上阅兵，只要变换队阵为三纵队，五纵队，七纵队，就能正确地估算出这支队伍的具体人数。孙子所以能点兵如神，使用的就是余数计算。

比如，三纵队时余二人；五纵队时余三人；七纵队时正好满员，余数为零；整支队伍估计约 500 人左右。按照“孙子点兵歌”可以算得：

$$X = 70 \times 2 + 21 \times 3 + 15 \times 0 = 203$$

然后递加 105，直到接近 500 为止，算得这支队伍的具体人数为 518 人。

程序一即为用上述算法编制的程序。其中 a 为三纵队余数，b 为五纵队余数，c 为七纵队余数，M 为阅兵队伍目测估计数，X 为计算所得具体数。

下面介绍另一种解法的孙子点兵程序（程序 2）

孙子点兵，实际上是求这样一个数，这个数用 Q 除余 M，用 R 除余 N，用 T 除余 L。

此题用数学解决的基本思想是：

令 A 是这个数 X 除 Q 的整数商，则

$$A = \frac{X - M}{Q}$$

B 是 X 除 R 的整商，则

$$B = \frac{X - N}{R}$$

C 是 X 除 T 的整商，则

$$C = \frac{X - L}{T}$$

其中，M，N，L 分别为余数。

可是，上述 A，B，C，X，M，N，L，Q，R，T 为正整数。则

$$A + B + C = \frac{(QR + QT + RT) X - (M \cdot R \cdot T + N \cdot Q \cdot T + L \cdot Q \cdot R)}{Q \cdot R \cdot T} = K$$

也是正整数，可得

$$X = \frac{Q \cdot R \cdot T \cdot K + M \cdot R \cdot T + N \cdot Q \cdot T + L \cdot Q \cdot R}{Q \cdot R + Q \cdot T + R \cdot T}$$

显然， $A + B + C \geq 1 + 2 + 3 = 6$

即 $K = 6, 7, 8, \dots$

只要将 K 的各个值代入上式求得 X 刚好是一个正整数即可，于是可编出程序 2。

其中第 10 语句中 Q，R，T 为模数，M，N，L 分别为被 Q 除，R 除，T 除的余数。此程序对于任何一个数，只要知道该数被任意三个数除的余数，该数即可求得。

源程序1 及运行结果

```

10 INPUT A,B,C,M          RUN
20 IF A>2 THEN 140
30 IF B>4 THEN 140       ? 2,3,0,500
40 IF C>6 THEN 140
50 X=70*A+21*B+15*C      518
60 IF X-M>52 THEN 110
70 IF X>M THEN 120      READY
80 X=X+105
90 IF X<M THEN 80        >_
100 IF X-M<52 THEN 120
110 X=X-105
120 PRINTX
130 GOTO150
140 PRINT"INPUT ERROR!"
150 END

```

源程序2 及运行结果

```

10 INPUT Q,R,T,M,N,L      RUN
20 U=Q*R                  RUN
30 V=Q*T                  ? 3,5,7,2,3,2   ? 3,5,7,2,4,6
40 W=R*T
50 K=6
60 K=K+1
70 X=(U*T*K+M*W+N*V+L*U)/(U+V+W)  23          104
80 IF X=INT(X) THEN 100
90 GOTO60                 READY
100 PRINT                 READY
110 PRINTX                >_
130 END                   >_

```

三十九、解九连环

王晓平 刘立新

解九连环是中国传统的一种锻炼智力的游戏，其结构是九个串连起来的环穿在一根铁丝上，只有第一个环可以随意装上拆下。要想装上或拆下第 i 个环，必须第 $i-1$ 个环在上边而其余第 $i-2, i-3, \dots, 2, 1$ 个环在下边。只能从一边拆，全拆下来为结束。传说中国古代曾用此代锁，可见其解法有一定复杂性。下面介绍一个解任意 N 个环的游戏程序。

程序说明

在程序中使用了数组 $K(N)$ 来记这 N 个环的上下情况，用 $J(N+2)$ 来记过程结束时的返回地址。此程序是用递归过程解决的。特征标志是用数组 $Z(N+2)$ ，当进入拆下过程时 $Z(N+2)$ 赋值1，进入装上过程时 $Z(N+2)$ 赋值0。判断语句判断当 $Z(N+2)=0$ 时即为从装上过程转回的，当 $Z(N+2)=1$ 时是从拆下过程转回的。

整个拆下 N 个环的过程是一个递归过程。语句7040~7160是拆下第 N 个环，语句7180~

7230是拆下其余的 $N - 1$ 个环。而拆下第 N 个环必须 $N - 1$ 个环装上，即语句7070~7090，其余 $N - 2$ 个环都拆下，即语句7110~7130。7040是判断语句，当 $N \leq 0$ 时表示不调用此过程，返回。7050判断语句，当 $Z(N + 2) = K(N)$ 时表示第 N 个环已装上或拆下，不须再调用过程，返回。7060语句，当 $N = 1$ 时表示是第一个环，可直接装上或拆下。每一次递归调用有三条语句：

$N = N - (1 \text{ 或 } 2)$ 移动指针

$Z(N + 2) = (0 \text{ 或 } 1)$ 存工作特征标志。“0”表示上，“1”表示下。

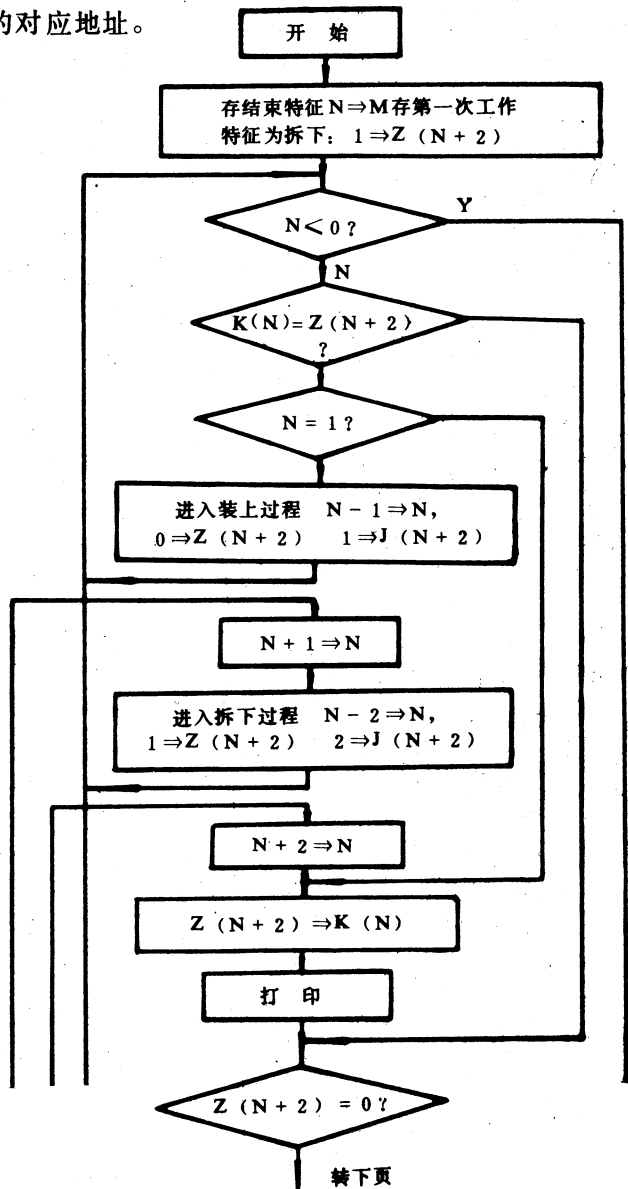
$J(N + 2) = (1, 2, 3)$ 存返回地址特征。

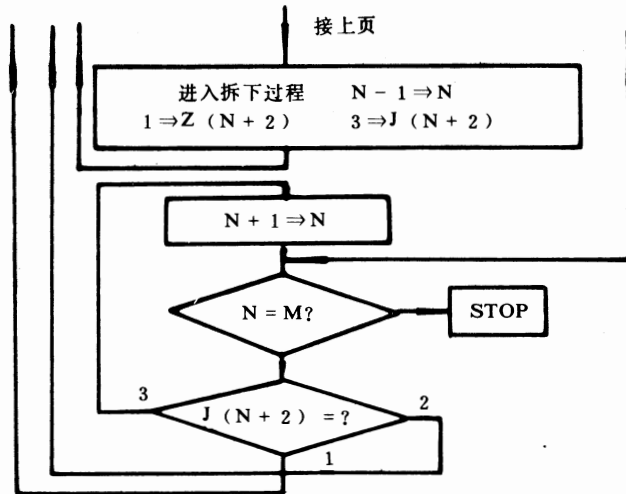
每一次过程返回后有一条

$N = N + (1 \text{ 或 } 2)$ 恢复指针

语句7220，如果 $N = M$ 表示整个拆下过程已结束。语句7230 是每一个子过程的最后一条，根据返回地址特征返回到主过程的对应地址。

程序框图





源程序及运行结果

```

7010 INPUT N
7020 DIM K(N), J(N+2), Z(N+2)
7030 M=N
7035 Z(N+2)=1
7040 IF N<=0 GOTO7220
7050 IF K(N)=Z(N+2) GOTO7165
7060 IF N=1 GOTO7150
7070 N=N-1
7075 Z(N+2)=0
7080 J(N+2)=1
7090 GOTO7040
7100 N=N+1
7110 N=N-2
7115 Z(N+2)=1
7120 J(N+2)=2
7130 GOTO7040
7140 N=N+2
7150 K(N)=Z(N+2)
7160 GOSUB 7240
7165 IF Z(N+2)=0 GOTO7220
7180 N=N-1
7185 Z(N+2)=1
7190 J(N+2)=3
7200 GOTO7040
7210 N=N+1
7220 IF N=M THEN END
7230 ON J(N+2) GOTO 7100,7140,7210
7240 FOR I=1 TO M
7250 PRINT K(I);
7260 NEXT I
7270 PRINT
7280 RETURN
  
```


源程序及运行结果

RUN
? 3_

```

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

RUN
? 4_

```

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

RUN
? 5_

```

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

四十、巧分牛奶

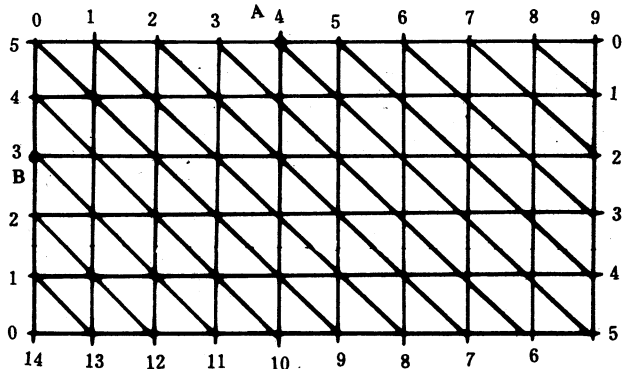
巫永群

在三个大小不等的器皿中，最大的一个装满了牛奶，两个小的空着，然后要求只利用这三个容器分出你所想要的份数，怎么分法？请计算机来帮助我们分析其中某些特殊情况，从而找出所有可能出现的分法及其相应步骤。

这里我们除了要求三个容器的体积不相等外，还要求两个小容器的体积之和正好等于大容器的体积。

此题解决要依靠下面的图形，如附图所示。例如大、中、小容器的体积分别为14、9和5。我们将长方形的底边表示为小容器的零度线，在它上面的横线分别称为一度线、二度线……、五度线。其中n度线表示容器中有n/5的牛奶，这样5度线也就为满容器了。

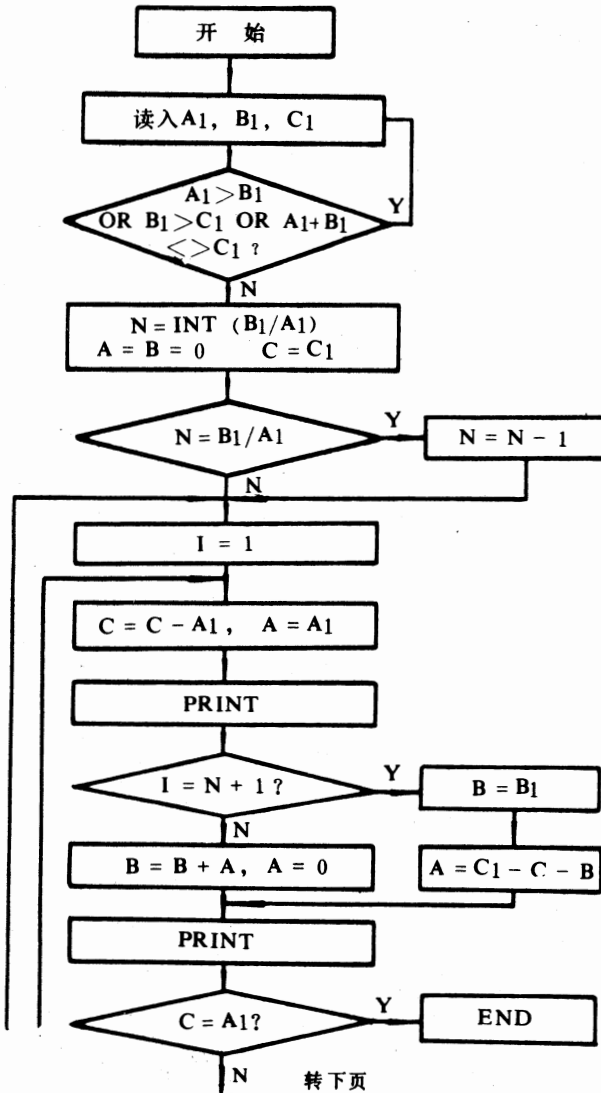
同样，将长方形的竖线表示为中等容器的刻度线，而画出的一系列斜线则表示了大容器的刻度线。这样，边框上任何一点即表示了一种具体的分法。因为实际可行的分法中必定有一容器为满态或零态，

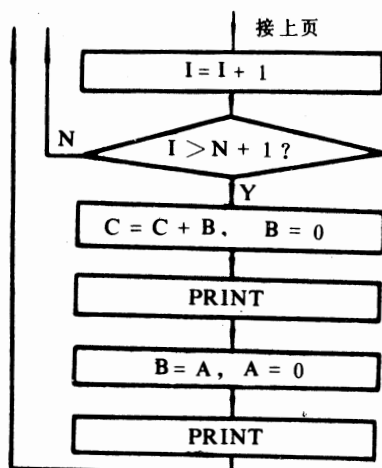


所以，除了边框上的点外，中间的那些分法在实际上是分不出来的。比如图中 A, B 点可分别表示为 (5, 4, 5) 与 (3, 0, 11) 的两种分法，其余各点依此类推。

在上面的打印结果中，有些分法的步骤较多，如按反向运行反而步数会少些，请读者自己做一做。

程序框图





源程序及运行结果

```

5 INPUT "A1,B1,C1";A1,B1,C1
10 IF A1>=B1 OR B1>=C1 OR A1+B1<>C1 THEN 5
15 LET N=INT(B1/A1)
20 LET A=0 :B=0:C=C1
25 IF N<>B1/A1 THEN 35
30 LET N=N-1
35 FOR I=1 TO N+1
40 LET C=C-A1
45 LET A=A1
50 PRINT "A=";A;" "; "B=";B;" "; "C=";C
55 IF I=N+1 THEN 120
60 LET B=B+A
65 LET A=0
70 PRINT "A=";A;" "; "B=";B;" "; "C=";C
75 IF C=A1 THEN 135
80 NEXT I
85 LET C=C+B
90 LET B=0
95 PRINT "A=";A;" "; "B=";B;" "; "C=";C
100 LET B=A
105 LET A=0
110 PRINT "A=";A;" "; "B=";B;" "; "C=";C
115 GOTO 35
120 LET B=B1
125 LET A=C1-C-B
130 GOTO 70
135 END
  
```

RUN
A1,B1,C1? 5,9,14

```

A= 5 B= 0 C= 9
A= 0 B= 5 C= 9
A= 5 B= 5 C= 4
A= 1 B= 9 C= 4
A= 1 B= 0 C= 13
A= 0 B= 1 C= 13
A= 5 B= 1 C= 8
A= 0 B= 6 C= 8
A= 5 B= 6 C= 3
A= 2 B= 9 C= 3
A= 2 B= 0 C= 12
A= 0 B= 2 C= 12
A= 5 B= 2 C= 7
A= 0 B= 7 C= 7
A= 5 B= 7 C= 2
A= 3 B= 9 C= 2
A= 3 B= 0 C= 11
A= 0 B= 3 C= 11
A= 5 B= 3 C= 6
A= 0 B= 8 C= 6
A= 5 B= 8 C= 1
A= 4 B= 9 C= 1
A= 4 B= 0 C= 10
A= 0 B= 4 C= 10
A= 5 B= 4 C= 5
A= 0 B= 9 C= 5
  
```

四十一、周游世界

袁继勋 宋锦金

1859年英国数学家威廉·汉密尔顿爵士在给他朋友的一封信中谈到了一个数学游戏，名叫周游世界游戏。他用一个正十二面体的20个顶点代表20个大城市，这个正十二面体同构于一个平面图（见图1）。要求沿着正十二面体的棱（即交通线），从一个城市出发，经过每个城市恰好一次，再回到原来的出发点。这个游戏曾风靡一时。对应图1有若干个解，每个解都称为汉密尔顿回路。下面的程序，不仅能判断任意图是否存在汉密尔顿回路，同时，若存在，

还能找出它的回路。

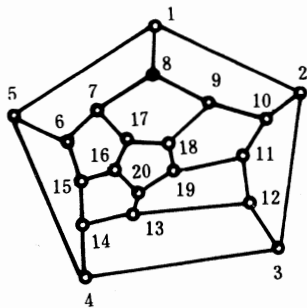


图1

B (n, m) 数组			对应的DATA语句	
1	2	8	5	DATA 2, 8, 5, 1, 3, 10
2	1	3	10	
3	12	2	4	DATA 12, 2, 4,
	⋮	⋮	⋮	DATA 16, 19, 13
20	16	19	13	

图2

程序设计思想及语句说明

①程序执行前的准备工作

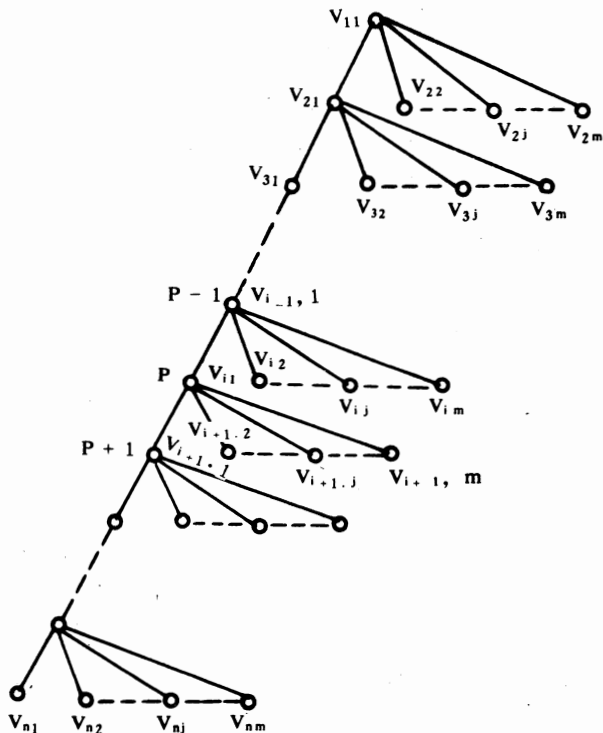
将图中结点按自然数顺序编号。根据图中结点个数及结点的最大度数（最大关联边数）构造 $B(n, m)$ 数组， B 的行标 i ($1 \leq i \leq n$) 对应图中第 i 个结点， $B(i, j)$ ($1 \leq j \leq m$) 的值是第 i 个结点所邻接的结点编号，若不邻接则置 0。填写时不必考虑第 i 个结点与相邻结点的邻接顺序。如图 2 就是根据图 1 构造的 $B(n, m)$ 数组。

②寻找回路的基本算法

程序在动态执行时，数据存放形式如图 3 所示，它是一棵变形的 m 叉完全树。沿着最左结点往下寻找一条无重复结点的汉密尔顿通路 $V_{11}V_{21}\dots V_{i-1,1}\dots V_{n1}$ （下面简称通路）。

若当前访问的结点是 P 级，则从左至右 ($j = 1$ 至 $j = m$) 寻找一个与已形成的通路中的结点无重复的结点。若能找到此结点，就将它置入通路中 ($V_{ij} \Rightarrow V_{i1}$)，同时将当前访问的级增为 $P + 1$ 级，继续构造通路。若不存在此结点，则将当前访问级减为 $P - 1$ 级，并删除 $V_{i-1,1}$ 结点，继续按上述方法，在 $P - 1$ 级寻找一个与通路中的结点无重复的结点。

上述算法执行后，如果图中存在汉密尔顿回路，则当程序完成后，在 $A(i, 1)$ ($1 \leq i \leq n$) 中形成了一条汉密尔顿回路；如果不存在，



则打印“NO Hamilton road”。

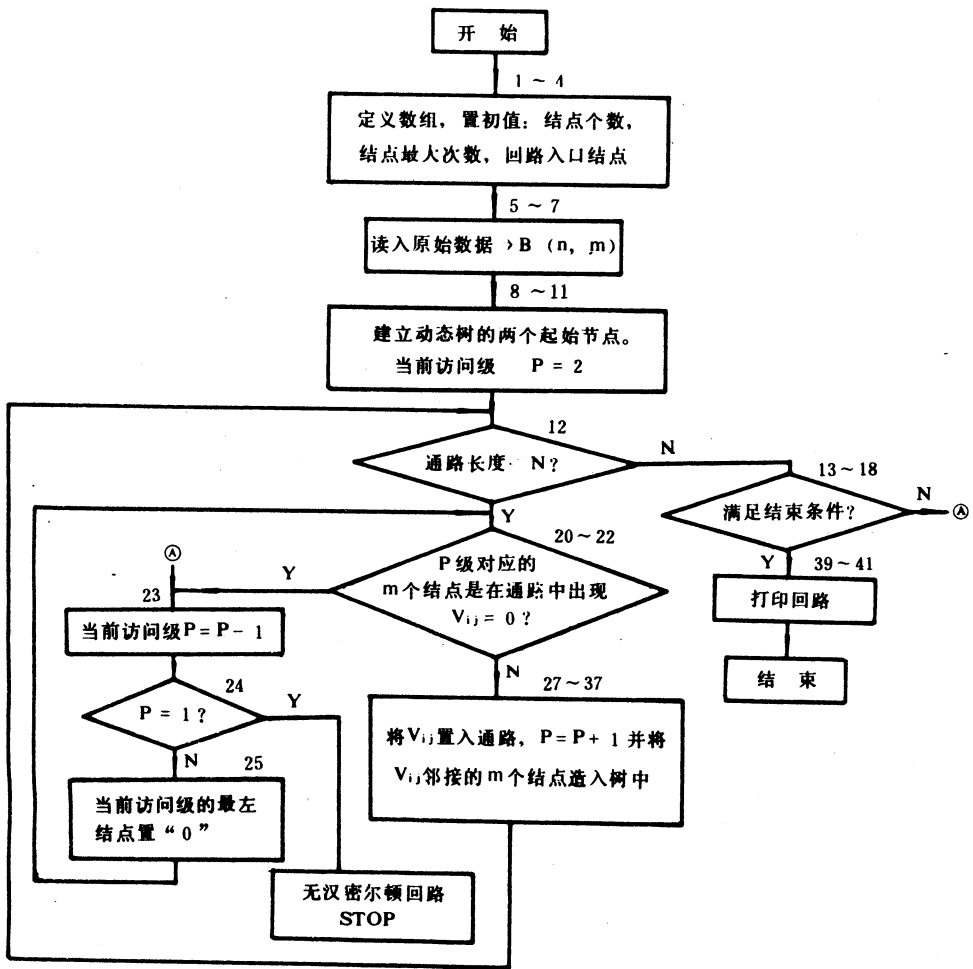
③程序的两点说明

a. A数组存放动态数据。

b. 本程序仅适用于结点数小于等于20, 结点最大度数小于等于10的图。若结点数或结点度数不满足上述条件, 则需根据具体问题重新说明A, B数组。

语句说明见框图。

程序框图



源程序及运行结果

```

1 DIM A(20, 10), B(20, 10)
2 PRINT " NODE="; INPUT N
3 PRINT "MAX DEGREE="; INPUT M
  
```

```

4 PRINT"ENTER NODE=""; INPUT A(1,1)
5 FOR I=1 TO N
6 FOR K=1 TO M:READ B(I,K):NEXT
7 NEXT I
8 W=A(1,1):P=2
9 FOR K=1 TO M
10 A(P,K)=B(W,K)
11 NEXT K
12 IF P<N THEN 20
13 FOR K=1 TO M
14 IF A(P,K)=0 THEN 18
15 FOR L=1 TO M
16 IF A(P,K)=B(W,L) THEN A(P,1)=B(W,L):GOTO39
17 NEXT L
18 NEXT K
19 GOTO 23
20 FOR K=1 TO M
21 IF A(P,K)<>0 THEN 27
22 NEXT K
23 P=P-1
24 IF P=1 THEN PRINT"NO HAMILTON ROAD":STOP
25 A(P,1)=0
26 GOTO20
27 A(P,1)=A(P,K)
28 IF K<>1 THEN A(P,K)=0
29 I=A(P,1):P=P+1
30 FOR K=1 TO M :A(P,K)=B(I,K):NEXT K
31 GOSUB 33
32 GOTO12
33 FOR J=1 TO P-1
34 FOR K=1 TO M
35 IF A(J,1)=A(P,K) THEN A(P,K)=0
36 NEXT K
37 NEXT J
38 RETURN
39 FOR P=1 TO N
40 PRINT A(P,1);"-";
41 NEXT P
42 PRINTA(1,1)
43 DATA 2,8,5,3,1,10,4,2,12,5,3,14,6,1,4,7,5,15
44 DATA 8,6,17,9,1,7,10,8,18,11,2,9,12,10,19
45 DATA 13,3,11,14,12,20,15,4,13,16,6,14,17,15,20,18,7
,16
46 DATA 9,17,19,18,4,20,16,19,13

```

RUN

NODE=? 20 输入结点个数

MAX DEGREE=? 3 输入结点最大度数

ENTER NODE=? 1 输入入口结点

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

READY

>_

四十二、传教士与吃人生番

钱文瀚

三个传教士和三个生番准备过河。只有一条船，每次最多乘2人。在穿梭运载过程中，无论此岸或对岸的生番多于传教士后者就被吃掉。试为传教士设计一个渡河方案。这是《人工智能导论》中一个典型课题。

程序设计思想及说明

①设计思想 我们用搜索树的方法求解。(如图1),每一节点代表一种状态。因为两岸的传教士、生番和船的总数为3、3、1,三者在此岸的数目就足以描述状态。我们设两个数组(M和N)和一个变量(B)来记录三者在此岸时的数目,并用 $M(0) = 3$, $C(0) = 3$ 和 $B = 1$ 来描述初始状态。从初始状态出发,每经一次操作(图1中直线),可变成一个新状态,搜索的目的在于找到目标状态(数组 $M = 0$ 、 $C = 0$,变量 $B = 0$,即此岸无传教士和生番)。在每种状态下理论上是有5种操作:渡河的传教士和生番分别为2与0,1与0,1与1,0与1,0与2,即每一节点可以扇出5个子节点,

每经过一个节点,都用数组元素 $M(L)$, $C(L)$ 记录下来,下标L是搜索树的层数或经过的操作数。B则交替地为1,0,1,0……。搜索时令 $B = 1, -1, 1, -1$ ……。以简化程序。另外,用 $I(L)$ 表示L层采取第几种操作。如果用 $M(L+1)$, $C(L+1)$ 表示新状态, $M(L)$, $C(L)$ 表示原状态,则新状态可表示为:

$$M(L+I) = M(L) - B * F1$$

$$C(L+I) = C(L) - B * F2$$

其中 $F1$ 、 $F2$ 是 $I(L)$ 的两个函数,其关系为:

$$F1 = \text{INT}((4 - I(L)) / 2)$$

$$F2 = \text{INT}(I(L) / 2)$$

$I(L)$ 种操作和渡河的传教士和生番的人数的对应关系如下:

操作 $I(L)$	0	1	2	3	4
传教士人数 $F1$	2	1	1	0	0
生番人数 $F2$	0	0	1	1	2

下列状态是不行的:

- (1) $M(L+1)$ 或 $C(L+1)$ 小于0或大于3;
- (2) 任意岸上生番多于传教士;
- (3) 该状态是从前有过的,它将导致死循环。

上述情况构成了判定一种操作是否合理的根据。

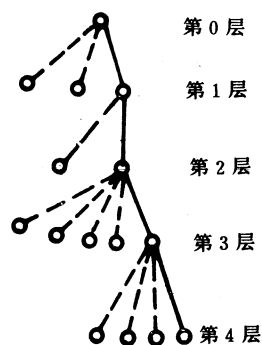
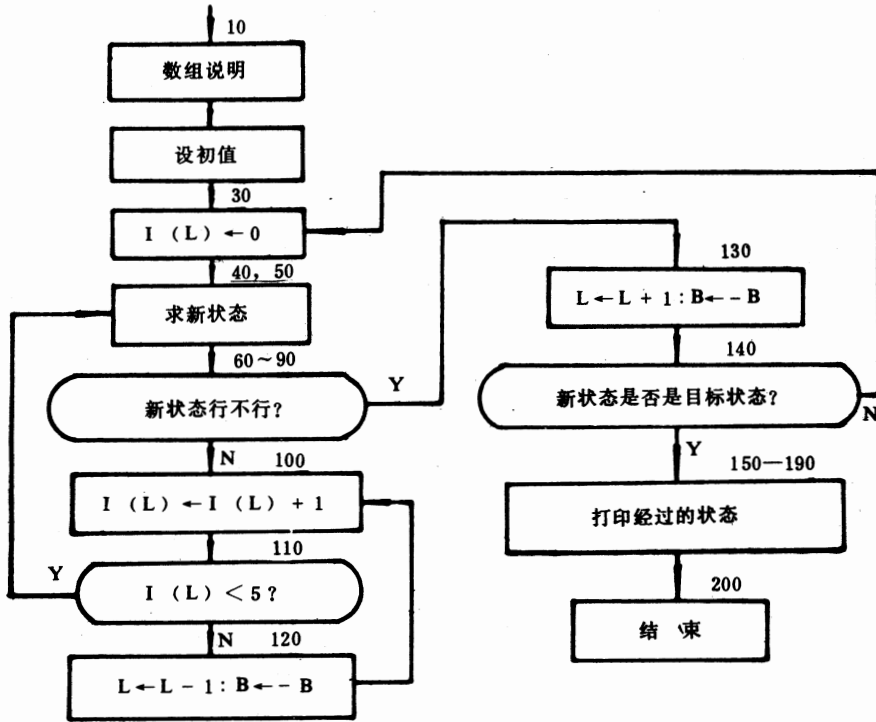


图1

我们首先向纵深搜索，若不属于上述情况就转向下一层，若属于上述情况之一，则换一种操作。如5种操作均已试过，不行，则回溯到上一层，试探另一种操作是否可行。这样不断搜索，不达目标不止。

② 语句说明 (见框图)

程序框图



源程序及运行结果

```

10 DIM M(16),C(16),I(16)
20 L=0:M(0)=3:C(0)=3:B=1
30 I(L)=0
40 M(L+1)=M(L)-B*INT((4-I(L))/2)
50 C(L+1)=C(L)-B*INT(I(L)/2)
60 FOR J=0 TO L
70 IF M(L+1)=M(J)AND C(L+1)=C(J) AND B=-(-1)[J THEN 10
0
80 NEXT J
90 IF (M(L+1)=0 OR M(L+1)=3 OR M(L+1)=C(L+1))AND C(L+1)
)>=0AND C(L+1)<4 THEN 130
100 I(L)=I(L)+1
110 IF I(L)<5 THEN 40
120 L=L-1:B=-B:GOTO100
130 L=L+1:B=-B
140 IF M(L)+C(L)>0 THEN 30
150 B=0
160 FOR J=0 TO L
170 B=1-B
180 PRINT"M"M(J) ",C"C(J)"B"B
190 NEXT J
200 END

```

```

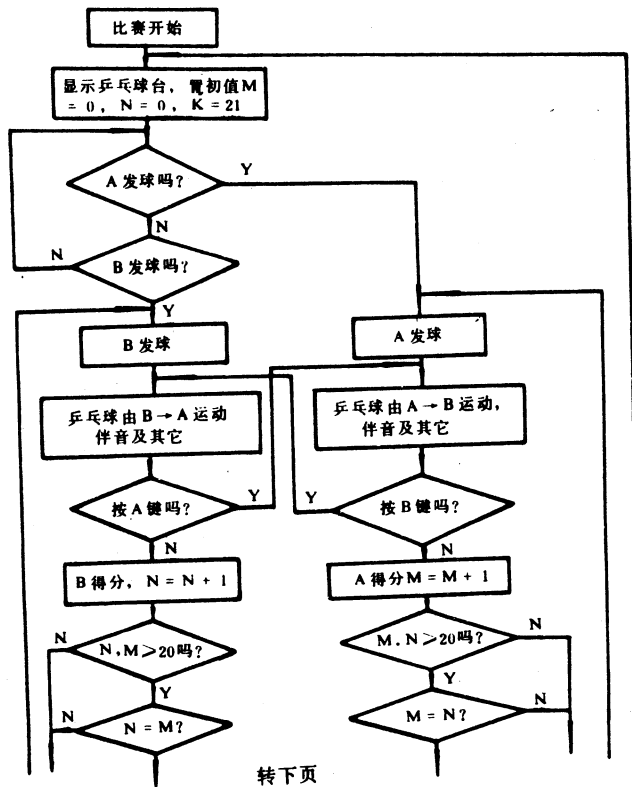
>RUN
M 3 ,C 3 B 1
M 2 ,C 2 B 0
M 3 ,C 2 B 1
M 3 ,C 0 B 0
M 3 ,C 1 B 1
M 1 ,C 1 B 0
M 2 ,C 2 B 1
M 0 ,C 2 B 0
M 0 ,C 3 B 1
M 0 ,C 1 B 0
M 1 ,C 1 B 1
M 0 ,C 0 B 0
READY
>

```

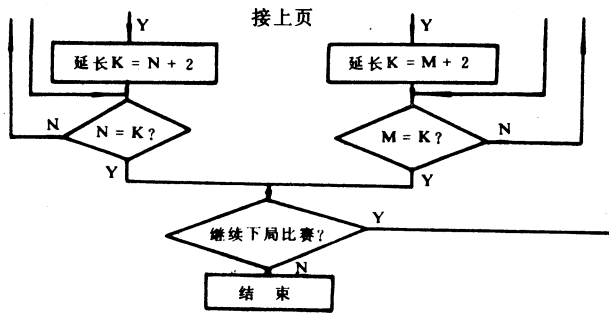

四十三、乒乓球比赛

潘林森

两个人(A和B)可以在计算机上进行乒乓球比赛。比赛中,两人各持一拍(A键或B键)在乒乓球台上把球打来打去,若时机掌握得好,能把球推向对方球台,否则对方得分。程序框图



转下页



程序设计思想与语句说明

程序中用了伴音的机器语言子程序，程序运行时，不断调用伴音子程序，使球落到球台或碰到球拍时发出声音来。程序的开始部分就是将子程序送入内存并定义入口地址。

程序中第90行至140行是在屏幕上作乒乓球台并置比赛初值。用POKE语句将图示块送入显示器内存显示图形，这在程序设计时经常用到，不再赘述。100语句中的 $K = 21$ 表示一局比赛定为21个球，170行至360行是执行B接发球的程序。370行至550行是执行A接发球的程序。560行至590行是判断是否继续下一局的比赛。600行至650行是一个乒乓球运动的子程序。每一方（A方或B方）发球，首先要在该方显示发球信号“*”，显示球拍，调用伴音子程序发声。同时就要调用600~650行的子程序，显示乒乓球运动的轨迹。调用时，给 M_1 、 M_2 、 H 、 L 置不同的值。其中 M_1 、 M_2 为乒乓球在X方向的坐标极限值， H 为步长， L 为乒乓球经过球网时X的坐标。球在球网两侧一端是上升的，一端是下降的。它的上升和下降是通过SET语句中Y坐标的变化来实现的。Y的变化通过两个公式得到，上升时用 $Y = X/9 + 26$ ；下降时用 $Y = 40 - 0.1 * X$ ，当X由大变小时，乒乓球从右向左方向运动的轨迹就可以显示出来了。相反，当X由小变大时，可以显示乒乓球从左向右的运动轨迹。X的变化是通过 M_1 、 M_2 、 H 的变化得到的。

比赛开始，程序扫描键盘，等待A、B任一方先发球，若A先发球转370行开始执行A接发球的程序，若B先发球则执行170行B接发球的程序，否则，程序在150行等待。比赛发球接球时，球拍自动转开球台。比赛中，当A方打出的球落在B方球台上刚弹起时，若正好B方按下B键，则转210行执行B接球的程序将球打回A方，否则，A得一分，马上更新比赛分数，并进一步判断A的分数是否等于K，若等于K，本局比赛结束，转560行等待回答是否继续下一局比赛。若不等于K则A方显示发球提示符“*”转370行，由A发球，比赛继续。进行分数判断时，若打到一局最后双方出现平分，则延长两分，比赛继续，若又出现平分，比赛继续延长，直到A、B双方比赛相差两分为止。同样，对于B方打出的球也作相同的分析判断转移。

程序中，键盘输入使用了INKEYS函数，使键盘输入不在屏幕上显示，并且程序对输入采取了自动校验，以防按键失误引起混乱。比赛中，该由谁发球也是根据比赛胜负情况确定的，双方运动员无法任意改变，确保比赛顺利进行。

程序是在TRS-80机上实现的。程序运行时，最好连接上录音机，其方法是将TRS-80微型机的输出信号插头插进录音机的MIX插孔按下PLAY键，使录音机处于放音状态，比赛才有声音。否则没有声音，但并不影响程序的运行。

源程序及运行情况

●源程序

```
10 DEFINT X
20 FOR X=31232 TO 31257
30 READ A:POKE X,A
40 NEXT
50 DATA 22,105,24,2,22,52,14,2,66,62,5,211,255,16
60 DATA254,66,62,2,211,255,16,254,13,32,239,201
70 CLS:CLEAR100
80 DEFUSR0=&H7A00:DEFUSR1=&H7A04
90 PRINT@216," PING-PONG MATCH"
100 M=0:N=0:Z$="":K=21
110 PRINT@839,STRING$(52,131)
120 PRINT@342,"A",@363,"B",@413,M;Z$:N
130 POKE16097,144:POKE16161,149:POKE16205,191
140 POKE16244,191:POKE16269,191:POKE16308,191
150 A$=INKEY$: IFA$="A" THEN PRINT@408,"*":GOTO370
160 IFA$<>"B" THEN 150
170 PRINT@425,"*"
180 M1=109:M2=18:H=-9:L=55
190 POKE16186,191:X=USR(0):FOR I=1 TO 60 : NEXT
200 X=USR(1):POKE16186,32
210 GOSUB600
220 X=USR(1)
230 FORX=18 TO 0 STEP -4
240 SET(X,29+X/2)
250 FOR I=1 TO RND(20):NEXT
260 RESET(X,29+X/2)
270 A$=INKEY$: IFA$<>"A"ORX<>14ANDX<>10THEN320
280 M1=27:M2=109:H=9:L=45
290 POKE16133,191:X=USR(0):FORI=1TO60:NEXT
300 POKE16133,32
310 GOTO400
320 NEXT X
330 N=N+1:PRINT@408," ";@413,M;Z$:N:@425,"*"
340 IFN>=K-1ANDM>=K-1THENK=M+2
350 IFN=KTHEN560
360 A$=INKEY$: IFA$<>"B"THEN360ELSE180
370 M1=18:M2=109:H=9:L=55
380 POKE16133,191:X=USR(0):FORI=1TO60:NEXT
390 X=USR(1):POKE16133,32
400 GOSUB600
410 X=USR(1)
420 FORX=110TO127 STEP 4
430 SET(X,54-X/6)
440 FORI=1 TO RND(20):NEXT
450 RESET(X,54-X/6)
460 A$=INKEY$: IFA$<>"B"ORX<>118ANDX<>114THEN510
470 M1=100:M2=18:H=-9:L=73
480 POKE16186,191:X=USR(0):FORI=1TO60:NEXT
490 POKE 16186, 32
500 GOTO 210
510 NEXT X
```

```

520 M=M+1:PRINT@408,"*";@413,M;Z$;N;@425," "
530 IF M>=K-1 AND N>=K-1 THEN K=N+2
540 IF M=K THEN 560
550 A$=INKEY$:IF A$<>"A" THEN 550 ELSE 370
560 A$=INKEY$:IF A$=" " THEN 560
570 IF A$="Y" THEN 70
580 IF A$="N" THEN END
590 GOTO560
600 FOR X=M1 TO M2 STEP H
610 IF X<=L THEN SET(X,40-.1*X) ELSE SET(X,X/9+26)
620 FOR I=1 TO RND(20):NEXT I
630 IF X<=L THEN RESET(X,40-.1*X) ELSE RESET(X,X/9+26)
640 NEXT X
650 RETURN

```

②运行情况

打入 RUN 命令后，在屏幕上显示出乒乓球台，按 A 键，表示 A 方先发球。乒乓球从 A 方向 B 方运动，待落到 B 方球台后，B 方要按 B 键，把球打回。如此反复。

四十四、打印万年历

杨舒

下面是打印任意日历的通用程序，它可以打印任意一年的年历，或者任意一月的月历，还可以打印出任意一天是星期几。而且打印年历的格式可以选择 3×4 阶矩阵或者 4×3 阶矩阵的形式。

程序的设计思想和语句说明

①设计思想 为使程序具有通用性，程序中采用了计算日历的通用公式：

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

其中，Y 为年号，C 为从 Y 年元旦起到要算那天的总天数。

用 S 除以 7 所得余数是几，那天即为星期几。

日历的编排是每 400 年一个大周期，如今年的某月，某日，和 400 年前完全一样。闰年的求法是：年号能被 4 整除的而年号末尾两位数不为零的是闰年；若年号末尾两位数为零，则此年号能被 400 整除的才是闰年。

欲打印某一年的年历，则根据上面公式即可求出 S（此时 $C = 1$ ），S 被 7 除，余数即为该年元旦是星期几。知道了元旦是星期几，一年的日历即可求得。

②语句说明 程序中第 10~250 语句是输入原始数据，对要输入打印的年号进行分析判断，并作相应处理。若是计算某日是星期几，则转 260 语句，在屏幕上显示出该日日历；若是打印月历，则先计算出该月 1 日是星期几，作为初值 K，然后转 280 语句，以 K 为基数计算出当月逐日是星期几，并在屏幕上显示该月月历；若是打印年历，则首先计算出该年元旦是星期几作为初值 K，然后转 370 语句。程序中，170、180 语句用来判断 y 年是否是闰年，若闰年，则修改二月份的天数为 $y(2) = 29$ 。

370~770 语句是专门打印年历的，打印中，采用了将 12 个月分段，逐段打印的办法。首先，

输入按月份矩阵格式打印的矩阵列数MM，然后以K作为基数，先算出MM个月逐日是星期几，暂存工作单元数组A中，等一段MM个数全部计算完后，先打印出MM个月的日历，打印完毕后，将数组A清零，再继续计算下一段MM个月逐日是星期几存数组A中，等全部算完后再次打印。这样重复12/MM次，即打印完全年年历。

年历打印中，还采用了定位打印的自选打印格式，程序中第40~60语句就是为年历打印格式而设计的读数字程序。在打印中，对年号、月号和日号采用几种不同的字体打印，使打印的年历更整齐美观。后面的程序是在TRS-80微型机 μ 80行印机上设计的打印格式，其它打印机打印格式将要相应有所变化。

3. 源程序1

```

10 CLS: CLEAR 200
20 DIM Y(12), A(5, 27), B$(27), C(27)
30 GOTO 70
40 FOR I=JJ TO JJ+6
50 READ B$(I): READ C(I)
60 NEXT I: RETURN
70 PRINT "THEN FORMAT OF THE DATE"
80 PRINT TAB(5) " (1) YYYY"
90 PRINT TAB(5) " (2) YYYY.MM"
100 PRINT TAB(5) " (3) YYYY.MM.DD"
110 INPUT "ENTER THEN DATE": A$
120 L=LEN(A$): Y=VAL(LEFT$(A$, 4)): YN$=MID$(A$, 3, 2)
130 M=VAL(MID$(A$, 6, 2)): D=VAL(MID$(A$, 9, 2))
140 IF D=0 THEN D=D+1
150 JJ=0 GOSUB 40
160 FOR I=1 TO 12: READ Y(I): NEXT I
170 IF YN$="00" THEN YM=400 ELSE YM=4
180 IF Y/YM=INT(Y/YM) THEN Y(2)=29
190 IF M=0 OR M=1 THEN 210
200 FOR I=1 TO M-1: C=C+Y(I): NEXT I
210 C=C+D: S=Y-1
220 S=S+INT(S/4)-INT(S/100)+INT(S/400)+C
230 K=S-INT(S/7)*7
240 IF L<5 THEN 370
250 IF L<8 THEN 280
260 CLS: PRINT @404, A$
270 PRINT @535, K: END
280 CLS: PRINT TAB(33); "***"; A$; "***": PRINT
290 PRINT TAB(18); "0      1      2      3      5      6"
300 PRINT
310 FOR N=1 TO Y(M)
320 PRINT TAB(17+6*K); N;
330 K=K+1
340 IF K=7 THEN PRINT: K=0
350 NEXT N
360 END
370 PRINT "INPUT FORMAT OF PRINTING"
380 PRINT TAB(5) "THEN COLUMN NUMBER OF ARRAY OF MONTHS (
'3' OR '4')";
390 INPUT MM: IF MM<>3 AND MM<>4 THEN 380 ELSE ME=MM
400 RESTORE: JJ=JJ+7: GOSUB 40

```

```

410 ME=ME-1:IF ME>1 THEN 400
420 DATA #####,0,###,1,###,2,###,3,###,4,###,5,###,6
430 DATA 31,28,31,30,31,30,31,31,30,31,30,31
440 LPRINT CHR$(31)TAB(22-3*MM)"=:=:=";A#;":=:=:="
450 LPRINT CHR$(33-MM):LPRINT
460 LPRINTAB(5)STRING$(25*MM-5,"-")
470 W=1
480 T=0:T1=7:T2=0
490 IF MM=3 THEN GOSUB 580 :GOTO510
500 GOSUB 560
510 K=K-7*MM:W=W+MM
520 IF W<=13-MM THEN 480
530 LPRINT TAB(5)STRING$(25*MM-5,"-")
540 LPRINT CHR$(30)TAB(78-15*MM)"THE COMPUTER ROOM OF
MATHEMATICS DEPARTMENT"
550 END
560 LPRINT CHR$(31)TAB(4)W;TAB(12)W+1;TAB(19)W+2;TAB(
27)W+3
570 LPRINT CHR$(29):GOTO590
580 LPRINT CHR$(31)TAB(7)W;TAB(20)W+1;TAB(32)W+2:LPRIN
T CHR$(30)
590 FOR I=0 TO 7*MM-1
600 LPRINT USING B$(I);C(I);:NEXT
610 LPRINT
620 FOR X=W TO W+MM-1
630 FOR N=1 TO Y(X)
640 A(T,K)=N :K=K+1
650 IF K=T1 THEN K=T2:T=T+1
660 NEXT N
670 T1=T1+7:T2=T2+7:K=K+7:T=0
680 NEXT X
690 FOR I=0 TO 5
700 FOR J=0 TO 7*MM-1
710 IF A(I,J)=0 THEN 730
720 LPRINT USING B$(J);A(I,J);:A(I,J)=0 :GOTO740
730 H=J+7:IF H/7=INT(H/7) THEN LPRINT" " ;ELSE LP
RINT" ";
740 NEXT J
750 LPRINT
760 NEXT I
770 RETURN

```

下面再介绍一个打印万年历的程序。此程序可连续打印几年的日历，并且还可以控制打印份数。程序运行时，计算机先问：起始年份，再问：终止年份，接着问每行月数和打印份数，当你给出（可只给出起始年份，其它缺省）后计算机即自动打出所要求的年历。由于计算万年历的公式和打印格式都基本相同，所以请读者自己分析程序，这里不再介绍。

此程序也是设计在TRS—80机和 μ 80打印机上运行的，如果使用其它机器和打印机，程序输出格式将有所变化。

源程序2

```

200 DEFINT I-Z:DIM T(12),M(12)
220 INPUT "YEAR, TO YEAR, NO./L, COUNT: ";Y,Y2,N,L
230 IF N>4 OR N<1 THEN N=3
240 A#="=====" :X=80

```

```

245 C$="-- * - . - : - . :- :!:-!:-!:-"
250 B$=" SUN MOU TUE WED THU FRI SAT "
260 FOR Q=1 TO L
270 FOR P=Y TO Y2
280 F=FIX((P-1)/4+P-1)/7;W=(F-FIX(F))*7+.5
290 FOR U=0 TO 12:M(U)=1 :NEXT
300 LPRINT TAB(37) P:LPRINT TAB(33);A$:LPRINT:LPRINT
305 LPRINT CHR$(27);CHR$(29):X=132
310 FOR I=1 TO 12 :READ T(I):NEXT
320 IF FIX(P/4)*4=P THEN T(2)=29
330 K=(X-N*30)/(N*2)
340 FOR J=1 TO 12 STEP N
350 Z=K:I=J-1
360 FOR V=1 TO N :READ D$
370 LPRINT TAB(Z);D$;TAB(Z+24);"(";I+V;")";
380 Z=(30+2*K)*V+K:NEXT:LPRINT :Z=K
390 FOR V=1 TO N
400 LPRINT TAB(Z);A$;A$;:Z=K+(30+2*K)*V
410 NEXT:LPRINT :Z=K
430 FOR U=1 TO N
440 LPRINT TAB(Z);B$;:Z=K+(30+2*K)*U
450 NEXT :LPRINT :Z=K
455 FOR U=1 TO N
456 LPRINT TAB(Z);C$;:Z=K+(30+2*K)*U:NEXT
457 LPRINT :Z=K:LPRINT
460 FOR U=1 TO 6
465 Z=K:I=J-1
470 FOR V=1 TO N
480 R=7:Z=Z+2:I=I+1
490 IF U<>16GOTO 520
500 R=7-W:Z=Z+4*W:W=W+T(I)
510 W=(W/7-FIX(W/7))*7+.5
520 FOR S=1 TO R
525 IF M(I)>T(I)GOTO570
530 LPRINT TAB(Z);M(I);
540 M(I)=M(I)+1:Z=Z+4
550 IF M(I)>T(I)THEN570
560 NEXT S
570 Z=(30+2*K)*V+K:NEXT V:LPRINT
575 NEXT U:Z=K
580 FOR V=1 TO N :LPRINT TAB(Z);A$;A$;
590 Z=(30+2*K)*V+K:NEXT V
600 LPRINT :LPRINT :LPRINT
610 NEXT J
620 FOR V=1 TO 10:LPRINT :NEXT
630 RESTORE
640 LPRINTCHR$(27); CHR$(30)
650 NEXT P,Q
657 STOP
660 DATA31,28,31,30,31,30,31,31,30,31,30,31
670 DATA"JANUARY ","FEBRUARY ","MARCH "
680 DATA"APRIL ","MAY ","JUNE "
690 DATA"JULY ","AUGUST ","SEPTEMBER"
700 DATA"OCTOBER ","NOVEMBER ","DECEMBER "
710 END

```

四十五、飞机投弹

潘林森

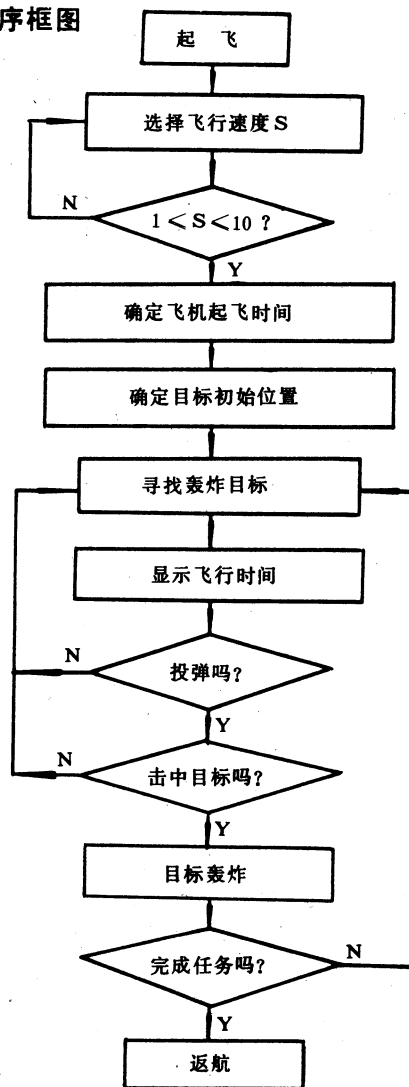
看电影常看到战争场面，飞机如闪电在空中来往穿梭，坦克像野牛在地上横冲直撞。忽然一辆坦克被击中，顿时火光冲天，爆炸声起。这些惊险场面都可以由计算机来模拟。运行本文给出的程序，就可以玩一场趣味性很强的飞机炸坦克的战争游戏。

程序说明

程序开始，需先确定飞机飞行速度因子。这个因子是1~10范围内的数字，它表示飞机由快到慢的各种速度，反映游戏的难度。在1~10的范围内选择回答一个数后，程序调用机器内部时间函数，记下飞机起飞时间（第70语句）。程序中，飞机（QS）用三个字符“—>—”表示，坦克（BS（1）和BS（2））用ASCII代码142的图示符CHR\$（142）表示，即程序中的80~90语句。在这两个语句行中，还确定了飞机起飞和坦克运行的初始位置（ K_1, K_2 ）。100语句是调用标号为250的子程序。250~320这段子程序为飞机飞行和坦克运行的子程序。其中，变量I控制着飞行区间，不让飞机飞出显示屏幕第一行。飞行速度由速度因子S确定，S值越大，飞行速度越慢。程序中还用了随机函数RND（20）控制飞机飞行，使飞机飞行速度随机地变化。同样，坦克运行中也用了随机函数RND（2）控制，使坦克运行速度也是随机的。坦克BS（1）和BS（2）分别由变量 K_1 和 K_2 控制，使它们也只在显示屏的最后第二行上运行。

程序运行中，不断调用子程序，使飞机和坦克处于不断运行中。同时，通过120语句扫描键盘，若未按下空格键，表示飞机未发现或者没有瞄准目标。此时程序在100~120行语句之间循环。若按下空格键，表示飞机已瞄准目标，程序转去执行130~160语句行的飞机投弹程序。投弹时，炮弹从飞机尾部以步长为65的速度向目标投下。炮弹用ASCII代码为136的图示符表示。接着是170~180两个条件语句，用来判断炮弹是否击中目标；若没有击中，则转100语句，调用子程序250，继续飞行；若击中目标，转200语句，让坦克爆炸，并延时约0.2秒时间。然后，由240语句判断是否还有轰炸目标。若已炸完，则程序结束，返回命令状态；否则，转100语句，继续执行飞行任务。

程序框图



在投弹时，由于飞机和坦克都是以随机速度运动的，为了提高飞机投弹的命中率，在170和180两条件语句中，采用了两个逻辑或（OR）表达式，使炮弹落到坦克的尾部、中部、头部任一部位都算击中目标。

在整个程序运行中，不断调用机器内部时间函数TIMES，由110行语句把飞机起飞时间和飞机已飞行时间不断显示在屏幕右上角。当目标轰炸完时，可以看到完成任务共用了多少时间。

源程序

```

10 CLS:DEFINTI-K
20 PRINTTAB(20)"BOMBER BOMBING"
30 PRINTTAB(15)"HIT 'SPACE BAR' TO BOMBING":PRINT:FRIN
T
40 PRINTTAB(5)"ENTER SPEED FACTOR(1--10)":
50 INPUT S
60 IF S<1 OR S>10 THEN 10
70 CLS:A#=TIME$
80 FOR X=1 TO 2 :C$(X)=" ":B$(X)=CHR$(142):NEXT X
90 I=0: K1=896:K2=920:P$=" ":Q$="->-"
100 GOSUB250
110 PRINT@246,RIGHT$(A$,8);@310,RIGHT$(TIME$,8)
120 Y#=INKEY$:IF Y#<>" " THEN 100
130 FOR K=I TO 896+I STEP 65
140 POKE15424+K,136:GOSUB250
150 POKE 15424+K,32
160 NEXT K
170 IF K=K1 OR K=K1-1 OR K=K1+1THEN I0=1 :GOTO200
180 IF K=K2 OR K=K2-1 OR K=K2+1 THEN I0=2 :GOTO200
190 POKE 15360+K,32:GOTO100
200 POKE 15359+K,35:POKE 15358+K,45:POKE 15360+K,45
210 POKE 15295+K,33:POKE 15423+K,33
220 FOR X=1 TO 100 :NEXT
230 CLS:B$(I0)=" ":I0=0
240 IF B$(1)=" "AND B$(2)=" " THEN END ELSE 100
250 IF I>=60 THEN PRINT@60," ":I=0
260 PRINT@I,P$, :I=I+RND(20)/(S+1):PRINT@I,Q$
270 IF K1>=958 THEN POKE 15360+K1,32:K1=896
280 IF K2>=958 THEN POKE 15360+K2,32:K2=920
290 PRINT@K1,C$(1),@K2,C$(2)
300 K1=K1+RND(2)/2:K2=K2+RND(2)/2
310 PRINT@K1,B$(1),@K2,B$(2)
320 RETURN

```

四十六、打魔星

于文泉

“打魔星”游戏是模仿流行的电子游戏编写的。游戏者通过键盘控制一门可左右移动的大炮，以空中魔星为目标进行射击。游戏中，魔星的初始位置，运动方向和运动速度都是随机产生的。给人以一种时隐时现、变化莫测的感觉。提高命中率的窍门是选择那些刚刚在你

面前出现的魔星为目标。不然的话,当你已经瞄准了,或者发炮了,魔星会突然不知去向,真的“魔”了。当然快速的反应,准确的控制是击中目标的前提条件。

游戏方法

打入RUN命令,程序开始运行。先显示出局数,然后显示出15颗魔星,炮口的初始位置。这时游戏者用左手击空格键即为开炮射击。如果要移动炮位,可用右手的食指和中指分别按快速数据区的2和3键或5和6键,把炮移到合适的位置,重新发炮射击。2和3键为炮口的快速移动键。每按动一次,炮口可左或右移动4个位置,5和6键为炮口的微动键。每按动一次,炮口只能左或右移动一个位置。右手对这4个键的控制,可使炮口位置随心所欲,看准时机立即发炮。命中后,该魔星即从屏幕上消失,计算机自动给游戏者记200分,并在屏幕左上角显示出所得的总分,游戏继续进行,游戏的结束是由程序的运行时间控制的。时间一到,该局即结束,并自动开始下一局。每次游戏进行三局。三局结束后,如果游戏者的得分超过6000分,计算机显示出“VERY GOOD, LET'S PLAY AGAIN...”,奖励游戏者再玩一盘,程序自动从头开始运行。否则程序结束。

程序设计思想与说明

①设计思想

本程序设计15个魔星。显示这15个魔星是通过TRS-80机的作图语句实现的。首先随机产生15组X和Y的值做为每个魔星的原始坐标值,并把它们分别放到 $A(I, 1)$, $A(I, 0)$ 中。然后,分别在X方向和Y方向给 $A(I, 1)$, $A(I, 0)$ 一个随机的坐标偏移量 $A(I, 3)$, $A(I, 2)$ 。这样得到的魔星坐标值 $A(I, 1) = A(I, 1) + A(I, 3)$ 和 $A(I, 0) = A(I, 0) + A(I, 2)$ 也是随机的。通过SET($A(I, 1)$, $A(I, 0)$)语句即把该图示块显示在屏幕上,15个魔星是通过0~14的循环产生的,每次产生一个。

炮台是通过打印位置变量CA控制的。开始,使 $CA = 992$,在屏幕下部中心处打印一个“*”,以后它的左右移动靠键盘上2、3、或5、6键来控制。程序不断扫描键盘,如果得到的是2或5,给变量CA一个负的修正值-4或-1,即 $CA = CA - 4$ 或 $CA = CA - 1$ 。这样,“*”即向左移动。如果得到的是3或6,则给变量CA一个正的修正值4或1,即 $CA = CA + 4$ 或 $CA = CA + 1$ 。这样,“*”即向右移动。移动中,“*”号不能超出该打印行,故使 $966 < CA < 1022$ 。

炮弹是通过图示块的方法给出的。在弹道上从下到上不断给出上边的图示块,同时抹除下边的图示块。这样,炮弹就在垂直的轨迹上前进了。炮弹的水平坐标是通过炮台的位置给出的;垂直坐标从44-1变化。有了水平坐标和垂直坐标,用SET语句不难表现炮弹的轨迹。

命中与否是通过POINT语句实现判断的。在炮弹的轨迹坐标中,如果有发光的块,则判为击中目标。这比较简单,不再详细说明。

②语句说明

100~110语句:程序初始化。

120~140语句:局数、初始炮位显示。

150~180语句:随机产生15个魔星的初始位置。

190~330语句:主程序,轮流改变15个魔星的运动方向,运动速度并显示出新位置;同时完成时间控制和监视键盘的工作。

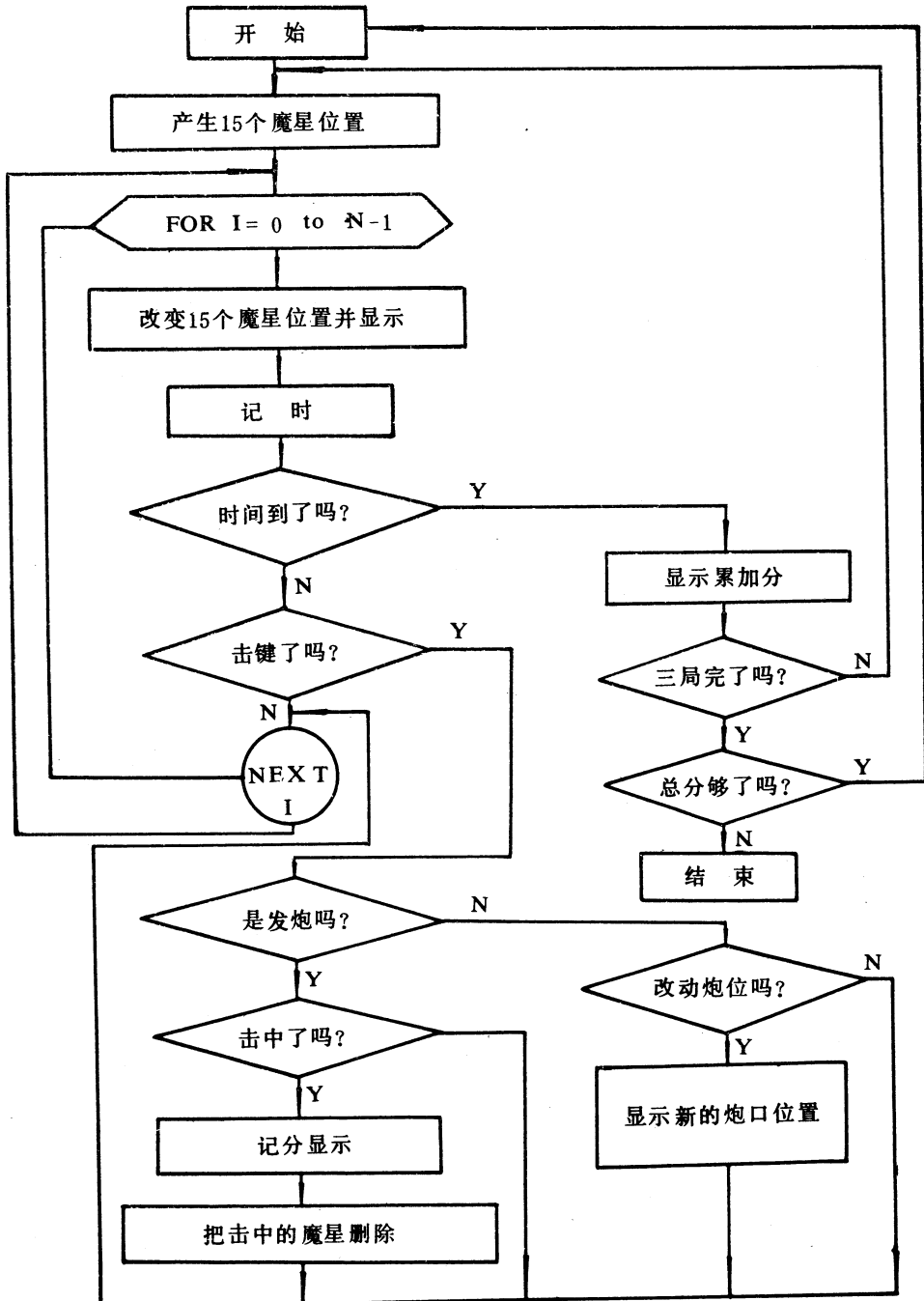
340~390语句：发炮后的处理子程序。产生炮弹的运动轨迹并判断是否击中目标。

400~550语句，炮口左右移动的子程序。

700~730语句，每局结束后的处理。

740~800语句，每盘游戏结束后的处理。

根据个人的要求，程序中的参数可自由改变。如想使魔星的数量增加或减少，只要修改110语句中的N值即可。由于计算机的运算速度是一定的，当N值太大时，每个魔星变动一次位置所需的时间就要变长。反之，魔星会给人一种连续跳跃的感觉。如果想改变魔星运动速



度的极值，只需将 240, 260 语句中 RND (20) 用更大或更小的随机数代替即能实现。如果想改变每局游戏的时间，仅仅改变 290 语句中 M 的比较值即可。如果想改变炮口的符号，将 140、680 语句中的 “*” 号用理想符号代替就可以了。

源程序

```

100 CLEAR
110 RANDOM: DIM A(15,3): B=1: N=15
120 CLS: PRINT@465, CHR$(23); B; "GAME"
130 FOR L=1 TO 1000 : NEXT
140 CLS: CA=992: PRINT@CA, "*"; : MX=64
150 FOR I=0 TO N-1
160 A(I,1)=RND(46)
170 A(I,0)=RND(127)
180 NEXT
190 FOR I=0 TO N-1
200 RESET(A(I,0),A(I,1))
210 IF A(I,0)=0 AND A(I,1)=0 THEN 320
220 A(I,3)=RND(20)*SGN(RND(10)-4.5)
230 A(I,2)=RND(20)*SGN(RND(10)-4.5)
240 IF A(I,1)+A(I,3)>46 OR A(I,1)+A(I,3)<0 THEN IF A(I,3)>0 THEN A(I,3)=-1*RND(20) ELSE A(I,3)=RND(20)
250 A(I,1)=A(I,1)+A(I,3)
260 IF A(I,0)+A(I,2)>127 OR A(I,0)+A(I,2)<0 THEN IF A(I,2)>0 THEN A(I,2)=-1*RND(20) ELSE A(I,2)=RND(20)
270 A(I,0)=A(I,0)+A(I,2)
280 SET(A(I,0),A(I,1))
290 M=M+1 : IF M=200 THEN 700
300 A$=INKEY$
310 IF A$<>" " THEN GOSUB 340
320 NEXT
330 GOTO 190
340 IF A$<>" " THEN 560
350 MY=44
360 SET(MX,MY)
370 IF MX<=20 THEN T=40 ELSE T=43
380 FOR U=1 TO T: RESET(MX,MY-U+1): SET(MX,MY-U): IF POINT(MX,MY-U-1) THEN 400 ELSE NEXT
390 RESET(MX,MY-U+1): RESET(MX,MY-U): RETURN
400 RESET(MX,MY-U)
410 FOR I=0 TO N-1
420 IF INT(A(I,0))=MX AND INT(A(I,1))=MY-U-1 THEN 440
430 NEXT
440 FOR J=1 TO 2
450 FOR K=1 TO 10
460 RESET(A(I,0),A(I,1))
470 NEXT
480 FOR K=1 TO 10
490 SET(A(I,0),A(I,1))
500 NEXT K,J
510 RESET(A(I,0),A(I,1))
520 A(I,0)=0: A(I,1)=0
530 W=W+200
540 PRINT@1, "* "; W; " *";
550 RETURN

```

```

560 IF A$<>"2" THEN 590
570 IF CA-4<=960 THEN RETURN
580 PRINT@CA," ";CA=CA-4:MX=MX-8:GOTO680
590 IF A$<>"3" THEN 620
600 IF CA+4>=1022 THEN RETURN
610 PRINT@CA," ";CA=CA+4:MX=MX+8:GOTO680
620 IF A$<>"5" THEN 650
630 IF CA-1<=960 THEN RETURN
640 PRINT@CA," ";CA=CA-1:MX=MX-2:GOTO680
650 IF A$<>"6" THEN RETURN
660 IF CA+1>=1022 THEN RETURN
670 PRINT@CA," ";CA=CA+1:MX=MX+2
680 PRINT@CA,"*";
690 RETURN
700 CLS:FOR H=1 TO 200
710 PRINT@471,"*** ";W;" ***"
720 NEXT:M=0
730 B=B+1:IF B<=3 THEN 120
740 IF W<6000 THEN 790
750 CLS:PRINT@465,"VERY GOOD"
760 PRINT@600,"LET'S PLAY AGAIN....."
770 FOR V=1 TO 3000:NEXT
780 W=0:GOTO100
790 CLS:PRINT@472,"*** END ***"
800 END

```

四十七、巧分工种

袁继勳

有 N 个人和 N 种工作，各人完成同一种工作所需工时数不同。现要找出一种最佳分配方案，使 N 个人完成 N 件工作所需总时数为最小。下面使用“匈牙利算法”来解决此问题*。对于算法不加证明直接使用。

算法简介

①建立分配矩阵 $A_{n \times n}$ ，元素 a_{ij} 的值表示第 i 个人完成第 j 项工作所需工时数。 $A_{n \times n} \Rightarrow B_{n \times n}$ ， B 矩阵作运算时用。

②逐行缩减。在 B 各行中，找出最小元素，将每行中的元素均减去该行最小元素。（此时 B 中各行均有一个 0 元素）形成 B_1 。

③逐列缩减。查看 B_1 中各列是否均有零。有，转④；无，对所有无零的列找出每一列中最小元素，并将该列中所有元素减去这列中最小元素，形成 B_2 ，转④。

经②，③步之后，各行各列至少有一个零元素，可能会包含最佳分配。

④分配检验。以最少的直线来覆盖 B_2 中的所有零元素所在的列和行。形成 B_3 。若直线数恰好等于 A 的阶数 n ，则能找到最佳分配方案转⑥，否则转⑤。

⑤检查覆盖线。在有覆盖线的 B_3 中，先找出无覆盖线的行与列中最小元素，将不在覆盖线上的行与列中的元素，减去这一最小元素，而把在两条覆盖线的交叉点的元素加上这最小元素。形成 B_4 。 $B_4 \Rightarrow B_2$ ，转④

⑥确定最优分配方案。具体如下：

(a) 在B3中依次选出有一个0的行进行分配,给定该0元素所在行(i)与列(j)的值,到A中找相应 a_{ij} 。 $D = D + a_{ij}$ 。

(b) 将B3中0元素所在列(j)元素全置1(以后不再对该列分配)。分配完成转⑦,否则转(a)。

⑦输出D。D为总工时数。

* 匈牙利数学家D·konirg 证明了这个基本定律,为解决分配问题作出了贡献,所以把这个解法称为匈牙利法。

实例及运行结果

某产品中有4个部件要分派给4个生产小组去完成,由于生产小组的生产能力和各部件的复杂程序不同,完成的时间也不一样。如果各小组完成不同的部件所需的时间如表1所示,计算机运算结果给出完成上述四个部件最好的分配办法,完成上述工作最少需要24个工时。

运行结果

PEOPLE? 4		
PEOPLE	KIND OF WORK	TIME
I = 2	J = 2	4
I = 4	J = 1	4
I = 1	J = 3	5
I = 3	J = 4	11

表 1

工时 小组	部件			
	1	2	3	4
A	2	10	5	7
B	15	4	14	8
C	13	14	12	11
D	4	15	13	9

THE TOTAL WORKING TIME IS: 24

源程序

```

1 DIMA(5,5),B(5,5),C(5),D(5),E(25,3)
2 PRINT"PEOPCE";
3 INPUT N
4 PRINT
5 PRINT"PEOPLE      KIND OF WORK      THME"
6 PRINT"-----"
7 FOR I=1 TO N
8 FOR J=1 TO N
9 READ A(I,J)
10 B(I,J)=A(I,J)
11 NEXT J
12 NEXT I
13 FOR I=1 TO N
14 A=1000
15 FOR J=1 TO N
16 IF A(I,J)<A THEN A=A(I,J)
17 NEXT J
18 FOR J=1 TO N
19 A(I,J)=A(I,J)-A
20 NEXT J
21 NEXT I
22 FOR J=1 TO N
23 FOR I=1 TO N
24 IF A(I,J)=0 THEN 33
25 NEXT I

```

```

26 A=1000
27 FOR I=1 TO N
28 IF A(I,J)<A THEN A=A(I,J)
29 NEXT I
30 FOR I=1 TO N
31 A(I,J)=A(I,J)-A
32 NEXT I
33 NEXT J
34 FOR I=1 TO N
35 C(I)=0
36 D(I)=0
37 NEXT I
38 FOR J=1 TO N
39 C=0
40 FOR I=1 TO N
41 IF A(I,J)=0 THEN C=C+1
42 IF C<2 THEN 45
43 C(J)=J
44 GOTO 46
45 NEXT I
46 NEXT J
47 FOR I=1 TO N
48 D=0
49 FOR J=1 TO N
50 IF A(I,J)<>0 THEN 55
51 IF J=C(J) THEN 55
52 D=D+1
53 IF D<2 THEN 54
54 D(I)=I
55 NEXT J
56 NEXT I
57 FOR J=1 TO N
58 IF J=C(J) THEN 65
59 FOR I=1 TO N
60 IF A(I,J)<>0 THEN 64
61 IF I=D(I) THEN 64
62 C(J)=J
63 GOTO 65
64 NEXT I
65 NEXT J
66 D=0
67 FOR I=1 TO N
68 IF C(I)<> 0 THEN D=D+1
69 IF D(I)<> 0 THEN D=D+1
70 NEXT I
71 IF D>=N THEN 91
72 E1=1
73 FOR I=1 TO N
74 IF I=D(I) THEN 82
75 FOR J=1 TO N
76 IF J=C(J) THEN 81
77 E(E1,1)=A(I,J)
78 E(E1,2)=I
79 E(E1,3)=J
80 E1 =E1+1

```

```

81 NEXT J
82 NEXT I
83 A=1000
84 FOR I=1 TO E1-1
85 IF E(I,1)<A THEN A=E(I,1)
86 NEXT I
87 FOR I=1 TO E1-1
88 A(E(I,2),E(I,3))=E(I,1)-A
89 NEXT I
90 GOTO 34
91 I1=0:J1=0:D=0:C=0
92 FOR I=1 TO N
93 I1=0:J1=0:Z=0
94 FOR J=1 TO N
95 IF A(I,J)<>0 THEN 99
96 IF Z=1 THEN 107
97 IF I1=0 THEN I1=I:J1=J
98 Z=1
99 NEXT J
100 IF Z<>1 THEN 107
101 PRINT"I=";I1,"J=";J1,B(I1,J1)
102 D=D+B(I1,J1)
103 FOR J=1 TO N
104 A(J,J1)=1
105 NEXT J
106 C=C+1
107 NEXT I
108 IF C<>N THEN 92
109 PRINT
110 PRINT"THE, TOTAL WORKING TIME IS:"D
111 DATA2,10,5,7,15,4,14,8,13,14,12,11,4,15,13,9
112 END

```

四十八、军官问题

方伟

十八世纪有人提出一个问题：有36个军官，他们分别在6个团，每团6人，且为6种不同的军衔，现要把他们排成方阵，使每行，每列的六个军官都来自不同的团队，而且军衔不同。

当时把这个问题求教于欧拉，欧拉的结论是：“36军官问题”是无解的，不仅如此，当阶数为 $4t + 2$ (t 为正整数)时也无解，后者被称为欧拉猜测，“36军官问题”的一般情形叫正交拉丁方问题。

1958年印度一数学家解决了欧拉猜测，他证明了当 $t=1$ 时无解，而当 $t \neq 1$ 时可解。这样“36军官问题”成了唯一的不可解正交拉丁方*。

正交拉丁方最初纯属游戏性质，后来才逐渐被用于试验设计，并在很多领域发挥作用，例如构造正交试验表等。

目前，对 $n = p^r$ (p 为素数， r 为正整数)阶的正交拉丁方是解决得最彻底的，本程序就能

生成这类正交拉丁方，不但如此还对有限域程序的构成提供了一个实例。

方法

若 $C_k (k = 1, 2, \dots, n)$ 是有限域 $GF(P^r)$ 的所有元素，设 $a_{ij}^{(e)}$ 是第 e 个正交拉丁方中 i 行 j 列的元素，则

$$a_{ij}^{(e)} = C_e \cdot C_i + C_j \quad (i, j = 1, 2, \dots, n; e = 2, 3, \dots, n)$$

当 $r = 1$ 时， n 是素数， n 阶剩余类环构成域，因而情况十分简单，此时 $C_k = k - 1$ ，加、乘运算为普通的数运算，且在 $\text{mod } n$ 下进行。这由程序 1 给出。

当 $r \neq 1$ 时，域的构成如下：

$$C_k = \begin{cases} 0 & k = 1 \\ X^{k-2} & k \neq 1 \end{cases}$$

元素的加、乘运算为普通的多项式运算，但要求：

- ① $x^r = f(x)$;
- ② 对各系数作 $\text{mod } p$ 处理。

最小函数 $f(x)$ 对不同的 p 、 r 也不尽相同。

程序说明

程序 2 只对 $p = 2$ 和 r 的几个特殊值给出了处理方案， $\text{mod } 2$ 运算被用按位进行的异或运算或非运算代替，但解决问题的基本骨架已建成，这包括：

70~120 语句，生成域的全部元素；

130~200 语句，生成域的加法表；

210~380 语句，生成域的乘法表。

其实矩阵 A 、 B 均是对称矩阵，因此只需生成它们的上三角矩阵，这样可减少一些运算量，但 390~490 的正交拉丁方输出部分就相应要复杂一点，大家可以试试。

源程序 1 及运行结果

10 INPUT P	RUN									
20 Q=P-1	? 5_									
30 FOR I=1 TO Q										
40 FOR J=0 TO Q										
50 M=I*J	1	2	3	4	5	1	2	3	4	5
60 IF M>Q THEN M=M-P*INT(M/P)	2	3	4	5	1	4	5	1	2	3
70 FOR K=0 TO Q	3	4	5	1	2	2	3	4	5	1
80 N=M+K	4	5	1	2	3	5	1	2	3	4
90 IF N>Q THEN N=N-P*INT(N/P)	5	1	2	3	4	3	4	5	1	2
100 PRINT USING "###"; N+1;										
110 NEXT K	1	2	3	4	5	1	2	3	4	5
120 PRINT	3	4	5	1	2	5	1	2	3	4
130 NEXT J	5	1	2	3	4	4	5	1	2	3
140 PRINT " "	2	3	4	5	1	3	4	5	1	2
150 NEXT I	4	5	1	2	3	2	3	4	5	1

源程序 2 及运行结果

```

10 INPUT R: IF R<20R R>7 OR R=5 THEN 10
20 N=2[R:DIMA(N,N),B(N,N),C(N,7)
30 FOR I=1 TO N
40 A(1,I)=I:B(2,I)=I:B(1,I)=1

```

```

50 A(I,1)=I:B(I,2)=I:B(I,1)=1
60 NEXT I
70 C(2,1)=1:C(3,2)=1:S=3:T=4
80 FOR I=2 TO R
90 C(T,I)=C(S,I-1)
100 NEXT I
110 IF C(S,R)=1 THEN C(T,1)=1:C(T,2)=1-C(S,1)
120 IF T<>N THEN S=T:T=T+1:GOTO 80
130 FOR I=2 TO N
140 FOR J=1 TO N
150 FOR K=1 TO R
160 D(K)=ABS(C(I,K)-C(J,K))
170 NEXT K
180 GOSUB 500
190 A(I,J)=H
200 NEXT J,I
210 FOR I=3 TO N
220 FOR J=3 TO N
230 FOR K=1 TO R
240 D(K)=0
250 NEXT K
260 FOR K=1 TO R
270 IF C(I,K)=0 THEN 350
280 FOR L=1 TO R
290 S=L+K-1:IF S>R THEN S=S-R ELSE 330
300 IF C(J,L)=0 THEN 340
310 D(S)=1-D(S):S=S+1
320 D(S)=1-D(S):GOTO 340
330 D(S)=ABS(D(S)-C(J,L))
340 NEXT L
350 NEXT K
360 GOSUB 500
370 B(I,J)=H
380 NEXT J,I
390 FOR I=2 TO N
400 FOR J=1 TO N
410 X=B(I,J)
420 FOR K=1 TO N
430 PRINT A(X,K);
440 NEXT K
450 PRINT
460 NEXT J
470 PRINT" "
480 NEXT I
490 END
500 H=1
510 FOR K=1 TO R
520 IF D(K)<>C(H,K) THEN H=H+1:GOTO 510
530 NEXT K
540 RETURN

```

RUN
? 2_

1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1

1	2	3	4
3	4	1	2
4	3	2	1
2	1	4	3

1	2	3	4
4	3	2	1
2	1	4	3
3	4	1	2

RUN
? 3_

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

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

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

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

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

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

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

四十九、梵塔问题

马松

传说在古代印度的贝拿勒圣庙里，安放有三根插在黄铜板上的宝石针。印度主神梵天在其中一根针上从下至上由大到小顺序串上了64片金属片，称为梵塔。然后要僧侣轮流值班把这些金属片在三根针上移来移去。移动时遵守以下规则：(1) 每次只能移动一个金属片；(2) 较大金属片永远不能放在较小的金属片的上面。又说如果这64片金属片全部移至另一根针上时，则世界在一声霹雳中毁灭。这个传说是有一定道理的。容易推出，N个针片从一根针座移至另一个针座需要 $2^n - 1$ 次，所以64片的移动次数为 $2^{64} - 1 = 18, 446, 744, 073, 709, 511, 615$ 。如果每秒钟移动一次，则需要近5800亿年。但科学家从能源角度推算，太阳系的寿命也

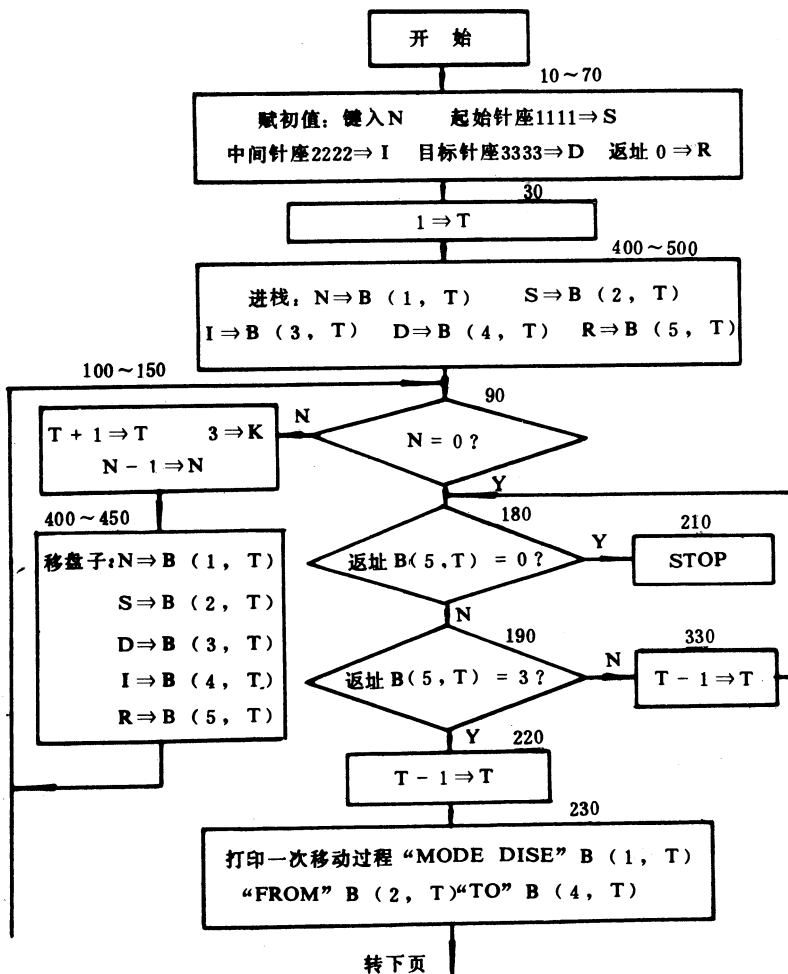
只有150亿年。

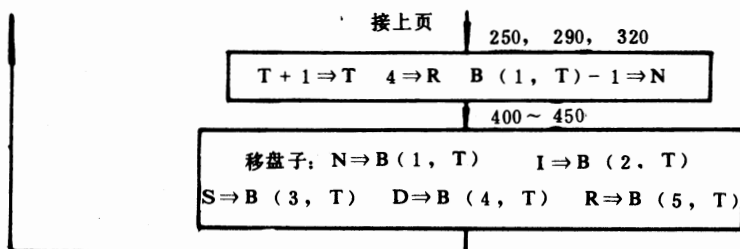
程序设计思想与语句说明

●设计思想 本程序中采用了三个针座，即起始针座（1111）、中间针座（2222）、目标针座（3333）。设有 N 个圆盘以大小递减的次序插在起始针座上，程序的算法一般需要三步：首先按大小递减的次序将 $N - 1$ 个圆盘从1111移至2222；其次把第 N 个圆盘从1111移至3333；然后使用递归过程将 $N - 1$ 个圆盘从2222移至3333。所谓递归过程就是自己调用自己的过程。显然，移动 N 个圆盘是建立在移动 $N - 1$ 个圆盘基础之上的，而移动 $N - 1$ 个圆盘又是建立在 $N - 2$ 个圆盘的基础上，如此等等。程序中数组 $B(5, n + 1)$ 作为堆栈，每个栈中有5个字段： N 字段是圆盘个数； S 、 I 、 D 字段分别是针座1111、2222、3333； R 字段是返回地址。每移完一个圆盘， N 就减一。至于移动哪一个圆盘则由当前的 N 值决定。 N 为奇数时，第一个盘从1111移至3333； N 为偶数时，第一个盘从1111移至2222。所以圆盘的移动位置是交替的。程序执行中，通过 $N = 0$ 控制转移（退栈一次）。圆盘的个数通过键盘输入。

●语句说明（见框图）

程序框图





④源程序 1 及运行结果

```

5 INPUT N
10 PRINT
20 DIM B(5,N+1)
30 LET T=1
40 LET S=1111
50 LET I=2222
60 LET D=3333
70 LET R=0
80 GOSUB 400
90 IF N=0 GOTO 180

100 LET S=B(2,T)
110 LET I=B(4,T)
120 LET D=B(3,T)
130 LET R=3
140 LET T=T+1
150 LET N=N-1
160 GOSUB 400
170 GOTO 90
180 IF B(5,T)=0 GOTO 210
190 IF B(5,T)=3 GOTO 220

200 GOTO 330
210 STOP
220 LET T=T-1
230 PRINT"MOVE DISK" B(1,T), "FROM" B(2,T), "TO" B(4,T)
)
240 PRINT
250 LET N=B(1,T)-1

260 LET I=B(2,T)
270 LET S=B(3,T)
280 LET D=B(4,T)
290 LET R=4
300 LET T=T+1
310 GOSUB 400
320 GOTO 90
330 LET T=T-1

340 GOTO 180
350 END
400 LET B(1,T)=N
410 LET B(2,T)=S
420 LET B(3,T)=I
430 LET B(4,T)=D
440 LET B(5,T)=R
450 RETURN

```

RUN

? 3_

```
MOVE DISK 1      FROM 1111      TO 3333
MOVE DISK 2      FROM 1111      TO 2222
MOVE DISK 1      FROM 3333      TO 2222
MOVE DISK 3      FROM 1111      TO 3333
MOVE DISK 1      FROM 2222      TO 1111
MOVE DISK 2      FROM 2222      TO 3333
MOVE DISK 1      FROM 1111      TO 3333
```

BREAK IN 210

READY

>

另外，再介绍一个梵塔问题的非递归无栈算法。

这一算法的基本思想是：设金属片（或称圆盘）数为 N ，设置一个 N 位的二进制计数器每次加 1 时，计数器中有且仅有一位由“0”变为“1”，而这一位的下标正好是本次要移动的盘的盘号。

定理 1：对于一个 N 位的非全“1”的二进制计数器，令最左位为最高位，最右位为最低位，从右到左每位的下标依次为 $1, 2, \dots, n$ 。给计数器加 1 时，有且仅有一位从“0”变“1”。

证明：因为此二进制数是非全“1”的，所以至少存在一位是“0”。

对于所有是“0”的那些位中，设其中下标最小的那一位的下标为 m ，如果 $m = 1$ ，则此二进制数加 1，仅引起第 1 位由“0”变“1”。

另一方面，如果下标为 m ， $1 < m \leq n$ ，则所有下标小于 m 的 $m - 1$ 位都为“1”，此二进制数加 1，则使 $m - 1$ 位的“1”变为“0”，产生的进位进入第 m 位，则第 m 位由“0”变为“1”，而高于 m 位的所有位都不变，于是定理得证。

定理 2：有 N 个圆盘的梵塔的最少移动步数是 $2^N - 1$ 次（这一结论对递归算法和本算法

都成立)。

证明: 为表示方便起见, 引入函数 $\text{step}(n)$, 表示 n 个盘的梵塔所需要的移动步数。

对于 $n < 0$, 定义 $\text{step}(n) = 0$

从梵塔问题的递归算法, 我们得出如下关系式:

$$\text{step}(n) = \text{step}(n-1) + 1 + \text{step}(n-1) = a \cdot \text{step}(n-1) + 1$$

使用数学归纳法:

对于 $n = 0$, $\text{step}(0) = 0 = a - 1$

事实上, 还可考虑 $n = 1$ 时, $\text{step}(1) = a \cdot \text{step}(1-1) + 1 = 1 = a - 1$

以此做为归纳基础, 设 $n > 0$ 时, $\text{step}(n) = a^n - 1$ 成立, 则 $\text{step}(n+1) = a \cdot \text{step}(n) + 1 = a(a^n - 1) + 1 = a^{n+1} - 1$

于是定理得证。

设梵塔的三根针分别标记为 1, 2, 3, 其中 1 号为开始针, 2 号为过渡针, 3 号为终止针, 此外盘从小到大标记为 1, 2, …… n , 现在的问题是每次要移动的是几号盘? 将移到几号针?

考虑 $n = 1$ 和 $n = 2$ 的情况, 若 $n = 1$, 则将 1 号盘移到 3 号针; 若 $n = 2$, 则将 1 号盘移到 2 号针, 2 号盘移到 3 号针。再将 1 号盘从 2 号针移到 3 号针。因此, 我们得到如下结论: 在最初开始移动 1 号盘的时候, 如果 n 为奇数, 则将它移到 3 号针, 若 n 为偶数, 则将它移到 2 号针。接下来我们就可以使用 $a^n - 1$ 次的循环来解决这一问题。其核心思想正是基于定理 1 和定理 2 的。而每次要移动的盘的编号等于二进制计数器加 1 后由“0”变为“1”的那一位的下标, 而循环次数为 $a^n - 1$ 次。例如 $n = 5$, 在进行第六次移动时, 二进制计数器由 5 (00101) 变为 6 (00110), 第 2 位由“0”变“1”, 所以本次要移动的是 2 号圆盘。

一旦我们确定了要移动的盘的盘号, 还要确定将它移到几号针, 显然, 只有两种选择, 但注意到梵塔的规则——不允许任何大盘放在小盘上面, 这样, 1 号盘总是位于某针的最上面。这样我们记住 1 号盘现在在何处, 以及本次要移动的盘现在何处, 我们就可确定本次将盘移到何处了。用下式表示

TO needle = 6 - Where-I-is - Where-disk-is

上述这些情况都考虑了以后, 就可以实现本算法了。下面给出程序清单和运行 $n = 3$ 的结果。

源程序 2

```
10 REM TOWERS OF HANOI ..... A NON-RECURSIVE ALGORITHM
M
20 INPUT "ENTER NUMBER OF DISKS WANTED ?"; T
30 IF T<=0 THEN GOTO20
40 DIM B(T),H(T)
50 FOR I=1 TO T
60 LET B(I)=0
70 LET H(I)=1
80 NEXT I
90 LET M=3-(T-INT(T/2)*2)
100 FOR J=1 TO 2[T-1
110 GOSUB 240
120 LET D=I
130 IF D<>1 THEN GOTO 180
140 LET F=H(1)
```

```

150 LET T2=6-F-M
160 LET M=F
170 GOTO200
180 LET F=H(D)
190 LET T2=6-H(1)-H(D)
200 PRINT"MOVE DISK";D;"FROM NEEDLE";F;" TO NEEDLE ";T
2
210 LET H(D)=T2
220 NEXTJ
230 STOP
240 LET I=1
250 IF B(I)<>0 THEN GOTO 280
260 LET B(I)=1
270 RETURN
280 LET B(I)=0
290 LET I=I+1
300 GOTO250
310 END

```

程序中，B是二进制计数器，H(I)表示第i个盘现在几号针，50~80号语句是给计数器清0，H全部赋1的意义为开始时所有盘都在一号针，240~300号语句是一个子程序，其功能是给二进制计数器加1，并把由“0”变“1”的那一位的下标赋给变量I，100~220号语句是一个完成 $2^n - 1$ 次移动盘子的循环。可以看到，本程序比上一个程序要简练一些。

运行结果

```

RUN
ENTER NUMBER OF DISKS WANTED ?? 3
MOVE DISK 1 FROM NEEDLE 1 TO NEEDLE 3
MOVE DISK 2 FROM NEEDLE 1 TO NEEDLE 2
MOVE DISK 1 FROM NEEDLE 3 TO NEEDLE 2
MOVE DISK 3 FROM NEEDLE 1 TO NEEDLE 3
MOVE DISK 1 FROM NEEDLE 2 TO NEEDLE 1
MOVE DISK 2 FROM NEEDLE 2 TO NEEDLE 3
MOVE DISK 1 FROM NEEDLE 1 TO NEEDLE 3
BREAK IN 230
READY
>_

```

五十、地图染色

蔡琪君

在地图染色过程中，一般要求相邻的区域使用不同的颜色，那么给一张地图染色至少要几种颜色？一百多年前，英国的格色里提出了用四种颜色对地图染色的猜想以来，吸引了许多数学家，但终未获得证明，直到1976年，美国数学家阿佩尔和黑肯借助电子计算机才证明了这一猜想，使之成为四色定理。

程序设计思想与说明

①设计思想。用最少的颜色（不一定是四种，可能多于四种）给地图染色的算法不少，下面用韦尔奇·鲍威尔方法来给地图染色。

由于地图是一种平面图，所以用一个结点代表图中的一个面（包括外界面），然后把相邻面所对应的结点联接起来，就得到与原图对应的对偶图，同时两个结点都编上号，这样就得到一组表示结点邻接关系的数据如下：

DATA<结点i>,<与结点i邻接的结点1>……,<与结点i邻接的结点k>,0

其中 k 就是结点 i 的度数，例如，图 1 所示的平面图所对应的对偶图和数据如图 2 所示

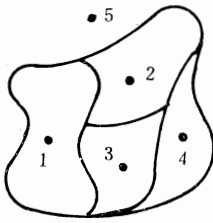


图 1

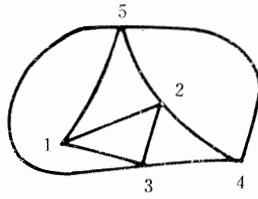


图 2

按度数的递减排列

DATA 2, 1, 3, 4, 5, 0

DATA 3, 1, 2, 4, 5, 0

DATA 5, 1, 2, 3, 4, 0

DATA 1, 2, 3, 5, 0

DATA 4, 2, 3, 5, 0

鲍威尔算法的基本思想：

(1) 将图中的结点按度数从大到小排列。

(2) 用一种颜色对第一点染色，并按排列顺序，对与前面已染色的结点不邻接的每一点染上同样的颜色。

(3) 用第二种颜色对未染色的结点重复第 2 步，再用第三种第四种……，继续上述步骤，直到所有结点都被染色为止。

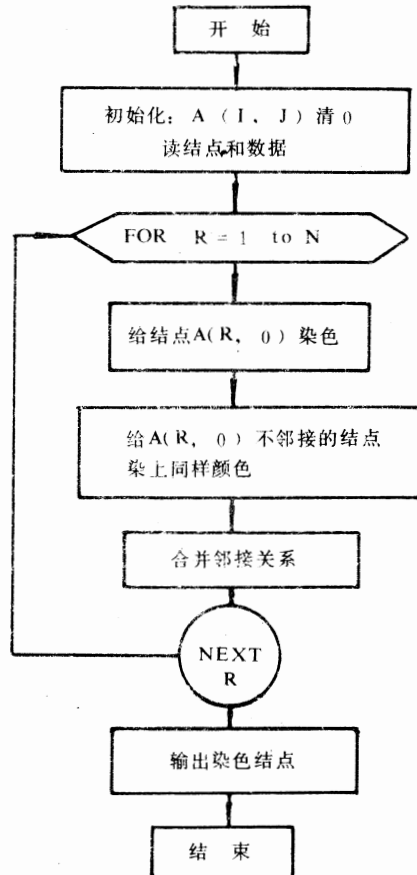
②程序语句说明 15~95 语句是对邻接矩阵 A 进行初始化。其中 A 的第 0 列的 $1 \sim N$ 个元素 $A(i, 0)$ 对应着排序后的结点；第 0 行的第 $1 \sim N$ 个元素 $A(0, i)$ 对应着第 i 个结点在排序后的结点序列中的位置；矩阵元素 $A(i, j)$ 代表与结点 $A(i, 0)$ 邻接的结点，即 $(i, j) \neq 0$ 则表示结点 $A(i, 0)$ 与结点 $A(0, j)$ 邻接。

105~175 语句就是实现鲍威尔算法的程序。当给结点序列中的第 R 个结点染上一种颜色以后，就对结点 $A(R, 0)$ 后的与之不邻接的结点也染上同样的颜色，并把它的邻接关系并到结点 $A(R, 0)$ 上去。重复上述过程，第一次循环结束就换一种颜色，再做上面的步骤，直到所有结点都被染色为止。

180~215 语句为输出染色结点。

235~295 语句是对中国地图进行染色的邻接关系的数据语句，格式同图 2；各省市自治区的代号为：京 1、津 2、冀 3、晋 4、蒙 5、辽 6、吉 7、黑 8、沪 9、苏 10、浙 11、皖 12、闽 13、赣 14、鲁 15、豫 16、鄂 17、湘 18、粤

程序框图



19、桂20、川21、贵22、滇23、藏24、陕25、甘26、青27、宁28、新29、台30、外界31。输出结果表明共使用了五种颜色，并且相邻省市自治区之间的颜色是不同的。

源程序

```
3 REM ALGORITHM OF COLORING A GRAPH USING "WELCH.POWEL
L METHOD"
10 REM INITIALIZATION
15 INPUT "HOW MANY NODES -->";N
20 DIM A(N,N)
25 FOR I=1 TO N
30 FOR J=1 TO N
35 A(I,J)=0
40 NEXTJ
45 NEXT I
50 FOR I=1 TO N
55 READ K
60 A(I,0)=K
65 A(I,K)=1
70 READ K
75 IF K=0 THEN GOTO90
80 A(I,K)=1
85 GOTO70
90 A(0,A(I,0))=I
95 NEXT I
100 REM COLOR THE GRAPH
105 C1=1
110 FOR R=1 TO N
115 IF A(R,0)<0 THEN GOTO 175
120 A(R,0)=-C1
125 FOR I=R+1 TO N
130 B=A(I,0)
135 IF B<0 THEN GOTO165
140 IF A(R,B)<>0 THEN GOTO165
145 A(I,0)=-C1
150 FOR J=1 TO N
155 A(R,J)=A(R,J)+A(I,J)
160 NEXT J
165 NEXT I
170 C1=C1+1
175 NEXT R
180 REM PRINT OUT THE GRAPH COLORED
185 PRINT
190 PRINT"THE GRAPH COLORED : "
195 PRINT"      NODE          COLOR      "
200 PRINT"-----"
205 FOR I=1 TO N
210 PRINT TAB(6);I; TAB(20);-A(A(0,I),0)
215 NEXT I
220 END
235 DATA 31,2,3,5,6,7,8,9,10,11,13,15,19,20,23,24,26,2
9,30,0
240 DATA 5,26,28,25,4,3,6,7,8,31,0,21,17,25,26,27,24,2
3,22,18,0
245 DATA 3,1,2,4,6,5,16,15,31,0,25,4,5,28,26,21,17,16,
0
```

```

250 DATA 26,5,29,27,21,25,28,31,0,12,10,15,16,17,14,11,0
255 DATA 14,11,12,17,18,19,13,0,16,15,3,4,25,17,12,0
260 DATA 17,12,16,25,21,18,14,0,18,14,17,21,22,20,19,0
265 DATA 11,9,10,12,14,13,31,0,10,9,15,12,11,31,0
270 DATA 15,3,16,12,10,31,0,19,13,14,18,20,31,0
275 DATA 20,19,18,22,23,31,0,23,20,22,21,24,31,0
280 DATA 24,23,21,27,29,31,0,6,3,5,7,31,0,4,3,5,25,16,0
285 DATA 22,18,21,23,20,0,27,26,28,24,21,0,7,8,5,8,31,0
290 DATA 13,11,14,19,31,0,29,26,27,24,31,0,28,25,5,26,0
295 DATA 2,1,3,31,0,8,5,7,31,0,9,10,11,31,0,1,3,2,0,30,31,0

```

五十一、键盘控制行印机作图

范幸义 张富蓉

目前，在微型机上作图，一般都是通过程序控制或者通过键盘控制，在显示器上作图。作好图后，再存入磁带或磁盘。要得到硬拷贝，再通过打印机把原图形打印出来。

本文介绍的程序可以由计算机键盘直接控制行印机打印各种图形、图表、函数曲线、以及打印各种中文字体、彩色花边、彩色图案。

微计算机PC—1500配有CE—150微型行印机，此微型行印机具有很强的打印功能。它属于喷墨式打印机，配有四种颜色的打印笔，分黑色、蓝色、红色和绿色。可以由计算机直接控制更换打印笔。它的打印纸只有5.8cm宽，最小字符每行可打印36个，是一种微型行印机。它有一个最大的特点，具有点控坐标打印功能，它的打印笔可以横向任意移动打印，还可以纵向任意移动打印（实际上是打印纸作上、下移动）。这是一般行印机所没有的功能。本程序利用这一特点，并利用PC—1500计算机作图语句的功能，做到直接用按键来控制打印图形，取得了比较满意的结果。

程序结构

本程序总的来说分为两部分。前一部分是键盘扫描，监视按键操作。除程序中所定义的按键以外，按其它键均不起作用（中断键除外）。后一部分为打印操作程序。

程序运行时不断扫描键盘，若打入的是“A、B、C、D、E、F、G、H、Q”键，计算机即控制行印机作出相应的打印操作。因此，以上各键都是作图功能转换键，其具体功能是：

“A”键，打印笔空走到指定位置（X，Y）处；“B”键，打印直线；“C”键，打印斜线；“Q”键，用来恢复作图初始状态。“D、E、F、G”键分别用来更换黑、蓝、红、绿色打印笔。“H”键，结束程序运行。

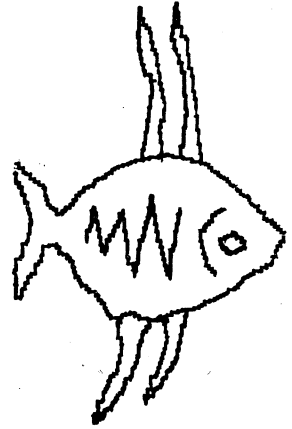
为了简化程序，打印时具体操作一律用四个键来控制，“<”键，打印笔左移（包括空走式打印，以下相同）；“>”键，打印笔右移；“*”键，打印笔上移；“+”键，打印笔下移。若需更换打印功能，一律用“Q”键来恢复功能选择。

例如运行本程序，先按“E”键，行印机立即更换成蓝色打印笔。然后按“A”，表示打印笔空走功能。这时可用“<、>、*、+”四个键来控制打印笔空走到指定的打印位置。然后按“Q”键来恢复功能选择。这时要按下“B”键，表示直线打印。可以用“<、>、*、+”来控制行印机打印横线和竖线。并可以分别用“<、*”和“>、+”来打印曲线。在打印的当前位置，你可以随时按“Q”键来作功能选择；更换打印笔，可以得到你要的彩色图案。

程序中，INKEY\$函数得到键盘瞬时输入的单个字符，GRAPH表示坐标打印；GLCURSOR(X, Y)表示打印从坐标(X, Y)处开始；LINE(X1, Y1) - (X2, Y2)表示从指定位置开始连续打印到(X1 - X2, Y1 - Y2)处。COLOR表示更换打印笔。

本程序采用的坐标增量为1，因此可以打印一个最小的基本点。PC—1500微型计算机的键盘缓冲寄存器具有自动清零功能。因此，本程序中的操作键具有全自动功能，只要你按住键不放，行打印机就一直连续打印。所以，作图具有一定的速度。

运行本程序，你不必花很大精力去构思“作图程序”，就能在键盘上很快地操作打印机，打印各种有趣的，具有四种颜色的图案。右图是用按键直接打印的“神仙鱼”图形。



源程序

```

5 : X=0:Y=0
10 :GRAPH:GLCURSOR(X1,Y1):SORGN
15 :A$=INKEY$
20 :ON ABS(64-ASC(A$)) GOTO100,200,300,30,35,40,45,800
25 :GOTO15
30 :COLOR 0:GOTO 15
35 :COLOR 1:GOTO15
40 :COLOR 2:GOTO15
45 :COLOR 3:GOTO15
100 :B$=INKEY$
103 :IF B$="Q"GOTO15
105 :ON ABS(39-ASC(B$))GOTO150,160,170,180
110 :GOTO100
150 :X=X-1:X1=X:GLCURSOR(X1,Y1):GOTO100
160 :X=X+1:X1=X:GLCURSOR(X1,Y1):GOTO100
170 :Y=Y+1:Y1=Y:GLCURSOR(X1,Y1):GOTO100
180 :Y=Y-1:Y1=Y:GLCURSOR(X1,Y1):GOTO100
200 :C$=INKEY$
205 :IF C$="Q"GOTO15
210 :ON ABS(39-ABS(C$))GOTO250,260,270,280
220 :GOTO200
250 :X1=X:X=X-1:LINE(X1,Y1)-(X,Y1):GOTO200
260 :X1=X:X=X+1:LINE(X1,Y1)-(X,Y1):GOTO200
270 :Y1=Y:Y=Y+1:LINE(X1,Y1)-(X1,Y):GOTO200
280 :Y1=Y:Y=Y-1:LINE(X1,Y1)-(X1,Y):GOTO200
300 :D$=INKEY$
305 :IF D$="Q" GOTO15
310 :ON ABS(39-ASC(D$))GOTO500,510,520,530
320 :GOTO300

```

```

500 :X1=X:X=X-1:Y1=Y:Y=Y+1:LINE (X1,Y1)-(X,Y):GOTO300
510 :X1=X:X=X+1:Y1=Y:Y=Y+1:LINE (X1,Y1)-(X,Y):GOTO300
520 :X1=X:X=X-1:Y1=Y:Y=Y-1:LINE (X1,Y1)-(X,Y):GOTO300
530 :X1=X:X=X+1:Y1=Y:Y=Y-1:LINE (X1,Y1)-(X,Y):GOTO300
800 :END

```

五十二、恭贺新禧

刘炳文

在显示器上写字的方法很多，下面介绍的方法是通过调用子程序来“写”出字的各个笔划。这个子程序能连接屏幕上任意两点画出一条直线。

程序设计思想与语句说明

①设计思想 语句 350~620 是连接两点画直线的子程序。它首先把y方向的长度($y_2 - y_1$)和在x方向的长度($x_2 - x_1$)进行比较，如果x的长度较大，那么该程序将从 x_1 到 x_2 ，把x的步长定为1，x每增加一个步长，就把y的一个“ Δ ”值加到当前的 y_1 值上。Dy的值系由y的长度被x步长的个数除求得。如果x的长度小于y的长度，该程序将从 y_1 到 y_2 ，把y的步长定为1，并通过增加 Δx 变动 x_1 。在Dx或Dy的递增过程中，有可能使 y_1 超过47或者 x_1 超过127，但是x或y值超过的部份将被舍弃。如果规定适当的参数， x_1 和 y_1 的值将始终不会超过127和47。

当 x_1 或 y_1 的值减小时，由于 Δ 是小数， x_1 或 y_1 可能会变成绝对值很小的负数，因此有必要对 x_1 或 y_1 是负数的情况进行测试。当计算得出的值是负数时，就令变量为零。不过负数是在打开最靠边的点时才有可能出现，即仅出现在屏幕的顶部或最左边。

利用这个子程序可以画出连接屏幕上任意两点的直线，其方法是，在DATA语句中设置数据，通过读入四个一组的数据，调用子程序就可以画出一条直线，反复读入若干组 $x_1, y_1; x_2, y_2$ 的值，就能画出一系列的直线。

②语句说明 10~250语句是组成恭贺新禧四个字的全部数据（-1除外）。

310~340语句是用READ语句读入这些数据，反复调用子程序，可按普通的书写笔顺，由计算机在屏幕上“写”出四个篆字来。

为了增强效果，程序用“白底黑字”显示。这需要在开始时将屏幕“画白”。语句270~300是将屏幕“画白”的程序，采用字符串法，所用时间不到一秒。另外，还要在“画白”的屏幕上关闭一些像素，因而要把子程序中的所有SET语句改成RESET语句。

源程序

```

10 DATA 4,12,31,12,11,10,11,15,23,10,23,15,11,15,23,15
20 DATA 5,17,5,20,5,20,15,20,15,17,15,20,10,18,13,18,1
0,18,10,22
30 DATA 6,22,10,22,6,22,6,34
40 DATA 19,17,19,20,19,20,29,20,29,17,29,20
50 DATA 21,18,23,18,23,18,23,22,23,22,27,22,27,22,27,3
4

```

```

60 DATA 14,23,14,32,14,32,18,32,18,23,18,32
70 DATA 10,24,10,28,10,28,14,28,23,24,23,28,18,28,23,2
8
80 REM *::*****:::*****::
90 DATA 37,10,40,10,37,10,37,13,37,13,40,13,40,13,40,2
0
100 DATA 35,15,35,20,35,15,46,15,46,15,46,20
110 DATA 49,10,49,20,49,20,60,20,60,10,60,20,49,11,60,
11
120 DATA 38,22,38,32,38,32,58,32,58,22,58,32,38,23,58,
23
130 DATA 38,26,58,26,38,29,58,29,38,32,58,32
140 DATA 44,32,44,34,52,32,52,34
150 REM :*****:::****:::***
160 DATA 66,11,75,11,64,14,77,14,66,14,66,16,66,16,74,
16,74,14,74,16
170 DATA 66,19,75,19,66,22,66,33,66,22,75,22,75,22,75,
33,70,16,70,34
180 DATA 79,10,91,10,91,10,91,14,79,14,91,14,79,14,79,
34,81,17,90,17,81,17,90,34
190 REM :::*****:::*****:
200 DATA 96,11,104,11,94,14,106,14,96,17,96,33,96,17,1
04,17,104,17,104,33,100,14,100,34
210 DATA 108,11,108,15,109,15,119,15,119,11,119,15,113
,10,113,18
220 DATA 105,18,121,18,109,20,103,24,109,24,117,24,117
,20,117,24,109,21,117,21
230 DATA 105,26,121,26,113,24,113,29,107,28,107,34,107
,34,119,34,119,28,119,34,107,29,119,29
240 DATA 2,3,125,3,125,3,125,43,125,43,2,43,2,43,2,3
250 DATA -1,-1,-1,-1
260 CLS: CLEAR 500
270 A$=STRING$(64,CHR$(191))
280 FOR I=1 TO 16
290 PRINT A$:
300 NEXT I
310 READ X1,Y1,X2,Y2
320 IF X1=-1 GOTO 320
330 GOSUB 350
340 GOTO 310
350 IF ABS(X2-X1)<= ABS(Y2-Y1) GOTO 490.

360 DY=(Y2-Y1)/ABS(X2-X1)
370 IF X2>X1 GOTO 440
380 FOR I=X1 TO X2 STEP -1
390 RESET(I,Y1)
400 Y1=Y1+DY
410 IF Y1<0 THEN Y1=0
420 NEXT I
430 RETURN
440 FOR I=X1 TO X2
445 RESET(I,Y1)
450 Y1=Y1+DY
460 IF Y1<0 THEN Y1=0
470 NEXT I
480 RETURN

```

```

490 DX=(X2-X1)/ABS(Y2-Y1)  560 RETURN
500 IF Y2>Y1 GOTO 570      570 FOR I=Y1 TO Y2
510 FOR I=Y1 TO Y2 STEP -1 580 RESET(X1,I)
520 RESET(X1,I)           590 X1=X1+DX
530 X1=X1+DX              600 IF X1<0 THEN X1=0
540 IF X1<0 THEN X1=0     610 NEXT
550 NEXT I                620 RETURN

```

运行该程序后，首先显示出四个篆字，然后在四周画上边框。如增加几行程序，还可以让让各种不同的小图形在字旁闪烁。

五十三、用功能键作图

张宝玺

这里介绍一个绘图程序，它只需在键盘上按几个功能键，就可以作出各种图形，比之用程序拼接的作图方法方便多了。它可以作为游戏程序，有利于培养绘画、写美术字的能力。该程序还可以把作出的图形存到磁盘上，需要时，可将单个或多个图形链接显示出来。

程序的设计思想和语句说明

该程序是基于TRS-80机BASIC语言具有的图示功能。点亮一个图形相应部份的图示块，(以下简称光点)就可以把一个图形完整地显示于荧光屏上，再把已点亮的图示块的坐标位置转换成相应的数据存到磁盘上，需要重新显示图形时，读入这些数据，去点亮图示块，从而使一幅图形再现。

为了作图的需要，设计了几个功能键。H、L、J、K四个键，用来使光点左右上下移动。按下@键后，再按H、L、J、K键可使光点经过路径上的图示块点亮；按下T键后，@键功能消失；按下Q键，把已作出的图形存到磁盘上；按下E键，取出磁盘上某个未完成的图形到内存显示，并转入作图部分。

①绘图程序说明

绘图程序由三部分组成。

a. 完成作图与修改功能。图形是作为数据文件存到磁盘上的，每一个图形有一个文件名，记作TXWJN，其中N是不同的，以区别不同的图形，供以后存取用。

30~60语句：是为方便作图设计的坐标。

70语句：使光点预置到坐标点(10, 10)位置。

80~120语句：移动光点到作图位置并记录下光点的坐标位置。

130~150语句：完成@、Q、E功能键的功能转换。

170~250语句：完成作图功能。

b. 输出图形数据到磁盘。

300语句：根据开机时输入的编号产生数据文件的文件名和文件标识符。

310语句：打开顺序输出文件。其中A=130是屏幕上一行数据结束的标志。

320~360语句：组成双重循环，扫描整个屏幕，得到亮点的图示块的X坐标值存盘。

310语句：以顺序输出方式建立文件，使用一号缓冲器。

c. 读入图形数据显示, 并转入修改完善。此部分程序实际上是一个完整地显示某一指定图形的程序。与第二部分存盘程序配合, 在绘图过程中断情况下, 可将未完成图形再现, 或者供修改图形用。

400语句: 要求键入欲显示图形的编号并根据输入的编号产生此图形的文件名。

440~470语句: 读入数据并显示原图形。

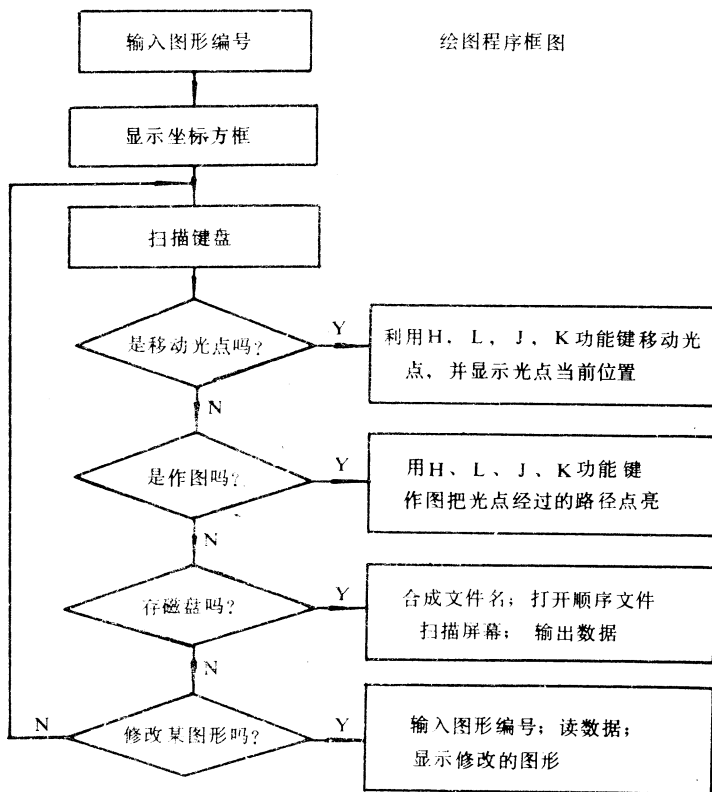
480语句: 关闭文件并转入程序的第一部分。

②链接程序说明

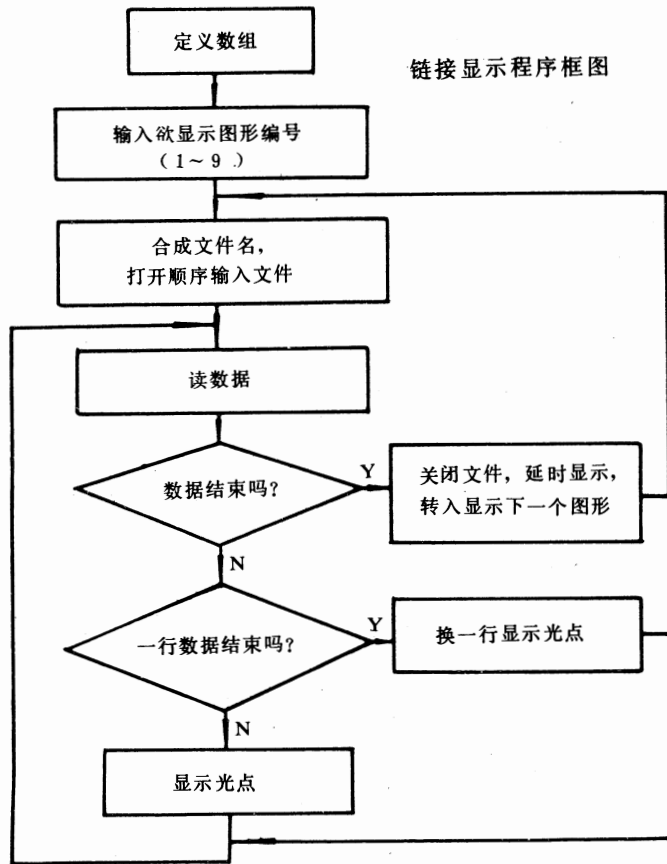
某些场合需要把绘出的图形(或字)接连不断地显示在荧光屏上, 要求有图形的显示和链接功能。这里采用一个一维数组A(9)存键盘输入的文件名, 再用循环的办法交替地取不同名字的数据到内存显示。考虑到磁盘空间的限制和实际需要, 每次显示图形最多为9个。键入图形编号时应先输入数字, 接着输入一逗号(或按回车), 再出现问号时, 再给数字并按回车。无输入时只简单按回车即可结束输入。程序则循环显示。

此程序和第一个程序的第三部份相似, 只是增加了130语句的延时功能, 目的是使显示的图形在屏幕上显示一段时间后再转入下一个图形的显示。

程序框图



链接显示程序框图



源程序

```

10 INPUT "BIAN XAO";C
20 CLS
30 FOR I=0 TO 126:SET(I,0):SET(I,47):NEXT
40 FOR I=0 TO 47 :SET(0,I):SET(127,I):NEXT
50 FOR I=3 TO 47 STEP 4 :SET(126,I):SET(1,I):NEXT
60 FOR I=15 TO 112 STEP 16:SET(I,1):SET(I,46):NEXT
70 X=10 :Y=10 :SET(X,Y)
80 A$=INKEY$: IF A$="" GOTO 80
90 IF A$="H" X=X-1: IF X<0 GOTO 80 ELSE SET(X,Y):RESET(X+1,Y):GOTO 80
100 IF A$="L" X=X+1: IF X>127 GOTO 80 ELSE SET(X,Y):RESET(X-1,Y):GOTO 80
110 IF A$="J" Y=Y-1 : IF Y<0 GOTO 80 ELSE SET(X,Y):RESET(X,Y+1):GOTO 80
120 IF A$="K" Y=Y+1: IF Y>47 GOTO 80 ELSE SET(X,Y):RESET(X,Y-1):GOTO 80
130 IF A$="@" GOTO 170
140 IF A$="Q" GOTO 260
150 IF A$="E" GOTO 360
160 GOTO 80
170 A$=INKEY$: IF A$="" GOTO 170
  
```

```

180 IF A$="H" X=X-1:IF X<0 GOTO 170 ELSE SET(X,Y):GOTO
  170
190 IF A$="L" X=X+1 :IF X>127 GOTO 170 ELSE SET(X,Y) :
GOTO 170
200 IF A$="J" Y=Y-1:IF Y<0 GOTO 170 ELSE SET(X,Y):GOTO
  170
210 IF A$="K" Y=Y+1 :IF Y>47 GOTO 170 ELSE SET(X,Y):GO
TO 170
220 IF A$="T" GOTO 80
230 IF A$="Q" GOTO 300
240 IF A$="E" GOTO 400
250 GOTO 170
260 A$="STJ"+RIGHT$(STR$(C),1)+"/BAS:1"
270 OPEN "O",1,A$:A=130
280 FOR Y=0 TO 47
290 PRINT#1,A;
300 CLOSE
310 GOTO 10
320 INPUT Y:A$="STJ"+RIGHT$(STR$(Y),1)+"/BAS:1":OPEN "
I",1,A$
330 Y=-1:CLS
340 INPUT #1,X
350 IF EOF(1)GOTO 400
360 IF X=130 Y=Y+1 ELSE GOTO 380
370 IF Y=48 GOTO 400 ELSE GOTO 340
380 SET(X,Y)
390 GOTO 340
400 CLOSE
410 GOTO 70

```

```

10 DIM A(9)
20 INPUT "BIAN XIAO 1,2,---,9":A(1),A(2),A(3),A(4),A(5
),A(6),A(7),A(8),A(9)
30 FOR I=1 TO 9
40 IF A(I)=0 GOTO140 ELSE D$=MID$(STR$(A(I)),2,1)
50 A$="STJ"+D$+"/BAS:1":OPEN"I",1,A$
60 Y=-1:CLS
70 INPUT#1,X
80 IF EOF(1) GOTO70
90 IF X=130 Y=Y+1:GOTO70
100 SET(X,Y)
110 GOTO70
120 CLOSE
130 FOR F=1 TO 1000:X=1:NEXT
140 NEXT I
150 GOTO30

```

五十四、模拟光笔

杨 舒

模拟光笔程序是在显示器上用功能键作图的又一例子。它利用微机上的辅助数字键盘作为走笔控制。以 4、5（或 1、2）代表“—”，4、1（或 5、2）代表“ ”，5、1代表

“/”，4、2代表“\”来形象地完成任一图字的基本笔划，使笔尖（光标）可顺向或逆向地在“米”字型的八个方向上自由移动，并能够随时进行走笔、描划、抹除以及定点存取图字、改变字模格式和消字等操作。这样，不仅可以实时写画修改，还可方便地复合出新的字句及图案，形成一种富有趣味性的图字处理功能。

程序中10—90行为初始准备；100—260行为走笔控制；270~590行为功能处理。程序运行后，首先询问字模大小，W为宽，H为高。以下操作均以字模为单位进行。此时设原点在（0，0）位置。操作的初态，光标在字模的左上角闪烁，且为抬笔状态。走笔时，先按一次某方向的组合数字键（如4、5为横走，亦可反向组合如5、4为逆向走等），而后只需屡屡按动该组合的后一个数字键，即可在该方向上每每向前移动一个光点，或随时按动功能键选通某种特定的操作。

各功能键的用途如下：

“.”：定笔键，按一次为落笔，再按一次为抬笔。抬笔后移动光标，不清除原有亮点。

“φ”：抹除键，抹掉光标所经之处的亮点。可由“.”键或以下的功能键解除。

“SPACE”：空格键，以字模的宽为单位横向移动字模。

“ENTER”：换行键，以字模的高为单位竖向移动字模。

“C”：清除键，抹去光标当前字模的图字显示。

“W”：写入键，将光标当前字模以用户定义的名字按点阵存贮方式（顺序存取）写入磁盘。

“R”：读出键，从磁盘中读出给定名字的图字文件，并在光标现行位置进行显示，且改变了原点和原字模格式，若遇到没有的文件名，将进行第二次询问。

“P”：拷贝键，在宽行上打印出光标当前字模的屏幕显示。

“Q”：退出键，退出当前字模方式，以光标现行位置为原点重新定义字模大小，若越界，以整个屏幕为一字模，且改变了原点。

“S”：停止键，停止功能C、W、R、P的操作，返回初态。

“BREAK”：中断键，中断执行，返回 BASIC 命令状态。

一旦掌握了模拟光笔，显示屏幕就将成一块具有记忆功能的黑板，任你随意写画和存取，必要时还能在宽行上得到一份拷贝。

源程序

```

10 DEFINT A-Y: DIM Z(385): X#=CHR$(13)+". 0CPRWQ": CLS: H=
-1
20 IF H=0 THEN GOSUB 570 : A=0 : B=0
30 T#="W=": GOSUB 530: A=VAL(N#): IF A<>0 T#="H=": GOSUB 53
0 : B=VAL(N#)
40 IF A=0 OR B=0 A=127 : B=48
50 X0=X: Y0=Y: A=A-1: B=B-1: IF H CLS
60 H=0: X1=X0: Y1=Y0
70 X=X1: Y=Y1: V=0: W=0: P=0: T=0: Q=0
80 IF X1+A>126 OR Y1+B>47 THEN X0=0: Y0=0: A=126: B=47: GO
TO 60
90 SET(X+A+1, Y): SET(X, Y+B): SET(X+A+1, Y+B)
100 SET(X, Y): RESET(X, Y): A#=INKEY#: IF A#="" THEN 100
110 IF P THEN SET(X, Y) ELSE IF T THEN SET(X, Y)
120 IF Q RESET(X, Y)

```

```

130 R=INSTR(X$,A$):IF R<>0 GOTO270
140 U=VAL(A$):IF U<>V THEN IF V-U<>0 THEN W=V-U:V=U:GOT
0100
150 ON ABS(W) GOTO160,180,200,220
160 IF W>0 X=X-1 ELSE X=X+1
170 GOTO230
180 IF W>0 X=X+1:Y=Y+1 ELSE X=X-1 :Y=Y-1
190 GOTO230
200 IF W>0 Y=Y+1 ELSE Y=Y-1
210 GOTO240
220 IF W>0 X=X-1:Y=Y+1ELSE X=X+1:Y=Y-1
230 IF X<X1 THEN X=X1 ELSE IF X>X1+A THEN X=X1+A
240 IF Y<Y1 THEN Y=Y1 ELSE IF Y>Y1+B THEN Y=Y1+B
250 IF P=0 THEN IF POINT(X,Y) THEN T=-1 ELSE T=0
260 GOTO100
270 ON R GOTO 310,280,300,390,360,320,480,400,20
280 P=NOT(P):Q=0
290 V=0:W=0:GOTO250
300 GOSUB570:X1=X1+A+1:IF X1+A>126X1=X0:GOTO70ELSE 70
310 GOSUB 570:X1=X0:Y1=Y1+B+1:IF Y1+B>47 Y1=Y0 :GOTO60
ELSE 70
320 LPRINT CHR$(29):GOSUB 570 :FOR I=Y1 TO Y1+B
330 LPRINT TAB(X1):FOR L=X1 TO X1+A
340 IF POINT(L,I)LPRINT CHR$(191);ELSE LPRINT CHR$(128
);
350 NEXT L:LPRINT :GOSUB 580:NEXT I:GOTO70
360 IF A>125 AND B>46CLS:GOTO70
370 FOR I=Y1 TO Y1+B :FOR L=X1 TO X1+A:RESET(L,I)
380 NEXTL:GOSUB 580 :NEXT I :GOTO70
390 Q=-1:P=-1:GOTO290
400 FOR I=1 TO 384:Z(I)=0:NEXT I:GOSUB 570:Y=Y1 :X=X1:
F=1
410 K=1:M=Y+B:IF M>47 M=47
420 FOR L=Y TO M :FOR N=X TO X+A STEP 16 :Z1=32768:FOR
J=0 TO 15
430 M1=N+J:IF M1>127 THEN 450 ELSE ON F GOTO 440,510
440 Z(K)=Z(K)*2:IF POINT(N+J,L) Z(K)=Z(K)+1
450 NEXT J:K=K+1:GOSUB 580 :NEXT N,L:ON F GOTO 460 ,70
460 M$="0":GOSUB 590:PRINT#1,A;B;K-1
470 FOR I=1 TO K-1 :PRINT#1,Z(I):NEXT I:CLOSE 1:GOTO7
0
480 M$="I":GOSUB 590:INPUT#1,A,B,C:F=2
490 IF X1<>X OR Y1<>Y THEN X0=X:X1=X:Y0=Y:Y1=Y
500 FOR I=1 TO C:INPUT#1,Z(I):NEXT I:CLOSE 1:GOTO410
510 IF Z(K)>=Z1 SET(M1,L):Z(K)=Z(K)-Z1
520 Z1=INT(Z1/2):GOTO450
530 N$="":PRINT@1012,T$:FOR I=1 TO 6
540 A$=INKEY$:IF A$="" THEN 540 ELSE IF A$=CHR$(13) THEN
560
550 N$=N$+A$:PRINT@1017+I,A$:NEXTI
560 PRINT@1012,STRING$(11," ");
570 RESET(X1+A+1,Y1):RESET(X1,Y1+B):RESET(X1+A+1,Y1+B)
:RETURN
580 A$=INKEY$:IF A$="S" THEN 70 ELSE RETURN
590 T$="NAME:":GOSUB 530:ON ERROR GOTO590:OPEN M$,1,N$
:RETURN

```

五十五、探索迷宫

王泽函

在古希腊神话故事中，勇士泰西必须穿过一个迷宫才能杀死妖怪，为了在迷宫中不致迷失方向，聪明的泰西进迷宫时一手持刀，另一只手中抓了一个线团，一边走一边放线，终于顺利地通过了迷宫，杀死了妖怪。

泰西的线团起了什么作用呢？

第一，凡是走过的路，他都铺上了一条线。

第二，每一个十字路口，他总是朝地上没有铺线的路走。

第三，当碰到一个死胡同要返回时，他就在返回的路上铺上第二条线，这样凡是遇到铺两条线的地方，前面一定是死胡同，不必在试探了。

第四，以上三点保证了每条路最多走两次，因迷宫是有限的，泰西在有限步内总能走出迷宫，不会在里面迷失方向走不出来。如果迷宫根本没有出口，那么也能在有限步内返回入口。

1. 程序设计思想与语句说明

泰西的线团启发了一个探索迷宫的有效算法，用数学语言表达如下：考虑右图所示的迷宫，其中阴影部分表示墙壁，白色的部分表示通道；又用 (I, J) 表示第 i 行第 J 列的小方块。

$$\text{令 } A(I, J) = \begin{cases} 0 & \text{若}(I, J)\text{是墙壁} \\ 1 & \text{若}(I, J)\text{是通路} \end{cases}$$

这里 $A(I, J)$ 可起到泰西的线团作用。

每到一小格 (I, J) ，察看它周围的四个小方格 $(I, J+1)$ ， $(I, J-1)$ ， $(I+1, J)$ ， $(I-1, J)$ ，若其中有三面是墙，则 (I, J) 是死胡同。令 $A(I, J) = 0$ ，以后就不再试探了。否则，记 $A(I, J) = 2$ ，表示 (I, J) 已经走过一次。

若 (I, J) 的周围有未走过的路，则按东、南、西、北的顺序向未走过的方格跨进一步，否则，就是从死胡同返回，

我们令 $A(I, J) = 0$ ，这就能防止以后再进入这个死胡同。

语句说明

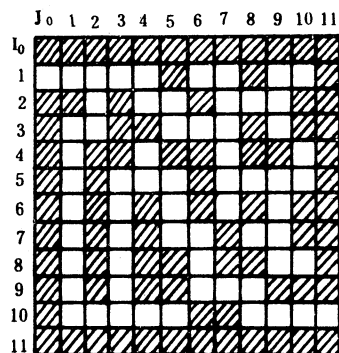
15~35语句，读数据

40语句，给出入口参数 I, J

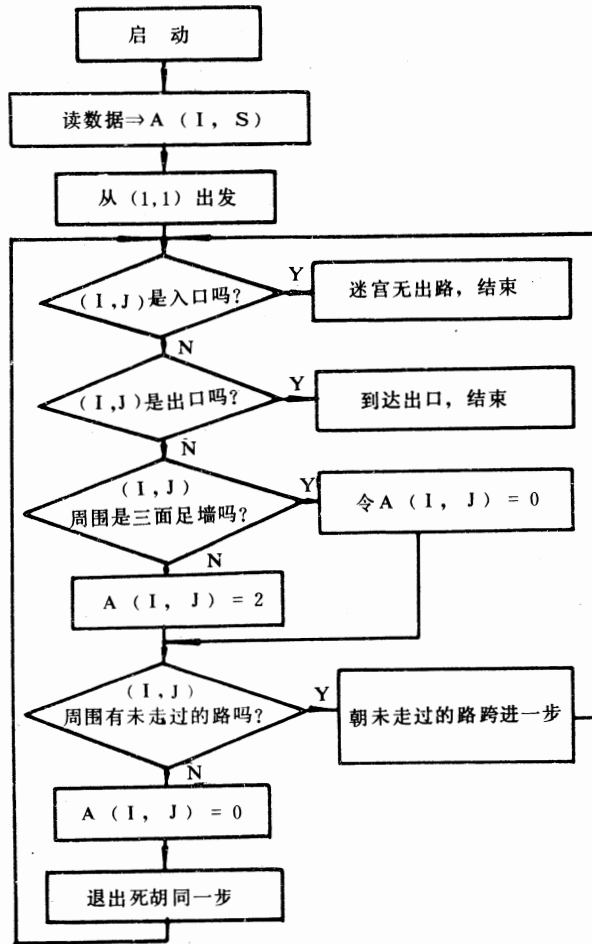
45语句，判 I, J 是否是入口

50语句，判 I, J 是不是出口

60~135语句，探索迷宫出口。



程序框图



源程序

```

5 N=11
10 DIM A(N,N)
15 FOR I=0 TO N
20 FOR J=0 TO N
25 READ A(I,J)
30 PRINT TAB(4*J);A(I,J);
35 NEXT J:PRINT:NEXT I
40 I=1:J=1:CO=0
45 IF I=1 AND J=0 THEN PRINT "NOT SOLVE":END
50 IF I=N-1 AND J=N THEN PRINT "CO";CO:END
55 PRINT I,J
60 CO=CO+1
65 P=0
    
```

```

70 IF A(I+1,J)=0 THEN P=P+1
75 IF A(I-1,J)=0 THEN P=P+1
80 IF A(I,J+1)=0 THEN P=P+1
85 IF A(I,J-1)=0 THEN P=P+1
90 IF P=3 THEN A(I,J)=0 ELSE A(I,J)=2
95 IF A(I,J+1)=1 THEN J=J+1:GOTO 45
100 IF A(I+1,J)=1 THEN I=I+1:GOTO 45
105 IF A(I-1,J)=1 THEN I=I-1:GOTO 45
110 IF A(I,J-1)=1 THEN J=J-1:GOTO 45
115 A(I,J)=0
120 IF A(I,J+1)=2 THEN J=J+1:GOTO 45
125 IF A(I+1,J)=2 THEN I=I+1:GOTO 45
130 IF A(I-1,J)=2 THEN I=I-1:GOTO 45
135 J=J-1:GOTO 45
140 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
145 DATA 1,1,1,1,1,0,1,1,0,1,1,0
150 DATA 0,0,1,0,1,1,0,1,1,1,0,0
155 DATA 0,1,1,0,0,1,1,1,0,1,0,0
160 DATA 0,1,0,0,1,0,0,1,0,0,1,0
165 DATA 0,1,0,1,1,1,0,1,1,1,1,0
170 DATA 0,1,0,1,0,1,0,1,0,1,0,0
175 DATA 0,1,0,1;0,1,1,0,1,1,0,0
180 DATA 0,1,0,1,0,0,1,0,0,1,1,0
185 DATA 0,1,0,1,0,0,1,1,1,0,0,0
190 DATA 0,1,1,1,1,1,0,0,1,1,1,1
195 DATA 0,0,0,0,0,0,0,0,0,0,0,0

```

五十六、哥尼斯堡桥问题

范幸义

古东普鲁士的哥尼斯堡城有七座桥，有个学士提出一个问题，能否把这些桥全部走一遍而每座桥都不重复走？

这是一个几何问题，即是否能在平面上一笔划好一个多边形。因此有人也叫一笔划问题。数学家Euler在理论上解决了这个问题。在实际中用人来解决，则需大量的时间和精力。若用电子计算机来解决，则是轻而易举的事。一笔划问题具有很强的实用性。对于一个公路网、铁路网、航空、航海，都具有现实和实用意义。诸如一个邮递员要把信送往每一条街，而又保证不重复走任何一条街，能做到的话，不但节约时间，而且少耗精力。

哥尼斯堡桥问题是一个几何问题，不是数值计算。用电子计算机可以把这个几何问题转化为数值问题来解决，並很快可以打印出结果。

程序设计思想与语句说明

1.设计思想 设有一个哥尼斯堡桥问题(见图一)，不妨把这些有关的边的交点进行编号分别为1、2、3、4、5、6、7、8。我们把这些交点叫做节点。奇数条边的交点叫奇节点。偶数条边的交点叫偶节点。根据Euler方法：哥尼斯堡桥问题的奇节点超过两个则无解。少于两个，则选一个奇节点作出发点。没有奇节点，则任选一个偶节点作出发点。然

后根据节点，作一个N阶方阵A。此方阵的第一行表示第一点；第二行表示第二点；……；第N行表示第N点。把和第i点相连的所有点在方阵A的对应元素 a_{ij} 置“1”，其余置“0”，则此方阵叫做特征方阵。例如，图1的特征方阵为：

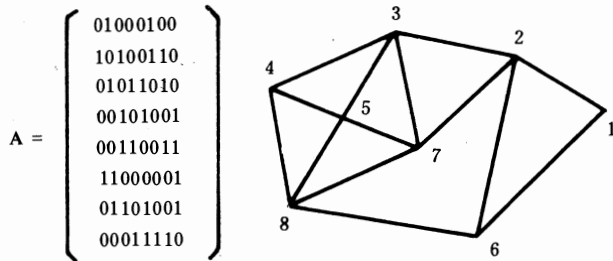
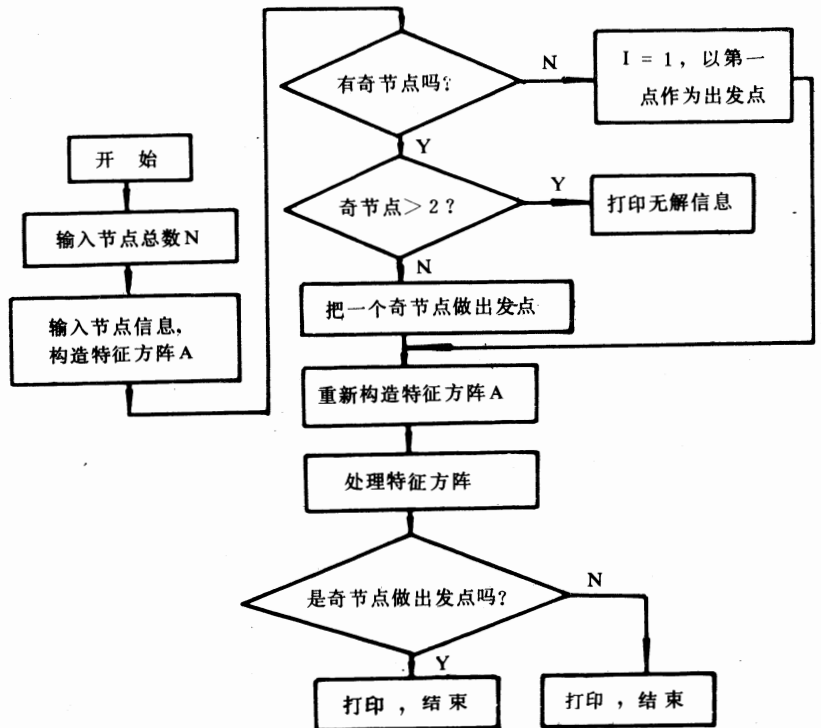


图1

此特征方阵A一定是对称方阵。从A的第一行起，逐个检查每个元素，遇到第一个非零元素，则记住列数 $B1$ ，然后把第一行第 $B1$ 列元素 $a_{1, B1}$ 、 $B1$ 与第 $B1$ 行第一列元素 $a_{B1, 1}$ 置0。表示第一点至 $B1$ 点的边已经走完。再从 $B1$ 行开始如法处理。若一次没有把A的元素都置0，则重复开始，直至A的元素全为0时止。这种方法称为特征矩阵法。

2. 语句说明 从通用性考虑，本程序设计了一些辅助程序。由键盘输入节点总数N。然后输入每个节点的节点信息（指和该节点相连的所有其它节点的编号），则程序自动按照节点信息建立特征方阵A。再由节点编号1~N，检查此节点是否是奇节点，若是奇节点，记录下编号。若奇节点多于两个，则打印出无解信息，程序停止运行。奇节点等于两个，则把第一个奇节点作为出发点。从而交换该行和第一行的所有元素，并改变与之有关的其它行的

程序框图



对应编号（指该编号和“1”号），使之合乎实际情况。再由主程序处理整个特征方阵。最后，打印由出发点起的每走一条边的节点编号。本程序的各语句功能如下：

10~70行，键盘输入节点信息，并构成特征方阵A。

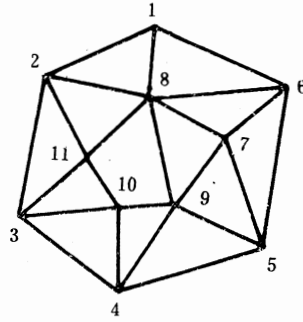


图2

80~180行，检查节点，如奇节点多于两个，则打印无解信息；若没有奇节点，则把第一点作为出发点。

190~330行，把第一个奇节点作为出发点，并改变节点信息，重新构成特征方阵A。

340~590行，主程序，处理特征方阵。

600~650行，打印结果，用于奇节点作出发点的情况。

660~680行，打印结果，用于偶节点作出发点的情况。

源程序 1

```

10 INPUT"THE LRUNODE MANER N=";N
20 DIM A(N,N),B(N*(N+1)/2),C(N*(N+1)/2),K(N)
30 FOR I=1 TO N
40 PRINT"I=";I:INPUT J:IF J=0 THEN 70
50 A(I,J)=1
60 GOTO 40
70 NEXT
80 FOR I=1 TO N
90 FOR J=1 TO N
100 IF A(I,J)<>0 THEN S=S+1
110 NEXT J
120 IF S=INT(S/2)*2 THEN S=0:GOTO 150
130 K(I)=I:X=X+1:S=0
140 IF X>2 THEN 180
150 NEXT I
160 IF X=0 THEN 340
170 GOTO 190
180 PRINT"ONT THE SOLUTION":END
190 FOR I=1 TO N
200 IF K(I)<> 0 THEN 220
210 NEXT
220 T=K(I)
230 FOR J=1 TO N
240 K(J)=A(T,J):A(T,J)=A(1,J):A(1,J)=K(J)
250 NEXT
260 FOR I=1 TO N
270 FOR J=1 TO N
280 IFJ<>T THEN 320
290 IFA(I,1)<>0AND A(I,J)<>0 THEN 330
300 IF A(I,1)<>0THEN A(I,1)=0:A(I,J)=1:GOTO 330

```

```

310 IF A(I,J)<>0 THEN A(I,J)=0:A(I,1)=1:GOTO330
320 NEXT J
330 NEXT I
340 I=1
350 FOR J=1 TO N
360 IF A(I,J)<>0 THEN K=J :GOTO 390
370 NEXT
380 L=L+1:B(L)=I: GOTO 410
390 A(I,K)=0:A(K,I)=0
400 L=L+1:B(L)=I:I=K:GOTO 350
410 FOR M=1 TO L
420 I=B(M)
430 FOR J=1 TO N
440 IF A(I,J)=0 THEN 470
450 IF V=0 THEN T=M:K=J: GOTO 500
460 K=J:GOTO 500
470 NEXT
480 IF U=0 THEN 510
490 R=R+1:C(R)=I: GOTO 530
500 A(I,K)=0:A(K,I)=0:R=R+1:C(R)=I:I=K:U=1:V=1:GOTO 43
0
510 NEXT M
520 IF X<>0 THEN 600 ELSE 660
530 FOR J=L TO T+1 STEP -1
540 B(R+T)=B(J)
550 NEXT
560 FOR J=T+1 TO T+R
570 B(J)=C(J-T)
580 NEXT
590 L=R+L:R=0:U=0:V=0: GOTO 410
600 FOR I=1 TO N*(N+1)/2
610 IF B(I)=0 THEN 650
620 IFB(I)=1 THEN B(I)=T: LPRINT B(I); CHR$(94); : GOT
D 650
630 IF B(I)=T THEN B(I)=1:LPRINT B(I); CHR$(94); :GOTO
650
640 LPRINT B(I);CHR$(94);
650 NEXT :END
660 FOR I=1 TO N*(N+1)/2
670 IF B(I)<>0 THEN LPRINT B(I); CHR$(94);
680 NEXT

```

程序 1 采用了特征阵法，其方阵中元素数为节点数的平方。当节点增加到 100, 1000 时，元素将增加到 10000, 1000000，这不仅占用很多内存，也大大增加了输入数据量，这是一种浪费。

下面介绍的程序中，输入数据是各边的端点号，一对对写在 DATA 之后。从一笔划的起点写起，有多少边，写多少对，末尾加一个 0 表示结束。如每一节点平均联 4 条边，则上述情况中输入数据约为 400 或 4000 个，它们通过 READ 语句赋值给 A(I, J)，这里 I 是边的序号，J = 0 或 1，这样一来，一笔划问题变成数据重排问题，最终要排成 $A(I, J) = A(I + 1, 0)$, $I = 0, 1, 2, \dots, N - 2$ 。

边的序号从 0 到 $N-1$ ，总数为 N 。从 M 到 $N-1$ 是尚未重排的边。接续点在 $A(L, 1)$ 。按 Euler 的解法，先从起点出发作一笔划，直到不能再作为止。在这过程中， $M = L + 1$ 。若仍有剩下的边，从中再找上述连接线中某一点为起点与终点的回路，取 $L = 0, 1, 2, \dots, N-2$ 。设置 M_0 是为了在一笔不能划出全图时避免死循环，这时从打印的结果可看出症结所在。

程序 2 中的数据是按图 1 的各边给出的。

源程序 2

```

10 DIM A(1000,1)
20 READ A(N,0)
30 IF A(N,0)=0 THEN 50
40 READ A(N,1):N=N+1:GOTO20
50 M=1
60 L=0:M0=M
70 FOR I=M TO N-1
80 FOR J=0 TO 1
90 IF A(L,1)<>A(I,J) THEN 170
100 A0 =A(I,0):A1=A(I,1)
110 A(I,0)=A(M,0):A(I,1)=A(M,1)
120 FOR K=M TO L+2STEP -1
130 A(K,0)=A(K-1,0):A(K,1)=A(K-1,1)
140 NEXT K
150 L=L+1:M=M+1
160 A(L,J)=A0:A(L,1-J)=A1:GOTO70
170 NEXT J,I
180 IF M=N THEN 210
190 L=L+1:IF L<M-1 THEN 70
200 IF M>M0 THEN 60
210 PRINTA(0,0);"-":A(0,1);
220 FOR I=1 TO N-1
230 IF A(I,0)<>A(I-1,1) THEN PRINT",":A(I,0);
240 PRINT"-":A(I,1);
250 NEXT I
260 DATA 1,2,1,8,1,6,6,8,6,7,2,8,8,7,6,5,2,3,2,11,8,11
,8,9,7,5,11,10,10,9,9,5,3,11,3,10,10,4,9,4,3,4,4,5,0

```

五十七、填牌游戏

许传国

游戏主持人拿出 N 张从 1 到 N 编号的牌。他把这些牌任意排成首尾相接的一圈，然后任意留下其中 M 张牌，把其余的牌收起。指定一个开始位置，要游戏的人把收起的牌按规定填进去。这个规定就是，在剩下的牌中，从指定的起始位置开始数牌。当数的数和被数牌的号码一致时，就把该牌拿掉。然后又从下一张牌开始数，当数的数和牌的号码又一致时，又把该牌拿掉。如此反复，要求能把所有的牌都拿掉。

用 BASIC 程序做这个游戏，要分两步走。第一步，要把填牌前的“牌局”摆出来。10 语句到 120 语句（包括 500 到 600 语句的子程序）完成这项工作。游戏的人只要输入牌的张数 N ，

程序就产生随机“牌局”。先由子程序产生一个随机数组 $C(N)$ 作为牌局中牌的摆放位置。如： $C(N) = (3, 1, 5)$ ，就表示在规定位置起数的第一、第三、第五个牌位上有牌。接着，再产生一个随机数组 $A(N)$ 作为牌局中牌的号码。把这两个随机数组一综合，就得到一个随机牌局数组 $Z(N)$ ($Z(C(I)) = A(I)$)。

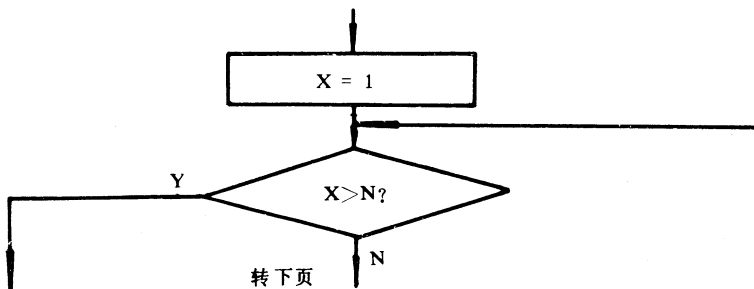
第二步是填牌。填牌的程序从130语句到400语句。用数组 $B(N)$ 来存放填入的每一张牌； j 代表牌的摆放位置号码，即数组 $B(N)$ 的单元号码； K 代表数牌时数的数； D 累计已填入的牌的张数。

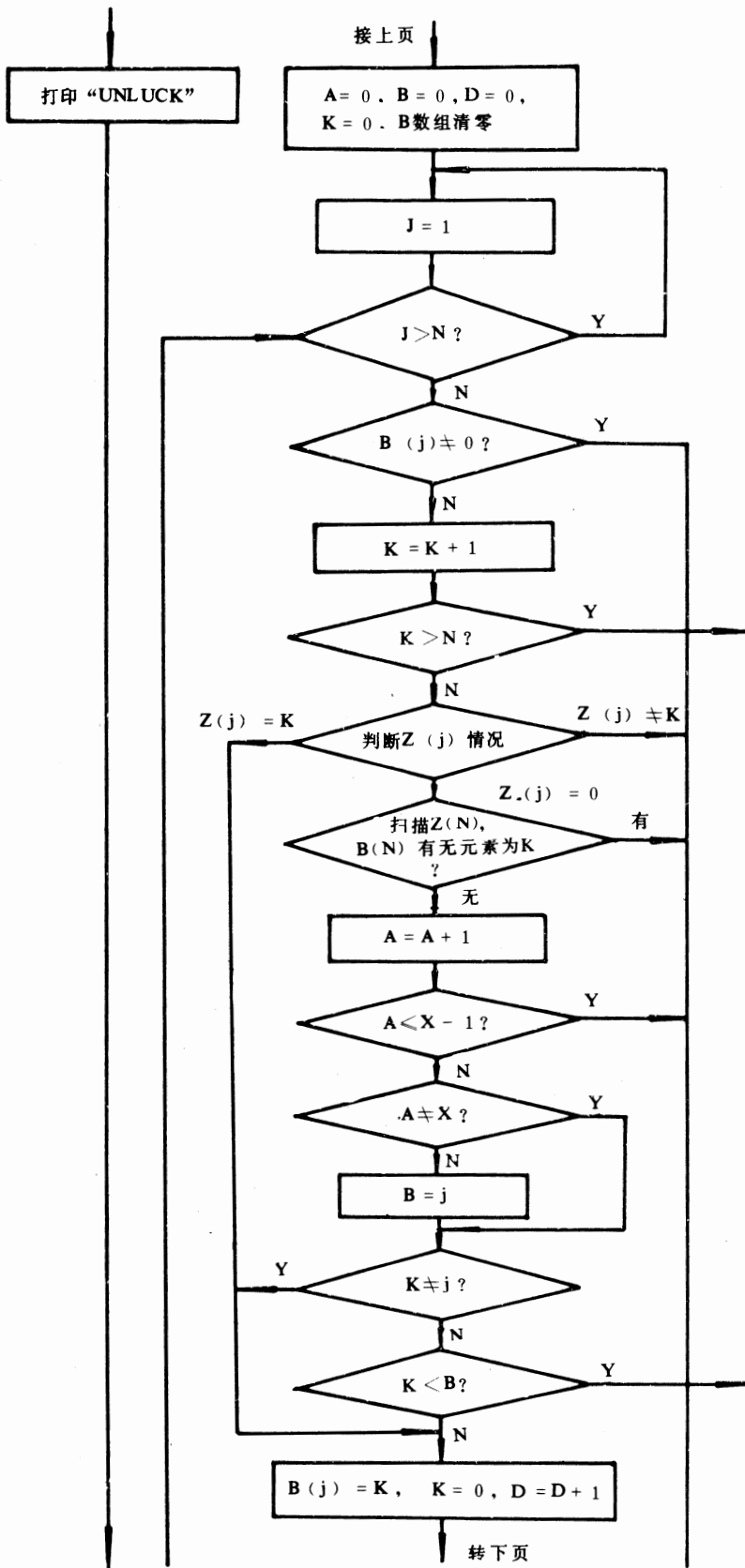
填牌的关键注意三点：一、 K 的数数规则：因为先填入的牌在数牌时先数“去”，故在 $B(j) = 0$ 处才数， $B(j) \neq 0$ 处要绕过不数；应首尾相接循环数牌。二、牌的填入规则：要在 $B(N)$ 中填入一张牌，比如 $B(j)$ 处，有三个条件（设数到这个位置的数为 K ）：1、 $B(j) = 0$ ，即这个位置原先没有填入过牌；2、已经填在 $B(N)$ 的牌中，没有 K 这个号码的牌；3、固定牌局 $Z(N)$ 中对应此位置没有放牌，而且 $Z(N)$ 中其它处的牌也不是 K 这个号码；或者 $Z(N)$ 中对应此位置虽然有牌，但牌的号码就是 K ，当这三个条件都满足时，才有可能把 K 号牌填入此处（令 $B(j) = K$ ），同时令 $K = 0$ ，使下次数牌从1开始数起。三、填入的第一张牌，一定有 $K = j$ ；以后填入的牌，若有 $K' = j'$ 、且 $K' < K$ 的话，则数牌时，先数“掉”的是“ K' ”，而不是“ K ”；这样“数牌”就不是严格跟踪“填牌”顺序，就不可能把全部牌数掉，要把这种情况区别出来。241~244语句就是作这种区别的。

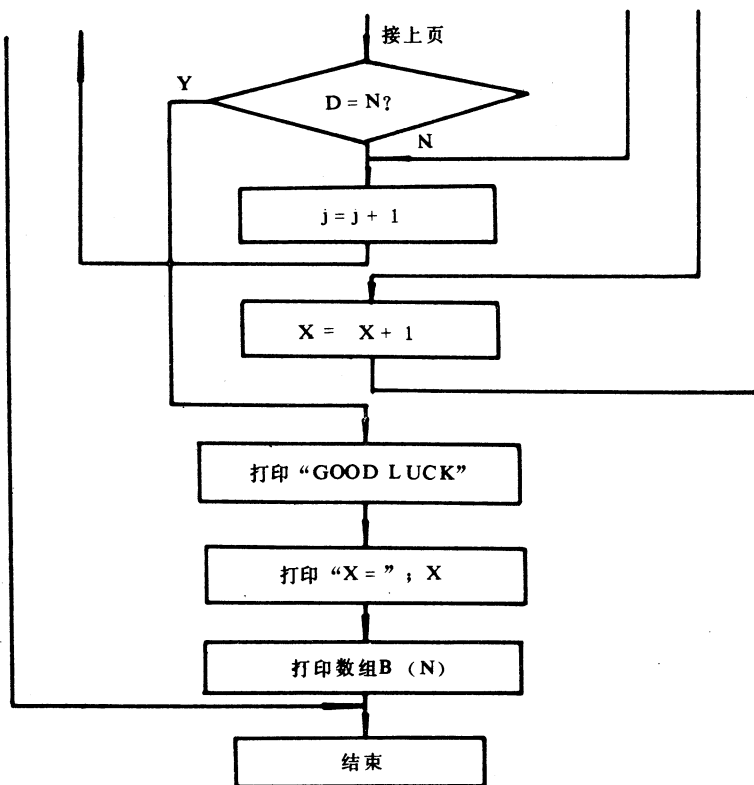
填牌可有多种模式。程序中对 N 张牌设计了 N 种填入模式，用 X 代表这些不同的模式。各种模式的不同区别，只在于第一张填入的牌所填入的可能位置不同而已，而其后（从填第二张牌起）的填法规则则完全一样。用 A 起这种标记作用。每遇一个可填牌的机会（符合上述三原则）， A 就自动加1（初值为0）。当 $A < X - 1$ 时，要填的第一张牌绕过这些能填牌的位置而不填；当 $A = X$ 时，填入第一张牌，同时用 $B = j$ 记住这第一张填入牌的位置；以后填入的牌，若有 $K = j$ 这种情况出现就必须判 $K < B$ 否？若是，则这样填出的牌必定失败，不必再填下去，可让 X 加1进入另一种模式的试填。

从 $X = 1$ 模式开始，如果此轮填牌不能按规定填完所有的牌（ $K > N$ ，而 $B(N)$ 仍未填满即 $D < N$ ），则此轮填牌（失败）作废（令 $MAT\ D = ZER, A = B = D = K = 0$ ），而 X 加1，开始下一轮填牌尝试。如果 N 次尝试都不成功，则打印“UNLUCK”而告结束。如果某一次尝试成功，则打印“GOOD LUCK”，而结束游戏。

程序框图







```

10 INPUT N
20 M=INT(RND(0)*N+1)
30 PRINT"N=";N,"M=";M
40 DIMA(N),B(N),C(N),Z(N)
50 GOSUB 500
60 FOR I=1 TO N
70 C(I)=A(I)
80 NEXT I
90 GOSUB 500
100 FOR I=1 TO N
110 Z(C(I))=A(I)
120 NEXT I
121 PRINT"Z(";N;")="
122 FOR I=1 TO N
123 PRINTZ(I);" ";
124 NEXT I
125 PRINT
130 FOR X=1 TO N
140 A=0:B=0:D=0:K=0
141 FOR Y=1 TO N
142 B(Y)=0
143 NEXT Y
150 FOR J=1 TO N
160 IF B(J)<>0 THEN 310
170 K=K+1
180 IF K>N THEN 325
190 IF Z(J)<>0 THEN 300
200 FOR I=1 TO N
210 IF B(I)=K THEN 310
220 IF Z(I)=K THEN 310
230 NEXT I
235 A=A+1
240 IF A<=X-1 THEN 310
  
```

源程序

```
241 IF A<>X THEN 243
242 B=J
243 IF K<>J THEN 250
244 IF K<B THEN 325
250 B(J)=K:K=0:D=D+1
280 IF D=N THEN 350
290 GOTO 310
300 IF Z(J)=K THEN 250
310 NEXT J
320 GOTO 150
325 NEXT X
330 PRINT"UNLUCK"
340 GOTO400
350 PRINT"GOOD LUCK"
352 PRINT"X=";X
355 PRINT"B(";N;")="
360 FOR I=1 TO N
370 PRINT B(I);" ";
380 NEXT I
400 END
500 FOR I=1 TO N
505 IF I>M THEN 530
510 A(I)=INT(RND(0)*N+1)
520 GOTO 540
530 A(I)=0
540 NEXT I
550 FOR I=1 TO M-1
560 FOR J=I+1 TO M
570 IF A(I)=A(J) THEN 500
580 NEXT J
590 NEXT I
600 RETURN
```

五十八、移子游戏

陈炜

我国古时有一移子游戏，称为“移棋相间”。即有N个白子与N个黑子(N>3)如图1。每次移动相邻两子，移动N次，使黑、白子相间排列，如图2。移动的规则是：

1. 每次只能移动相邻两子，并不得交换位置；
2. 移动中，棋间不能同时空出四子位置；
3. 移子时，不能将两子分开插入棋间。

N较小时，移动较容易。当N较大时，就比较困难。运行下面的程序，使输入任何N时，计算机都可将每步摆法显示或打印出来。

程序说明

N=3, N=4, N=5, N=6为基本模式，当N>6时，可以移动数次，转化为这四种模式。程序中语句5~55是存入初始数据；70~80是处理N=3, 4, 5, 6的典型模式；95~105是处理余下的棋子；115~130是处理N=3时的特殊摆法，135~150是把N>6的模式转化成N=3, 4, 5, 6的典型模式。程序中有很多地方要用到打印格式，所以，把



图 1

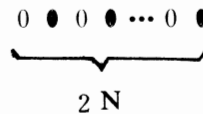
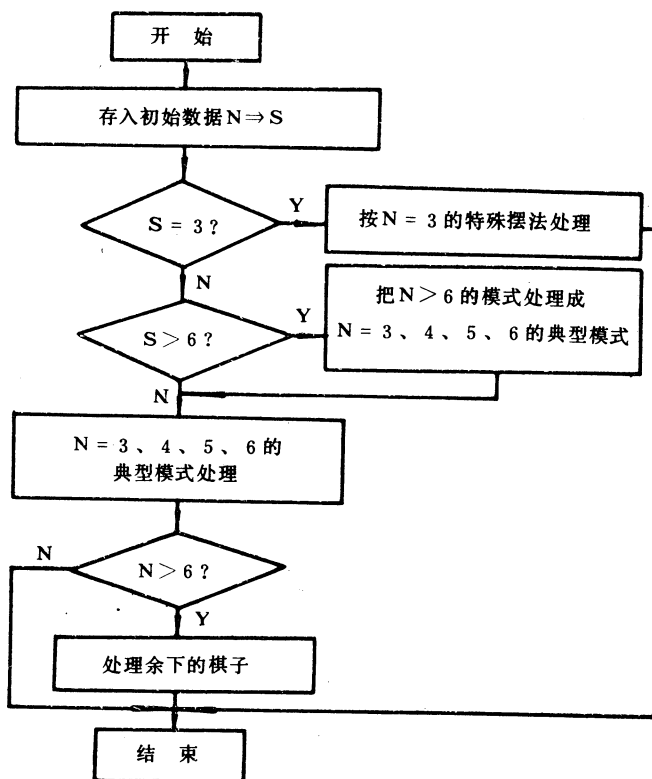


图 2

它们都放在155~170的子程序中。

程序框图



源程序及运行结果

```

5 DIM B(6,12)
10 FOR I=1 TO 6:READ B(2,I):NEXT
15 FOR I=3 TO 6:FOR J=1 TO I*2:READ B(I,J):NEXT J,I
20 DATA 4,-2,6,3,3,0
25 DATA 4,-2,-2,3,3,0
30 DATA 5,-2,-2,1,1,4,4,-3
35 DATA 6,-3,-3,3,3,0,0,5,5,-4
40 DATA 7,-4,-4,2,2,-2,-2,3,3,6,6,-5
45 INPUT N
50 DIM A(N*2+4)
  
```



```

55 GOSUB 165 :S=N:Z=N+4:IF S<3 THEN 45
60 IF N=3 THEN 115
65 IF S>6 THEN 135
70 FOR I=1 TO S*2 STEP 2
75 A(N+B(S,I))=A(N+B(S,I+1)):A(N+B(S,I)+1)=A(N+B(S,I+1)
)+1):A(N+B(S,I+1))=0:A(N+B(S,I+1)+1)=0:GOSUB 155
80 NEXT I
85 IF N<7 THEN 110
90 X=N+B(S,I-1)
95 Y=N+Z:A(X)=A(Y):A(Y)=0:A(X+1)=A(Y+1):A(Y+1)=0:GOSUB
155
100 X=Y:Y=N-Z+1:A(X)=A(Y):A(Y)=0:A(X+1)=A(Y+1):A(Y+1)=
0:GOSUB 155:X=Y:Z=Z+4
105 IF Y>1 THEN 95
110 END
115 FOR I=1 TO N*2 STEP 2
120 A(N+B(2,I))=A(N+B(2,I+1)):A(N+B(2,I)+1)=A(N+B(2,I+
1)+1):A(N+B(2,I+1))=0:A(N+B(2,I+1)+1)=0:GOSUB 155
125 NEXT I
130 END
135 Z=Z-4
140 A(N+Z+1)=A(N-Z+2):A(N+Z+2)=A(N-Z+3):A(N-Z+2)=0:A(N
-Z+3)=0:GOSUB 155
145 S=S-4:A(N-Z+2)=A(N+Z-3):A(N-Z+3)=A(N+Z-2):A(N+Z-3)
=0:A(N+Z-2)=0:GOSUB 155
150 GOTO 65
155 FOR II=1 TO N*2+4:PRINT USING"##";A(II);:NEXT:PRIN
T
160 RETURN
165 FOR I=1 TO N:A(I)=1:A(I+N)=2:NEXT:GOSUB 155
170 RETURN

```

>RUN

? 3

```

1 1 1 2 2 2 0 0 0 0
0 0 1 2 2 2 1 1 0 0
0 0 1 2 2 0 0 1 2 1
0 0 0 0 2 1 2 1 2 1

```

READY

>_

RUN

? 4

```

1 1 1 1 2 2 2 2 0 0 0 0
1 0 0 1 2 2 2 2 1 1 0 0
1 2 2 1 0 0 2 2 1 1 0 0
1 2 2 1 2 1 2 0 0 1 0 0
0 0 2 1 2 1 2 1 2 1 0 0

```

READY

>_

五十九、解奇数阶魔方

赵恒友

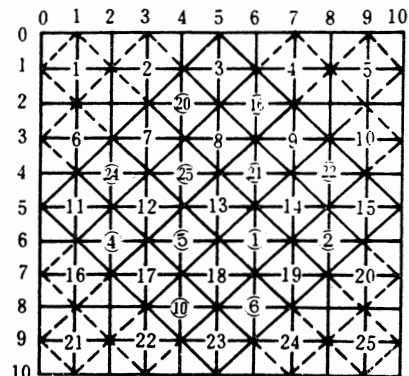
魔方 (Magic Square) 也称幻方或纵横图。它的一般定义是将 $1 \sim n^2$ 的自然数排列成纵横各几个数的正方形, 使在同一行、同一列或同一对角线上的几个数的和都相等。最早的魔方起源于我国。我国汉代就有了三阶魔方 (见图 1), 称为九宫。它的每一行, 每一列和两条对角线的三个数之和均为 15。

魔方这一“自然之迷”自古以来就引起人们的极大兴趣, 陆续发现了各式各样的魔方, 有些魔方还有待于人们去揭示它们的规律。

魔方, 按其元素个数, 分为奇数阶魔方和偶数阶魔方。这里介绍一个生成奇数阶魔方的程序。

解奇数阶魔方的程序的基本思想是:

- 设置一个 $2N \times 2N$ 的方阵;
- 在奇数格子上赋 $C = C + 1$ 的值, 一直到 $C = N^2$;
- 再把 C 值填到下图的粗框中去, 即 $A(K, L)$ 位置上;
- 算法的关键在于下标 K, L 的计算。已经证明了, 恰好所希望的值落在粗框内, 而不希望的值落在粗框之外, 这是根据对称性得到的;



·按粗框内的斜线方向打印出来,就是所要求的魔方图。

语句说明

140 语句:定维。

150~190语句:输入数据检查。

230~420语句:计算下标,给数组A赋值。

440~600语句:输出结果。

源程序

```
140 DIM A(80,80)
150 INPUT N
160 IF N/2<>INT(N/2) THEN 190
170 PRINT"ENTER ODD NUMBER"
190 IF N/2=INT(N/2) THEN 150
230 C=1
240 FOR I=1 TO 2*N-1 STEP 2
250 FOR J=1 TO 2*N-1 STEP 2
260 IF I>N THEN 300
270 K=I+N
280 GOTO320
300 K=I-N
320 IF J>N THEN 360
330 L=J+N
340 GOTO380
360 L=J-N
380 A(I,J)=C
390 A(K,L)=C
400 C=C+1
410 NEXT J
420 NEXT I
440 'OUTPUT RESULT
460 I=1
480 IF I>N THEN 620
490 K=I
510 P=0
520 FOR L=N-I+1 TO 2*N-I
530 PRINT TAB(10+P);A(K,L);
540 K=K+1
550 P=P+6
560 NEXT L
570 PRINT
580 PRINT
590 I=I+1
600 GOTO480
620 END
```

六十、解奇偶阶魔方

潘林森

魔方是组合数学研究的一个重要内容,有许多种计算编排方法,排列也不是唯一的。例如 4×4 魔方可排成:

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

或

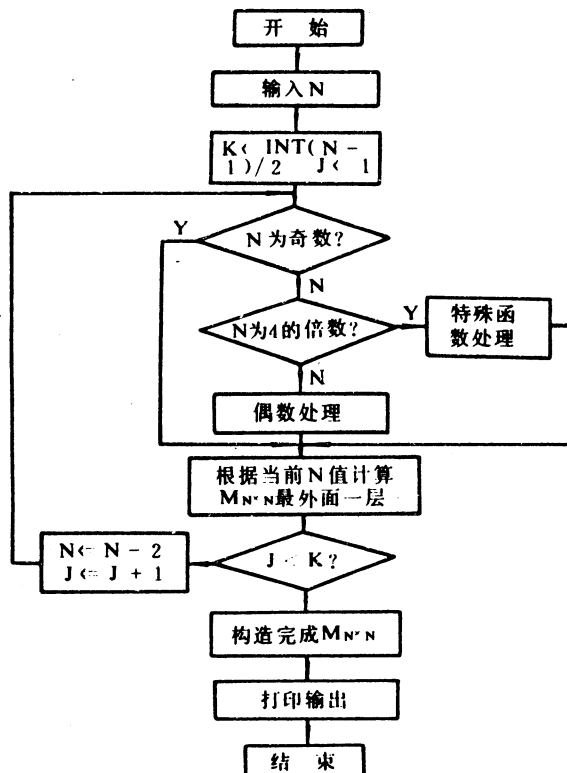
12	6	3	13
9	7	2	16
5	11	14	4
8	10	15	1

有不少计算、打印魔方的程序，但大多数程序只能计算打印奇数阶魔方，或者一些特殊偶数阶魔方。下面介绍的程序可以计算打印出大于等于3的任意阶魔方。

在程序设计中，对魔方元素的计算采用了由外向里收缩算法（见程序框图）。首先，根据当前的N值，计算出 $M_{N \times N}$ 魔方最外面一层各元素的值，然后再向里推进一层计算。每推进一层，魔方的阶数减少2。经过 $\lfloor (N-1)/2 \rfloor$ 次推进后，便可构造出一个 $M_{N \times N}$ 魔方。

程序中，第300~360，370~440语句是两个子程序，用来对偶数进行特殊计算处理。子程序中第320语句用来进一步判断N是否为4的倍数，若是4的倍数，继续作330~360语句的相应计算，否则返回。

程序框图



源程序

```
5 REM THE PROGRAM OF MAGIC SQUARE COMPUTATION
10 INPUT"INPUT INTEGER NUMBER N(N>=3)";N
20 DIM M(N,N)
30 X=N*N+1:Y=INT(N/2)+1
40 E=2*Y-N:D=(E-2)*N+X+1
50 A=E*(1-Y):B=1-A:C=N
60 FOR J=1 TO INT((N-1)/2)
70 A=A+B+E-1:M(1+J,J)=A
80 E1=E:B=E-B
90 FOR I=J TO C-2
100 M(J,I)=X-A+B:B=-B
110 M(C,I)=A+B:A=D-A-E1
120 M(I+2,J)=A:E1=E-E1
130 NEXT I
140 A=X-A:M(J,C)=A
150 IF E=2 GOSUB 300
160 M(C,C)=D+A-X:D=D+2-E1
170 M(J,C-1)=A+Y-C:M(C,C-1)=X-M(J,C-1)
180 FOR I=1+J TO C-1:M(I,C)=X-M(I,J):NEXT I
190 C=C-1
200 NEXT J
210 M(J,J)=A+B
220 IF E=2 GOSUB 370
230 PRINT"MAGIC SQUARE N=";N
240 FOR I=1 TO N
250 FOR J=1 TO N
260 PRINT USING"#####";M(I,J);
270 NEXT J:PRINT
280 NEXT I
290 END
300 M(Y,J)=A-B
310 A=X-A:B=-B
320 IF INT(B/4)*4+1<>B RETURN
330 M(1+J,J)=X-M(1+J,J)
340 M(Y,J)=X-M(Y,J)
350 M(Y+1,J)=X-M(Y+1,J)
360 RETURN
370 M(J,J+1)=A-2
380 M(J,J-1)=A+5
390 M(J,J+2)=A+12
400 FOR K=J-1 TO J+2
410 M(J+1,K)=X-M(J-1,K)
420 M(J+2,K)=X-M(J,K)
430 NEXT K
440 RETURN
```

六十一、魔方的判别

程胜利

一个方阵是否为魔方阵，一般要计算方阵的每一行、每一列及其两条对角线中的所有元素之和，如果它们都相等，我们称该方阵为魔方阵。当方阵阶数较大时，计算起来就比较麻烦。下面的程序可以判别任意阶方阵是否为魔方阵，并且可以打出检验结果。

程序的设计思想

设任意输入的方阵的阶数为N。我们可以先求出其 $2 \times (N + 1)$ 个和，即求出每行、每列和两条对角线的和，然后再来比较这些和的值是否相等，相等即为魔方。这种方法容易考虑得到，编程也比较直观，但却是不可取的。因为在某种情况下它占用的机时较多。

实际上，我们可以采用在求这些和值的过程中进行比较判别的方法，即一面求值，一面比较。一旦发现某两个和值不等，则立即输出检验结果：“THE MATRIX IS NOT A MAGIC SQUARE”(这个矩阵不是魔方)，结束程序的运行。如果输入的方阵恰好是一个魔方，这种边求值边比较的过程一直要进行到底，直到输出信息“THE MATRIX IS A MAGIC SQUARE”(该矩阵是个魔方)。

为了输出格式的需要，本程序使用了STR\$、LEN两种函数，可以将任意阶方阵以最大元素的位数为间隔打印出来，便于阅读。

语句说明

- 10 ~ 20语句：定义数组。
- 30 ~ 70语句：输入方阵。
- 80语句：打印方阵提示信息。
- 90 ~ 130语句：求方阵中数的最大位数。
- 140~230语句：打印方阵。
- 240~280语句：计算两对角线的和。
- 290~300语句：打印对角线的值。
- 310语句：判两对角线是否相等。
- 335语句：有一行的和不等于对角线的和，结束。
- 340~360语句：矩阵转置，列变成行。
- 390~430语句：判各行、列以及对角线的和是否相等。
- 440语句：打印“这是魔方”信息。
- 450语句：打印“这不是魔方”信息。
- 470~600语句：子程序。计算并打印某行（或列）的和。

源程序

```
10 CLEAR 200:INPUT N
20 DIM S(N,N),Q(N),W(2*(N+1)),AA1(N,N),AB$(N,N)
30 FOR R=1 TO N:FOR C=1 TO N
40 INPUT S(R,C)
50 AB$(R,C)=STR$(S(R,C))
60 AA1(R,C)=LEN(AB$(R,C))
70 NEXT C,R
80 PRINT"THE MATRIX IS:"
90 X=AA1(1,1)
```

```

100 FOR R=1 TO N:FOR C=1 TO N
110 IF X>AA1(R,C)THEN 130
120 X=AA1(R,C)
130 NEXT C,R
140 FOR R=1 TO N
150 FOR C=1 TO N
160 PRINTS(R,C);
170 X1=X-AA1(R,C)
180 FOR II=1 TO X1+2
190 PRINT" ";
200 NEXT II
210 NEXT C
220 PRINT
230 NEXT R:PRINT
240 A=0:B=0:F=N
250 FOR L=1 TO N
260 A=A+S(L,L)
270 B=B+S(L,(P+1-L))
280 NEXT L
290 PRINT"THE SUM OF DIAGONAL-1 IS ";A
300 PRINT"THE SUM OF DIAGONAL-2 IS ";B
310 IF A<>B THEN 450
320 W(1)=A:W(2)=B
330 GOSUB 470
335 IF B1=-1THEN 450
340 FOR I=1 TO N-1:FOR J=I+1 TO N
350 C=S(I,J):S(I,J)=S(J,I):S(J,I)=C
360 NEXT J,I
370 A1=-1
380 GOSUB470
385 IF B1=-1 THEN 450
390 M=W(1)
400 FOR K=2 TO 2*(N+1)
410 IF W(K)<>M THEN 450
420 M=W(K)
430 NEXT K
440 PRINT"THE MATRIX IS A MAGIC SQUARE":GOTO460
450 PRINT"THE MATRIC IS NOT MAGIC SQUARE"
460 END
470 FOR R=1 TO N
480 Q(R)=0
490 FOR C=1 TO N
500 Q(R)=Q(R)+S(R,C)
510 NEXT C
520 W(R+2)=Q(R)
530 IF A1=-1THEN 545
535 IF Q(R)<>B THEN PRINT"THE SUM OF ROW";R;" IS ";
Q(R):GOTO580
540 PRINT"THE SUM OF ROW ";R;" IS ";Q(R):GOTO570
545 IF Q(R)<>B THEN PRINT"THE SUM OF COL ";R;" IS
";Q(R):GOTO580
550 PRINT"THE SUM OF COL.";R;" IS ";Q(R)
560 W(R+2+N)=Q(R)
570 GOTO590
580 B1=-1:GOTO600
590 NEXT R
600 RETURN

```

六十二、任意函数的图象

刘炳文

函数图象是数学的重要内容之一。从图象中我们可以迅速而准确地看出函数的重要特性

及其变化情况，可以很方便地求得函数的最大值、最小值等。因此，在某种程度上来说，一个有价值的图象抵得上几百甚至上千的文字叙述。

这个程序可以显示由用户提供的任意函数的二维图象，要求函数的格式必须是 $Y=f(X)$ ，即 Y 是 X 的函数。自变量 X 将画在横坐标（水平轴）上，因变量 Y 画在纵坐标（垂直轴）上。在 X 轴和 Y 轴上的刻度可以根据实际需要描绘出来。

程序用法

运行程序之前，必须首先输入需要描绘图象的函数，该函数放在从语句5000开始的子程序中。如前所述，要把 Y 定义为 X 的函数，即 $Y=f(X)$ 。通过设置不同的 X 值来调用子程序，同时在程序运行中要相应设置 Y 的最小值和最大值（方法见后）。需要描绘图象的函数的子程序根据实际情况编写，可以和需要定义的函数一样简单或复杂，它可能只需要一行（例如5000 $Y=\text{SIN}(X)$ ），也可能需要几行，几十行乃至上百行。语句5999作为返回语句已经写在子程序中，因此用户无须再写RETURN语句。

输入上面的子程序后，就可以准备运行程序了。必须重申，该程序是在假定函数预先输入的前提下开始运行的。打入RUN后，程序将询问 X 的区间，即用户想要画的图象中自变量 X 的最小值和最大值，它们可以是任何实数，只要最大值大于最小值就可以。

接着，程序要求选择 Y 的刻度值。为了选择得更准确一些，用户最好能知道在 X 的区间里函数 Y 所取的最小值和最大值，这些值将由程序根据输入的 X 的最小值、最大值求出并显示出来供用户参考。然后由用户选择想要使用的 Y 的最小和最大刻度值。只要最大刻度值大于最小刻度值，任何两个值都是合法的。

在做完了上述这些准备工作之后，程序就开始显示函数的图象。与普通的函数图象不同的是，程序把坐标原点规定为 X 和 Y 的最小刻度值。 X 轴和 Y 轴的长度各为40个字符，每个轴上各有20个小点作为刻度。最小值、中间值和最大值将在各个适当的刻度位置标明。

当 Y 值超过规定的最大或最小刻度时，将根据对应的 X 值显示星号。如果实际的 Y 值太大，这些星号将在规定的 Y 的最大值处显示；类似地，如果 Y 的值太小，则星号在 Y 的最小值处显示。

有的图象可能会沿着 X 轴画出一串圆圈，把 X 轴断开。这些圈是与 Y 值对应的点，而这些 Y 值与沿 X 轴的值相等。

图象画完后，程序将提示用户按下任一个键继续运行。按键后，程序提供有关的刻度信息，即分别给出函数在 X 轴和 Y 轴上的最小值、中间值和最大值。

现在面临两种选择：按下“G”键，重新显示一次刚刚画过的图象；或者按下其它任何键以结束程序的运行。

可以对该程序进行如下的简单修改：

1. 增加下面两行语句，可以让程序自动地在 Y 轴上定出刻度，即使用程序所求出的 Y 值作为 Y 轴上范围的最小、最大值。

```
833 IF MX <= MN THEN 840
835 YU = MX:YL = MN:RETURN
```

2. 有的用户在程序运行前可能会忘记输入从语句5000开始的子程序，倘若如此，程序将画出直线 $Y=0$ 。如果想绝对避免出现这种情况，可以把语句5000改为

```
5000 Y = I/O
```

这样，如果在运行程序前没有输入实际需要的子程序，则在打入 X 的区间后，程序将停

止运行，並显示如下信息：

? / 0 ERROR IN 5000

程序语句及变量说明

120~130 预置常量。

200~210 显示程序说明。

220~260 主程序一得到由用户输入的X的区间，然后调用各子程序。

300~410 画坐标轴和刻度的子程序。

500~590 绘制函数曲线的子程序。

700~740 确定作图位置 (X, Y) 中Y值的子程序。

800~880 确定Y的最小值、最大值的子程序；得到由用户输入的Y的刻度值。

900~980 显示刻度参数(包括最小值、中间值和最大值)，询问是否要重画图象的子程序。

1000~1070 显示程序说明的子程序。

5000~5999 由用户提供的函数 $Y = f(x)$ 的子程序。

XL, XM, XU: X的最小值、中间值和最大值。

YL, YM, YU: Y的最小值、中间值和最大值。

DX, DY: X, Y的刻度增量。

X, Y: X, Y的当前值。

XP, YP: X, Y的作图位置。

A\$, B\$, C\$:

H\$, M\$, L\$: 在坐标轴和刻度中使用的字符串。

F\$ 坐标轴以外的字符串。

X\$ X轴的字符串。

Q 坐标轴字符串的长度。

F 特殊作图字符标志 (1 = F太高; 2 = F太低; 3 = F在X轴上)。

V X或Y刻度单位的值。

D V的整数值。

MN, MX Y的最小值，最大值。

Q\$ 用户打入的字符串。

3. 源程序

```
100 REM:GRAPH
120 CLEAR 50:A$=CHR$(191):B$=CHR$(143):C$=CHR$(179)
130 H$=CHR$(131):M$=CHR$(140):L$=CHR$(176):F$="*":X$="
0"
200 CLS:PRINT CHR$(23);TAB(11)"GRAPH":PRINT
210 GOSUB 1000:PRINT
220 PRINT:INPUT"LOWEST VALUE OF X":XL
230 INPUT"HIGHEST VALUE OF X":XU
240 IF XU<=XL THEN PRINT:PRINT " -- BAD X RANGE --"
:GOTO 220
250 GOSUB 800:GOSUB 300:GOSUB 500
260 GOSUB 900:CLS:END
300 CLS:FOR J=80 TO 912 STEP 64:PRINT@J,A$::NEXT
310 FOR J=913 TO 952:PRINT@J,H$::NEXT:PRINT@14,L$:L$:L
```

```

#:
320 FOR J=79 TO 719 STEP 128:PRINT@J,M$;:PRINT@J+64,C$
;
330 NEXT:PRINT@847,M$;:PRINT@270,H$;:PRINT@462,M$
340 PRINT@654,L$;:PRINT@910,H$;H$;:FOR J=914 TO 952 ST
EP 2
350 PRINT@J,B$;:NEXT:FOR J=922 TO 952 STEP 10:PRINT@J,
A$;:NEXT
360 PRINT@14,"Y";:PRINT@954,"X";
370 XM=(XL+XU)/2;YM=(YL+YU)/2;Q=LEN(STR$(YU))
380 PRINT@12-Q,YU;:Q=LEN(STR$(YM)):PRINT@460-Q,YM;
390 Q=LEN(STR$(YL)):PRINT@908-Q,YL;:Q=LEN(STR$(XL))/2
400 PRINT@976-Q,XL;:Q=LEN(STR$(XM))/2:PRINT@996-Q,XM;
410 Q=LEN(STR$(XU))/2:PRINT@1016-Q,XU;:RETURN
500 DX=(XU-XL)/80;DY=(YU-YL)/40;FOR J=0 TO 80
510 X=XL+DX*(J);XP=32+J;GOSUB 700
520 IF F=1 THEN PRINT@912+J/2,F$;:GOTO 560
530 IF F=2 THEN PRINT@16+J/2,F$;:GOTO 560
540 IF F=3 THEN PRINT@912+J/2,X$;:GOTO 560
550 SET(XP,YP):SET(XP+1,YP)
560 NEXT
570 PRINT@193,"HIT ANY KEY";:PRINT@321,"TO CONTINUE";
580 Q$=INKEY$:IF Q$=" " THEN 580
590 RETURN
700 GOSUB 5000:V=(Y-YL)/DY;D=INT(V)
710 IF Y<YL THEN F=1:RETURN
720 IF Y>YU THEN F=2:RETURN
730 IF D=0 THEN F=3:RETURN
740 F=0:YP=42-D:RETURN
800 DX=(XU-XL)/80;X=XL;GOSUB5000:MN=Y;MX=Y;FOR J=1 TO
80
810 X=XL+DX*J;GOSUB 5000:IF Y>MX THEN MX=Y
820 IF Y<MN THEN MN=Y
830 NEXT
840 PRINT:PRINT"OVER THIS RANGE OF X":PRINT"  MINIMUM
Y =";MN
850 PRINT"  MAXIMUM Y =";MX:PRINT"NOW CHOOSE THE SCALE
FOR Y"
860 PRINT:INPUT"MINIMUM Y SCALE VALUE";YL
870 INPUT"MAXIMUM Y SCALE VALUE";YU:IF YU>YL THEN RET
URN
880 PRINT:PRINT"  -- BAD Y SCALING --":GOTO 840
900 CLS:PRINT CHR$(23);"X SCALING":PRINT"  MINIMUM - "
;XL
910 PRINT"  MID      - ";XM:PRINT"  MAXIMUM - ";XU:PRIN
T
920 PRINT"Y SCALING":PRINT"  MINIMUM - ";YL
930 PRINT"  MID      - ";YM:PRINT"  MAXIMUM - ";YU:PRINT
940 PRINT"HIT 'G' TO SEE THE GRAPH AGAIN"
950 PRINT"  ANY OTHER KEY TO QUIT"
960 Q$=INKEY$:IF Q$="" THEN 960
970 IF Q$="G" THEN GOSUB 300:GOSUB 500
980 RETURN
1000 PRINT@128,A$;:FOR J=130 TO 148:PRINT@J,B$;:NEXT
1010 PRINT@152,"WARNING!";:FOR J=170 TO 188:PRINT@J,B$

```

```

: NEXT
1020 PRINT@190,A$::PRINT@200,"THE SUBROUTINE AT LINES"
;
1030 PRINT@264,"5000-5999 IS ASSUMED TO";
1040 PRINT@324,"DEFINE Y AS A FUNCTION OF X";
1050 FOR J=192 TO 320 STEP 64:PRINT@J,A$::K=J+62:PRINT
@K,A$;
1060 NEXT:PRINT@384,B$::PRINT@446,B$::FOR J=386 TO 444
1070 PRINT@J,B$::NEXT:RETURN
4950 REM
4960 REM *****
4970 REM * SUBROUTINE AT LINE 5000 MUST BE SET *
4980 REM *****
4990 REM *** Y=F(X) GOES HERE ***
4999 REM
5000 REM *** Y=F(X) GOES HERE ***
5010 Y=SIN(X)
5999 RETURN

```

六十三、 JOT

刘炳文

JOT是两个人玩、含有大量智力推导的文字游戏，由用户和计算机进行“对抗”竞赛。

这个程序之所以有趣，主要原因是：在游戏过程中，你会觉得计算机和人一样，有较强的分析、推理和判断能力，即具有一定的“智能”。它可以根据提示信息，逐步推出你的心里想的是一个什么词；如果你和计算机“耍花招”，它还可以指出前后矛盾的情况。总之，给人的印象是，你不是在与计算机玩游戏，而是和一个十分聪明的人进行智力竞赛。

JOT的规则十分简单。游戏中所使用的词一律由三个字母组成，词中的字母不能重复，例如，不能使用诸如“ZOO”、“SEE”之类的词。其余规定请参见后面的“修改”部分。

游戏开始时由计算机和用户各自选择一个“秘而不宣”的单词，看谁能首先“推导”出对方的“保密词”。

比赛双方轮流猜测。每次猜测之后，要告诉对方，他所猜的词与自己的“保密词”有多少个字母相同（即猜中），只告诉数目，而不管字母所在位置是否正确。例如，假定保密词是“OWN”，猜“WHO”则为猜中两个字母。先猜出对方保密词的一方为胜者。

程序的用法

打入RUN后，屏幕上将显示简短的“开场白”，同时让用户考虑自己的保密词是什么，接着计算机询问用户是否想首先猜测。在这之后，用户和计算机将轮流猜测对方的保密词。

计算机每猜完一次，将要求用户对它的猜测作回答。可能的回答是0，1，2，3或R。回答R(Right)意即计算机已准确地猜到了用户的保密词(真叫人扫兴)；回答数字则表明计算机所猜中的保密词中的字母数。回答“3”意味着所有的字母都是正确的，但需要按字母在词中的位置进行调整。例如，保密词是“TEA”，计算机猜的是“EAT”。

计算机得到用户的“回答”之后，将需要一定的时间为自己下一步的操作采取相应的“措施”。如果这个时间不是很短，将显示出如下信息：

“I'M THINKING...”(我在思考)。如果计算机在处理过程中发现前后有矛盾，它将

向用户索取保密词，然后分析错在什么地方。

轮到用户猜测时，除猜保密词外，还可以使用两种特殊的回答，各用一个字母来表示，即“S”和“Q”。回答“S”将显示“猜测一览”，列出前面各次猜测及每次相应猜中的字母数，这对尽快地猜到保密词有时是很有用的。显示“猜测一览”之后，程序将提示用户进行下一次猜测。

如果不使用上述特殊回答，则输入所猜测的保密词。当用户输入的词不合法时，计算机将立即作出反映；输入合法的词之后，计算机将告诉用户猜中了多少个字母。如果用户首先准确地猜到了计算机的保密词，它会及时向你表示“祝贺”，然后索要用户的保密词，进行验证。对该程序可作如下修改：

①一般来说，“猜测一览”有助于较快地猜到正确的保密词。如果想让程序在每次猜测之前都能自动地提供“猜测一览”，请把500和510语句改为：

```
500 IF G1 > 0 OR G2 > 0 THEN GOSUB 1000
510 PRINT: INPUT "YOUR GUESS (OR Q) " : P$
```

②当前程序中允许猜测的最多次数为25，它由160语句中的M确定。实际上，这个值比所需要的猜测数要大一些。在具体的游戏中，几乎很少有多于15次的猜测。因此，可以适当减小M的值。例如，可修改M=15。

③要修改数据清单是很容易的。现有数据中的单词组成规则是：由3个不同的字母组成，并且不能是：

- 用大写字母开头的专用词
- 缩略词
- 感叹词（如“ogh”，“hey”等）
- 专业词汇（如“Ohm”，“yaw”等）

语句160中的N是数据清单所具有的词汇的总数，数据清单从5000开始。

要增加词汇，可在现有的数据之后，用DATA语句加上新的词汇，所用的语句行号要大于5570，并重新规定语句610中的N值，即把406加上新增加的词汇数。例如，为了在数据里增加“OHM”和“YAW”这两个词，可以把语句160改为：

```
160 M = 25: N = 408
```

并增加一个新的DATA语句行：

```
5580 DATA OHM, YAW
```

要删除词汇，就得按相反的方法去做。即从适当的语句中去掉不需要的词汇，并相应地减小N的值。

④在游戏过程中增加一些内容，可以使程序更加人格化。另外，L的值是计算机把用户的保密词缩小范围之后的词数，可以在程序中随时测试它的值，并在它缩小时宣布：

```
BE CAREFUL, I'M CLOSING IN ON YOU.
```

程序和变量说明

150~180 定义数组

200~330 显示“开场白”，计算机选定保密词。

500~570 用户猜测计算机的保密词。

600~680 计算机猜测用户的保密词。

800~870 计算机判断用户所选定的保密词，并进入数组A\$中。

- 900~970 根据用户给出的信息,对前后矛盾的情况进行处理。
 1000~1090 显示当前的“猜测一览”。
 1100~1130 询问用户是否再玩一次游戏。
 1500~1610 把用户的猜测与计算机保密词的各个字母进行比较。
 1800~1830 检查用户所猜测的词是否合法。
 2200~2240 随机变动数组 A\$ 中词汇的内容。
 2400 交换数组 A\$ 中的元素 A 和 B。
 2600~2610 把单词 Q\$ 分解为独立的字母。
 3000 把数据(词)装入数组 A\$。
 3200~3220 计算机获胜后的处理。
 3400~3490 用户获胜后的处理。
 3600~3610 猜测次数超过规定之后的出错处理。
 5000~5570 数据清单。

- N 数据(词汇)总数。
 M 允许猜测的最多次数。
 A\$ 存放词汇的数组。
 G1\$, G2\$ 在猜测时用户和计算机所使用的字符串数组。
 H1(M), H2(M) 在每次猜测中用户和计算机猜到的字母数。
 G1, G2 用户和计算机当前猜测的次数。
 M\$ 计算机所选定的保密词
 M1\$, M2\$, M3\$ 计算机保密词中的第一、第二、第三个字母。
 P\$, Q\$ 字符串中间变量和工作变量。
 L 计算机在猜测中逐步缩小保密词的范围, L 为当前的词数。
 F 输入词是否合法的特征值。
 A, B 在交换数组 A\$ 中的元素时所使用的变量。
 H, J, P, Q 中间变量、数组和循环变量。

源程序及运行情况

```

100 REM JOT
150 CLEAR 200 :DEFINT A-Z:RANDOM
160 M=25:N=406
170 DIM A$(N)
180 DIM G1$(M),G2$(M),H1(M),H2(M)
200 G1=0:G2=0
210 L=N
250 CLS:PRINTTAB(25)"J O T":PRINT
260 PRINT"JUST AMOMENT PLEASE .....":GOSUB 3000:PRINT:
Q=RND(N)
270 PRINT"THANKS,NOW LET'S EACH THINK OF OUR SECRET WO
RD"

```

```

280 PRINT"(THIS TAKES ME A WHILE ...)"
290 GOSUB 2200:M#=A$(Q):PRINT:PRINT"OK, ";
300 INPUT"DO YOU WANT TO GO FIRST";Q$
310 Q$=LEFT$(Q$,1):IF Q$="N" THEN 600
320 IF Q$="Y" THEN 500
330 PRINT:PRINT"YES OR NO PLEASE":PRINT:GOTO 300
500 PRINT:INPUT"YOUR GUESS (OR S OR Q)";P$
510 IF P$="S" THEN GOSUB 1000:GOTO 500
520 IF P$="Q" THEN 1100
530 IF P$=M$ THEN G1=G1+1:G1$(G1)=P$:H1(G1)=9:GOTO 340
0
540 GOSUB 1800:IF F=0 THEN PRINT" THAT'S NOT ALEAGAL W
ORD";
550 IF F=0 THEN PRINT" --TRY AGAIN":GOTO 500
560 Q#=M$:GOSUB 2600:Q#=P$:GOSUB 1500:PRINT"# OF HITS
IS";Q
570 G1=G1+1:G1$(G1)=Q$:H1(G1)=Q:IF G1=M THEN 3600
600 Q#=A$(L):G2=G2+1:G2$(G2)=Q$
610 PRINT:PRINT"MY GUESS IS -- ";Q$
620 INPUT "HOW DID I DO (0-3 OR R)";P$
630 P$=LEFT$(P$,1)
640 IF P$="R" THEN H2(G2)=9:GOTO 3200
650 P=VAL(P$)
660 IF P>3 OR (P=0ANDP$<>"0")THEN PRINT"BAD ANSWER":GO
TO 610
670 IF L>100 THEN PRINT"I'M THINKING..."
680 H2(G2)=P:GOSUB 800:GOTO 500
800 Q#=G2$(G2):H=H2(G2):J=0:GOSUB2600:L=L-1:IFL<1 THEN
900
810 J=J+1:IF J>LTHEN 870
820 Q#=A$(J):GOSUB 1500
830 IF Q=H THEN 810
840 A=J:B=L:GOSUB2400:L=L-1
850 IF L<1THEN 900
860 IF L>=J THEN 820
870 RETURN
900 PRINT:PRINT"SOMETHING'S WRONG !!"
910 PRINT:INPUT"WHAT'S YOUR SECRET WOED";P$:GOSUB 1800
920 IF F<>0 THEN 940
930 PRINT"ILLEGAL WORD -I NEVER HAD A CHANCE":GOTO1100
940 PRINT:PRINT"YOU GAVE A BAD ANSWER SOMEWHERE --"
950 PRINT"HIT ANY KEY TO CHECK THE SUMMARY"
960 Q#=INKEY$:IF Q$="" THEN 960
970 GOSUB 1000:GOTO1100
1000 PRINT:Q=G1:IF G2>G1 THEN Q=G2
1010 IF Q=0 THEN PRINT"NO GUESSES YET":RETURN
1020 FOR J=1 TO 41:PRINT"--";:NEXT:PRINT"--"
1030 PRINT"YOUR GUESS SUMMARY MY GUESS"
1040 PRINT"WORD HITS";TAB(32)"WORD HITS"
1050 FOR J=1TO Q :K=1:IF J>9 THEN K=0

```

```

1060 IF J>G1 THEN PRINTSTRING$(19+K,32);J;STRING$(10,3
2);G2$(J);STRING$(2,32);H2(J):GOTO1090
1070 IF J>G2 THEN PRINT"";G1$(J);" ";H1(J);STRING$(10
+K,32);J:GOTO1090
1080 PRINT" ";G1$(J);" ";H1(J);STRING$(10+K,32);J;STRIN
G$(10,32);G2$(J);" ";H2(J)
1090 NEXT:RETURN
1100 PRINT:INPUT"HOW ABOUT ANOTHER GAME";Q$
1110 Q$=LEFT$(Q$,1):IF Q$="Y"THEN 200
1120 IF Q$="N"THEN END
1130 PRINT:PRINT"YES OR NO PLEASE":GOTO 1100
1500 P$=LEFT$(Q$,1):Q=0:GOSUB1600
1510 P$=MID$(Q$,2,1):GOSUB 1600
1520 P$=RIGHT$(Q$,1):GOSUB 1600:RETURN
1600 IF P$=M1$ OR P$=M2$ OR P$=M3$ THEN Q=Q+1
1610 RETURN
1800 F=0
1810 FOR J=1 TO N
1820 IF A$(J)=P$ THEN F=1:RETURN
1830 NEXT:RETURN
2200 FOR A=N TO 100 STEP -1:B=RND(A):GOSUB 2400
2210 NEXT:PRINT
2220 PRINT"I'VE ALMOST GOT IT ..."
2230 FOR A=99 TO 2 STEP -1 :B=RND(A)
2240 GOSUB 2400:NEXT :RETURN
2400 Q$=A$(B):A$(B)=A$(A):A$(A)=Q$:RETURN
2600 M1$=LEFT$(Q$,1):M2$=MID$(Q$,2,1)
2610 M3$=RIGHT$(Q$,1):RETURN
3000 RESTORE:FOR P=1 TO N:READ A$(P):NEXT:RETURN
3200 PRINT:PRINT"IT SURE FEELS GOOD"
3210 PRINT:PRINT"MY WORD WAS -";M$
3220 GOTO 1100
3400 PRINT:PRINT"CONGRATULATIONS - THAT WAS IT":PRINT
3410 INPUT"WHAT WAS YOUR WORD";P$:GOSUB 1800:J=1
3420 IF F<>0 THEN 3440
3430 PRINT"ILLEGAL WORD - I HAD NO CHANCE":GOTO1100
3440 IF A$(J)=P$ THEN PRINT"NOCE WORD":GOTO1100
3450 J=J+1:IF J<=L THEN 3440
3460 PRINT:PRINT "YOU MADE AN ERROR SOMEWHERE --"
3470 PRINT"HIT ANY KEY TO CHECK SUMMARY"
3480 Q$=INKEY$:IF Q$="" THEN 3480
3490 GOSUB 1000:GOTO1100
3600 PRINT"SORRY, I'M OUT OF MEMORY":PRINT
3610 PRINT"MY WORD WAS -";M$:GOTO1100
5000 DATA ACE,ACT,ADE,ADO,ADS,AFT,AGE
5010 DATA AGO,AID,AIL,AIM,AIR,ALE,ALP
5020 DATA AND,ANT,ANY,APE,APT,ARC,ARE
5030 DATA ARK,ARM,ART,ASH,ASK,ASP,ATE
5040 DATA AWE,AWL,AXE,AYE,BAD,BAG,BAN
5050 DATA BAR,BAT,BAY,BED,BEG,BET,BID

```

5060 DATA BIG, BIN, BIT, BOA, BOG, BOW, BOX
5070 DATA BOY, BUD, BUG, BUM, BUN, BUS, BUT
5080 DATA BUY, BYE, CAB, CAD, CAM, CAN, CAP
5090 DATA CAR, CAT, COB, COD, COG, CON, COP
5100 DATA COT, COW, COY, CRY, CUB, CUD, CUE
5110 DATA CUP, CUR, CUT, DAB, DAM, DAY, DEN
5120 DATA DEW, DIE, DIG, DIM, DIN, DIP, DOE
5130 DATA DOG, DON, DOT, DRY, DUB, DUE, DUG
5140 DATA DYE, DUO, EAR, EAT, EGO, ELK, ELM
5150 DATA END, ELF, ERA, FAD, FAG, FAN, FAR
5160 DATA FAT, FED, FEW, FIG, FIN, FIR, FIT
5170 DATA FIX, FLY, FOE, FOG, FOR, FOX, FRY
5180 DATA FUN, FUR, GAP, GAS, GAY, GEM, GET
5190 DATA GIN, GNU, GOB, GOD, GOT, GUM, GUN
5200 DATA GUT, GUY, GYP, HAD, HAG, HAM, HAS
5210 DATA HAT, HAY, HEN, HEX, HID, HIM, HIP
5220 DATA HIS, HIT, HER, HEM, HOE, HOG, HOP
5230 DATA HOT, HOW, HUB, HUE, HUG, HUM, HUT
5240 DATA ICE, IEY, ILK, INK, IMP, ION, IRE
5250 DATA IRK, ITS, IVY, JAB, JAR, JAW, JAY
5260 DATA JOB, JOG, JOT, JOY, JUG, JAG, JAM
5270 DATA JET, JIB, JIG, JUT, KEG, KEY, KID
5280 DATA KIN, KIT, LAB, LAD, LAG, LAP, LAW
5290 DATA LAY, LAX, LED, LEG, LET, LID, LIE
5300 DATA LIP, LIT, LOB, LOG, LOP, LOT, LOW
5310 DATA LYE, MAD, MAN, MAP, MAR, MAT, MAY
5320 DATA MEN, MET, MID, MOB, MOP, MOW, MUD
5330 DATA MIX, MUG, NAB, NAG, NAP, NAY, NET
5340 DATA NEW, NIL, NIP, NOD, NOT, NOR, NOW
5350 DATA NUT, OAF, OAK, OAR, OAT, ODE, OIL
5360 DATA OLD, ONE, OPT, ORE, OUR, OUT, OVA
5370 DATA OWE, OWL, OWN, PAD, PAL, PAN, PAR
5380 DATA PAT, PAW, PAY, PEA, PEG, PEN, PET
5390 DATA PEW, PIE, PIG, PIT, FLY, POD, POT
5400 DATA POX, PER, PIN, PRO, PRY, PUB, PUN
5410 DATA PUS, PUT, RAG, RAM, RAN, RAP, RAT
5420 DATA RAW, RAY, RED, RIB, RID, REV, RIG
5430 DATA RIN, RIP, ROB, ROD, ROE, ROT, ROW
5440 DATA RUB, RUE, RUG, RUM, RUN, RUT, RYE
5450 DATA SAD, SAG, SAP, SAT, SAW, SAY, SET
5460 DATA SEW, SEX, SHY, SEA, SIN, SHE, SIP
5470 DATA SIR, SIT, SIX, SKI, SKY, SLY, SOB
5480 DATA SOD, SON, SOW, SOY, SPA, SPY, STY
5490 DATA SUE, SUM, SUN, TAB, TAD, TAG, TAN
5500 DATA TAP, TAX, TAR, TEA, TEN, THE, THY
5510 DATA TIC, TIE, TIN, TIP, TOE, TON, TOP
5520 DATA TOW, TOY, TRY, TUB, TUG, TWO, URN
5530 DATA USE, UPS, VAN, VAT, VEX, VIA, VIE
5540 DATA VIM, VOW, YAK, YAM, YEN, YES, YET

5550 DATA YOU, WAD, WAG, WAN, WAR, WAS, WAX
5560 DATA WAY, WEB, WED, WET, WHO, WHY, WIG
5570 DATA WIN, WIT, WOE, WON, WRY, ZIP, FIB

MY GUESS IS -- KID

HOW DID I DO (0-3 OR R)? R

IT SURE FEELS GOOD (的确感到高兴)

MY WORD WAS -SIN (我的保密词是SIN)

HOW ABOUT ANOTHER GAME? _ (还想再玩一次吗?)

RUN_

J O T

JUST AMOMENT PLEASE (请稍等一会)

THANKS, NOW LET'S EACH THINK OF OUR SECRET WORD (谢谢, 现在让我们彼此都想一下保密词)

(THIS TAKES ME A WHILE ...)

I'VE ALMOST GOT IT ... (我已经想好了)

OK, DO YOU WANT TO GO FIRST? Y (好, 你要先猜吗?)

YOUR GUESS (OR S OR Q)? ACE (你猜的是什么?)

OF HITS IS 0 (未猜中)

MY GUESS IS -- ICE (我猜的是ICE)

HOW DID I DO (0-3 OR R)? 1_ (我猜的怎么样? 键入1说明猜中1个字母)

I'M THINKING... (我在想)

YOUR GUESS (OR S OR Q)? FIX (你猜的是什么?)

OF HITS IS 1

MY GUESS IS -- IVY

HOW DID I DO (0-3 OR R)? 1

I'M THINKING...

YOUR GUESS (OR S OR Q)? FLY

OF HITS IS 0

MY GUESS IS -- INK

HOW DID I DO (0-3 OR R)? 2

YOUR GUESS (OR S OR Q)? S_ (键入S, 显示猜测一览)

YOUR GUESS	SUMMARY	MY GUESS
WORD HITS		WORD HITS
ACE 0	1	ICE 1
FIX 1	2	IVY 1
FLY 0	3	INK 2

YOUR GUESS (OR S OR Q.)? BID

OF HITS IS 1

MY GUESS IS -- KID

六十四、单人击球游戏

韩启明

下面介绍的是在TRS 80 II型机上运行的“单人击球游戏”。游戏开始后,首先输入游戏级别选择,一共有3级,“1”为最高级,玩起来比较困难,“2”、“3”级次之。接着在显示器上画出游戏场地和游戏器具。场地呈长方形,四周用档板围住,右边有七道厚墙,左边是球

拍。球拍可以上下移动。按下“Q”键，球拍向上移动；按下“P”键，球拍向下移动。小球在场内来回运动，碰到四周的挡板时，会自动弹回。在弹向右边时，小球如果将碰到的第一道墙击穿，则在下一道墙前面弹回，每击穿一道墙可得10分。当球弹回左边时，游戏者应迅速移动球拍将其档回，如果没有击中球，则会前攻尽弃，分数又从0开始计。游戏时的得分不断显示在显示器的右上角。

程序设计思想与语句说明

① 设计思想

游戏场地与器具的设计主要是用符号或图示块拼接成的，通过循环语句不难实现，不做进一步说明了。

球拍的运动主要靠键盘输入命令，给球拍位置X一个增量。当按下“Q”键时，给X一个负增量，即 $X = X - 1$ ；当按下“P”键时，给X一个正的增量，即 $X = X + 1$ 。这样，通过打印位置的变化，即可使球拍上下移动。

程序中给出行，列的极限值，即1，22和1，78。1，22是球在行方向上的极限值，1，78是球在列方向的极限值。某一时刻，当球处于所在行或列的极限值时，程序给出行或列的负的增量，即 $R1 = -R1$ ， $C1 = -C1$ 。这样就使球向相反方向运动。球在游戏场地的运动即是通过这种办法实现的。

那么球是怎样穿透墙并且得分的呢？程序中给出七道墙，它们的列坐标分别为35、40、45、50、55、60、65。每当球向右运动，并且列坐标等于上述某一个值时（程序中用320~330语句进行判断），表示碰到了该道墙。每道墙都用数组A(I, R-1)记录该墙的破损状态。当该墙未破损时，球碰到该墙弹回，同时记录球已撞到该墙一次（B=0），当球第二次碰到该墙时，可穿透该墙得分，同时置该墙为破损状态（A(I, R-1)=0）。当球继续前进，碰到下一道墙时，由于下一道墙没有被球撞过（B=1），所以球弹回，在弹回的同时，置B=0，表示该墙已被球撞过，即下一道墙处于可穿透状态。如果你能很好地控制球拍，使每次球从厚墙处弹回时都能挡回，则可连续使墙破损得分。否则，只要有一次未挡回，都会前功尽弃。程序置分数S=0，又要重新开始。

② 语句说明

- 10~60 语句，输入游戏级别，初始化。
- 70~170 语句，显示游戏场地与器具。
- 180~200 语句，显示小球以及得分。
- 210~260 语句，控制球拍运动。
- 270~300 语句，显示球碰到挡板时的运动轨迹。
- 310~360 语句，球击穿厚墙的判断与处理。
- 370~390 语句，游戏重新开始的程序。

主要变量说明

- R： 球所处的行。
- C： 球所处的列。
- X： 球拍所处的位置。
- B： 记录墙的可穿透状态。
- R1： 球位置的行增量。
- C1： 球位置的列增量。

S: 记录分数。

A (6,22) : 记录七道墙的破损情况。

3 源程序

```
10 DIM A(6,22)
20 CLS:PRINT@(12,25),"SELECT THE GAME LEVEL (1-3)"
30 INPUT T:IF T>3 OR T<1 THEN 20
40 FOR I=0 TO 6 :FOR J=0 TO 22
50 A(I,J)=1
60 NEXT J,I :CLS
70 PRINT@(0,0),STRING$(79,"-"); 即在79个“-”的字符串
80 PRINT@(23,0),STRING$(79,"-");
90 FOR I=1 TO 23
100 PRINT@A(I,0),CHR$(149);:PRINT@A(I,79),CHR$(149);
110 NEXT I
120 FOR I=0 TO 6: FOR J=1 TO 23
130 PRINT@(J,35+5*I),CHR$(158);
140 NEXT J,I
150 PRINT@(12,1),CHR$(158);:PRINT@A(13,1),CHR$(158);
160 X=12:S=0:A#=CHR$(154)
170 R=RND(22)+1:C=RND(32)+1:R1=1:C1=1
180 PRINT@(R,C),A#;
190 IF C=1 AND NOT(X=R OR X+1=R) THEN S=0
200 PRINT@A(4,70),S;:PRINTCHR$(2);
210 FOR I=1 TO T
220 C#=INKEY$
230 IF C#="Q" AND X>1 THEN PRINT@(X+1,1)," ":X=X-1
240 IF C#="P" AND X<21 THEN PRINT@(X,1)," ":X=X+1
250 PRINT@(X,1),CHR$(158);:PRINT@(X+1,1),CHR$(158);
260 NEXT I
270 PRINT@A(R,C)," "
280 IF R=22 OR R=1 THEN R1=-R1
290 IF C=78 OR C=1 THEN C1=-C1
300 R=R+R1:C=C+C1
310 IF C<35 THEN 180
320 I=(C-35)/5:IF I>6 THEN 180
330 IF I<>INT(I) OR A(I,R-1)=0 :GOTO180
340 IF C1<0 THEN A(I,R-1)=0: GOTO 180
350 IF B=0 THEN B=1 ;A(I,R-1)=0:S=S+10: GOTO180
360 C=C-2:C1=-1:B=0:GOTO180
370 C#=INKEY$
380 IF C#<>" " THEN 370
390 GOTO20
```

六十五 “S 先生和 P 先生” 谜题

邢祖畴

在我国计算机学会第六届年会上曾经有人提到一个著名的谜题:

选择两个自然数 i 和 j , 使得 $2 < i < j < 990$ 。将这两个数的和 x 告诉 S 先生; 而将两

数的积 y 告诉 P 先生。随后 S 先生和 P 先生进行了如下的对话:

S 先生:“我知道你不知道这两个数是什么,但我也不知道。”

P 先生:“现在我知道这两个数是什么了。”

S 先生:“现在我也知道了。”从以上的对话来看,这两个数是什么?

算法分析

由于 $x = i + j$, $y = i \times j$ 。所以欲知 i 、 j , 必须先求出 x 、 y 。因此,问题转化为求 x 、 y 。在对话开始前, x 存在的集合 s 是 $4 \sim 98$, y 的集合是 $2^2 \sim 99^2$ 。而三句话后就能找到 x 、 y , 说明经过这三句话就使 S 和 P 先生缩小到仅有的唯一选择。下面就来分析此过程。

第一句话首先将 s 缩小成 s_1 , 并说明了 s_1 中数的两条性质: (1) 对于 x 的全部分解形式, 不能有一种使得 i 、 j 为一对素数, 否则 P 就有知道 i 、 j 的可能。如 $x = 10 = 2 + 8 = 3 + 7 = 4 + 6 = \dots$, 其中 3 、 7 为一对素数, 而当 $y = 21$ 时, P 先生就可以知道 i 、 j 了。(2) x 的分解法不是唯一的, 如 x 不能是 4 、 5 、 197 和 198 。如 $x = 197$, 则 i 、 j 只能为 98 、 99 , 这样 S 先生就可以知道 i 、 j 了。

根据自己手中的数和 S 先生的话, P 先生找到了 x 、从而知道了 i 、 j 。听了 P 先生的话, 我们虽然还不能确定 y , 但可以分析出缩小后的 P 集合 (即 p_1) 的性质: 由于 $x = i + j$, $y = i \times j$, 如果 x 不在 s_1 中, 则 P 先生就不能解出 i 、 j ; 如 y 可分解成 $i_1 \cdot j_1$ 或 $i_2 \cdot j_2$, 于是 $x_1 = i_1 + j_1$, $x_2 = i_2 + j_2$ ($x_1 \neq x_2$; 若 x_1 、 x_2 都属于 s_1 , 则 P 先生就不能确定 i 、 j 。因此, p_1 中的 y 所分解的各对 i 、 j , 只能在 s_1 中有唯一对应的 x 。当然不排除两个以上不同的 y , 在 s_1 中有同一个 x 与之对应。而当 S 先生第二次说话时表明他已从 s_1 中选出了排除这种情况的 x 。如果 x 在 p_1 中可以对应两个以上的 y , 则 S 先生就不能确定 i 、 j 。因此, S 先生所找的 x , 在 p_1 中只能有唯一的 y 与之对应。

经过上述三步筛选, 就确定了 x 、 y 。由于 $1 \leq i$, $y = i \times j$, 可解出

$$i = \frac{x}{2} - \sqrt{\left(\frac{x}{2}\right)^2 - y}, \quad j = \frac{x}{2} + \sqrt{\left(\frac{x}{2}\right)^2 - y}$$

程序说明

根据上面分析, 可以编制出 BASIC 程序(附后)。程序共分三部分:

① 10~190 语句。建立 x 的集合 s_1 , 同时将其存入数组 $s_1(k)$ 。根据前面的分析, s_1 的范围是 6 至 196 。其中任何一个数所分解的各对组合数, 不能有一对是素数。40~140 语句完成一个数字的全部分解。对一个素数的判定, 需调用程序段 440~480。

② p_1 集合的建立依赖于 s_1 。其对应关系如前所述。任务是要在它们中选出有一一对应关系的 x 、 y 。200~420 语句建立这种关系, 并找到了这对数。办法是对 $s_1(A)$ 中的任何一个数的各种分解, 看其中有几对乘积属于 p_1 ; 找到只有一种分解的乘积属于 p_1 的 $s_1(A)$, 它和这个乘积就是所求的 x 、 y 。

程序中用了四层循环: 200 语句考察所有的 $S_1(A)$; 210 语句对当前所考察的 $S_1(A)$ 进行全部分解; 240 语句考察其中一种分解所产生的数是否属于 p_1 ; 260 语句扫视全部 $s_1(A)$ 。其中以 Q 的值是否为 1 来判断该数应否属于 p_1 , 如属于 p_1 , 将其送入相应的 $p_1(A)$ 。 $s_2(A)$ 是计数器, 记录查找过程中 $s_1(A)$ 的全部分解数的乘积属于 p_1 的个数。当一个 $S_1(A)$ 被考察完之后, $p_1(A)$ 记录了 $S_1(A)$ 的最后一种分解在 p_1 的值。

③380~410语句, 对于找到的 x 、 y , 计算出 i 、 j , 并打印。

源程序

```
10 DIM S1(200),S2(200),P1(200)
20 K=0
30 FOR U=6 TO 196
40 FOR V=2 TO U/2
50 P=0
60 N=U-V
70 IF N>99 THEN 140
80 GOSUB 440
90 IF P=0 THEN 140
100 N=V
110 P=0
120 GOSUB 440
130 IF P=1 THEN 190
140 NEXT V
150 K=K+1
160 S1(K)=U
170 S2(K)=0
180 P1(K)=0
190 NEXT U
200 FOR A=1 TO K
210 FOR B=2 TO S1(A)/2
220 Q=0
230 M=B*(S1(A)-B)
240 FOR U=2 TO SQR(M)
250 IF INT(M/U)=M/U THEN N=M/U+U ELSE 310
260 FOR C=1 TO K
270 IF N=S1(C) THEN 300
280 NEXT C
290 GOTO310
300 Q=Q+1
310 NEXT U
320 IF Q<>1 THEN 350
330 S2(A)=S2(A)+1
340 P1(A)=M
350 NEXT B
360 PRINTS1(A),
370 IF S2(A)<>1 THEN 420
380 R=SQR((S1(A)/2)*(S1(A)/2)-P1(A))
390 I=S1(A)/2-R
400 J=S1(A)/2+R
410 PRINT"S=";S1(A),"P=";P1(A),"I=";I,"J=";J
420 NEXT A
430 END
440 FOR L=2 TO SQR(N)
450 IF INT(N/L)=N/L THEN 480
460 NEXT L
470 P=1
480 RETURN
```

六十六 快速掷骰子游戏

刘炳文

不少人可能玩过掷骰子游戏。由于它需要“赌”输赢，因而具有一定的刺激性和吸引力，容易引起人们的兴趣。我们当然反对把骰子作为“赌具”来使用；但是，如果用它来丰富我们的业余生活，还是值得提倡的。

利用这个程序，你可以和计算机玩掷骰子的游戏。计算机尽管很“高明”，你也未必会输给它，因为这个游戏在很大程度上不是靠技巧，而是靠“运气”。只要认真对付，要战胜计算机不是很困难的。游戏的唯一技巧，就是在是否把掷（滚动）骰子的权力交给对手时所做的判断。经过判断后的操作，可能使你（或计算机）加倍得分，但也可能会输掉一些分，甚至前功尽弃。

计算机将为双方滚动骰子——不过请勿担心，它很“公正”，不会偏向任何一方。

这个游戏的名字也可以叫“叹息”，在玩过之后就会明白它的含义。

程序用法

程序中使用两颗骰子，每颗骰子六个面，分别有1到6个点。但是，与普通骰子不同的是，1点所在的那一面不是点，而是一幅画，上面画的是“皱着眉的脸”，其它五面分别有2到6个点。

程序运行后，首先询问取胜的分数是多少，任何正整数都是允许的。不过50~100之间的值有可能产生最好的游戏效果。输入分数之后，程序进行抛掷硬币的模拟，以决定谁先滚动（掷）。

每颗骰子的滚动用一个图形在屏幕上的移动来表示，程序“驱动”两颗骰子向前滚动，然后停下来，并显示出每颗骰子朝上的一面的点数（或者皱着眉的脸）。在每次滚动之前，程序指明将要滚动的是哪一方。

游戏取胜的条件是，看谁能第一个达到游戏开始前双方“商定”的得胜分数。游戏在双方交替进行，轮到的一方至少要滚动一次骰子，可能要滚动几次，其规则如下：

只要在滚动后两颗骰子都没有出现“皱眉”，滚动的一方就可以在记分牌上记下当前一次滚动所得的分数。每次没有皱眉的滚动之后，是再滚动一次还是把“滚动权”传给对方，可以任意选择。如果传给对方，则当前一次滚动中的得分就加到前面已经获得的总分上。如果滚动权不交给对方，在再次滚动时有一颗骰子出现了“皱眉”，就要发生“叹息”（GROAN），并取消当前滚动中所得的分数。在这种情况下，前面的得分（总分）仍将保留。如果两颗骰子同时出现“皱眉”，那就要把前面所得的分数以及当前的分数统统取消。这样，所得的总分又恢复为零——真扫兴。

每次滚动后，记分牌要显示得分。这个记分牌将给出全部有关信息，诸如获胜所需要的分数，在当前一次滚动之前双方各自取得的分数，以及当前一次滚动中所取得的分数等。

如果滚动后有一颗骰子发生“皱眉”，记分牌上显示当前滚动的得分为0；在极其可怕的两只骰子出现“皱眉”时，所有的得分（当前的和前面的）都要变成0。在以上两种情况下，都将自动把骰子的滚动权传给对方。

如果在一次滚动后得了分（没有皱眉），滚动的一方必须确定是再滚动一次还是让对方滚动。对计算机来说，程序有作出这种选择的内部对策；对于用户（就是你）来说，在记分牌

上显示有关信息后，程序将会提出问题让用户回答，以作出选择，或者回答“R”，或者回答“P”。回答“R”意味着你想再滚动一次，回答“P”则是要把滚动权传给计算机。即使得分已经达到取胜的分数，也要把滚动权传出去，以便把当前的得分加到前面的总分上去。

在把滚动权传给对方后，所得总分大于或等于取胜分数的一方即为取胜方，这无疑会使得对方“叹息”。在结束游戏之前，计算机将确认并显示出谁是胜利者。

程序在开始时需要输入取胜的分数，如果想把它设置为固定值，可删除语句220，并修改语句210。只要把W置为所需要的数值就行了。例如

210 W = 100.

当前的程序有“骰子”滚动的图形显示，如果想取消这种显示，可以对语句500和540作如下修改：

500 D = 266; R1 = RND (6); R2 = RND (6); GOTO 540

540 C = D + 48; GOSUB 650; C = C + 384; GOSUB 650

游戏玩过几次之后，你可能想使游戏的速度加快，这可以通过修改语句830和840的时间延迟常数来实现。例如

830 FOR K = 1 TO 400: NEXT: RETURN

840 FOR K = 1 TO 1000: NEXT: RETURN

可使游戏的速度快一些。当然也可把常数设置得大一些，以降低游戏速度。

程序及变量说明

120~190 预置常量（字符串）。

200~270 显示标题和提示信息，得到取胜分数。

300—390 用户滚动骰子。

400—470 计算机滚动骰子。

500—590 骰子滚动和显示。

600 时间延迟。

700—770 显示两颗骰子向上的一面的点（或皱眉）。

800—820 显示“叹息”的信息。

830—840 延时循环。

850—960 显示记分牌。

970—990 结束处理。

1000—1110 计算机的对策，置 $x = 0$ ，停止滚动，或者 $x = 1$ 再滚动一次。

W 取胜所需的分数。

H 前面几次滚动中用户的得分。

P 前面几次滚动中计算机的得分。

T 当前滚动的得分。

X 计算机对策的当前值（0—停止滚动，1—再滚动一次）。

L 计算机内部对策中所使用的临界值。

V 计算机把滚动权传走后所得的分数

DL 延迟长度。

Q, Q \$ 工作变量和工作字符串变量。

J, K 循环变量。

- P\$ 当前滚动者名字的字符串。
 R1, R2 第一, 第二颗骰子的滚动结果。
 R 两颗骰子的滚动结果。
 F 滚动结果的特征值 (“0”没有皱眉; “1”, 1次皱眉; “2”, 2次皱眉)。
 C、D、G 屏幕打印位置。
 NT TAB函数的自变量。
 B\$—U\$ 图示字符串。

3. 源程序

```

100 REM: GROAN
120 CLEAR 250: RANDOM: DEFINT W
140 B$=CHR$(191): C$=CHR$(179): L$=CHR$(176): M$=CHR$(140)
)
150 U$=CHR$(131): N$=CHR$(32): F3$=B$+STRING$(40, 140)+B$
160 D1$=B$+STRING$(7, 131)+B$: D2$=B$+STRING$(7, 32)+B$
170 D3$=B$+STRING$(7, 176)+B$: E1$=STRING$(4, 32)+L$+M$+L$
$+STRING$(4, 32)
180 E2$=N$+L$+M$+U$+N$+N$+U$+M$+L$+N$: E3$=U$+M$+L$+
STRING$(5, 32)+L$+M$+U$
185 E4$=N$+N$+N$+U$+M$+L$+M$+U$+N$+N$+N$
190 F1$=B$+STRING$(40, 131)+B$: F2$=B$+STRING$(40, 176)+B$
$
200 CLS: PRINT TAB(25) "G R O A N"
210 PRINT "HOW MUCH NEEDED TO WIN"
220 INPUT "(BETWEEN 50-100 IS BESE) "; W: IF W<=0 THEN 200
0
230 PRINT: PRINT "LET'S TOSS FOR FIRST ROLL": GOSUB 830
240 PRINT: PRINT "THE COIN IS IN THE AIR AND"
250 Q$="YOU": Q=RND(2): IF Q=2 THEN Q$="I"
260 PRINT TAB(8);: FOR J=1 TO 5: PRINT ".":: GOSUB 830:
NEXT
270 PRINT Q$; " GET FIRST ROLL": GOSUB 840: T=0: IF Q=2 THEN
400
300 P$="YOU": CLS: PRINT TAB(25) "YOU'RE ROLLING": GOSUB 830
: GOSUB 500
310 T=T+R1+R2: IF F>0 THEN T=0
320 IF F=2 THEN H=0
330 GOSUB 850: IF F>0 THEN PRINT "DICE PASS TO ME": GOSUB
840: GOTO 400
340 PRINT "(P=PASS DICE - R=ROLL AGAIN)": PRINT "YOUR DEC
ITION (P OR R)?"
350 Q$=INKEY$: IF Q$="" THEN 350
360 IF Q$="R" THEN 300
370 IF Q$<>"P" THEN 350
380 PRINT: H=H+T: IF H>=W THEN 970
390 T=0: F=1: CLS: GOTO 330
400 T=0: P$="I"
410 CLS: PRINT TAB(26) "I'M ROLLING": GOSUB 830: GOSUB 500
420 T=T+R1+R2: IF F>0 THEN T=0
430 IF F=2 THEN P=0
440 GOSUB 850: IF F>0 THEN PRINT "DICE PASS TO YOU": GOSU
B 840: T=0: GOTO 300
450 GOSUB 1000: IF X=1 THEN PRINT "I'LL ROLL AGAIN": GO

```

```

SUB 840:GOTO410
460 PRINT"I'LL STOP WITH THIS":GOSUB 830:P=P+T:IF P>=W
  THEN 970
470 PRINT:PRINT"DICE PASS TO YOU":T=0:GOSUB 840 :GOTO
300
500 D=266:DL=5:R1=RND(6):R2=RND(6):FOR K=0 TO 40 STEP
  8
530 C=D+K:GOSUB650:C=C+384:GOSUB650:GOSUB600:CLS
535 GOSUB 660:C=C-384:GOSUB 660:GOSUB 600:CLS
540 NEXT :C=D+48 :GOSUB 650 :C=C+384 :GOSUB 650
550 C=D+48 :R=R1 :GOSUB 700
560 C=C+384 :R=R2:GOSUB 700:F=0 :IF R1=1 THEN F=1 :GOS
UB 800
570 IF R2=1 THEN F=F+1 : GOSUB 810
580 IF F=2 THEN GOSUB 820 :GOSUB 830
590 RETURN
600 FOR J=1 TO DL :NEXT :RETURN
650 PRINT@C-136,D1$:PRINT@C-72,D2$:PRINT@C-8,D3$:RETUR
N
660 PRINT@C-197 ,E1$:PRINT@C-133,E2$:PRINT@C-69,E3$
670 PRINT@C-5,E4$:RETURN
700 ON R GOSUB 710,730,740,750,760,770:RETURN
710 PRINT@C-134,C$::PRINT@C-130,C$::PRINT@C-68,"(":PR
INT@C-6,CHR$(182);
720 FOR Q=3 TO 5 :PRINT@C-Q,C$::NEXT:PRINT@C-2,CHR$(18
5)::RETURN
730 PRINT@C-134,C$::PRINT@C-2,C$::RETURN
740 PRINT@C-68,CHR$(140)::GOSUB 730:RETURN
750 PRINT@C-130,C$::PRINT@C-6,C$:GOSUB 730:RETURN
760 GOSUB 740:GOSUB 750:RETURN
770 PRINT@C-70,CHR$(140)::PRINT@C-66,CHR$(140)::GOSUB
750:RETURN
800 PRINT@52,"GROAN"::RETURN
810 PRINT@820,"GROAN"::RETURN
820 PRINT@434,"-DESPAIR-":RETURN
830 FOR K=1 TO 800:NEXT:RETURN
840 FOR K=1 TO 2000:NEXT:RETURN
850 PRINT@64,F1$:G=128:GOSUB 960:PRINT@135,"--S C O R
  E B O A R D --";
860 PRINT@192,F3$::G=256:GOSUB 960: PRINT@264,W;"POINT
  S NEEDED TO WIN";
870 PRINT@320,F3$::G=384:GOSUB 960
880 PRINT@386,"  POINTS SCORED          YOU          ME"::
  G=448
890 GOSUB 960:PRINT@450,"BEFORE THIS SERIES":PRINT@47
  5,H;
900 PRINT@482,P::PRINT@512,F3$::G=576:GOSUB 960
910 PRINT@582,P$;" HAVE";T;"POINTS THIS SERIES";
920 PRINT@640,F2$::PRINT@768,"":RETURN
960 PRINT@G,B$::PRINT@G+41,B$::RETURN
970 T=0:CLS:GOSUB 850: IF P>=W THEN PRINT@ 768,"SKILL
  WINS AGAIN"
980 IF H>=W THEN PRINT@768,"YOU WIN - IT WAS SHEER LUC
  K"

```

```

990 END
1000 V=P+T:IF V>=W THEN 1100
1010 IF (W-H)<10 THEN 1110
1020 IF P>=H THEN L=T/25:GOTO 1050
1030 IF V<H THEN L=T/35:GOTO 1050
1040 L=T/30
1045 S=RND(0)
1050 IF S>L THEN 1110
1100 X=0 :RETURN
1110 X=1 :RETURN

```

六十七、Wari 游戏

刘炳文

Wari 是美国一个相当古老的游戏。不但趣味性强，而且对锻炼智力，增强思维能力大有补益。和国际象棋、西洋跳棋一样，它需要较高的技巧，而不是靠侥幸和运气取胜。

下面的程序就是人与计算机玩Wari 游戏的程序。具体游戏方法如下：

程序运行后，屏幕上首先显示出Wari 棋盘并询问你(you)是否先走。Wari 棋盘分为12个格子，由两行6列组成。你的一方在下面，从左到右编号为1到6；计算机的一方在上面，从右至左编号为7到12。

游戏开始时，每个格子里放有4块“石子”。无法区分哪些石子属于你，哪些石子属于计算机，在比赛过程中，对双方都是一样的，并能从一方移到另一方。

先走的棋手“抓起”自己一方一个格子中的全部石子并把它们撒出去。撒开后，从该格子的下一个格子开始，这些石子依次连续“落入”各个格子，每个格子中放一块。例如，把第二格的4块石子“撒开”，第三、四、五、六格中的石子各增加一块，第二格中的石子数变为0。在需要时也可以进入对方的格子。例如第五个格子中的4块石子撒开后，分别落入第六、七、八、九格；而第十一个格子中的5块石子撒开后，将分别落入第十二、一、二、三、四格。第十二格之后是第一格。

如果撒出去的最后一块石子落到了对方的格子中，并且当时那个格子里的石子数变为2块或3块，则这些石子被走的一方吃掉，并从棋盘上拿走；同时，如果在被吃掉石子的前一个格子里其棋子数也为2块或3块，则该格石子也被吃掉，再往前一个格子也是如此。总之，如果最后一个格子符合“吃子”的条件，则从这个格子向前类推，直到遇到不是2块或3块石子的格子为止。

举例来说，假定操作员一方的第6个格子中有5块石子，计算机一方的第七、八、九、十、十一个方格中的石子分别为0、3、1、1、1。此时操作员走第六个方格的步，则撒出去的最后一块石子落到第十一格，这个格的石子数变为2，即被操作员一方吃掉；以此向前类推，第十、第九（即倒数第二、三）个格的石子也要被吃掉；再向前推，第八个格子虽属计算机一方，但由于石子数为4，因而“吃子”不再继续。

在比赛过程中，无论那一方吃子，总是由两个对奕者轮流着棋。

比赛获胜的条件是：一个棋手（计算机或人）“吃掉”24块石子（或者更多），谁先达到这一目标谁就为胜方。这个数字是比赛开始时棋盘上所具有的48块石子的一半。

针对程序运行中可能出现的情况，有一些特殊的规则。首先，不允许吃掉对方所有格子中的全部石子，因为这将使得对方在下一轮中无步可走；其次，当对方的每个格子里都没有石

子时（由于在前面一轮中被迫撤出了最后的石子），如果可能的话，你至少必须“送”给他一块石子，这样他才能在下一轮中有步可走。如果不能这样做，则比赛结束，并认为这一局是“和棋”（平局）。

比赛过程中，在一个格子里累积的石子有可能达到12块或者更多，此时如果用这个格的石子应着，则会使得石子散布到棋盘的所有各格子中。当出现这种情况时，将跳过应着的这个格子，因此，在走出当前的一步之后，这个格子是空的（石子数为0）。

计算机为了走出一步棋，通常都需要大约5到40秒的“考虑”时间，时间的长短取决于棋局的复杂程度。在这段时间里显示“THINKING”，并在后面加上“.”，用一串点（7到12个）来表示计算机正在“考虑”可能要走的棋步。

下面是程序运行的示例：

COMPUTER						W A R I	
12	11	10	9	8	7		
4	4	4	4	4	4		CAPTURED
4	4	4	4	4	4		COMPUTER 0
							YOU 0
1	2	3	4	5	6		
YOU							

WANT GO FIRST? YES- (图1)

程序开始运行后，画出比赛的Wari 棋盘，并询问谁先走。操作员（you）决定先走，键入YES。（见图1）

COMPUTER						W A R I	
12	11	10	9	8	7		
4	4	4	5	5	5		CAPTURED
4	4	4	4	0	5		COMPUTER 0
							YOU 0
1	2	3	4	5	6		
YOU							

YOUR MOVE? 5
THINKING..... (图2)

操作员决定用第5个格子应着，计算机正在“考虑”着法，屏幕上显示“THINKING...”（见图2）。

操作员已“吃掉”23块石子，比赛继续进行。操作员用第五个格子应着，吃掉2块石子，从而在比赛中获胜（见图3）。

程序设计思想与语句说明

W A R I

COMPUTER						
12	11	10	9	8	7	
0	2	0	0	0	1	
3	4	2	5	2	0	
1	2	3	4	5	6	
YOU						

CAPTURED
COMPUTER 6
YOU 23

YOUR MOVE? 5_ (图 3)

① 设计思想

设置一个数组，记录棋盘上12个格子中的石子数和操作员“吃掉”的石子数，以及计算机“吃掉”的石子数。当操作员应着时，依其键入的格子数，找出相对应的数组元素，并取出其中的石子数，按其石子数的多少，给该数组元素后面的数组元素所存数据（石子数）分别加1，如果最后一个被加1的数组元素的值等于2或3，则判操作员“吃子”得分，并把得分存到数组中去，否则只改变棋盘状态。计算机走着则依棋局状态的几个评价系数（详见变量说明）来确定。当确定走棋方法后，其状态变换和吃子情况同上。

② 程序流程和语句说明

- 120~150 预置变量，显示棋盘。
- 160~180 询问操作员是否先走，并对其回答进行测试。
- 190~220 确定计算机的着法，显示新的棋局。
- 230~240 判断计算机是否获胜（吃子数大于等于24），如是则显示相应的信息。
- 250~300 得到操作员走的棋步，检查是否合法，显示新的棋局。
- 310~320 判断操作员是否获胜。
- 330 在企图走非法的棋步时所显示的信息。
- 350~500 把M送入T数组的子程序。
- 360~390 把数组T复制为数组Y（在计算机走步时转换）。
- 400~430 在数组Y中的棋步。
- 440~450 测试吃子数，拿掉石子，检查拿掉石子的格子。
- 460~470 看一看对方的棋步是否合法。
- 480~500 把数组Y再复制回数组T。
- 510~570 确定计算机着数的子程序。
- 810~880 显示结束信息，询问是否再玩一次。
- 900~960 显示棋盘上的石子及吃子的子程序。
- 990 把光标移到屏幕上“YOUR MOVE”位置的子程序。
- 1050 把光标移到“MY MOVE”位置的子程序。
- 1100 光标回车并显示一个空格字符的子程序。
- 1200~1270 显示棋盘（不包括石子）的子程序。
- 2000 计算机走了非法的棋步之后显示的信息。

③变量的含义

- J、K 下标变量。
- Q、P、F、D 分别存放常数14、13、50和12。
- T、Y、W Wari 棋盘中临时复制的数组。
- V 操作员所作的6种可能回答的计值数组。
- E 计算机所要使用的6种可能的着法的计值数组。
- B 存放Wari棋盘的数组。第13个元素中含有操作员“吃掉”的石子数，第14个元素中含有计算机“吃掉”的石子数。
- ZA、ZB 求值操作的加权系数。
- MN 棋步数。
- R S 操作员的应着，也作开关用（C——计算机，H——人）。
- M 要走的棋步（1~6为操作员，7~12为计算机），负数为非法棋步。
- C 棋盘上失掉的石子数的下标。
- L 最后一个格子里失掉的石子。
- S 走了一步之后，棋盘上对方的石子数。
- A 为指明计算机当前6种可能的着法而使用的下标。
- FA 走出一步之后，在确定棋局的“理想值”时所使用的第一个评价系数（指出计算机所占据的方格数）。
- FB 第二个评价系数（棋盘上计算机一方总的石子数）。
- FC 第三个评价系数（有两个或者少于两个石子的格子数）
- FD 第四个评价系数（计算机一方最“拥挤”的格子的石子数）
- FE 棋盘上总的石子数。
- AS、BS、CS用来显示Wari棋盘的图示字符串。
- DS 32个空格的字符串。

③两点说明

a. 如果计算机的对手棋艺不高，则计算机无须用过多的时间考虑着法。插入下面的语句，可以使程序运行得更快一些。

```
550 GOTO 600  
665 E( A ) = V( K ) : GOTO 690
```

在标准的游戏模型中，当确定着法时，计算机需考察它自己要走的每一步和对方应着的各种可能性。经过上面的修改之后，计算机只考虑它自己的着法，而不考虑对手各种可能的着法。这样作，使得计算机的着法比较粗糙，不过走出一步也需要几秒钟的考虑时间。

b. 修改语句行690，可以看到计算机“思考”（THINING）和应着的规则。

```
690 PRINT E( A ) ; : NEXT: M = 0 : FA = - A : FOR J = 1 TO 6
```

经过这样修改，程序将在“THINKING”之后依次显示每个格的评价参数（从第七格开始），然后根据评价参数确定在每一轮中要走的棋步——它将选择6个格子中评价参数最大的那个格子。

如果你的应着很好，使计算机输掉了宝石，它就显示一个负数。-50表明是非法棋步（即相应的格子中的石子数为0）；大于1的正数代表由计算机吃掉的石子数。

源程序

```

100 REM :WARI
120 CLEAR 180:DEFINT A-D,J-T,W-Y:J=1:K=1:Q=14:P=13:F=5
0:D=12
130 DIM T(Q),Y(Q),W(Q),V(6),E(6),B(Q)
140 RANDOM:ZB=RND(0):ZB=ZB/Q:ZA=.25+ZB:ZB=.25-ZB:GOSUB
750
150 FOR J=1 TO D:B(J)=4:NEXT :B(P)=0:B(Q)=0:MN=0:GOSUB
1200:GOSUB 900
160 GOSUB 990:INPUT"WANT GO FIRST";R$
170 GOSUB 990:PRINT D$:R$=LEFT$(R$,1):IF R$="Y" THEN 2
50
180 IF R$<>"N" THEN 160
190 GOSUB 1050 :PRINT D$:D$:D$:GOSUB 1050:PRINT"THINK
ING":GOSUB 510
195 IF M<1 THEN 2000
200 GOSUB 1050 :PRINTD$:GOSUB 1050:PRINT"MY MOVE IS";
M
210 FOR J=0 TO Q :T(J)=B(J):NEXT :GOSUB 350
220 FOR J=1 TO Q :B(J)=T(J):NEXT :GOSUB 900
230 IF B(Q)<24 THEN 250
240 GOSUB 1050 :PRINT"I WIN!";D$:GOTO810
250 GOSUB 990:PRINTD$:D$:GOSUB 990:INPUT" YOUR MOVE
";R$
260 M=INT(VAL(R$)):IF M>6 OR M<1 THEN 330
270 FOR J=1 TO Q : T(J)=B(J):NEXT
280 GOSUB 350 :IF M<0 THEN 330
290 FOR J=1 TO Q :B(J)=T(J):NEXT
300 MN=MN+1:GOSUB 900
310 IF B(P)<24 THEN 190
320 GOSUB 1050 :PRINT "YOU WIN!";D$:GOTO810
330 PRINT@852," ILLEGAL " :FOR J=1 TO 3000:NEXT:GOTO25
0
350 IF T(M)<0 THEN M=-1 :RETURN
360 R$="H":IF M>6 THEN R$="C":GOTO 380
370 FOR J=1 TO Q:Y(J)=T(J):NEXT:GOTO 400
380 FOR J=1 TO 6 :Y(J)=T(J+6):Y(J+6)=T(J):NEXT
390 Y(P)=T(Q):Y(Q)=T(P):M=M-6
400 C=M:N=Y(C):FOR J=1 TO N:C=C+1
410 IF C=P THEN C=1
420 IF C=M THEN C=C+1 : GOTO 410
430 Y(C)=Y(C)+1 :NEXT :Y(M)=0 :L=C
440 IF L<7 OR Y(L)>3 OR Y(L)<2 THEN 460
450 Y(P)=Y(P)+Y(L):Y(L)=0 : L=L-1:GOTO 440
460 S=0: FOR J=7 TO D:S=S+Y(J):NEXT
470 IF S=0 THEN M=-2 :RETURN
480 IF R$="H" THEN FOR J=1 TO Q:T(J)=Y(J):NEXT:RETURN
490 FOR J=1 TO 6: T(J)=Y(J+6):T(J+6)=Y(J):NEXT
500 T(Q)=Y(P):T(P)=Y(Q):RETURN
510 FOR A=1 TO 6 :M=A+6 : IF B(M)=0 THEN E(A)=-F:GOTO
690
690
530 FOR J=1 TO Q :T(J)=B(J):NEXT : GOSUB 350

```

```

540 IF M<0 THEN E(A)=-F:GOTO 690
550 IF T(Q)>23 THEN M=A+6 :RETURN
560 FOR J=1 TO Q:W(J)=T(J):NEXT :FOR K=1 TO 6
570 IF T(K)=0 THEN V(K)=F:GOTO 670
580 FOR J=1 TO Q:T(J)=W(J):NEXT :M=K:GOSUB 350
590 IF M<0 THEN V(K)=F : GOTO 670
600 FA=0 :FB=0:FC=0:FD=0:FOR J=7 TO D
610 FB=FB+T(J): IF T(J)>0 THEN FA=FA+1
620 IF T(J)<3 THEN FC=FC+1
630 IF T(J)>FD THEN FD=T(J)
640 NEXT :FE=FB:FOR J=1 TO 6 :FE=FE+T(J):NEXT
650 FA=FA/6:FD=1-FD/FB:FC=1-FC/6:FB=FB/FE
660 V(K)=ZA*(FA+FB)+ZB*(FC+FD)+T(Q)+B(P)-B(Q)-T(P)
670 NEXT :E(A)=F:FOR J=1 TO 6 : IF V(J)<E(A) THEN E(A)
=V(J)
680 NEXT
690 PRINT". " :NEXT :M=0: FA=-F:FOR J=1 TO 6
700 IF E(J)>FA THEN F(A)=E(J):M=J+6
710 NEXT :RETURN
750 A$="":FOR J=1 TO 36 :A$=A$+CHR$(176):NEXT
760 B$=CHR$(170):FOR J=1 TO 6 :B$=B$+" "+CHR$(170)
:NEXT
770 C$=CHR$(170):D$=CHR$(176)+CHR$(176): FOR J=1 TO 6
780 C$=C$+D$+D$+CHR$(176)+CHR$(186)
790 NEXT :D$="":FOR J=1 TO 5 :D$=D$+D$:NEXT :RETURN
810 PRINT"GOOD GAME!"
840 INPUT"WANT TO PLAY AGAIN":R$
850 R$=LEFT$(R$,1):IF R$="Y" THEN 120
860 IF R$<>"N" THEN 840
870 PRINT:PRINT"SEE YOU LATER"
880 PRINT:END
900 FOR J=0 TO 5
910 PRINT@(322+J*6),B(12-J);
920 IF B(12-J)=0 THEN GOSUB 1100
930 NEXT :PRINT@367,"COMPUTER";B(Q);
940 FOR J=0 TO 5
950 PRINT@(514+J*6),B(J+1);:IF B(J+1)=0 THEN GOSUB 110
0
960 NEXT :PRINT@564,"YOU";B(P);:RETURN
990 PRINT@832," " :RETURN
1050 PRINT@896," " :RETURN
1100 PRINTCHR$(24);" " :RETURN
1200 CLS:PRINTTAB(30)"W A R I"
1220 PRINTTAB(14)"COMPUTER"
1230 FOR J=0 TO 5 :PRINTTAB(6*J+2)12-J;:NEXT
1240 PRINT@192,CHR$(160);:PRINTA$;TAB(50)"CAPTURED"
1250 FOR J=1 TO 2 : PRINTB$:PRINTB$:PRINT C$:NEXT
1260 FOR J=0 TO 5 :PRINTTAB(6*J+2);J+1;:NEXT
1270 PRINT:PRINTTAB(17)"YOU":RETURN
2000 "NO LEAGAL MOVES.":GOTO 840

```


六十八 打靶游戏

张仲文

这里介绍一个打靶游戏程序。它可在TRS-80微型计算机LEVEL II BASIC或磁盘BASIC下运行，操作简单，玩起来颇有兴趣。可以锻炼游戏者的思维反应及判断能力。

该程序运行后，显示屏上将显示出有趣的画面：由带花边的大方框围成一个靶场；有一小方框（称为靶子）在靶场内随机变换位置出现；不断延伸移动的粗线条（追击轨迹）在追逐靶子。

游戏者可用计算机键盘上的 \uparrow \downarrow \leftarrow \rightarrow 四个键来控制改变追击的方向。显示屏将出现迂迴曲折的追击轨迹。当追击轨迹与靶子外框接触时，计算机将判断为命中靶子。此时靶子闪烁，在靶子方框中央显示出命中成绩。同时计算出命中率（RATING），累加成绩（SCORE），追击时间（TIME），显示在靶场方框的左边。当追击轨迹碰触靶场外框或游戏者企图沿与追击方向相反的方向追击时，计算机将判为‘犯规’，结束此次操作并将总成绩以最高成绩（HIGH SCORE）的形式显示在靶场边框的右边。

游戏者须集中精力，快速灵活地操作 \uparrow \downarrow \leftarrow \rightarrow 四个键，改变追击方向，才能击中靶子，取得好成绩。有条件的话，读者不妨玩玩，看你的最高成绩是多少？

该程序虽不长但层次清楚，很简炼。主要由七个功能部分组成。

1. 程序参数初始化子程序

由标号10的各程序语句组成。对各程序参数、变量予以定义并置成初始值。

2. 操作提示说明子程序

由标号20—48的各程序语句组成。进入该子程序，显示屏上显示出操作提示说明信息：命中成绩取决于中靶的数目；追击轨迹触碰靶场边框或沿追击方向的反向运动为犯规；命中率是中靶次数与追击时间的比值；控制追击方向的命令是： \uparrow 向上， \downarrow 向下， \leftarrow 向左， \rightarrow 向右。

3. 靶场方框生成子程序

由标号8000—8070的各程序语句组成。其中8000—8060的各程序语句生成靶场方框，置于显示屏中间位置。标号8070的各程序语句，用于在靶场方框的左侧显示追击时间（TIME）、成绩（SCORE）、命中率（RATING）及在右侧显示最高分数等提示信息。

4. 靶子生成子程序

由标号3000—3030各程序语句组成。靶子（小方框）主要由3030的各程序语句实现。靶子在靶场内的随机位置是由标号3025的程序语句确定的。为了确保靶子出现时不至于落在靶场边框及追击轨迹上，使用了3026—3028各程序语句。从TRS-80机的内存图可以看出，它的显示存贮区是由15360开始的。显示屏上的1024个字符位置与15360开始的1024个存贮单元一一对应。如果显示屏上有显示的信息，则对应的显示存贮单元必定不是空格（ASCII代码为32或128）信息。因此可利用判断 $15360 + F4$ 、 $15361 + F4$ 、 $15362 + F4$ 三个单元是否为“空格”的方式，来确定在 $F4 - 1$ 、 $F4 + 1$ 各显示位置上是否可以显示靶子。标号3026—3028的各程序语句恰好完成了这个判断工作。

在操作中，我们会发现靶子始终不会落在靶场方框之外。这是由程序语句 $F4 = RND$

(F7) *F9 + RND (FZ) 来保证的。

靶子在靶场内的随机出现，有两种情况：中靶后，清除旧靶，显现新靶；追靶中，为不让游戏者容易地命中靶子，要按随机的时间变换靶的位置。追靶中变换靶位的随机时间是由随机变量IB控制的，旧靶的清除是由3010中的 PRINT @ F4-IP, “┌┌┌”来完成的。中靶时旧靶的清除是由2540中相应语句完成的。

5. 追击轨迹控制子程序

由标号1000—1060中的各语句组成。追击方向的改变如前所述用 \uparrow \downarrow \Leftarrow \Rightarrow 四个键控制。标号1010、1020、1030、1040的各语句用来判别键入的是哪个控制键，从而为SET (X, Y) 语句设不同的X、Y增量 X_1 、 Y_1 ，进而达到控制追击方向的目的。不同键的 X_1 、 Y_1 增量如下：

$X_1 = -1$	$Y_1 = 0$	(\Leftarrow 键时)
$X_1 = 1$	$Y_1 = 0$	(\Rightarrow 键时)
$X_1 = 0$	$Y_1 = -1$	(\uparrow 键时)
$X_1 = 0$	$Y_1 = 1$	(\downarrow 键时)

6. 追击轨迹处理及命中处理子程序

由标号2000—2540中的各程序语句组成。这个子程序是整个程序的核心。

(1) 追击轨迹的延伸和显示 由标号2000中的 $X = X + X_1$ 、 $Y = Y + Y_1$ 及标号2050中的SET (X, Y) 语句完成。每当程序进入该子程序时，就计算出新的X、Y坐标值并在显示屏的相应位置上显示出轨迹图符。由于X、Y坐标值是+1递增(减)的，因此在显示屏上显示出了延伸的追击轨迹。

(2) 追击轨迹的尾部抹除 是由下述各语句共同完成的：

```
2000 F3 = F3 + IP; IF F3 > IT LET F3 = IZ
2020 IF F0 < IZ; LET F0 = F0 + IP; GOTO 2040
2030 F2 = F2 + IP; IF F2 > IT LET F2 = IZ
2040 RESET (I (F2, IZ), I (F2, IP)); I (F3, IZ) = X;
      IF POINT (X, Y) THEN 2500
2050 I (F3, IP) = Y; RETURN
2500 IF P-1 = F4 OR P = F4 OR P+1 = F4 THEN 2520 ELSE 2600
2520 F0 = F0 - Q * 2
2540 GOTO 2050
```

从上述可以看出，追击轨迹的当前X、Y坐标值分别由 I (F3, IZ), I (F3, IP)记录的。而要抹除的轨迹坐标是由 I (F2, IZ), I (F2, IP)记录的。由于使用了同一个多维变量I，因而保证了抹除的轨迹一定是前进中划过的轨迹。在2520中Q为记录每次中靶成绩的变量，F0是随Q变化而变化的。可想而知，击中次数越多，成绩越高，执行2030的机会越少，F2增长的越慢，那么抹除的轨迹位置落后于前进轨迹的距离越远。因而画面上随着命中成绩的增加，追击轨迹呈现出迂迴曲折的现象。

(3) 命中及犯规判断 靶场外框、靶子小方框是在一系列确定的X、Y坐标下，置入了图形字符信息用SET语句显示出来的。因而可用判断要置入轨迹的X、Y坐标位置中是否有非空格信息的方法，确定置入轨迹信息并判断是否犯规、击中目标。判断过程是这样的：

```
2040 IF POINT (X, Y) THEN 2500
```

```
2050 SET (X, Y); RETURN
```

```
2500 P = INT (Y / 3) * 64 + INT (X / 2); IF P - 1 = F4 OR P = F4 OR P + 1 = F4  
THEN 2520 ELSE 2600
```

首先判断X、Y坐标点是否已有非空格信息，若无信息，则置轨迹图符，若有，进一步判断是否中靶，若不是中靶则为犯规。

(4) 中靶处理 主要完成：计算当次中靶成绩并显示在靶框内；靶框闪烁并发出声响；清除旧靶显示新靶；记录追击时间、累加成绩及命中率。

当次中靶成绩用Q表示。Q = RND(9)，Q是随机数值，因而每次命中成绩不尽相同。命中后是利用标号2530中PRINT@F4+IN, STR\$(Q)语句将成绩置于靶框中的。靶框的闪烁是由标号2530的循环语句实现的。其中PRINT@F4+IN, STR\$(Q);“ ”语句中的两个空格是用于清除靶框的右边框；此后转而执行800语句，使之保持一定时间，回转执行后续的语句，重显靶子。在2530循环语句的循环周期内，周而复始，便形成了靶子的闪烁。

追击时间TIME是用IQ来记录的。命中成绩SCORE是用IS来记录的。命中率 = $\text{INT}(I_8 * 1000 / IQ + F_3)$ 这里I₈是命中次数。IQ = IQ + F₃。

$$\text{亦即命中率} = \frac{\text{命中次数}(I_8) \times 1000}{\text{追击时间}}$$

7. 犯规处理子程序

由标号2600—2640各程序语句组成。判定犯规操作后进入该子程序。其功能是：清除屏幕，以每行32个字符的大字符形式显示出：此次操作的总成绩、命中率、最终成绩；提示按ENTER键可重新开始游戏。

该游戏只适于一人操作或多人串行操作，以命中成绩评定各人名次：由于该程序中击靶命中的分数是用随机数产生的，有时会出现很容易击中靶时，取得的成绩较多，而追击路程较长，却得很少成绩的现象。

注：变量说明

1. X、Y——当前轨迹图符坐标。

2. X₁、Y₁——改变追击轨迹方向时X、Y坐标的增量。

3. F₃：变化范围0—550。

F₂：变化范围0—550。

F₂、F₃均为I变量的下标。其中用I(F₂, IZ)，I(F₂, IP)记录抹除追击轨迹的坐标。用I(F₃, IZ)，I(F₃, IP)记录当前追击轨迹的坐标。

4. Q：为中靶成绩。其值由随机数RND(9)产生。

5. I₈：中靶次数计数器，每中一次加1。

6. I₅：中靶成绩累加器，记录中靶成绩之和 I₅ = I₅ + Q。

7. IQ + F₃：追击时间计数器。

8. INT(I₈ * 1000 / IQ + F₃)：用来计算命中率。

9. F₄：决定靶子在靶场中位置的基础变量。

10. IB：决定改变靶子位置的初始变量。每进入一次靶子生成程序减1，当IB < IZ并且RND(0) ≥ 0.9时，将抹除旧靶、置新靶。

源程序

```

1  '      -- H U S T L E --
2  'WRITTEN BY PETER TREFONAS, U.N.O. (12/6/78)
3  GOTO20
10  RANDOM:CLS:DEFINT I,F,X,Y:A$="":IZ=0:IP=1:IN=-1:X=6
2:Y=24:X1=1:Y1=0:F3=0:F2=0:F0=-3:F4=0:IS=0:Q=0:I0=0:IB
=-1:P=0:P1=.9:IT=550:IQ=0:F7=12:F8=79:FZ=32:FY=15360:F
X=15359:FW=15361:IB=0:FV=128:F9=64:RETURN
20  GOSUB10:DIM I(IT,1):INPUT"DO YOU WANT DIRECTIONS";A
$:IFA$="NO"ORA$="N"GOTO190
23  PRINT:PRINT:PRINTTAB(30);"H U S T L E":PRINT:PRINT
26  PRINT"THE OBJECT IS TO SCORE AS MANY POINTS AS POSS
IBLE BY
29  PRINT"REACHING THE TARGET WITHOUT TOUCHING A LINE O
R REVERSING
30  PRINT"APON YOURSELF. YOUR RATING REFERS TO THE NUMB
ER OF TARGETS
31  PRINT"THAT YOU GOTTEN DIVIDED BY THE TIME.
32  PRINT"COMMANDS ARE:
35  PRINTCHR$(91),"MOVE UP
38  PRINTCHR$(92),"MOVE DOWN
41  PRINTCHR$(93),"MOVE LEFT
44  PRINTCHR$(94),"MOVE RIGHT
48  FORI=1TO4000:NEXT
190  GOSUB8000:'DRAW SCREEN
200  GOSUB1000:GOSUB2000:GOSUB3000:GOTO200
800  FORI0=1TO4:NEXT:RETURN
999  'NEXT SBR. PROCESSES COMMANDS
1000 A$=INKEY$:IFA$=""RETURN
1010 IFASC(A$)=8LETX1=IN:Y1=IZ:RETURN
1020 IFASC(A$)=9LETX1=IP:Y1=IZ:RETURN
1030 IFA$="["LETX1=IZ:Y1=IN:RETURN
1040 IFASC(A$)=10LETX1=IZ:Y1=IP:RETURN
1060 RETURN
1999 'NEXT SBR. RESETS POINTERS AND MOVES DOT
2000 X=X+X1:Y=Y+Y1:F3=F3+IP:IFF3>ITLETF3=IZ:IQ=IQ+IT
2020 IFF0<IZLETF0=F0+IP:GOTO2040
2030 F2=F2+IP:IFF2>ITLETF2=IZ
2040 RESET(I(F2,IZ),I(F2,IP)):I(F3,IZ)=X:IFPOINT(X,Y)T
HEN2500
2050 SET(X,Y):I(F3,IP)=Y:RETURN
2500 P=INT(Y/3)*64+INT(X/2):IFF-1=F4ORP=F4ORP+1=F4THEN
2520:ELSE2600
2520 Q=RND(9):IS=IS+Q:F0=F0-Q*2:IB=IB+1
2530 FORI=IPT08:OUT255,4:PRINT@F4+IN,STR$(Q);" ";:GOSU
B800:OUT255,3:PRINT@F4+IN,CHR$(183);CHR$(179);CHR$(187
);:NEXTI
2540 PRINT@F4-IP," ";:F4=IP:IB=IN:PRINT@197,IQ+F3;:P
RINT@326,IS;:PRINT@515,INT(18*1000/(IQ+F3));:GOTO2050
2599 'YOU LOSE HERE
2600 CLS:PRINTCHR$(23);:PRINT:PRINT:PRINT
2610 PRINTTAB(10);"GAME OVER":PRINT:PRINT:PRINTTAB(5);
"FINAL SCORE=";IS:PRINT:PRINTTAB(5);"RATING=";IB*1000/

```

```

(IQ+F3):FORI=1TO100:NEXT
2620 FORI=IZTOTI:I(I,IZ)=IZ:I(I,IP)=IZ:NEXT
2630 IFIS>HILETHI=IS
2635 INPUT"HIT ENTER FOR NEW GAME";A$
2640 GOSUB10:GOTO190
2999 'THIS SUBR PUTS BOXES ON SCREEN
3000 IFIB>IZLETIB=IB-IP:RETURN
3010 IFIB=IZPRINT@F4-IP," ";:IB=IN
3020 IFRND(IZ)<P1RETURN
3025 F4=RND(F7)*F9+F8+RND(FZ)
3026 IFPEEK(FX+F4)<>FZANDPEEK(FX+F4)<>FVTHEN3025
3027 IFPEEK(FY+F4)<>FZANDPEEK(FY+F4)<>FVTHEN3025
3028 IFPEEK(FW+F4)<>FZANDPEEK(FW+F4)<>FVTHEN3025
3030 PRINT@F4-1,CHR$(183);:PRINT@F4,CHR$(179);:PRINT@F
4+1,CHR$(187);:IB=RND(65)+25:RETURN
7999 'FOLLOWING SUBR. DRAWS SCREEN
8000 CLS:FORI=1TO46:SET(22,I):SET(103,I):NEXT
8010 FORI=3TO44:SET(26,I):SET(99,I):NEXT
8020 FORI=23TO102:SET(I,IP):SET(I,46):NEXT
8030 A$=CHR$(183):FORI=76TO114:PRINT@I,A$;:NEXT
8040 FORI=908TO946:PRINT@I,A$;:NEXT
8050 FORI=2TO13:PRINT@64*I+12,A$;:PRINT@64*I+50,CHR$(1
87);:NEXT
8060 SET(101,4):SET(101,43)
8070 PRINT@192,"TIME:";:PRINT@320,"SCORE:";:PRINT@448,
" RATING";:PRINT@181,"HIGH SCORE";:PRINT@247,HI;:RETU
RN

```

六十九、射击游戏

张宝玺

这是一个很有趣味的游戏。游戏时，屏幕上方用锯齿形方框镶嵌着“靶场”几个字。两边信号灯闪烁，下面一排排“星星”在天空移动；一个个“小丑”在画面中央跳跃；移动的“烟斗”好象要送到“小丑”咀边，叫他去吸烟；一对对“鸭子”在水中游动；天空中不时飞过一靶子，等待你去射击。整个画面显得生动活泼。

你一共有20发炮弹，可以射击飞速前进中的靶子；或者射击“星星”、“小丑”、“烟斗”，“鸭子”。根据射击各种目标的难易，分别给100分、50分、25分、15分、10分。当然，最好射击飞来的靶子，因为击中它可得100分。每次击中目标的得分，累计在画面的右下方的得分栏里，发一发炮弹，总的炮弹数减1。炮弹数是采用倒计时办法，剩多少发炮弹，不断显示在画面左下方的炮弹数目栏里。计算机给你提供一个“十字”瞄准器。按下键盘上↑、↓、←、→、键，可使瞄准器上、下、左、右移动。连续按下某一键，可使瞄准器往某方向不停地移动。要击中移动中的目标，必须事先把瞄准器对准目标，然后按空格键发炮。如果击中目标，这个目标消失，同时，这个目标中的分值被加到总分中去。发炮时，在画面上部中心处显示“BANG!!!”字样，同时在磁带机中发出“呼”的声音。如果击中目标，磁带机中除发出“呼”的声音外，还奏一段乐曲，以示祝贺。飞行靶子一出现，磁带机中即发出隆隆声。好象飞机从天空飞过。整个情况，仿佛真是在炮声隆隆的战场，由你驾驶一门火炮去轰击敌人的飞机或地面目标。飞靶是随机的，它可能从画面的不同地方出现。你要不断地注意移动瞄准器

去迎击它。击中各种目标的得分分别显示在画面下方。

1. 程序设计思想与说明

①设计思想

程序中，固定图案的设计是采用图示块拼接的办法，即把显示的图形分成很多个小图示块，把这些小图示块相加后，赋值给字符串变量。打印字符串变量，图形即显示在显示器上。比较复杂一点的图案设计大都采用此法，这里不再详细说明。

移动图案一共有 4 个。包括星星、小丑、烟斗、鸭子。我们以“*”号的移动为例加以说明，其余大同小异。

程序中， $DS(0) = \text{"*~~~~~"}$ 。其中 $DS(0)$ 是用定义类型语句定义的字符串数组元素。 $DS(1)$ 等于 9 个 $DS(0)$ 相加，一共 72 个字符的串。在程序第 220 语句中，当第 1 次执行此语句时， $XS = 1$ ， $MID\$(DS(0), XS, 1)$ 得到 $DS(0)$ 中第 1 个字符“*”，函数 $RIGHT\$(DS(1), 63)$ 得到 $DS(1)$ 串中后 63 个字符，把上面两函数得到的字符串相加，即得到 64 个字符，形成 $DS(1)$ 串：“~~~~~*~~~~~*~~~~~*”，其中，最后一个“*”是由“ $MID\$(DS(0), XS, 1)$ ”函数得到的；最后一个“*”前面共 63 个字符是由“ $RIGHT\$(DS(1), 63)$ ”函数得到的。第二次执行此语句时， $XS = 2$ ， $MID\$(DS(0), XS, 1)$ 得到一个“~”加到 $DS(1)$ 串的末尾。实际上显示的“*”就前移一格。每执行一次 220 语句，“*”都前移一格，整个一行中的“*”就显得不断前移了。

其余移动图案的道理都是一样的，只是在向左移动时，在字符串的后边加上一个符号；向右移动时，在字符串的前面加上一个符号。

飞行中的靶子的移动是把靶子赋值给一个字符串变量 DT，即 $DT = \text{"---100|>"}$ 。程序反复运行一段子程序 570~630。每次运行都给 DT 的打印位置变量 YT 一个增量 YG，从而每次运行都在新的位置打印 DT，同时，通过 $PRINT @YT, CHR\$(30)$ 语句，抹除原来位置靶子图案。用 $RND(3)$ 函数控制靶子出现的位置。当 $Y0 = RND(3) = 1$ 时， $YT = 192$ ； $Y0 = 2$ 时， $YT = 320$ ； $Y0 = 3$ 时， $YT = 512$ 。由于 YT 值随机变化，故靶子出现位置也随机改变。靶子出现的时间由 570 行的语句： $IF RND(89) < > 50 THEN RETURN$ 决定，当 $RND(89) = 50$ 时，靶子出现。靶子出现后置变量 $YA = 1$ 。以后控制程序给靶子前进方向一个增量，使靶子不断前进。

瞄准器的显示是通过程序中 260-270 行的语句实现的。运行一次该段程序，根据瞄准器的坐标变量 Z 的值，在相应位置打印一个“+”。等到下次再运行该段程序时，首先看该坐标位置是否是“+”，如果是“+”，则打印该位置无“+”号时的图案；否则，打印“+”。这样，通过程序的一次次运行，图案不断交替变化，“+”就闪烁起来了。

瞄准器可在画面上移动。通过 ↑、↓、←、→ 四个键控制它的移动。每次移动都要改变它的坐标值。每右移一格，坐标加 1；左移一格，坐标减 1；上移一行，坐标减 64；下移一行坐标加 64。

为了使发射炮弹时，达到声情并茂的效果，除了按空格键时，在屏幕中央显示“BANG !!!” (呼) 的字样外，还搞了一个机器语言发声的子程序。接上录音机后，通过调用该子程序，可使录音机发出声音来。有关机器语言子程序不再介绍。

②语句说明

20~25 语句，程序参数初始化。

- 40~110语句, 打印游戏画面。
- 120 语句, 给变量赋初值。
- 140 语句, 打印靶场闪烁的信号灯。
- 150 语句, 打印游戏结束信息。
- 160~170语句, 打印活动的小丑图案。
- 180~190语句, 打印活动的鸭子图案。
- 200~210语句, 打印活动的烟斗图案。
- 220 语句, 打印活动的星星图案。
- 260~270语句, 子程序, 显示闪烁的瞄准器。
- 290~350语句, 子程序, 计算瞄准器的坐标值。
- 370~480语句, 子程序, 瞄准器在不同部位打靶时的处理。
- 370 语句, 打靶显示和发声; 计算瞄准器“十”在画面上的区间, 转相应的程序段处理。
- 380 语句, 中飞靶时的处理。
- 390 语句, 显示子弹数和累计中靶分数, 以及游戏结束后的显示。
- 400 语句, 瞄准器在“*”区打靶时的处理。
- 410 语句, 瞄准器在小丑区打靶时的处理。
- 420 语句, 瞄准器在烟斗区打靶时的处理。
- 430~450语句, 瞄准器在鸭子区打靶时的处理。
- 470~480语句, 瞄准器在画面右部打中鸭子和小丑时的处理。
- 500~550语句, 子程序, 打印游戏名称和说明。
- 570~630语句, 子程序, 打印活动的靶子。

③ 主要变量说明

- DC(0)、DC(1)、DC(2)、DC(3) 字符串数组元素, 用于显示小丑图案。
- DD(0)、DD(1)、DD(2)、DD(3) 字符串数组元素, 用于显示鸭子图案。
- DT 字符串变量, 用于显示飞行中的靶子。
- DP(0)、DP(1) 字符串数组元素, 用于显示烟斗图案。
- DS(0)、DS(1) 字符串数组元素, 用于显示“*”号。
- YB 炮弹数
- YS 中靶累计分数。
- YA 飞行靶子出现后的控制变量。当YA = 1时, 程序转去给出靶子前进方向增量, 使靶子前进。
- YO 飞靶出现位置的工作变量。
- YG 飞靶坐标增量。
- YT 飞靶从不同位置出现时的原始坐标。
- YL 飞靶在视频RAM中的地址。
- ZL 瞄准器在显示器上下行的行数。
- ZS 瞄准器的相对坐标值 (以191为0点)。
- QQ\$ 一共692个字符串, 用来确定机器语言子程序的入口地址。

磁带机使用说明

如果使用磁带机发声, 应先将磁带机和主机之间联接好。磁带机和主机之间只接一根主

机输出信号线，插到磁带机MIC插座中去。磁带和输出线不接，控制线可不接。注意使磁带机处于录音状态。磁带机中不要放录音带，以免录上声音。

源程序

```

20 CLEAR2000:DEFSTRD:DEFINTX-Z:RANDOM
23 QQ#=""
LETE>ORISA!85E>ORISA!85
  CINTINKEY$. ...."
25 I=VARPTR(QQ#):X=FEEK(I+1)+256*FEEK(I+2):FORY=XTOX+2
2:READZ:POKEY,Z:NEXTY:IFPEEK(16396)=201THENPOKE16526,P
EEK(I+1):POKE16527,FEEK(I+2)ELSECMD" T":DEFUSR0=X:POKE1
4308,0
30 GOSUB500
40 DC(0)="      "+CHR$(183)+CHR$(187)+CHR$(143)+CHR$(18
3)+CHR$(187)+"      ":DC(1)="      "+CHR$(191)+CHR$(178
)+CHR$(179)+CHR$(177)+CHR$(191)+"      ":DC(2)=DC(0)+D
C(0)+DC(0)+DC(0)+DC(0):DC(3)=DC(1)+DC(1)+DC(1)+DC(1)+D
C(1)
45 DATA205,127,10,14,140,69,62,1,211,255,16,254,69,62,
2,211,255,16,254,13,32,239,201
50 DD(0)="      "+CHR$(180)+"      "+CHR$(175)+CHR$(159)+CH
R$(140)+"      ":DD(1)="      "+CHR$(139)+STRING$(4,CHR$(
191))+CHR$(159)+"      ":DD(2)=DD(0)+DD(0)+DD(0)+DD(0)
+DD(0):DD(3)=DD(1)+DD(1)+DD(1)+DD(1)+DD(1):DT="--100--
>"
60 DP(0)=CHR$(171)+"      ":DP(1)=DP(0)+DP(0)+DP(0)+D
0(0)+DP(0)+DP(0)+DP(0)+DP(0):DS(0)="*      ":DS(1)=DS
(0)+DS(0)+DS(0)+DS(0)+DS(0)+DS(0)+DS(0)+DS(0)+DS(0)
70
80 CLS:D0=CHR$(188)+CHR$(191):D1=CHR$(140)+CHR$(143):D
2=D1+D1+D1+D1+D1+D1+D1+D1+D1:D3=CHR$(143)+CHR$(131):D4
=STRING$(2,CHR$(191))
90 PRINT@0,D0;D2;D0;D1;D2;D0;D2;D0;D4;"  ";STRING$(3,C
HR$(143));"      ";STRING$(3,CHR$(143));"  ";D4;"  S
HOOTING GALLERY  ";D4;"  ";STRING$(3,CHR$(143));"
      ";STRING$(3,CHR$(143));"  ";D4:FORX=128TO190STEP2:P
RINT@X,D3;:NEXTX
100 PRINT@832,STRING$(8,CHR$(191));STRING$(18,CHR$(143
));STRING$(15,CHR$(191));STRING$(15,CHR$(143));STRING$(
16,CHR$(191));" SHOTS LEFT: 20 ";STRING$(15,CHR$(19
1));" SCORE: 0      ";STRING$(8,CHR$(191));
110 PRINT@640,STRING$(64,CHR$(131));:PRINT@960,"<<<>> S
TARS = 50      PIPES = 25      DUCKS = 15      CLOWNS = 10 <
>>>";
120 YB=20:YS=0:XC=16:XD=16:XS=1:XP=1:D4=STRING$(2,CHR$(
143)):Z=351:YA=0
130 'START GAME
140 GOSUB260:FORX=0TO3:IFPOINT(20,3)THENPRINT@74,"<>";
:PRINT@116,D4;ELSEPRINT@74,D4;:PRINT@116,"<>";
150 IFYB=0THENIFPEEK(14656)=1THEN30ELSEPRINT@348,"
      ":FORYS=0TO20:NEXTYS:PRINT@348,"ENTER";
160 GOSUB290:IFX=0ORX=2THENDC(2)=MID$(DC(0),XC,1)+LEFT

```



```

$(DC(2),79):DC(3)=MID$(DC(1),XC,1)+LEFT$(DC(3),79):XC=
XC-1:IFXC=0THENXC=16
170 PRINT@384,RIGHT$(DC(2),64);RIGHT$(DC(3),64)::GOSUB
260
180 GOSUB290:DD(2)=MID$(DD(0),XD,1)+LEFT$(DD(2),79):DD
(3)=MID$(DD(1),XD,1)+LEFT$(DD(3),79):XD=XD-1:IFXD=0THE
NXD=16
190 PRINT@704,RIGHT$(DD(2),64);RIGHT$(DD(3),64)::GOSUB
260
200 GOSUB290:IFX<>3THENDP(1)=RIGHT$(DP(1),63)+MID$(DP(
0),XP,1):XP=XP+1:IFXP=9THENXP=1
210 PRINT@576,DP(1)::GOSUB260
220 GOSUB290:DS(1)=RIGHT$(DS(1),63)+MID$(DS(0),XS,1):X
S=XS+1:PRINT@256,DS(1)::IFXS=9THENXS=1
230 GOSUB260
240 NEXTX:GOTO140
250 'PRINT SIGHT AND BACKGROUND
260 GOSUB570:IFPEEK(15360+Z)<>43THENYL=PEEK(15360+Z):P
RINT@Z,"+":RETURN
270 PRINT@Z,CHR$(YL)::RETURN
280 'MOVE SIGHT AND SEE IF GUN FIRED
290 GOSUB260:IFPEEK(15360+Z)=43THENPRINT@Z,CHR$(YL);
300 DB=INKEY$:IFDB="" THEN370ELSEY=PEEK(14656):IFY>127
THENY=Y-128
310 IFY>32THENY=Y-64:ZL=ABS(Z/64):IFZ-ZL*64<63THENZ=Z+
1
320 IFY>16THENY=Y-32:ZL=ABS(Z/64):IFZ-ZL*64>0THENZ=Z-1
330 IFY>8THENY=Y-16:IFZ<768THENZ=Z+64
340 IFY>4THENIFZ>255THENZ=Z-64
350 RETURN
360 'SHOOT THE GUN
370 IFYB=0THENRETURNELSEPRINT@348,"BANG!!!":A=USR(20)
:ZL=ABS((Z-192)/64):ZS=Z-(ZL*64)-191:ONZLGO10400,380,4
10,410,380,420,390,430,440
380 IFYL<>32THENYS=YS+100:PRINT@192+ZL*64,CHR$(30)::FO
RN=0TO3:A=USR(30):A=USR(25):A=USR(20):A=USR(15):NEXT:Y
A=0
390 YB=YB-1:PRINT@917,YB;PRINT@945,YS;PRINT@348,CHR$
(30)::IFYB=0THENPRINT@331,"GAME OVER, HOLD ";CHR$(34);
"ENTER";CHR$(34);" DOWN TO PLAY AGAIN":RETURNELSERETU
RN
400 IFMID$(DS(1),ZS,1)="*"THENDS(1)=LEFT$(DS(1),ZS-1)+
" "+RIGHT$(DS(1),64-ZS):PRINT@256,DS(1)::YS=YS+50:FORN
=0TO2:A=USR(25):A=USR(35):NEXT::GOTO390ELSE390
410 IFMID$(DC(2),ZS+16,1)=" "THEN390ELSEYS=YS+10:FORN=
1TO2:A=USR(40):A=USR(60):NEXT:DT(0)=DC(2):DT(1)=DC(3):
GOSUB470:DC(2)=DT(0):DC(3)=DT(1):PRINT@384,RIGHT$(DC(2
),64);RIGHT$(DC(3),64)::GOTO390
420 IFMID$(DP(1),ZS,1)=" "THEN390ELSEDP(1)=LEFT$(DP(1)
,ZS-1)+" "+RIGHT$(DP(1),64-ZS):PRINT@576,DP(1)::YS=YS+
25:FORN=1TO2:A=USR(30):A=USR(50):NEXT:GOTO390
430 IFMID$(DD(2),ZS+16,1)=" "THEN390ELSE450
440 IFMID$(DD(3),ZS+16,1)=" "THEN390
450 YS=YS+15:FORN=1TO2:A=USR(40):A=USR(60):NEXT:DT(0)
=DD(2):DT(1)=DD(3):GOSUB470:DD(2)=DT(0):DD(3)=DT(1):PR

```

```

INT@704,RIGHT$(DD(2),64);RIGHT$(DD(3),64);:GOTO390
460 'BLANK OUT OBJECT IN STRING
470 IFZS>55THEN480ELSEFORXX=0TO1:DT(XX)=LEFT$(DT(XX),Z
S+9)+STRING$(13,"")+RIGHT$(DT(XX),58-ZS):NEXTXX:RETUR
N
480 FORXX=0TO1:DT(XX)=LEFT$(DT(XX),70)+STRING$(10," ")
:NEXTXX:RETURN
490 'SEE IF THEY WANT INSTRUCTIONS
500 CLS:DB="N":PRINT@24,"SHOOTING GALLERY":PRINT"
      BY FRANK B. ROWLETT, JR.":PRINT@212,"D
O YOU WANT INSTRUCTIONS":INPUTDB:IFASC(DB)<>89THENRET
URNELSEPRINT@64,CHR$(31);
510 'PRINT INSTRUCTIONS
520 PRINT@128,"YOU AIM AND SHOOT AT THE MOVING TARGETS
ON THE SCREEN. THE GUN SIGHT IS THE BLINKING +. YOU
CAN MOVE IT RIGHT WITH THE ";CHR$(94);" KEY, LEFT WIT
H THE ";CHR$(93);" KEY, UP WITH THE [ KEY AND DOWN WIT
H THE ";CHR$(92);" KEY."
530 PRINT"HOLDING THESE KEYS DOWN CONTINUES MOVING THE
SIGHT, AND ANY":PRINT"COMBINATION OF THESE KEYS CAN B
E PRESSED AT ONE TIME.":PRINT@512,"YOU CAN SHOOT THE G
UN BY PRESSING THE SPACE BAR--ONE SHOT FOR"
540 PRINT"EACH TIME YOU PRESS IT. IF THE SIGHT IS ON
A TARGET WHEN YOU":PRINT"SHOOT, THE TARGET DISAPPEARS
AND YOU GET THE SCORE VALUE OF":PRINT"THAT TARGET ADDE
D TO YOUR SCORE. YOU HAVE 20 SHOTS.":PRINT@832,"--GOOD
SHOOTING !"
550 PRINT@977,"(PRESS ";CHR$(34);"ENTER";CHR$(34);" T
O BEGIN GAME)":INPUTDB:RETURN
560 '
570 IFYA=1THEN590ELSEIFRND(89)<>50THENRETURN
580 YA=1:YO=RND(3):YG=-2
590 YG=YG+2:ONYOGOTO610,620
600 YT=512:GOTO630
610 YT=192:GOTO630
620 YT=320
630 PRINT@YT,CHR$(30);:A=USR(13):IFYG=58THENYA=0:RETUR
N:ELSEPRINT@YT+YG,DT::RETURN

```

七十、弹球游戏

张宝玺

这个弹球游戏程序一共可进行四种游戏。运行此程序，首先列出弹球游戏的目录。一共有五种选择，第五种选择是退出本程序，结束游戏。第1~4种选择分别为4种游戏的名字。当你键入1~4任意一个数时，就开始了相应的一种游戏。下面先介绍一下1~4种游戏的方法。

第一种游戏：非洲东部出猎

进入本游戏时，首先在显示器上显示出一个方框。在方框内，站着一个人，他的面前站着一头象。程序运行时，还会出现两条蛇。方框下边中间部分开一个大闸门。在大闸门的两侧开两个小门。一个球从右边框内侧向前运动，当球碰到任意边框，或者大象、猎人、两条蛇时都要改变方向（弹回）同时，在录音机中发出清脆悦耳的“呼”的声音。当球从闸

门跑出去，则发出“扑”的声音。显得很扫兴。当象被球打中或球离开方框时，象都要前进一步。当猎人被球打中，他的枪就要伸长一些。当球要从闸门跑出去，如你封死闸门，球将被弹回到猎人方向。游戏者的目的是通过控制“Z”“1”两个键，在闸门的左半部分和右半部分设置障碍物，封住闸门，使球弹回到猎人方向。並不断击中猎人，从而使猎人伸长枪。当枪伸长到一定程度时，就发子弹把大象打死。但是，如果在球跑出闸门时，你封不住闸门，或者大象被球击中，大象都要前进一步，同时在录音机中发出清脆的“呼”声，大象走、停20次后，就会不停下来，随着录音机中发出悦耳的伴奏曲一直走下去，直到把猎人踩死。在游戏的过程中，要不断注意两条移动的蛇，球可能突然击中它而反射回来，以至来不及封闸门，就跑出闸门之外。每次跑出闸门后，你可按空格键，使游戏继续下去。

运行本程序，首先显示游戏目录，接着显示：你要进行困难的表演吗？你若选择比较困难的一种，可键入Y，此时，球只允许跑出六次，否则键入N（或按回车）你可玩八次。

当猎人把大象打死时，录音机中发出蜂鸣声，同时显示祝贺信息：“你已经捕获了美好的胜利品”。当大象把猎人踩死时，显示：“你被象踩中，祝你下次旅行愉快”。两种情况都使游戏结束，转去显示游戏目录。这时你可再键入你要进行的游戏代号，重新开始游戏。

第二种游戏：随机目标弹球游戏

在此游戏中，屏幕也是显示如前的方框。方框下面也是开一个闸门和两个小门。不过，此时在框内显示的是九个随机移动的目标，球也是在方框内弹来弹去，当球碰到边框、目标或从闸门跑出去，都发出不同频率的蜂鸣声。球不断击中目标而得分。总分数不断显示在方框左下方，而玩的次数不断显示在框右下方。整个分数是由每次击中靶子的环数决定的。每次击中靶子后，靶子将随机地跑到游戏区的另一个位置。无论何时，球离开游戏区时，目标也移动，游戏者的目的也是通过控制“Z”“1”二键，设置障碍物，不让球出去。

进入本游戏时，首先显示游戏说明，接着提示：游戏最多10次，请你输入游戏次数。你输入数据后，游戏即开始。球不断击中目标得分；由于判断错误，封不住闸门，球不断跑出闸门，当球跑出闸门次数大于键入的游戏次数时，游戏结束。程序自动累计、显示得分总数。并根据你的得分给出三种评价：大于300分，打印“评价：冠军水平”；大于200分；打印“评价：比较好”；小于200分，打印“评价：严格地说，你差一些”。同时程序转到显示游戏开始时的目录。

第三种游戏：固定目标弹球游戏

此种游戏和第二种游戏无论从方法，游戏规程和输出结果上都大致一样。只是目标不再是随机移动的靶子，而是固定的。这里不在赘述。

第四种游戏：歼灭游戏

进入本游戏，也是显示如上的方框，方框的下部开一个大闸门，两边不开小门，在方框内有4排符号排列在游戏区上部，每当球“消灭”一个符号，录音机中发生蜂鸣声，分数增加30分。当每排只有5个符号时，下次再打中该行符号，该行符号将全部消失。游戏者要不断封住屏幕底端的闸门使球保持在游戏中。随着游戏进行，符号将逐渐移向闸门，这样要求游戏的技术更高。如果玩得好，可以把屏幕所有符号“消灭掉”。

进入本游戏时，首先显示游戏说明，接着也是提示你：游戏最多10次，请你输入游戏次数。输入游戏次数后，游戏即开始。球不断击中符号，並且把符号消灭掉而得分。当你游戏次数超过预设的游戏次数时，游戏结束，此时程序依据你的得分给出评价：当得分大于900时打出“评价：冠军水平”。当得分大于550时打出“评价：比较好”。当得分小于550时打出

评价。“严格地说，你差一些”；当全部符号均被消灭时，显示“这是理想游戏”。

程序说明

- 5 ~ ^ 变大字体打印。
- 20 延迟。
- 40 定义字符串空间。
- 50 给字符串变量赋值。
- 55 转子程序9000。
- 100 ~110 打印随机数发生器；定义整型变量；打印“弹球游戏目录”。
- 120 输入游戏项目；给球坐标 X 和位移坐标 IX, IY 赋初值。
- 130 输入游戏难易要求。
- 141 变量赋初值。
- 150 ~155 给出游戏难度参数 U、W。
- 160 给 PS、A 赋初值。
- 170 判断转向。
- 180 ~240 给工作变量赋值；打印游戏画面；对于第一种游戏，打印人，同时转到相应的子程序打印大象和蛇。
- 245 判断转向。
- 250 对于第三种游戏，转去打印 8 个障碍物。
- 260 对 1 ~ 3 种游戏在下边框开一个闸门，在边框左下部打印分数提示符和分数。
- 265 对 1 ~ 3 种游戏，在闸门两侧开两个小便门。
- 267 对第 4 种游戏，在下边框开一大闸门。
- 270 在画面右下部给出游戏次数；在边框内给出球体。
- 340 机器语言发声子程序。
- 500 ~1120 游戏主控程序。
- 2020 ~2095 对第二种游戏，在画面上打印 8 个不同形状的障碍物，以及当球撞到障碍物时该物闪现出来的图案。
- 2130 ~2150 球出界后的处理。
- 4000 ~4050 2 ~ 4 种游戏结束后的处理（给出评价）。
- 5000 ~5090 对第一种游戏赋初值。
- 5100 ~5160 打印非洲出猎游戏的说明。
- 5170 ~5190 赋初值。
- 5300 ~5310 大象成像程序和前进驱动程序。
- 5350 ~5360 第 1 种游戏结束后的处理。
- 5530 ~5830 球碰到人、蛇、边框的处理。
- 5600 ~5620 第 1 种游戏（枪伸出来）人打死象的处理。
- 5700 ~5710 球碰到大象后的处理。
- 5800 ~5830 第 1 种游戏打印蛇。
- 6000 ~6060 第 4 种游戏说明。
- 6090 ~6120 第 4 种游戏的数据准备。
- 6125 打印游戏控制说明。

6130 输入本游戏一局的次数。
 6135 给 E、D 赋初值转 180。
 6140 第 4 种游戏在方框内打印 4 排障碍物。
 6200~6300 第 4 种游戏球运动和撞到障碍物等情况的转向判断处理程序。
 6400~6420 第 4 种游戏, 游戏胜利后的显示。
 7000~7070 打印第 3 种游戏说明。
 7100~7140 定义数组, 读数据到 PS 数组。
 7160~7170 第 2 种游戏, 打印 9 个障碍物。
 7300~7310 第 2 种游戏, 球碰到障碍物后的处理。
 7400~7410 第 2 种游戏, 球出界后移动 9 个障碍中的几个障碍物的位置。
 8000~8060 第 3 种游戏说明。
 8070 给 D、E 赋值, 转游戏开始程序。
 9000~9040 读使录音机发声的机器语言子程序, 定义入口地址。

主要变量说明

X——球的横向坐标。
 Y——球的纵向坐标, 每下一格增量为 64。
 ZX——X 方向坐标增量。
 ZY——Y 方向坐标增量。
 U = 2~4 游戏难度参数, U = 4 时难。
 W = 5~3 游戏难度参数, W = 3 时难。
 A——显示器入口地址。
 Q——游戏项目选择。
 BA——每种游戏预定玩的次数。
 CT——已玩的次数。
 PS——封闸门符号。
 E 1 \$、E 2 \$、E 3 \$ 非洲大象成像图示块。
 M 1 \$、M 2 \$、M 3 \$ 猎人成像图示块。
 X、Y、L、R、S、LZ、RZ; D, PR, PL, PD, M, Z, XE
 YE, XZ S1, S2, GX, GN, QA, QB, BA, E, —工作变量。
 QQ\$ 654 个空格字符串, 用来定义机器语言子程序入口地址。

源程序

```

5 CLS:PRINTCHR$(23):PRINT:PRINT
10 PRINT" MICRO-80 PINBALL MACHINE
  COPYRIGHT 1980 BY G. R. JENSEN,
  ALL RIGHTS RESERVED"
20 FORN=1TO2000:NEXT
40 CLEAR400
50 QQ$=""
LETE>ORISA!85E>ORISA!85
  CINTINKEY$....."
55 GOSUB9000
100 RANDOM:DEFINTA-Z:CLS:PRINTTAB(4);CHR$(23);"MICRO-8
0 PINBALL MACHINE":PRINT:PRINT

```

```

110 PRINT" 1 - AFRICAN SAFARI":PRINT" 2 - RANDOM TAR
GETS PINBALL":PRINT" 3 - NORMAL PINBALL":PRINT" 4 -
WIPEOUT":PRINT" 5 - STOP":PRINT
120 INPUT" ENTER GAME";Q:IFQ<10RQ>STHEN120ELSEP$="":X
=0:IX=0:IY=0:IFQ=5THENCLS:END
130 INPUT" WANT DIFFICULT VERSION? Y/N";T$:IFLEFT$(T$
,1)<>"Y"THEN150
141 Y=0:L=0:R=0:S=0:LZ=0:RZ=0:W=0:A=0:D=0:PR=0:PL=0:M
=0:E=0
150 G=0:IFLEFT$(T$,1)="Y"THENU=4:W=3:ELSEU=2:W=5
155 IFG=1THENU=2
160 P$=STRING$(14-U,61):A=15360
170 ONG GOTO5000,7000,8000,6000
180 X=59:Y=768:Z=64:IX=-RND(3):IY=-64:S=32:M=16258:PD=
62:LZ=916:RZ=926+U
190 L=0:R=0:R$=STRING$(14-U,32):PL=-1:PR=-1:IFG=4 PD=9
1
195 IFG=4 R$=STRING$(22-U,5):P$=STRING$(22-U,91):LZ=90
6:RZ=928+U
200 T$=STRING$(4,191)+STRING$(56,32)+STRING$(4,191)
210 CLS:FORI=0TO63:POKEA+I,191:POKEA+I+896,191:NEXT
220 FORI=64TO832STEP64:PRINT@I,T$::NEXT:T$=""
230 IF G=1 THEN 245 ELSE PRINT@243,M1$::PRINT@305,M2$
::PRINT@370,M3$:
240 GOSUB5300:GOSUB5800:GOTO260
245 IFG=2GOTO7160ELSEIFG=4GOTO6140
250 FORI=1TO8:ONIGOSUB2020,2030,2040,2050,2060,2070,20
80,2090:NEXTI
260 PRINT@916,STRING$(24,S)::PRINT@960,STRING$(56,S)::
IFG>1PRINT@960,"SCORE";STR$(SC);
265 IFG<>4PRINT@908," ";PRINT@944," ";
267 IFG=4 PRINT@906,STRING$(44,S);
270 PRINT@1015,"BALL";CT+1::POKEA+X+Y,D:GOTO500
340 DATA 205,127,10,14,140,69,62,1,211,255,16,254,69,6
2,2,211,255,16,254,13,32,239,201
500 Q=A+X+IX+Y+IY:IFPEEK(Q)<>STHEN1000
505 IFPEEK(Q)<>STHEN1000
512 L=L+PL:IFL>WGOTO700
514 R=R+PR:IFR>WGOTO750
520 POKEQ-IX-IY,S:POKEQ,D:Y=Y+IY:X=X+IX:T$=INKEY$:IFT$
=""THEN500
600 IFQ>MTHEN500ELSEIFT$="Z"PRINT@LZ,P$:L=0:PL=1:GOTO
500
610 PRINT@RZ,P$::R=0:PR=1:GOTO500
700 L=0:PL=-1:PRINT@LZ,R$::GOTO514
750 R=0:PR=-1:PRINT@RZ,R$::GOTO520
1000 IFQ>16255ANDQ<16320THEN QQ=USR(30):IFPEEK(Q)=PDTH
EN1090ELSEIFX>57ORX<6THEN1085ELSEIY=-IY:IX=(RND(U)+E)*
SGN(IX):GOTO500
1010 IF Q>16319THENCT=CT+1:QQ=USR(255):IFCT>BATHEN4000
ELSE2130
1020 V=PEEK(Q):IFV<1650RV=191THEN1060ELSEV=V-164:SC=SC
+10*V:PRINT@960,"SCORE";STR$(SC);
1025 ONG GOTO1060,7300,1030,6200
1030 ON(V-1)GOSUB2025,2035,2045,2055,2065,2075,2085,20
95
1035 QQ=USR(35+4*V)
1040 ON(V-1)GOSUB2020,2030,2040,2050,2060,2070,2080,20
90
1060 IF PEEK(Q-IX)<>S THEN QQ=USR(30):IY=-IY:C=IX:IX=
(RND(U)+E)*SGN(IX):IFG=1THEN5500ELSE500

```

```

1080 IF PEEK(Q-IY)<>S THEN QQ=USR(30): C=IX:IX=-IX:IFG
=1THEN5500ELSE500
1085 IY=-IY:C=IX:IX=-(RND(U)+E)*SGN(IX):IFG=1THEN5500E
LSE500
1090 IFG=1 IY=-IY:IX=RND(2)+1:GOTO500
1100 IFNT>20RCT<30RRND(0)>.15THEN1120ELSEFORI=11TO1STE
P-1:PRINT@64*I,ZQ$(I);:NEXTI:NT=NT+1
1120 IY=-IY:IX=RND(U)*SGN(IX):GOTO500
2020 PRINT@606,STRING$(5,166);:RETURN
2025 PRINT@606,STRING$(5,191);:RETURN
2030 PRINT@622,STRING$(4,167);:RETURN
2035 PRINT@622,STRING$(4,191);:RETURN
2040 PRINT@591,STRING$(4,168);:RETURN
2045 PRINT@591,STRING$(4,191);:RETURN
2050 PRINT@407,STRING$(4,169);:RETURN
2055 PRINT@407,STRING$(4,191);:RETURN
2060 PRINT@422,STRING$(4,170);:RETURN
2065 PRINT@422,STRING$(4,191);:RETURN
2070 PRINT@238,STRING$(4,171);:RETURN
2075 PRINT@238,STRING$(4,191);:RETURN
2080 PRINT@206,STRING$(4,172);:RETURN
2085 PRINT@206,STRING$(4,191);:RETURN
2090 PRINT@223,STRING$(3,173);:RETURN
2095 PRINT@223,STRING$(3,191);:RETURN
2130 IFG=2GOSUB7400
2140 X=59:Y=12*64:IX=-RND(3):IY=-64
2143 IFG=1GOSUB5300:GOSUB5800:IFW=3THENIFRND(4)>2GOSUB
5300
2144 PRINT@980,"HIT SPACE BAR TO CONTINUE";
2146 T$=INKEY$:IF T$<>" "THEN2146
2150 GOTO 260
4000 IFG=1THEN5350ELSEFORI=1TO900:NEXT:CLS:PRINTCHR$(2
3)
4010 PRINT@82,"SCORE :";STR$(SC);
4015 E1$="RATING : CHAMPIONSHIP QUALITY":E3$="RATING
: STRICTLY MINOR LEAGUE":E2$="RATING : COULD BE BETT
ER,
      COULD BE WORSE!"
4020 CT=SC/CT:IFG=2ORG=3THENIFCT>300THENPRINT@192,E1$;
ELSEIFCT>200THENPRINT@192,E2$;ELSEPRINT@194,E3$;
4030 IFG=4THENIFCT>900THENPRINT@192,E1$;ELSEIFCT>550TH
ENPRINT@192,E2$;ELSEPRINT@194,E3$;
4035 GOTO4050
4050 FORI=1TO2700:NEXT:RUN40
5000 E1$=STRING$(2,32)+CHR$(184)+STRING$(9,191)+CHR$(1
87)
5010 E2$=CHR$(32)+CHR$(168)+CHR$(171)+STRING$(8,191)+C
HR$(159)+CHR$(171)+CHR$(134)
5020 E3$=STRING$(2,32)+CHR$(142)+CHR$(129)+CHR$(143)+S
TRING$(3,32)+CHR$(136)+CHR$(135)+CHR$(138)+CHR$(133)+C
HR$(130)
5030 M1$=CHR$(143)+CHR$(191)+CHR$(133)
5040 M2$=CHR$(141)+CHR$(140)+STRING$(3,191)
5050 M3$=CHR$(160)+CHR$(186)+CHR$(163)+CHR$(149)
5060 DIMG$(13):G$(1)=CHR$(143):G$(2)=CHR$(144):G$(3)=C
HR$(184)
5070 FORI=4TO9:G$(I)=CHR$(176):NEXT:G$(10)=CHR$(180)
5080 S1=532:S2=724:S1$=CHR$(172)+STRING$(5,176)+CHR$(1
56)+CHR$(142)+STRING$(2,131)
5090 S2$=CHR$(131)+CHR$(139)+STRING$(4,140)
5100 CLS:PRINT@22,"AFRICAN SAFARI":PRINT

```

```

5105 PRINTTAB(5)"YOU ARE A HUNTER ON A AFRICAN SAFARI,
    FACING A WILD"
5110 PRINTTAB(5)"ELEPHANT. THE ELEPHANT WILL CHARGE E
    ACH TIME IT IS"
5120 PRINTTAB(5)"HIT BY THE BALL AND WHEN THE BALL LEA
    VES PLAY EACH TIME"
5130 PRINTTAB(5)"YOUR HUNTER IS HIT BY THE BALL. HE EX
    TENDS HIS GUN AND"
5140 PRINTTAB(5)"WILL EVENTUALLY SHOOT THE ELEPHANT.
    TO REWARD YOUR SKILL,"
5150 PRINTTAB(5)"THE PADDLES RETURN THE BALL IN THE HU
    NTER'S DIRECTION."
5152 PRINTTAB(5)"BEWARE OF TWO MOVING SNAKES IN THE VI
    CINITY, "
5153 PRINTTAB(5)"YOU HAVE UP TO";:IFW=3PRINT" 6 ";ELSE
    PRINT" 8 ";
5154 PRINTTAB(5)"PLAYS TO BAG YOUR ELEPHANT. ":PRINT
5155 PRINTTAB(16)"USE << Z >> FOR THE LEFT PADDLE":PRI
    NTTAB(16)"USE << / >> FOR THE RIGHT PADDLE":PRINT
5160 PRINTTAB(5)::INPUT"HIT << ENTER >> TO CONTINUE";X
    E:CLS
5170 XE=20:YE=256:X2=7:GX=51:GN=0:QA=15555:QB=15737
5190 BA=5:D=42:F#=STRING$(14-U,62):E=1:IFW=5 BA=7:GOTO
    180
5195 GOTO180
5300 IFXE>39THEN5350
5305 XE=XE+1:X2=X2+1:QQ=USR(20):QQ=USR(20):QQ=USR(20)
5310 PRINT@YE-Z+X2,E1#;:PRINT@YE+X2,E2#;:PRINT@YE+Z+X2
    ,E3#;:RETURN
5350 FORJ=1TO35:GOSUB5305:IFXE>57THEN5355ELSEFORI=1TO1
    :NEXTI,J
5355 FORI=1TO300:NEXTI:CLS:PRINTCHR$(23):PRINT"YOU HAV
    E JUST BEEN TRAMPLED BY THE ELEPHANT !!!"
5360 PRINT:PRINT"BETTER LUCK ON YOUR NEXT SAFARI":GOTO
    4050
5500 IFQ<QAORQ>QBTHEN500ELSEIFX+C>X2ANDX+C<=XEGOSUB570
    0:GOSUB5300:GOTO500
5510 IFX+C<41ORX+C>53THEN500ELSEGN=GN+1:IFGN>10THEN560
    0
5520 GOSUB5700:IFGN=1THENPRINT@306,G$(1);ELSEPRINT@244
    -GN,G$(GN);
5530 GOTO500
5600 QQ=USR(30):QQ=USR(200):FORI=233TO197STEP-1:PRINT@
    I-1,CHR$(176);:PRINT@I," ";:FORW=1TO1:NEXTW
5610 NEXTI:PRINT@I," ";:FORW=1TO400:NEXTW
5620 CLS:PRINTCHR$(23):PRINT"YOU HAVE BAGGED A NICE TR
    OPHY !!!":GOTO4050
5700 Q=A+X+IX+Y+IY:IFPEEK(Q)<>STHEN5710ELSEPOKEQ-IX-IY
    ,S:POKEQ,D:Y=Y+IY:X=X+IX:GOSUB5800:RETURN
5710 IY=Z:IX=-RND(3):Q=A+X+IX+Y+IY:POKEQ-IX-IY,S:POKEQ
    ,D:Y=Y+IY:X=X+IX:GOSUB5800:RETURN
5800 IFRND(0)>.85GOTO5830
5810 PRINT@S1,STRING$(10,32);:IFRND(0)>.95THEN5830ELSE
    S1=521+RND(33):PRINT@S1,S1#;
5830 IFRND(0)>.85THENRETURNELSEPRINT@S2,STRING$(6,32);
    :IFRND(0)>.9THENRETURNELSES2=713+RND(37):PRINT@S2,S2#;
    :RETURN
6000 CLS:PRINTTAB(25)"WIPEOUT":PRINT
6010 PRINT"IN THE GAME OF WIPEOUT, FOUR ROWS OF NUMBER
    S ARE IN THE TOP"

```



```

6025 PRINT"IS REMOVED FROM THE SCREEN.  WHEN ONLY FIVE
NUMBERS REMAIN"
6030 PRINT"IN A ROW, THE NEXT HIT ON THAT ROW REMOVES
THE REST OF"
6035 PRINT"THOSE NUMBERS.  PINBALL PADDLES AT THE BOTT
OM OF THE SCREEN"
6045 PRINT"HELP KEEP THE BALL IN PLAY.  RANDOMLY AS TH
E GAME PROGRSSES,"
6050 PRINT"THE NUMBERS SHIFT DOWN CLOSER TO THE PADDLE
S,  THUS DEMANDING"
6055 PRINT"GREATER SKILL FROM THE PLAYER.  IT IS POSSI
BLE TO PLAY A"
6060 PRINT"PERFECT GAME AND * WIPEOUT * ALL THE NUMB
ERS FROM THE SCREEN.":PRINT
6090 DIMZQ$(11):FORI=1TO11:ZQ$(I)="." :POKEVARPTR(ZQ$(I
)),64:READZ:POKEVARPTR(ZQ$(I))+1,Z:READZ:POKEVARPTR(ZQ
$(I))+2,Z:NEXTI
6094 DATA 0,60,64,60,128,60,192,60,0,61,64,61,128,61,1
92,61,0,62,64,62,128,62
6096 MO=Y+1:ME=Y+4
6100 NT=0:DIM N(9)
6110 FORI=2TO9:READN(I):NEXTI
6120 DATA 517,54,389,54,261,54,133,54
6125 PRINT"USE << Z >> FOR LEFT PADDLE, AND << / >> FO
R RIGHT PADDLE."
6130 INPUT"ENTER NUMBER OF BALLS (10 MAX)":BA:IFBA<10R
BA>10THEN6130ELSEBA=BA-1
6135 D=140:E=0:GOTO180
6140 FORI=3TO9STEP2:PRINT@N(I-1),STRING$(54,164+I):NE
XTI:PRINT@0,STRING$(64,191):GOTO260
6200 POKE0,191:QQ=USR(15+2*V):N(V)=N(V)-1:IFN(V)<5PRIN
T@N(V-1)+NT*Z,STRING$(54,S);
6220 IFN(3)+N(5)+N(7)+N(9)<17 THEN FORI=1TO500:NEXTI:G
OTO6400
6300 POKE0,S:GOTO1060
6400 CLS:PRINTCHR$(23):FORI=1TO30:PRINT@470,"WIPEOUT";
:FORJ=2*ITO40:NEXTJ
6410 PRINT@470," " :FORJ=2*ITO40:NEXTJ:NEXTI
6420 PRINT@144,"PERFECT GAME":FORI=1TO800:NEXTI:GOTO4
035
7000 CLS:PRINTTAB(20);"RANDOM TARGET PINBALL":PRINT
7010 PRINTTAB(8)"IN * RANDOM TARGET PINBALL * , THERE
ARE NINE NUMERICAL"
7020 PRINTTAB(8)"TARGETS IN THE PLAYING ARE.  WHEN THE
BALL HITS"
7030 PRINTTAB(8)"ONE OF THE TARGETS,  THE TOTAL SCORE
IS INCREASED"
7040 PRINTTAB(8)"BY TEN TIMES THE VALUE OF THE TARGET.
THEN THE"
7050 PRINTTAB(8)"TARGET ITSELF IS MOVED RANDOMLY TO AN
OTHER LOCATION"
7060 PRINTTAB(8)"IN THE PLAYING AREA.  WHENEVER THE BA
LL LEAVES PLAY,"
7070 PRINTTAB(8)"SOME OF THE TARGETS ARE MOVED. ":PRIN
T

```

```

7100 DIM N(9),PS(30),N$(9):P2=30
7110 FORI=1TO30:READD:NEXTI:FORI=1TOP2:READ PS(I):NEXT
7120 FORI=1TO9:N$(I)=STRING$(5,164+I):NEXT
7130 DATA 202,214,226,238,328,337,344,352,359,369,458,
465,472,484,496
7140 DATA 585,597,609,621,722,729,739,196,245,628,479,
750,714,491,233
7150 GOTO8050
7160 FORI=1TO9
7165 J=RND(22):IFPEEK(A+PS(J))<>STHEN7165ELSEN(I)=PS(J
)
7170 PRINT@N(I),N$(I):NEXTI:GOTO260
7300 PRINT@N(V),STRING$(5,191):QQ=USR(25+2*V)
7310 J=RND(22):IFPEEK(A+PS(J))<>STHEN7310ELSEPRINT@N(V
),STRING$(5,S):N(V)=PS(J):PRINT@N(V),N$(V):GOTO1060
7400 ZZ=RND(9):FORI=1TOZZ:PRINT@N(I),STRING$(5,S):NEX
TI:FORI=1TOZZ
7410 J=RND(P2):IFPEEK(A+PS(J))<>STHEN7410ELSEN(I)=PS(J
):PRINT@N(I),N$(I):NEXTI:RETURN
8000 CLS:PRINTTAB(26):"PINBALL":PRINT
8010 PRINTTAB(8)"MICRO-80 PINBALL IS A FAST-PACED SIMU
LATION OF"
8020 PRINTTAB(8)"THE REAL-WORLD GAME IN WHICH A BOUNCI
NG BALL"
8030 PRINTTAB(8)"COLLIDES WITH TARGETS IN AN ENCLOSED
PLAYING"
8035 PRINTTAB(8)"AREA. THE PADDLES AT THE BOTTOM OF T
HE SCREEN"
8037 PRINTTAB(8)"HELP KEEP THE BALL IN PLAY. WHEN A T
ARGET (MADE"
8038 PRINTTAB(8)"UP OF NUMBERS) IS HIT, TEN TIMES THE
VALUE OF"
8039 PRINTTAB(8)"THAT TARGET IS ADDED TO THE SCORE.":P
RINT
8050 PRINTTAB(8)"USE << Z >> FOR LEFT PADDLE,"
8055 PRINTTAB(8)"AND << / >> FOR RIGHT PADDLE":PRINT
8060 PRINTTAB(8):INPUT"ENTER NUMBER OF BALLS (10 MAX)"
;BA:IFBA<1ORBA>10THEN8060ELSEBA=BA-1
8070 D=140:E=0:GOTO180
9000 I=VARPTR(QQ#):X=PEEK(I+1)+256*PEEK(I+2)
9010 FORY=XTOX+22:READZ:POKEY,Z:NEXTY
9030 IFPEEK(16396)=201THENPOKE16526,PEEK(I+1):POKE1652
7,PEEK(I+2)ELSECMD"T":DEFUSR0=X:POKE14308,0
9040 RETURN

```

附录一 BASLC II 程序语句一览表

语 句	功 能 及 举 例
(类型定义) DEFDBL 字母表或范围 DEFINT 字母表或范围 DEFSNG 字母表或范围 DEFSTR 字母表或范围	定义为双精度型, 所有与指定字母或指定范围内的字母作开头的变量都是双精度型。 例: DEFDBL J DEFDBL X, Y, A-E 所有以指定字母或指定范围内的字母作开头的变量都被定义为整型。 例: DEFINT A, B, C DEFINT S, X-Z 所有以指定字母或指定范围内的字母作开头的变量都被定义为单精度型。 例: DEFSNG A, B, C, DEFSNG A-L, P 所有以指定字母或指定范围内的字母作开头的变量都被定义为串变量 例: DEFSTR A, B, C DEFSTR X-2, M
(赋值和分配) CLEAR DIM 数组 (维数 1, , ... 维数 K) LET 变量 = 表达式	为字符串存储开辟指定数目 n 字节内存。 例: CLEAR 750 CLEAR MEM/10 CLEAR ϕ 为 K 维数组分配内存, 数组的每一维具有指定长度: 维数 1, ... 维数 K, DIM 后的数组表可用逗号隔开。 例: DIM A (2, 3) DIM A1 (X+2), C (J, K) DIM T (3, 3, 5) 把表达式的值赋给变量, 在 BASIC II 中 LET 可有可无。 例: LET A/\$ = D = X"CHARLIS" , *Y/Z
(执行的顺序) END STOP GOTO 行数 GOSUB 行数	结束执行, 回到命令状态。 例: 99END 停止执行, 打印出有关中断行语句标号的中断信息。 例: 100 STOP 转移到指定行数。 例: GOTO 100 转移到指定行数开始的子程序。

附录一 BASIC II 程序语句一览表(续)

语 句	功 能 及 举 例
RETURN	例: GOSUB 3000
ON表达式GOTO行数 1, 行数 K	子程序出口, 转移到最近执行的GOSUB下边的语句。 计算表达式的值, 如果表达式的整数部分等于 1 至K之间的一个数, 则转移到相应的行数, 否则到下一条语句。
ON表达式GOSUB行数 1, 行数 K	例: ON K + 1 GOTO 100, 200, 300 同上, 根据表达式的整数数值, 转移到相应的行数开始的子程序。
FOR 变量 = 表达式 TO 表达式 STEP表达式	例: ON J GOSUB 330, 7000 打开FOR—NEXT循环, STEP可有可无, 若不用时增量为 1。
NEXT变量	例: FOR I = 1 TO 50 STEP1.5 FOR 19% = J% TO K - 1% 关闭FOR—NEXT循环, 变量可省略, 要关闭嵌套的循环可用变量表。
ERROR (代码)	例: NEXT, NEXT I, J, K 模拟用代码指定的错误。
ON ERROR GOTO行数	例: ERROR (14) 在以后的程序行中如果产生错误, 转移到该行数开始的错误处理程序。
RESUME n	例: ERROR GOTO 999 由错误处理程序回到指定行 n, 如果 n 等于或没有, 则回到到含有错误的语句, 如果 n 是“NEXT”则回到错误语句的下一条语句。
RANDOM	例: RESUME, RESUME O, RESUB100RESUME NEXT 开启随机数发生器。
REM	例: RANDOM 注释说明。BASIC解释程序将忽略程序行其余部分。
(条件语句) IF 表达式 1 THEN 语句 1 ELSE 语句 2	例: REM A Is ALTITUDE 测试表达式 1: 如果是真, 执行语句 1, 否则跳到下一程序行(除非语句1是GOTO)如果是伪, 直接跳到ELSE语句并执行后边的语句。
(图示语句) CLS	例如: IF A = 0 THEN PRINT “ZERO” ELSE PRINT “NOT ZERO” 清除电视显示器并成为每行64字符。 例如: CLS

附录一 BASIC II 程序语句一览表

语 句	功 能 及 举 例
RESET (X, Y)	关闭水平坐标 X 垂直坐标 Y 的图示块。 $0 \leq X < 128, 0 \leq Y < 48$ 。
SET (X, Y)	例: RESET (8 + B, 11) 打开坐标为 X, Y 的图示块, 自变量限制同上。
(特殊语句) POKE 地址, 值	例: SET (A * 2, B + C) 将值输入到指定地址的内存单元中。
OUT 出口, 值	例: POKE 15635, 34 POKE 17770, A + N 两自变量都以十进制形式表示, $0 \leq \text{变量} \leq 255$ 将值送到出口 (两自

附录二 BASIC II 字符串函数一览表

ASC (串)	取自变量第一个字符的 ASCII 代码。串可以是串变量, 表达式或常数。(下同)。
CHR\$(代码表达式)	例: ASC (B\$), ASC ("H") 取回由代码规定的一个字符串, 如果代码规定一个控制功能, 那么实施这功能。
FRE (串)	例: CHR\$(34), CHR\$(11) 取回可用于串存储的内存数量, 自变量是一个虚变量。
INKEY\$	例: FRE (A\$) (注: 函数取决于 CLEAR n 的数 n) 选通键盘, 得到一个在选通脉冲期间新按键输入的字符 (如果不按键, 则为零串)。
LEN (串)	例: 100PRINT@ 540, INKEY\$: GOTO 100 求串的长度 (零串为 0)。
LEFT\$(串, n)	例: LEN (A\$ + B\$), LEN ("HOURS") 取串的开头 N 个字符。
MID\$(串, P, n)	例: LEFT\$(A\$, 1), LEFT\$(L1\$, + C\$, 8), LEFT\$(A\$, M + L) 取串的子串, 其长度为 n, 在串中的起始位置是 P。
RIGHT\$(串, n)	例: MID\$(M\$, 5, 2), MID\$(M\$ + B\$, P, L - 1) 取串的最后 n 个字符。
STR\$(数学表达式)	例: RIGHT\$(NA\$, 7), RIGHT\$((AB\$, M2) 计算自变量并把其值表示为串。
STRING\$(n, 字符)	例: STR\$(1.2345), STR\$(A + B * 2) 得到一个字符串, 由几个规定的字符所组成。
VAL (串)	例: STRING\$(30, ".") , STRING\$(5, C\$) STR\$函数的逆函数, 根据数值串求数值。 例: VAL ("1" + A\$ + "." + C\$), VAL (G1\$)。

附录三 BASIC II 数学函数一览表

函 数	功 能 及 举 例
ABS (表达式)	取绝对值。表达式可以是任何数学表达式或常数。(注:下同) 例: ABS (L * 7), ABS (SIN (x))
ATN (表达式)	求反正切 (弧度表示)。 例: ATN (2.7), ATN (A * 3)
CDBL (表达式)	取表达式的双精度表示。 例 CDBL (A), CDBA (A + 1 / 3 #)
CINT (表达式)	取不大于表达式的最大整数。 限制: $-32768 \leq \text{表达式} \leq +32767$ 例 CINT (A # + B)
COS (表达式)	求表达式的余弦, 表达式用弧度表示。 例: COS (2 * A), COS (A / 57.2958)
CSNG (表达式)	取单精度表示, 当表达为双精度数时, 其最小有效小数四舍五入。 例: CSNG (A #), CSNG (33 * B #)
EXP (表达式)	取自然指数, 即 $e^{\text{表达式}} = \text{EXP}(\text{表达式})$ 。 例: EXP (34.5), EXP (A * B * C - 1) 取不大于表达式的整数。自变量不受 $-32768 \sim +32767$ 范围的限制。 例: INT (A + B * C)
LOG (表达式)	取表达式的自然对数 (以 e 为底)。 限制: 表达式必须是正数。 例: LOG (12.33), LOG (A ↑ B + B)
RND (0)	取 0.000001 到 0.999999 之间的伪随机数。 注乙: 有些机器括弧内是 1, BASIC II 为 0。 例: RND (0)
RND (表达式)	取 1 到 INT (表达式) 之间的伪随机数。 限制: $1 \leq \text{表达式} \leq 32768$ 例: RNO (40), RND (A + B)
SGN (表达式)	符号函数, 表达式为负数时得 -1; 零时得 0, 正数时得 +1 例: SGN (A * B + 3), SGN (COS (x))
SIN (表达式)	求表达式的正弦。表达式用弧度表示。 例: SIN (A / B), SIN (90 / 57.29578)
SQR (表达式)	求表达式的平方根。 限制: 表达式必须是非负数 例: SQR (A * A - B * B)
TAN (表达式)	求表达式的正切, 表达式是弧度。 例: TAN (x), TAN (X * .0174533)

注: 除非有其它说明, 函数自变量限于: $-1.7E+38 \leq \text{表达式} \leq 1.7E38$ 。

附录四 BASIC II 特殊函数一览表

函 数	功 能 及 举 例
ERL	取产生错误的行数。 例: ERL
ERR	取与错误代码有关的值 (如果产生错误的话)。 $ERR = (\text{错误代码} - 1) * 2$, 也就是: $(ERR/2) + 1 = \text{错误代码}$ 。 例: $ERR/2 + 1$
INP (入口)	从指定入口输入一个数值。自变量和结果都在 0 到255范围内。 例: INP (55)
MEM	取未用的和未保留的内存字节总数。 例: MEM, IF MEM < 800 THEN 90
PEEK (地址)	取回存在指定内存的一个字节的数值。地址必须是十进制形式的有效内存地址。 例: PEEK (15370)
POINT (x, y)	检查由水平坐标 x 和垂直坐标 y 指定的图示块, 如果图示块“开着”, 得真 (-1); 如果“关着”, 得假 (0)。 限制: $0 \leq X \leq 128, 0 \leq y \leq 48$ 例: 10SET (50, 28) : IF POINT (50, 28) THEN PRINT "ON" ELSE PRINT "OFF"
POS (O)	取一个表明光标目前位置的数, 自变量 0 是虚变量。 例: POS (O)
USR (n)	转移到机器语言子程序。n 是二字节整数, $-32768 < n < 32767$ 。 例: USR (O)
VARPTR (变量)	取一个地址, 该处存着指定变量的名称、数值和指数。变量必须是有效的变量名。如果变量没有被赋值, 则得 0。 例: VARPTR (A/S), VARPTR (NI)

附录五 导出函数表

函 数	用BASIC II 函数表示
正 割	$SEC(X) = 1 / \cos(X)$
余 割	$CSC(X) = 1 / \sin(X)$
余 切	$COT(X) = 1 / \tan(X)$
反正弦	$ARCSIN(X) = \text{ATN}(X / \sqrt{-X * X + 1})$
反余弦	$ARCcos(X) = -\text{ATN}(X / \sqrt{X * X - 1}) + 1.5708$
反正割	$ARCSEC(X) = \text{ATN}(\sqrt{X * X - 1}) + (\text{SGN}(X) - 1) * 1.5708$
反余割	$ARCCSC(X) = \text{ATN}(1 / \sqrt{X * X - 1}) + (\text{SGN}(X) - 1) * 1.5708$
反正切	$ARCCOT(X) = -\text{ATN}(X) + 1.5708$
双曲正弦	$\sinh(X) = (\exp(X) - \exp(-X)) / 2$
双曲余弦	$\cosh(X) = (\exp(X) + \exp(-X)) / 2$
双曲正切	$\tanh(X) = -\exp(-X) / (\exp(X) + \exp(-X)) * 2 + 1$
双曲正割	$\text{SECH}(X) = 2 / (\exp(X) + \exp(-X))$

附录五 导出函数表 (续)

函 数	用BASIC II函数表示
双曲余割	$CSCH(X) = 2 / (EXP(X) - EXP(-X))$
双曲余切	$COTH(X) = EXP(-X) / ((EXP(X) - EXP(-X)) * 2 + 1)$
反双曲正弦	$ARGSINH(X) = LOG(X + SQR(X * X + 1))$
反双曲余弦	$ARGCOSH(X) = LOG(X + SQR(X * X - 1))$
反双曲正切	$ARGTANH(X) = LOG((1 + X) / (1 - X)) / 2$
反双曲正割	$ARGSECH(X) = LOG((SQR(-X * X + 1) + 1) / X)$
反双曲余割	$ARGCSCH(X) = LOG((SGN(X) * SQR(X * X + 1) + 1) / X)$
反双曲余切	$ARGCOTH(X) = LOG((X + 1) / (X - 1)) / 2$

附录六 控制、ASCII、图形符代码对照表

1. 控制代码 (0—31)

0—7	不用
8	退格并删除该字符
9	不用
10—13	回车
14	开光标
15	熄光标
16—22	不用
23	字符显示转换成32个/行
24	光标左移
25	光标右移
26	向下换行
27	向上换行
28	把光标恢复到显示屏(0, 0)位置
29	把光标移至行的开始处
30	删除至行的末尾
31	清除至这帧结束

2. ASCII 字符代码 (32—128)

代 码	字 符	代 码	字 符	代 码	字 符
32	<空格>	54	6	76	L
33	!	55	7	77	M
34	"	56	8	78	N
35	*	57	9	79	O
36	\$	58	:	80	P
37	%	59	;	81	Q
38	&	60	<	82	R
39	'	61	=	83	S
40	(62	>	84	T
41)	63	?	85	U
42	*	64	Ⓢ	86	V

2. ASCII 字符代码 (32—128)

代 码	字 符	代 码	字 符	代 码	字 符
43	+	65	A	87	W
44	,	66	B	88	X
45	-	67	C	89	Y
46	。	68	D	90	Z
47	/	69	E	91	↑或 [
48	0	70	F	92	↓
49	1	71	G	93	←
50	2	72	H	94	→
51	3	73	I	95	—
52	4	74	J	96—127	小写代码
53	5	75	K	128	< 空格 >

3. 图形符代码 (129—191)

代 码	图 形	代 码	图 形	代 码	图 形
129	█	153	█	177	█
130	█	154	█	178	█
131	█	155	█	179	█
132	█	156	█	180	█
133	█	157	█	181	█
134	█	158	█	182	█
135	█	159	█	183	█
136	█	160	█	184	█
137	█	161	█	185	█
138	█	162	█	186	█
139	█	163	█	187	█
140	█	164	█	188	█
141	█	165	█	189	█
142	█	166	█	190	█
143	█	167	█	191	█
144	█	168	█		
145	█	169	█		
146	█	170	█		
147	█	171	█		
148	█	172	█		
149	█	173	█		
150	█	174	█		
151	█	175	█		
152	█	176	█		

用户可以利用如下程序，打印出上列的图形符：

```

10 FOR X = 129 TO 191
20 PRINT; PRINT CHR$(X); LPRINT$(X); LPRINT$(CHR$(X))
30 NEXT X
    
```

4. 压缩空格代码 (192—255)

代 码	功 能
192—255	分别代表 0—63 个空格

附录七 BASIC II 错误信息表

代 码	缩 写 符	错 误 性 质
1	NF	NEXT与FOR不配对
2	SN	句法错误
3	RG	RETURN与GOSUB不配对
4	OD	数据完
5	FC	非法调用功能
6	OV	溢出
7	OM	存贮空间用完
8	UL	标号未定义
9	BS	下标超界
10	DD	数组定维重复
11	/O	除数为零
12	ID	非法直接指令
13	TM	类型不一致
14	OS	字符串空间完
15	LS	字符串太长
16	ST	字符串公式太复杂
17	CN	不能继续
18	NR	无法RESUME
19	RW	RESUME与ERROR不配对
20	UE	无法打印的错误
21	MO	遗漏被运算数
22	FD	文件数据出错
23	L 3	只能用于磁盘BASIC

对错误信息的解释

- NF NEXT没有与其相匹配的FOR语句。如果NEXT语句的变量在嵌套层里颠倒，也同样显示这个错误信息。
- SN 该错误通常是由不正确的标点符号、括号不配对、非法字符、语句或命令拼写错误而引起的。
- RG 在执行与之相匹配的GOSUB之前就遇到RETURN。
- OD 执行READ或 INPUT #时，可供使用的数据不够。可能遗漏了DATA语句；或有磁带数据、DATA中的数据都已读完。
- FC 企图使用一个非法的参数来执行运算。如负变元的平方根、负的数组维数、LOG (X)系数的变量是负或是零、没有先用POKE语句给出入口地址就调用USR等等。
- OV 输入或标出的值过大或过小，超过了计算机所能处理的范围。
- OM 所有可用的存贮单元都被占用或被保护。如使用了过大的数组维数，GOTO、GOSUB、FOR/NEXT等嵌套或转移过多。
- UL 企图涉及到或转移到不存在的行。
- BS 下标超界。引用的数组元素下标超出维数界域。
- DD 由DIM语句或其它违反规定的语句重复给已定过维的数组定维。

- /0 分母的值为零。
- ID 把INPUT语句作为直接命令。
- TM 企图将非字符串变量赋值给字符串或反之。
- OS 字符串空间用完。超出已分配的字符串空间的总数。
- LS 赋给字符串变量的字符串超过255个。
- ST 字符串运算太复杂，难于处理。解决办法是将运算分为n个较小的段。
- CN 发出CONT命令后，程序无法执行。这是由于程序已执行了END，或在 BREAK 后进行程序的修改。
- NR 在纠正错误子程序结束前又陷入错误状态。
- RW 在执行ON ERROR GOTO之前就遇到RESUME语句。
- UE 企图在ERROR语句使用一个无数的代码模拟错误信息。
- MO 进行运算时，没有提供所需要的运算数。
- FD 从外部设备输入的数据不正确，或是一个不合适的序列等。
- L3 只有连接了扩展器和磁盘驱动器后，才可以使用的语句、函数或命令。

附录八 磁盘BASIC错误信息表

代 码	信 息	注 释
50	FIELD OVERFLOW	分配给随机存取缓冲区内容过多，已超过255个字符。
51	INTERNAL ERROR	磁盘操作系统本身出错或磁盘输入输出失败。
52	BAD FILE NUMBER	文件——缓冲器编号使用不当。OPEN语句没有把此编号赋给文件。
53	FILE NOT FOUND	磁盘上没有指定该文件。请检查指定的文件标识符是否正确。
54	BAD FILE MODE	对磁盘文件进行I/O方式与打开文件时指定的方式不相符。
57	DISK I/O ERROR	计算机和磁盘文件间传输的数据出错。
61	DISK FULL	磁盘上可用空间已用完。
62	INPUT PAST END	由磁盘以顺序方式读出数据赋给变量时，还没有读到数据就遇到文件结束标志。
63	BAD RECORD NUMBER	PUT语句中记录号超出范围。
64	BAD FILENAME	文件标识符不正确。
66	DIRECT STATEMENT IN FILE	要读入、运行或合并的磁盘文件不是BASIC程序。
67	TOO MANY FILE	在磁盘上存放的文件太多，已超过48个。
68	DISK WRITE—PROTECTED	要在一个已有写保护的磁盘上写入文件。
69	FILE ACCESS DENIED	使用错误的通行字来存取现有文件。

注：磁盘BASIC错误信息不能用ERROR语句（模拟错误语句）进行模拟。

附录九 BASIC II 内存存储器分配图

地 址		内 容
十 进 制	十 六 进 制	
0	0000	
12288	3000	LEVEL II BASIC ROM
		保 留 区
14302	37DE	—— 通讯状态地址
14303	37DF	—— 通讯数据地址
14304	37E0	—— 中断锁寄存器地址
14305	37E1	—— 磁盘驱动选择锁寄存器地址
14308	37E4	—— 盒式录音机选择锁寄存器地址
14312	37E8	—— 行式打印机的地址
14316	37EC	—— 软磁盘控制器的地址
14336	3800	
15360	3C00	TRS-80键盘存储器
16383	3FFF	TRS-80CRT显示存储器
16384	4000	LEVEL II BASIC 固定 RAM
		中断矢量 (RST 1-7)
16402	4012	
16405	4015	—— 键盘装置的控制块
		DCB+ 0 = DCB型式
		+ 1 = 驱动地址
		+ 2 = 驱动地址
		+ 3 = 0
		+ 4 = 0
		+ 5 = 0
		+ 6 = 'K'
		+ 7 = 'I'
16413	401D	—— 显示控制块
		DCB+ 0 = DCB型式
		+ 1 = 驱动地址 (LSB)
		+ 2 = 驱动地址 (MSB)
		+ 3 = 光标位置N (LSB)
		+ 4 = 光标位置N (MSB)
		+ 5 = 光标字符
		+ 6 = 'D'
		+ 7 = '0'
16421	4025	—— 行式打印机控制块

附录九 BASIC II 内存储器分配图 (续)

地 址		内 容
十 进 制	十 六 进 制	
		DCB + 0 = DCB型式 + 1 = 驱动地址 (LSB) + 2 = 驱动地址 (MSB) + 3 = 行/页 + 4 = 行计数器 + 5 = 0 + 6 = 'P' + 7 = 'R'
		保 留 区
16429	402D-	FDC 中断矢量
16463	404F-	
16464	4050 -	通信中断矢量
16466	4052 -	} 保留区
16468	4054 -	
	405C -	
16476	405E -	25毫秒字符中断
16478		[保留区]
16512	4080	
		LEVEL II BASIC 可供使用RAM
		[保留区]
16870	41E6	I/O 缓冲器
17127	42E7	始终保持零
17129	42E8-	
		[↓ 程序文本 [↓ 简单变量 [↓ 数 组 [↓ 字符串变量名称和整理操作 [空闲存储器 [↑ 堆 栈 [↑ 字符串空间 [给出MEMORY SIZE? 这是留给与BASIC混用的机器语言子程序的保留区
20479	4FFF	4 K 存储器终点
32767	7FFF	16 K 存储器终点
49151	BFFF	32 K 存储器终点
65535	FFFF	48 K 存储器终点