

• WEIDIANNAO • BASIC YUYAN •

• 微电脑 • BASIC语言 • 趣味程序

方 鸿 辉 沈 美 新

浙江科学技术出版社

• QUWEI CHENGXU •

责任编辑 王晓明
封面设计 张瑞邦
插图 陈聿强 肖征波等

微电脑 BASIC语言 趣味程序

方鸿辉 沈美新

*

浙江科学技术出版社出版

浙江新华印刷厂印刷

浙江省新华书店发行

开本787×1092 1/32 印张13.5 字数281,000

1985年10月第一版

1985年10月第一次印刷

印数：1—23,350

统一书号：7221·83

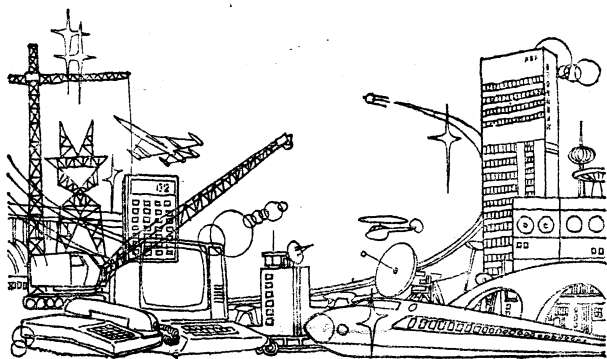
定 价：1.65 元

微 电 脑

BASIC 语言

趣味程序

方鸿辉 沈美新



浙江科学技术出版社

内 容 提 要

本书是介绍微电脑知识的入门书。

全书共分三个部分：第一部分介绍微电脑的一般知识，并着重叙述了APPLE-Ⅰ型微电脑系统的结构以及如何上机操作；第二部分简明扼要地介绍了基本BASIC语言；第三部分汇集了100个用BASIC语言编写的趣味程序。

本书图文并茂、内容充实，语言通俗易懂，叙述深入浅出，具有较强的实用性和趣味性，可供青少年电脑爱好者和电脑技术初学者阅读，也可供广大中、小学教师及各地少年宫、少科站、青少年电脑活动中心的辅导员参考。本书既可作为中学开设微电脑课程的教学参考书，也可作为开展科技活动的有益资料。

前 言

当前，世界正面临一场以电脑为中心的新技术革命。为了迎接挑战，推动我国“四化”建设，努力学习、掌握和应用电脑技术已成为我们十分迫切的任务。

我国的建设事业既需要有一批高级的电脑技术人材，也需要大批初、中级的电脑人材。因此，及早地在青少年中进行电脑知识普及教育，开展多种形式的电脑科技活动，对于培养大量的电脑人材，加快我国现代化建设具有重大的战略意义。

中、小学时期是学习科学文化知识的最好时期，无疑也是学习电脑技术的最好时期。实践证明，广大青少年只要了解了一些微电脑的基本概念，有一些数学和英语基础知识，再加上一定的逻辑思维能力，是完全可以实践学会运用电脑的。

青少年利用微电脑开展课外学习，解算一些趣味数学、物理题，既有利于开阔思路，发展智力，也有助于激发学习科学文化知识的热情和积极性。信息时代的青少年应该借助电脑来“思考”和解算各种物理、数学问题，并且编制简易程序，上机调试，初步掌握微电脑技术，打破对微电脑的神秘感，培养进一步学习电脑技术的浓厚兴趣，以适应信息时代对人材素质的要求。

正是鉴于以上思想，我们试着编写了这本实用性较强的

微电脑普及读物，奉献给广大青少年读者。

全书分为微电脑的一般知识、基础 BASIC 语言介绍和趣味程序等三部分。考虑到 APPLE-Ⅱ 型微电脑是国家优选机种(国内型号为 DJS—033),全国各地拥有该机种的单位也较多,因此,我们从动手实践的角度出发,介绍 APPLE-Ⅱ 微电脑的操作、基础浮点 BASIC 语言以及用该语言编写的趣味程序,旨在面向那些不懂电脑程序设计,但又希望尽快掌握 BASIC 语言,学会操作微电脑的广大青少年和电脑技术初学者。为此,我们在编写时尽可能写得通俗、简洁。书中提供的100例趣味程序,有些问题虽然看来很简单,然而,它能启迪思路,有助于读者理解和掌握 BASIC 语言和程序设计基本思想,并在今后解决较复杂问题时能运用这些方法。

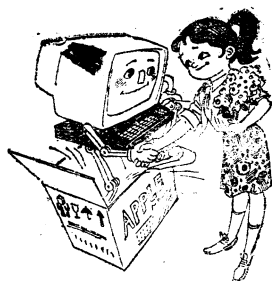
书中的趣味程序除作者自己编制外,还编译和选用了国内外有关资料。全部程序均在 APPLE-Ⅱ 微电脑上调试通过。

本书完稿后,承浙江大学数学系陈育樟同志审阅,并提出了许多宝贵意见,上海计算技术研究所孙维强同志和中国科学院声学研究所李明方同志给予了很多帮助,在此一并致谢。由于作者水平有限,错误在所难免,欢迎广大读者不吝指正。

作 者

1984年5月于上海

目 录



微电脑

I. 微电脑的一般知识	(2)
1. 什么是微电脑	(2)
2. 电脑基础知识	(4)
3. 微电脑的结构特点	(11)
II. APPLE-II 微电脑的使用与操作	(14)
1. APPLE-II 基本系统及扩展	(14)
2. 如何上机操作	(53)
3. 键盘输入的技能	(62)
4. 程序的编辑与修改	(72)

BASIC 语言

I. BASIC语言的基本概念	(80)
1. BASIC语言的基本特点	(80)
2. BASIC语言的基本符号	(81)
3. BASIC语言的基本结构	(82)
4. BASIC语言中的数	(84)
5. BASIC语言中的变量	(86)

6. BASIC 语言中的标准函数	(88)
7. BASIC 语言中的算术表达式	(89)
8. 编制简易程序的一般方法	(91)
I. BASIC语言的基本语句	(95)
1. LET 语句	(95)
2. INPUT 语句	(97)
3. READ 和 DATA 语句	(99)
4. RESTORE语句	(100)
5. GOTO语句	(101)
6. IF-THEN 语句	(103)
7. FOR-NEXT 语句	(106)
8. GOSUB 和 RETURN 语句	(108)
9. REM 语句	(111)
10. DEF 语句	(112)
11. DIM语句	(113)
12. PRINT 语句和打印格式	(115)
II. 键盘命令与程序调试语句	(123)
1. 常用键盘命令	(123)
2. 常用程序调试语句	(125)

趣味程序

冬冬抛球	(132)
扔给猴子的胡桃	(134)
卡尔解题的巧妙方法	(136)
神机妙算变“方阵”	(138)
“韩信点兵”问题	(141)
刁藩都的年龄之谜	(143)
猪八戒分西瓜	(144)

分苹果.....	(147)
历经创伤的勇士.....	(148)
无法赏赐的米.....	(150)
小猴与桃子.....	(151)
马克思做过的数学题.....	(153)
爱因斯坦解难题.....	(155)
移圆盘.....	(159)
买水果.....	(162)
高利贷.....	(163)
笑哈哈与愁煞人.....	(164)
高度单位转换.....	(166)
温度转换.....	(168)
人口的增长.....	(170)
求最大公因数.....	(172)
求多个自然数的最大公约数.....	(173)
整数的因子分解.....	(175)
到花园的路有几条.....	(177)
何日再相会.....	(178)
乏味的车号.....	(180)
逃不掉.....	(182)
黄浦江上的渡口.....	(184)
猜数.....	(186)
四个整数.....	(187)
十个数.....	(189)
巧排数组.....	(190)
移位.....	(191)
N阶乘.....	(193)
双胞胎数7744.....	(194)

电阻·电容·电压·时间·····	(196)
电阻并联·····	(197)
求串并联网络的等效电阻·····	(201)
求多电阻并联电路中的电流·····	(203)
一传十 十传百·····	(204)
百鸡问题·····	(206)
翻两番·····	(208)
算术测验·····	(210)
考场指南·····	(212)
统计考试成绩·····	(214)
择优录取·····	(217)
谁的分数最高·····	(218)
解一元二次方程·····	(221)
打印数学用表·····	(224)
打印正弦曲线·····	(226)
打印课程表·····	(228)
求椭圆周长·····	(230)
菲波那奇数列·····	(232)
阿姆斯特朗数·····	(233)
毕达哥拉斯数·····	(236)
特殊的数·····	(239)
完全数·····	(240)
有趣的幻数·····	(243)
奇妙的六位数·····	(245)
有趣的数·····	(247)
583是素数吗·····	(249)
找素数·····	(251)
孪生素数·····	(254)

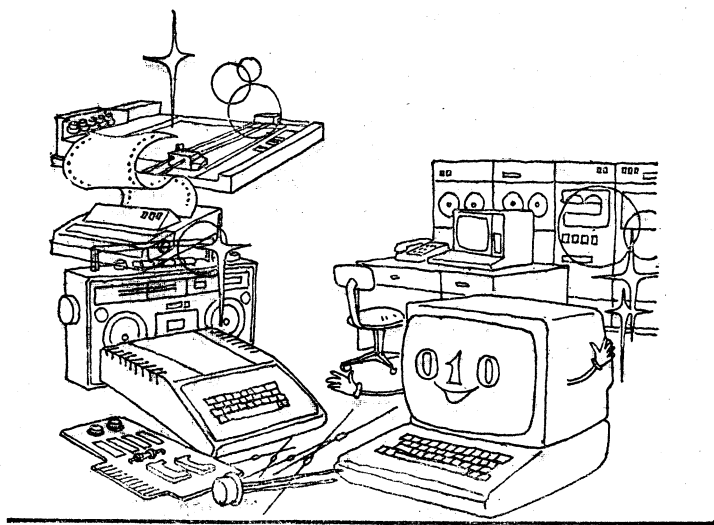
逐一“验证”哥德巴赫猜想的正确性·····	(256)
数制转换·····	(258)
两数平方和·····	(260)
捣蛋鬼3792·····	(262)
能被16除尽的数·····	(264)
整数A,B,C,D·····	(265)
整数和·····	(267)
被墨迹覆盖了的数字·····	(269)
玩纸牌·····	(271)
奇怪的数学题·····	(273)
掷硬币·····	(276)
7或11·····	(277)
非六点的几率·····	(279)
掷骰子·····	(280)
取火柴·····	(282)
打靶·····	(288)
巧分牛奶·····	(291)
钟面上的算术·····	(293)
每边数字之和都相等的三角形·····	(296)
帕斯卡三角形·····	(297)
巧填数字·····	(299)
填方格·····	(300)
螺旋方阵·····	(303)
奇妙的幻方·····	(306)
从任意数开始的幻方·····	(310)
血型检索·····	(313)
测定你的生物节律·····	(316)
图书管理·····	(319)

计算星期几.....	(321)
打印一张年历.....	(324)
打印变量图表.....	(327)
工资统计.....	(331)
让 APPLE 奏乐.....	(334)
美妙的电子琴.....	(335)
快乐的小天鹄.....	(338)
打印图案.....	(341)
祝您成功.....	(345)
附 1984年全国青少年计算机程序设计竞赛(基础知识)	
试题.....	(348)
参考答案.....	(362)

附录

I. APPLE-Ⅱ微电脑主机内存贮器概况.....	(386)
II. ASCII 码表.....	(388)
III. 软磁盘的格式化步骤.....	(392)
IV. 浮点 BASIC 错误信息表	(393)
V. 浮点 BASIC (APPLESOFT) 与整数 BASIC (INTEGER BASIC)区别	(395)
VI. 浮点 BASIC 的保留字及其代表数字	(397)
VII. 浮点 BASIC (APPLESOFT) 语言一览表.....	(399)
VIII. 微电脑常用术语.....	(409)

微 电 脑



I. 微电脑的一般知识

1. 什么是微电脑

什么是微电脑？简单地说，就是利用大规模集成电路技术，使电脑的各个主要组成部分，包括中央处理单元（CPU）、存储器、输入/输出接口电路等都集中在一块或几块只有手指头大小的芯片上（图1·1—1），每块芯片上拥有几千到几

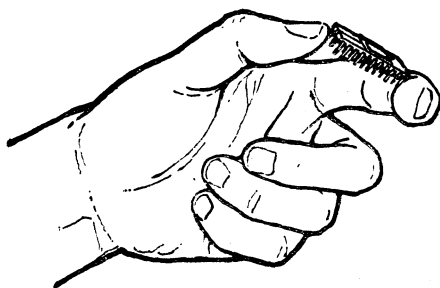


图1·1—1 象手指头大小的集成电路

十万个晶体管或其他电子元件，由这些芯片组成如同普通电视机大小的超小型通用电子计算机，这就是微电脑。目前，微电脑的功能已经赶上甚

至超过了六十年代的小型电子计算机。

1971年，美国英特尔（Intel）公司的 I4004 微处理机的问世，揭开了微电脑发展的帷幕。以后的十年间，随着大规模集成电路技术的不断突破，微电脑技术的发展也日新月异。仅以美国英特尔公司的产品为例，1971年该公司生产的 4 位微处理器 4004 和 4040，集成度为 2000 晶体管/片；1974 年生产的 8 位微处理器 8080，其集成度为 5000 晶体管/片；1976 年

生产的改进型 8 位微处理器 8085 集成度则是 9000 晶体管/片；1978 年生产的 16 位微处理器 8086，集成度已是 29000 晶体管/片；随着超大规模集成电路的出现，该公司 1981 年又研制成 32 位微处理器 IAPX432，这种微处理器是由三片电路组成，即指令译码部件、微指令执行部件和输入/输出接口处理器，三者总共集成了 22 万 5 千多个晶体管。32 位微处理器的出现，标志着微电脑的发展进入了一个新时代。目前，64 位微电脑的研制和开发工作正在进行。据统计，世界上不同类型的微电脑已达 500 多种。

微电脑由于具有体积小、价格低、功能强、工作可靠、应用灵活、使用方便等优点。因此，从七十年代末开始，它的应用范围越来越广。

微电脑能把宇航员引向遥远的空间站和外星球，能将科学家从繁琐的计算中解放出来，能帮助工程师设计机器、飞机和桥梁，能替银行及企业管理帐务，对生产线进行自动控制，甚至能帮助售货员出售商品等等。微电脑不仅深入到了工农业生产、航天、军事等各个领域，而且日益渗透到教学、医学、家庭和个人生活之中。实践证明，无论什么地方，只要用上了微电脑，那个地方的工作效率和经济效益就会显著提高。

目前，个人电脑 (PC) 正在全世界普及。不久的将来，不仅在生产、工作上离不开微电脑，人们的生活、学习也离不开微电脑。这就预示着，微电脑作为一种“智能”工具，将引起人类生活方式的极大变化。

2. 电脑基础知识

在讲述微电脑的结构原理之前，我们先来介绍一下一般电脑（又称计算机）的基础知识。

电脑是一个能进行算术运算和逻辑运算，并能够存取大量信息的自动装置。这种自动装置的设计，模拟了人脑处理信息的过程。现在让我们以算盘算题为例，回想一下这个过程。首先，我们需要有一个算盘作为运算的工具；其次要有纸和笔，用来记录原始数据、中间以及最后运算结果；当然整个运算是在大脑的控制下进行的。计算时，人们先要把待算的问题和数据记录下来，然后决定计算方法和步骤，可能还得查阅有关资料以找出计算公式；计算过程中还要不断地将中间结果记在纸上；计算完毕，再将最终结果记录在纸上（见图1·1—2）。

用这样的比喻来认识一台电脑，就比较清楚了。我们要用一台电脑来完成上述计算过程，显然，首先得要有代替算盘进行运算的部件，这就是运算器；其次还要有能记忆题目、原始数据、中间结果以及各种公式的器件，这就是存贮器；还要有代替大脑控制作用的控制器；此外，还必须要有能将题目及原始数据输给电脑和将计算结果输出的输入/输出设备。这样就构成了电脑的基本系统。从外表来看，整个系统如图1·1—3所示。

在电脑中，基本上有两股信息在流动，我们用一个方框图来表示（见图1·1—4）。图中实线表示数据信息的流动方向，虚线表示控制信息的流动方向。从图中可以看出，数据



图1·1—2 用算盘算题时人脑处理信息的过程

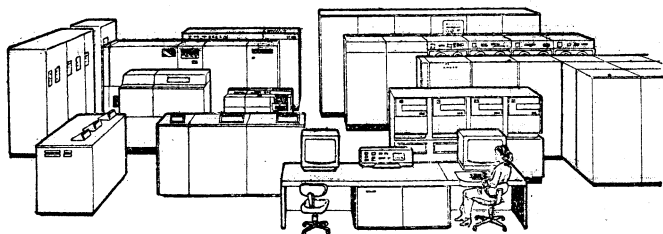


图1·1—3 电脑系统

信息（即各种原始数据、中间结果、程序等）由输入装置输入至内存贮器（存贮器有内存贮器与外存贮器之分），运算时，数据从内存贮器中取出，由运算器进行运算，运算的中间结果也要存入内存贮器。控制信息则由控制器发出，用来控制输入装置的启动或停止，控制运算器按规定的步骤一步

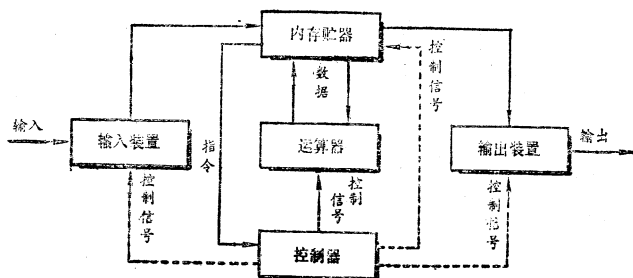


图1.1—4 电脑中信息流动方框图

步地进行各种运算和处理，控制存储器的存入和读出，控制输出设备以输出结果。

电脑的基本系统共有五个主要部件，其中运算器、内存贮器和控制器合起来，叫做“主机”，而运算器和控制器又合称为中央处理单元（CPU）。输入/输出装置以及外存贮器称作“外部设备”。一台电脑所配备的外部设备的品种和数量，是根据机器的功能大小和使用者的要求而定的。下面，将这五个部件作简单介绍：

控制器 是整个电脑的指挥系统，是电脑的司令部。通过它向电脑的各个部分有节奏地发出控制信号以指挥整台机器协调地自动工作。

运算器 是直接完成各种算术运算和逻辑运算的装置。它是电脑的骨干部门。运算器的基本组成部分是全加器和寄存器，它具备三个最基本的功能：数码的寄存、移位、相加。运算器在控制器的控制下和内存贮器交换信息。

存贮器 是存放计算程序、原始数据以及中间和最后结果的装置。它是电脑的后勤部门。它在控制器的控制下，和运算器、输入/输出装置交换信息。内存贮器常用磁芯或半导

体来构成存贮体，存贮容量较小，但存取速度快；外存贮器一般由磁鼓、磁带、磁盘等构成，存贮容量大，但存取速度较慢。

输入装置 是用来输入计算程序、原始数据及各种字符信息的装置。常用的输入装置有光电纸带输入机、卡片输入机、键盘、光笔显示器及模-数转换器等。

输出装置 是用来输出计算结果（信息或图形）的装置。常用的有阴极射线显示管、电传打字机、纸带穿孔机、自动绘图仪和数-模转换器等。

以上介绍的只是一般电脑的硬件。但是，要电脑替我们工作，还得告诉电脑做什么和怎样做。这就得靠包括各种程序的软件来帮忙。这两者的关系如同钢琴与琴谱一样，没有很好的琴谱，钢琴奏不出动听的乐曲；同样的道理，没有软件，电脑也不能“思考”。

电脑不懂得人类的各种语言。对于采用二进制数的电脑来说，它只懂“0”和“1”。这是由于电脑中的电子元件都是双稳元件，它只有两个状态（即电平“高”与“低”、开关的“开”与“关”、脉冲的“有”或“无”），具有两个稳定的运行方式。因此，最初人与电脑对话也就采用由“0”和“1”组成的一连串的机器码，这就是机器语言。例如，我们举一个小学生都会算的题目： $A = B + C \times D$ ，看看电脑是怎样运算的？我们可以把电脑存贮器里的四个单元看成是一幢房子的四个房间，房间号分别为A、B、C、D。其中B、C、D里各放了一个数据，要求计算 $B + C \times D$ ，将结果存放到A单元里去。电脑将分四步来计算这个题目（见图1.1—5）：

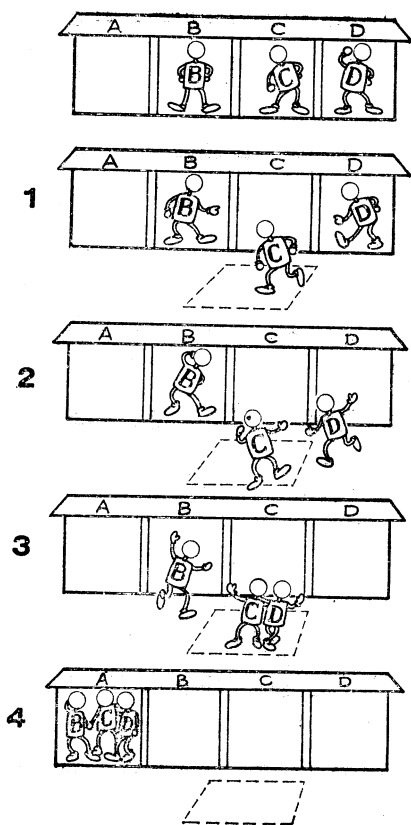


图1·1—5 电脑算题步骤示意图

码设定为1000、0101、0001、1101（称操作码）。内存贮器的各个单元也都进行编码，如果假定A、B、C、D四个房间的编码（也称操作数）分别为1100、1000、1001、1010，那么，以上四个步骤就可以写成：

1000	1001	（取数C）
0101	1010	（乘D）

第一步，从C单元取数C放到累加器里。

第二步，将累加器里的数与D单元的数相乘，也就是 $C \times D$ ，结果仍暂存在累加器里。

第三步，把累加器里的数与B单元里的数相加，将结果 $B + C \times D$ 暂存在累加器里。

第四步，把累加器里的数送到A单元里。

以上四步每一步都由一条指令加以控制。为了区别各种不同的操作，将各种指令都编上号码。这种号码是用二进制来表示的，如我们可以将“取数”、“乘”、“加”、“送数”的编

0001 1000 (加 B)

1101 1100 (送至 A)

这就是机器指令程序，也叫机器语言程序。电脑能“懂得”并执行它。然而，用机器语言表达一个意思必须用一大堆的“0”和“1”，不仅一般人难以理解和记忆，就是程序编制人员，日子一久也会被弄得莫名其妙。机器语言不仅不便于交流，用它编制程序还很容易出错。以后，人们就设法用一些助记符来代替机器语言中的操作码，如用 ADD 表示“加”，SUB 表示“减”等，这就是汇编语言。但汇编语言的语句与机器指令是一一对应的。用汇编语言编写的程序，其语句仍然很多，编写程序还是十分麻烦。为了解决这个问题，于是就发展了各种类似数学公式或简单句子的高级语言，如 FORTRAN、ALGOL 等。为了使电脑能在银行、财会、行政管理等领域里发挥作用，人们发明了 COBOL 这种进行数据处理的语言。为了便于使用者和电脑直接对话，又发明了交互式会话语言，BASIC 就是很典型的一种。这种语言对微电脑尤为适合，因为微电脑本身就是专为一入配备的。高级语言通用性大，易于理解、学习和掌握，为使用电脑提供了很多方便。当然，使用高级语言时，也需要配备“翻译”，即需要配备一套语言处理程序。这种存放在电脑内的语言处理程序，能根据语法和语义把各种程序设计语言翻译成电脑能懂得和执行的机器语言。语言处理程序除了汇编程序外，还有编译程序和解释程序。编译程序能对用高级语言写的源程序的各个语句进行识别和分析，根据情况考虑确定运算和处理的先后顺序。解释程序就不同，它不是一下子把所有源程序全部变换成目的程

序,而是按源程序语句的动态顺序逐句进行分析、解释,并立即予以执行,一直到结束。编译程序的特点是程序执行的效率高,FORTRAN、ALGOL 等就是经编译程序翻译的,它的缺点是在运行过程中不能修改。而解释程序由于是一边解释,一边执行,因而速度较慢。但它的优点是发现错误时,修改方便。把 BASIC 语言写的源程序变为机器能看懂的目的程序,就是通过解释程序逐句解释的。因此,它可以进行人机对话,随时修改。

电脑语言还在发展,近年来出现了专为儿童设计的适合于电脑教育的LOGO 语言,语句简单,使用方便。面向各种问题的专用语言也相继出现。据统计,目前世界上电脑语言不下1000种。下表列出了一些常用的高级语言及应用范围。

主要高级语言一览表

序号	语言名称	研制年代	应用范围
1	FORTRAN	1954	科学、工程计算
2	ALGOL	1958	科学、工程计算(欧洲)
3	LISP	1959	人工智能
4	COBOL	1960	事务管理,商业
5	APL	1960	科学、工程计算
6	BASIC	1965	电脑程序教育、电脑辅助教 育、一般应用
7	PL/I	1965	一般应用
8	PASCAL	1970	一般应用
9	ADA	1975	一般应用(美国政府标准)
10	PILOT	1975	电脑辅助教学
11	LOGO	1969	图形学和电脑教学
12	SNOBOL	现在	语言学、编辑、字符串处理 过程控制
13	FORTH		

人们还期待出现一种能在各种不同特点的语言和使用者之间建立联系的接口语言，如果能实现的话，人们与电脑交流就更为方便了。

3. 微电脑的结构特点

在了解电脑的基础知识之后，我们再来看微电脑的结构原理就比较容易了。微电脑与普通电脑的最大区别，就是其中央处理单元是由一块或几块大规模集成电路芯片构成的。

微电脑按集成电路中位数的多少进行分类，可分为4位、8位、12位、16位、32位等等。根据所配的存贮器容量的不同以及输入/输出接口和设备的多少，又分为单片式、多片式和位片式。从制造工艺来看有MOS和双极型两大类，而每一类又有若干分支。

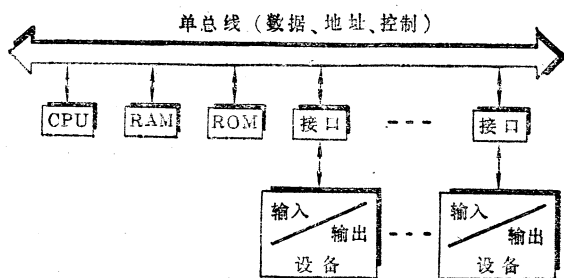


图1.1—6 典型的单总线微电脑系统结构图

图1.1—6所示的是一个典型的单总线微电脑系统结构图。单总线由地址总线、数据总线和控制总线构成。由于中央处理单元、存贮器和输入、输出装置都接在单总线上，所

以整个系统结构较简单，并且容易扩展。下面简单介绍微电脑四个组成部分：

中央处理单元 它包括控制器和运算器。微电脑的控制器的主要用来产生对微电脑各部件进行控制的微操作命令，这种微操作是比通常的指令更基本、更小的操作步骤。控制器中还包括指令寄存器和程序计数器。微电脑的运算器是进行算术和逻辑运算的部件。它还包括累加器和其他一些运算中存放数据的寄存器。除此之外，在一般微电脑系统中，主存储器的数据寄存器和存储地址寄存器也算作中央处理单元的一部分。

存储器 微电脑的存储器按其性质不同，可分成只读存储器 (ROM) 和随机存储器 (RAM)。只读存储器，顾名思义，只能将贮存的信息“读出”，而不能“写入”。当电源切断时，存放于只读存储器中的信息是不会消失的。因此，它所存储的往往是一些与管理和控制电脑有关的程序。而随机存储器就不同了，它能“读出”也能“写入”信息。当电源切断时，存放于随机存储器内的信息将全部消失，但它很容易地重新将信息“写入”。所以随机存储器中贮存的主要是数据。随机存储器按其性质的不同又可以分为静态和动态两种。静态随机存储器中的信息在电源不断的情况下是不会消失的。而动态随机存储器中的信息，在电源不切断的情况下，过了几个毫秒或微秒后也会消失，所以运算系统在使用这种随机存储器时，必须在信息还没消失之前，周期性地将其信息重新刷新，以免消失。虽然动态随机存储器有以上缺点，但由于它价格便宜、存取速度快和集成度高，还是为人们广泛采用。

输入/输出装置 微电脑的输入输出装置往往很简单,输入装置一般为键盘,输出装置通常是屏幕显示器和打印机。

总线 微电脑的总线包括数据总线、地址总线和控制总线。中央处理单元将数据传输到存贮器和输入/输出装置,或者由存贮器和输入/输出装置将数据资料传输至中央处理单元,都要通过数据总线。大部分的微电脑的数据总线为8位,即相当于有8条共用导线,并且是双向的。一般来说,电脑的字长位数也就等于数据总线的位数。地址总线用来在中央处理单元和存贮器以及输入/输出装置之间传送地址。在一般微电脑中,地址总线都是16位,但却是单向的。控制总线用来传送控制信号,它不但能控制中央处理单元的操作,也能控制存贮器和输入/输出装置的操作。控制总线一般也是单向的。

II. APPLE-Ⅱ 微电脑的使用与操作

APPLE-Ⅱ 微电脑是一种字长为 8 位的微型电脑。几年来, 经过不断发展与改进, 系统的软件、电路板与电源等方面日臻完善。目前, APPLE-Ⅱ 微电脑主要有标准 APPLE-Ⅱ 与 APPLE-Ⅱ PLUS 两种型号。由于后者将自动监控程序及浮点 BASIC 解释程序固化在主机的 ROM (只读存储器) 中, 并且具有允许电压在 110/220V 之间进行选择等特点, 因此, APPLE-Ⅱ PLUS 型微电脑操作较简便, 处理能力较强, 适用范围较广。我国引进和组装的 APPLE-Ⅱ 微电脑绝大多数是 APPLE-Ⅱ PLUS 型, 并命名为 DJS-033 型微电脑。

鉴于上述情况, 我们准备就 APPLE-Ⅱ PLUS 型微电脑的操作和使用作些简单介绍。本书所介绍的基本 BASIC 语言也是适用于 APPLE-Ⅱ PLUS 型微电脑的浮点 BASIC 语言。当然, 所介绍的内容基本上也适合于其他微电脑系统。

1. APPLE-Ⅱ 基本系统及扩展

图 1·2—1 是 APPLE-Ⅱ 微电脑的基本系统。它包括这样三个部分: 主机、与主机相连的键盘以及屏幕显示器 (简称 CRT)。

主 机

APPLE-Ⅱ 微电脑与一般微电脑一样, 也由下述五个

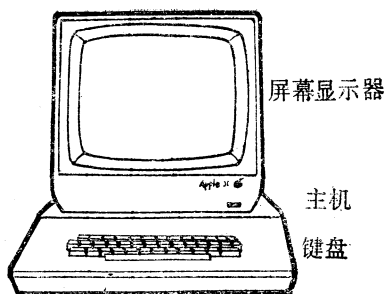


图1·2—1 APPLE-Ⅱ微电脑的基本系统

部分组成，即运算器、控制器、存贮器，输入设备和输出设备（见图1·2—2）。其中运算器与控制器往往集成在一块称为中央处理单元（CPU）的大规模集成电路芯片上。以CPU为核心，加上一些由大规模

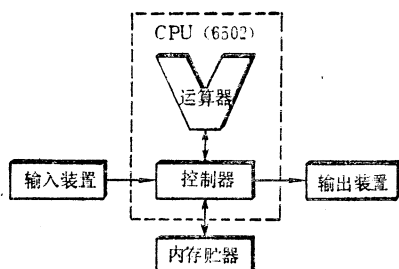


图1·2—2 APPLE-Ⅱ微电脑框图

模集成电路组成的内存贮器和输入/输出接口电路，就构成了APPLE-Ⅱ微电脑的主机。可见，主机是APPLE-Ⅱ的心脏部分。图1·2—3是APPLE-Ⅱ主机的外型，看上去象只扁平的盒子。

为了了解主机的构成，我们可以打开主机顶面盖板，将盖板后端稍稍往上提起，直到拉开两边的扣闩（见图1·2—4），然后将它往后推动，直至脱离机壳，整个主机内部结构就一目了然了（见图1·2—5）。

几十块集成电路和其他一些零件井井有条地排列在一块

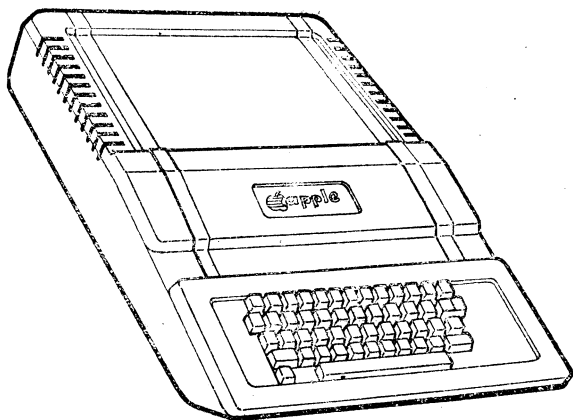


图1·2—3 APPLE-Ⅱ微电脑主机外型

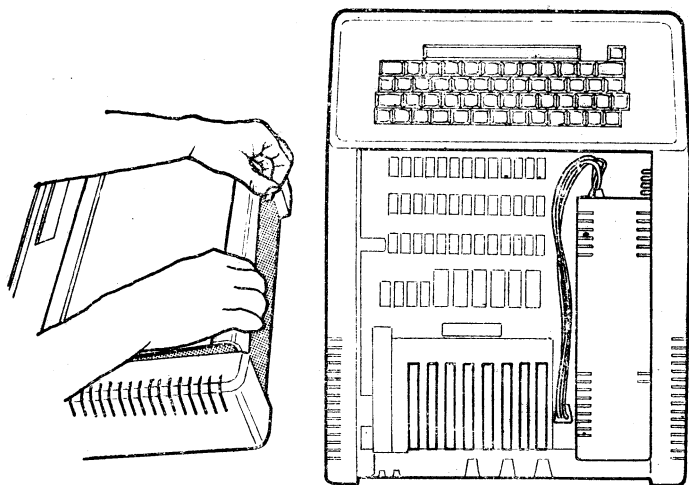


图1·2—4 打开主机顶面
盖板

图1·2—5 APPLE-Ⅱ主机内
部结构

绿色的印刷电路板上，这块电路板称作APPLE-Ⅱ的主机板。

在众多的集成电路中，有一块最大的集成电路，上面标有“6502”的字样（见图1·2—6），这就是APPLE-Ⅱ的CPU，它起着传输和处理算术与逻辑运算指令、数据的作用。“6502”是一块有40只引出脚的大规模集成电路。它每秒可进行50万次加减运算，懂得56条基本指令，会采用13种方式寻找内存地址，寻找地址的范围为64K字节（即65536个字节）。

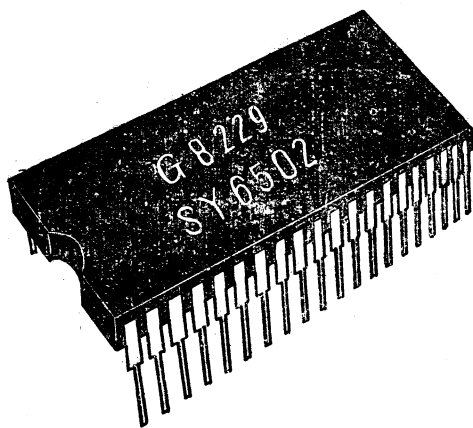


图1·2—6 APPLE-Ⅱ的CPU(6502)

在“6502”电路的正前方有6块24个引出脚的大规模集成电路，这些集成电路是APPLE-Ⅱ微电脑的一种内存贮器——只读存贮器（简称ROM）。也就是说，只能从ROM中“读”出信息，而不能“写进”信息，但ROM中已经存放的信息，一般是不会被改变或消失的。APPLE-Ⅱ主机的ROM有12K存贮容量，其中2K用来存放监控程序（即不论什么时候打开电脑，它们能自动指挥电脑运行），其余10K均存放浮点BASIC解释程序。

在 ROM 电路的正前方，有一个用白色（或黄色）线条画的矩形框，框内整齐地排列着24块（三排，每排8块）具有16个引出脚的大规模集成电路，这些集成电路是 APPLE-Ⅱ 微电脑的另一种内存贮器——随机存取存贮器（简称RAM）。也就是说，它们可以听凭操作者的指挥，既可将程序或数据“写”进去，也可将其中已存放的程序或数据“读”出来。但是，一旦电脑断电后，这一类存贮器中所存放的信息马上就会消失。这24块 RAM 电路共有 48K 存贮容量。

APPLE-Ⅱ 微电脑另有 4K 存贮容量是专用于输入/输出 (I/O) 的，所以整个主机的内存贮器为 64K。

读者也许会问：这里的“K”指的是什么意思？

这要从存贮器的功能说起。存贮器如上一部分介绍的，是一种存放数据或指令的部件，这些数据或指令都是以二进制数代码表示的。整个存贮器（不论 ROM 还是 RAM）都由许多存贮单元组成，每个单元是存放代码的最基本单位。在存贮器中，我们将每个存贮单元编上号码，这种号码就是存贮器的地址。这如同一个大旅馆内有一间间房间，而每间房内铺位是一定的，并且每间房间都标有房间号码（譬如几楼几号等等）。存贮器中的每个存贮单元就相当于一间房间，每个单元中所允许存放的“0”或“1”的代码个数相当于房间的铺位，而存贮单元的地址就相当于房间号码。APPLE-Ⅱ 主机中每个存贮单元允许存放一个8位的二进制数代码（即一个由8位“0”与“1”组成的代码），我们称这样一个8位二进制数代码为一个“字节”（Byte）；称每个代码的位数（字节长度）为“字长”。因此，APPLE-Ⅱ 主机内存贮器的字长为8位。在电脑术语中，将1024个存贮

单元(即可存放1024个字节)称为1K(即 2^{10})。APPLE-Ⅱ主机内存贮器具有64K容量,也就是说可以存放 $64 \times 1024 = 65536$ 个8位二进制数组成的代码。微电脑运算时,凡要读出或写入一个信息,必须先指出存贮单元的地址。在APPLE-Ⅱ微电脑中,内存贮器的存贮单元地址编号是从0到65535,并且用十六进制表示成\$0000~\$FFFF。我们也可以将APPLE-Ⅱ的内存贮器看作为一本书,共有256页,每页有256个存贮单元,譬如在页次\$15(即十进制的第21页)中,具有地址为\$1500到\$15FF之间的256个存贮单元,有关APPLE-Ⅱ内存贮器的安排,请见附录I。

在“6502”电路的后面,有一排(8个)40芯的插槽,从左自右分别编号为0*,1*,2*,3*,……7*。其中每个插槽内可插入一块接口电路卡,各种外部设备就通过接口电路卡与主机沟通。因此,这8个插槽起到了扩展APPLE-Ⅱ主机功能的作用。

对这8个插槽一般作了如下安排:

0*插槽 规定只能插语言接口电路卡(见图1·2—7)。插上语言接口电路卡可使APPLE-Ⅱ主机的内存随机存取存贮器扩展到64K(即增加16K RAM),并可在APPLE-Ⅱ PLUS电脑中使用整数BASIC语言及FORTRAN、COBOL、PASCAL、FOTH等语言。

1*插槽 一般习惯于插入打印机接口电路卡。接上打印机后,可使显示屏上的内容全都打印在纸上,以便长久保存。

2*插槽 一般习惯于插入Z-80语言卡。插上该语言卡后,APPLE-Ⅱ微电脑不仅能执行用“6502”的指令系

统编制的程序，还能执行用“Z-80”的指令系统编制的程序。

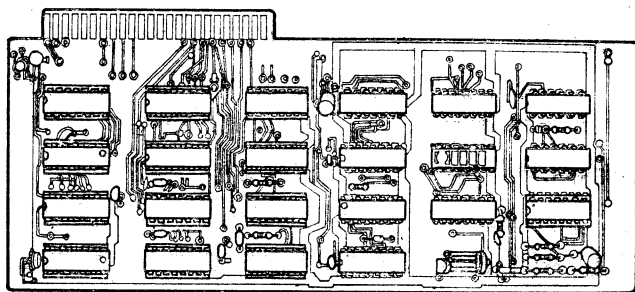


图1·2—7 16K RAM扩展卡

在有绘图仪的情况下，绘图仪接口电路卡一般也插在2*插槽上。接上绘图仪后(见图1·2—8)，APPLE-Ⅱ便能绘制各种图形和曲线了。

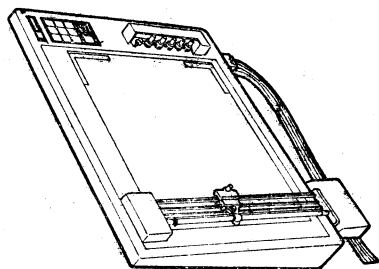


图1·2—8 绘图仪

3*插槽 一般习惯于插入汉字卡。插上汉字卡后，就可使APPLE-Ⅱ显示和打印汉字了。

6*插槽 一般习惯于插入软磁盘驱动器接口电路卡(见图1·2—9)。假如具有两台磁盘驱动器，一般规定将驱动器接口电路卡插入6*插槽(因为一块卡上可控制两台驱动器)，若有四台驱动器，通常就将两块驱动器接口电路卡分别插入6*与5*插槽之中。

一般说来，除0*插槽为语言(RAM扩展)卡专用外(绝

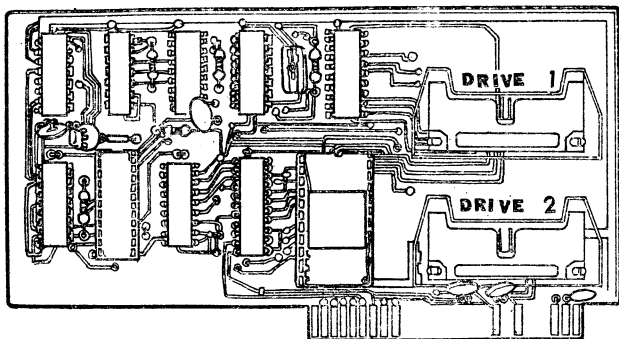


图1·2—9 软磁盘驱动器接口电路卡

对不准插入其他外部设备)，其他外部设备可以自由选择从1*到7*中的任意一个插槽。假如外部设备的接口电路卡不按上述习惯安排，就需要更改许多牵涉到外部设备号的通用软件。因此，我们主张读者在进行外部设备与主机连接时，考虑采用上述习惯安排，这样做通用性强，操作不易出错。

位于主机板左面的金属壳盒子，是电源供应装置（见图1·2—5）。它能将交流市电220V（允许输入交流电范围200~264V）转换成APPLE-II微电脑所需要的直流工作电源：+5V(2.5A)；+11.8V(1.5A)；-12V(250mA)；-5V(250mA)。

电源供应装置前方装有一个扬声器。该扬声器通过软件控制能够发出种种不同的声音，诸如：演奏音乐，发出“噪声”，甚至能“讲话”。

在主机板右后方有一个标有“GAME I/O”的插座，是用来连接游戏操纵杆的。游戏操纵杆主要用于游戏程序中输入连续的模拟量（见图1·2—10），游戏操纵杆控制着两个

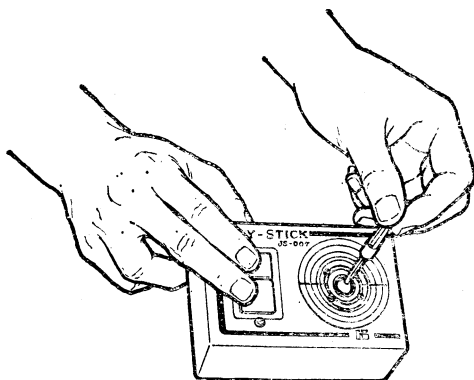


图1·2—10 游戏操纵杆

160k Ω 的电位器。摇动操纵杆，这两个电位器的值相应发生变化，从而达到改变输入电压的目的。游戏操纵杆与主机连接时，一定要将插头上带有标记的一端朝向插座的缺口处，如图1·2—11所示。

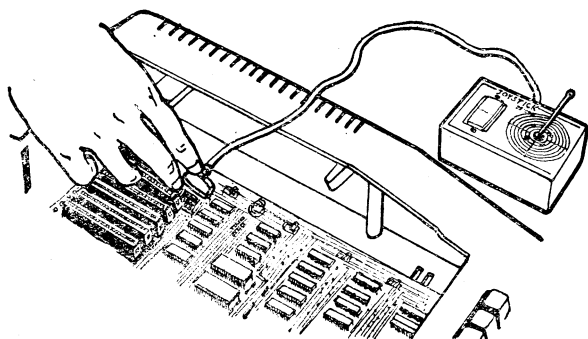


图1·2—11 游戏操纵杆与主机的连接

图1·2—12是 APPLE-Ⅱ 主机机箱的背面图。最右面是主机电源开关；旁边是电源连接插座。左边一排三个插孔，

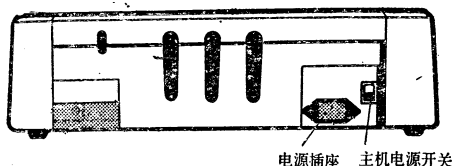


图1·2—12 APPLE-Ⅱ 主机机箱背面图

最左面的一个是屏幕显示器与主机的连接插孔；其余两个分别是卡式磁带机的输入、输出插孔。

图1·2—13是整个APPLE-Ⅱ主机机箱的解剖示意图。

键 盘

键盘是电脑的输入设备。APPLE-Ⅱ微电脑处理的数据和指令，大多数是经过键盘输入的。因此，键盘起到了人一机相联系的纽带作用。

图1·2—14是APPLE-Ⅱ微电脑的键盘，一共有52个键，分列为五排，与普通的英文打字机十分相似。但是击下这些键时，电脑并不能象打字机那样将字符直接打印在纸上，而只是将所键入的字符显示在显示屏上。键盘与主机相连，通过一些电路，将键入的字符信号转换成相应电信号。APPLE-Ⅱ微电脑系统采用ASCII码（国际上绝大多数电脑系统均采用ASCII码），也就是说每键入一个或几个键的组合，就可产生一个相应的ASCII码，整个键盘可组合成95个ASCII码（见附录Ⅱ）。

下面简单介绍各个键的排列及功能。

第一排是0到9十个数字键。为了区分数字“0”与英文字母“O”，键盘上将数字“0”上加一斜杠，标为“φ”，并且在屏幕上显示或打印机打印时，也都呈现“φ”。

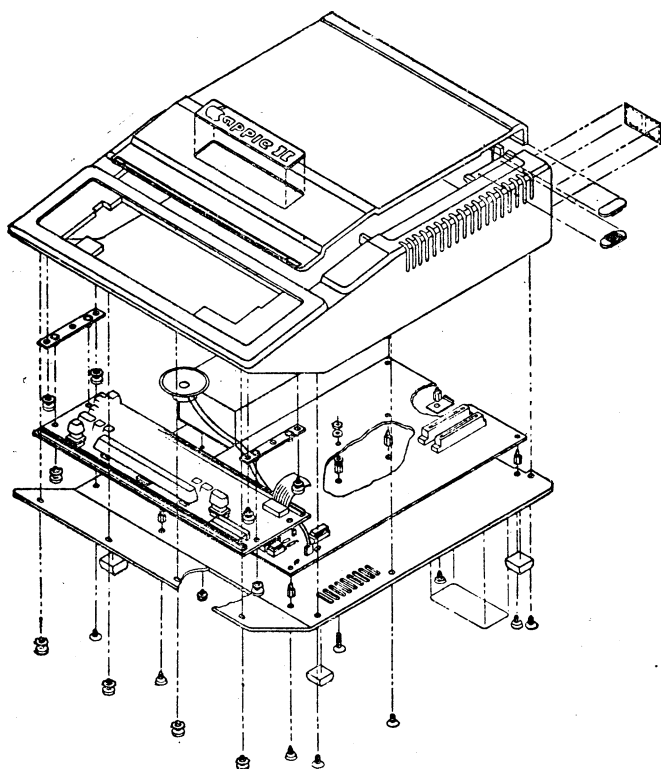


图1·2—13 APPLE-Ⅱ 主机机箱的解剖示意图

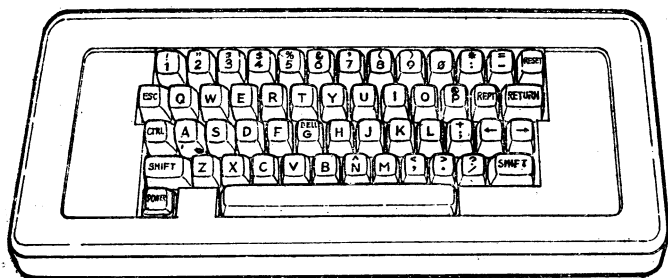


图1·2—14 APPLE-Ⅱ 微电脑的键盘

第二、三、四排分别排列着26个大写的英文字母键。对于标准APPLE-Ⅱ微电脑来说,它只能输入(并显示)大写的英文字母,而APPLE-Ⅱ PLUS型微电脑却与普通英文打字机相同,既能输入(并显示)大写英文字母,也能输入(并显示)小写英文字母,即平时按下字母键时,输入的是大写字母,当按下左下角的POWER指示键时,输入便呈现小写字母了。

此外,键盘上还排列着各种运算符键、标点符号键及一些特殊功能键。为了避免与英文字母“X”相混,键盘上将乘号标为“×”,在屏幕显示和打印机打印乘号时,也都呈现“×”。下面谈谈几个特殊功能键。

SHIFT 键 有些键上标有上下两个符号(例如 $\frac{\&}{6}$ 键),表明这个键具有两种功能。当直接键入 $\frac{\&}{6}$ 键时,屏幕上显示“6”,但在键入SHIFT键的同时再键入 $\frac{\&}{6}$ 键,就可显示出“&”。但也有例外,在键入SHIFT的同时再键入 $\frac{\text{BELL}}{\text{G}}$ 键,却并不显示“BELL”,而仍然显示“G”。

SHIFT 键在键盘的第四排左右两边对称出现。

REPT 键 “REPT”是英语单词“REPEAT”(重复)的缩写。若将任意一个字符按下的同时,按下REPT键,该字符便会在屏幕上重复出现(重复的速度为每秒10个字符),直到释放这两个键中任意一个或两个键为止。这个键对于编辑和修改程序特别有用。

←键 称作回格键(又称左位移键),每键入一次,光标便向左退回一个字符的位置,并将光标左移时经过的字符

从存储器中抹掉（尽管屏幕显示器上暂时仍保存）。但是该键不能使光标移至屏幕的最左一列上（欲使光标移至最左一列位置，可通过键入 **ESC** 键并同时键入 **B** 键来实现）。

→键 称作重打键（又称右位移键），每键入一次。光标便向右移动一个字符的位置，并将光标右移时经过的字符重新复制到内存存储器中，就如同从键盘上逐个键入这些字符键一样。

左、右位移键对于编辑和程序修改特别有用。

ESC 键 “ESC” 是英语单词 “ESCAPE”（抹去）的缩写。击一次 **ESC** 键后，在键入 **SHIFT** 键的同时再键入 **@** 键，结果会使整个屏幕显示器上的内容全部抹去。至于 **ESC** 键与其他键配合使用，能起到编辑与修改程序的作用，这将在下文叙述。

RETURN 键 相当于英文打字机的回车键，即要求电脑将光标移到下一行的最左端。**RETURN** 键作为每一程序行或每个命令的终结符号，它通知电脑：本行指令输入完毕；或者说，可以执行给出的指令了。

CTRL 键 “CTRL” 是英语单词 “CONTROL”（控制）的缩写。**CTRL** 键与 **SHIFT** 键相类似，必须与别的字符键配合使用。控制字符本身并不能显示于屏幕上，而是以电脑执行某个动作来回答。例如：

键入 **CTRL** 的同时，再键入 ^{BELL}_G 键，电脑会发生“嘟”的声音。

键入 **CTRL** 的同时，再键入 **C** 键，可以中断电脑的运行。操作者时常会应用这个命令让电脑停止运行，而使用

CONT 命令可使中断的程序继续执行。

键入 **CTRL** 键同时，再键入 **X** 键，电脑会将刚输入的一行程序从内存中抹去，并使光标立即返回到该行的最左端。

CTRL 键与其他键能够组合出许许多多控制命令。

RESET 键 这是一个复位键。一旦按下 **RESET** 键，电脑就停止当前正在处理的一切工作；释放后，电脑便起动一个复位操作过程。当程序运行不正常时，可以按 **RESET** 键，使电脑重新开始操作。为防止不慎误按了 **RESET** 键，而使当前的操作全被破坏，操作 **APPLE-II RLUS** 型微电脑必须先按 **CTRL** 键，再按 **RESET** 键，电脑才执行“复位”的指令。

POWER 键 这是一个表明 **APPLE-II** 微电脑主机已接通电源的指示键。在 **APPLE-II PLUS** 型微电脑中，该键还起着控制输入英文字母大、小写的作用，即按下 **POWER** 键后，输入的所有英文字母都为小写，否则为大写。

空格键 这是位于键盘最后一排的唯一没有标志的长条键，每击一下，光标向右移动一个字符的位置，屏幕上留下一个空格。

键盘与主机的连接 有些 **APPLE-II** 微电脑的键盘是与主机连体的，也有些是与主机分离的。对于后者，在主机板最前端有一个键盘连接插孔，键盘通过一根16芯扁平电缆线直接插入该插孔，实现主机与键盘的连接。

屏幕显示器

屏幕显示器是显示程序、数据和图形的一种输出设备。**APPLE-II** 微电脑的屏幕显示器既可用普通家用 **NTSC** 制的彩色电视接收机(可以用黑白电视接收机，但不能显示彩

色)，也可采用专门为电脑设计的屏幕显示器。由于专用屏幕显示器的清晰度比普通家用电视接收机高得多。因此，大多数APPLE-Ⅱ微电脑都配有专用的屏幕显示器。专用屏幕显示器又分为彩色显示屏和绿色显示屏两种。

一般来说，由微电脑输入屏幕显示器的信息不外乎三种型式，相应地也就有三种屏幕显示格式。

文本格式

文本格式能显示24行，每行最多能显示40个字符，所以满屏幕为960个字符。假如要显示的内容多于24行，那么，显示屏会自动向上翻动，即屏幕上部的字符逐行上移消去，而下部字符逐行显示。

屏幕上每个字符分别代表存贮区内每个单元的内容。字符组成为：26个大写英文字母、10个数字、28个专用符号，共64个字符。每个字符在屏幕上由5（宽）×7（高）点阵形成，各字符间留有一个点的间距。

当在主机的输入/输出插槽上插入特殊的80列字符接口电路卡后（该接口卡一般插于3*或4*插槽中）屏幕的文本格式就成为每行最多能显示80个字符了。

字符的显示有下述三种方式：

- （1）黑底白字的正常显示方式；
- （2）白底黑字的反显示方式；
- （3）字符的闪动显示方式。

低分辨率图形方式

APPLE-Ⅱ微电脑具有低分辨率和高分辨率彩色绘图功能。只要给出绘图命令和有关绘图程序^[注]就能在屏幕显示器上勾画出种种图形。由于绘图命令有低分辨率和高分

分辨率的区别，因此，屏幕显示方式也就有低分辨率与高分辨率两种。

所谓低分辨率图形显示方式，就是将整个显示屏的上部40行、40列用来显示图形，底部留出4行位置用来显示文字资料（见图1·2—15）。也就是说除底部4行外，整个显示屏

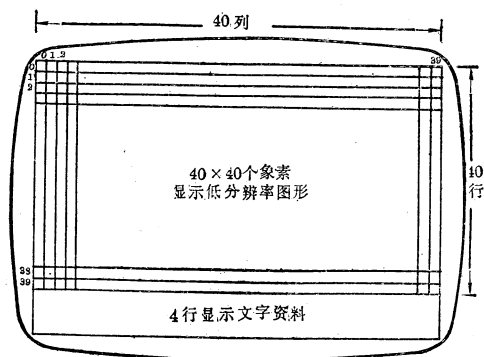


图1·2—15 低分辨率图形显示方式

被划分为 40×40 个像素（小方块），每个像素可以变换16种彩色，相邻两个像素间（即上、下、左、右小方块间）没有空格。若相邻像素有相同颜色，就连成一整块。图1·2—16是

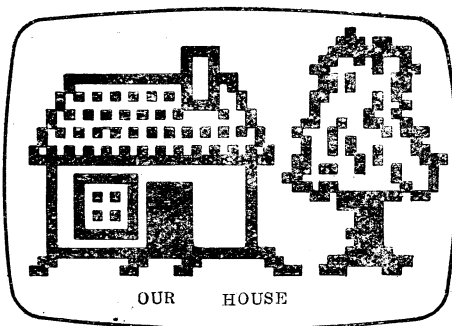


图1·2—16 低分辨率图形

〔注〕有关绘图命令参阅《APPLE-Ⅱ微电脑使用手册》。

低分辨率图形。

高分辨率图形方式

所谓高分辨率图形显示方式，就是将整个显示屏留出底部 4 行显示文字资料外，其余被划分成宽 280 列、高 160 行的高分辨率图形显示区，即 280×160 个像素的点阵区域（见图 1.2—17）。其中每个像素（即小点子）可以变换 6 种彩色。

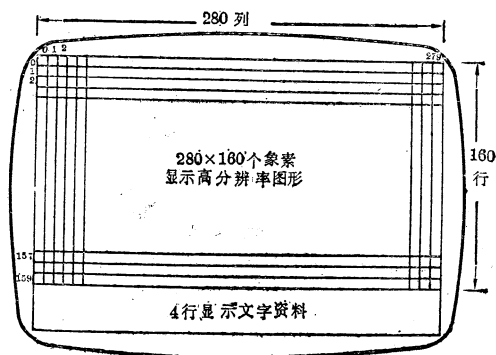


图1.2—17 高分辨率图形显示方式

通过一定的命令，还可将显示屏底部 4 行文字显示区域也用于图形显示，从而使整个屏幕扩展成宽 280 列、高 192 行的（即 280×192 点阵区域）全屏幕高分辨率图形显示。

图 1.2—18 是一幅高分辨率图形。

屏幕显示器与主机的连接

在 APPLE-Ⅱ 主机的背面有一个屏幕显示器插孔 (VIDEO OUT)，而屏幕显示器背面也有相应的 (VIDEO INPUT) 插孔。使用时只要用一根两头带有特殊插头的单芯屏蔽电缆将两个插孔连接起来就可以了（见图 1.2—19）。

如果使用特殊的 80 列显示电路卡，那么屏幕显示器就不

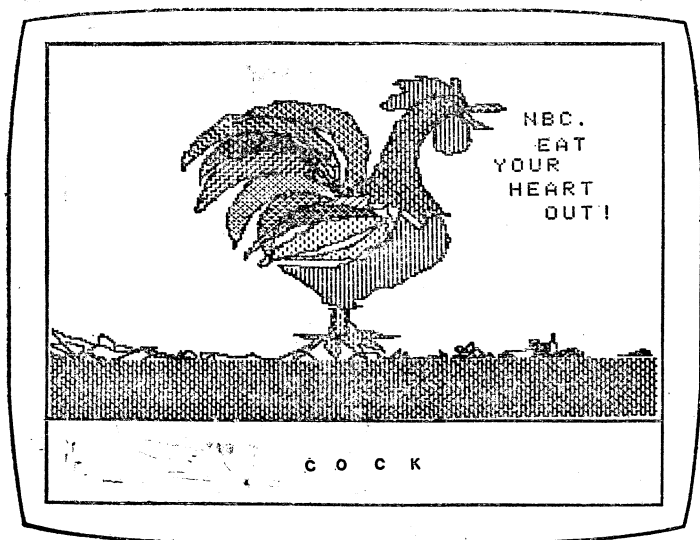


图1·2—18 高分辨率图形

能直接与主机相连，而应该与插于主机板上的80列显示接口电路卡相连，即通过该接口电路卡实现屏幕显示器与主机的连接。

主机、键盘、屏幕显示器构成了 APPLE-Ⅱ 微电脑的基本系统。

通过上述对主机情况的介绍，我们知道，APPLE-Ⅱ 微电脑的功能可以进一步扩展。譬如在基本系统上，可以通过主机板上的 8 个输入/输出插槽和有关的插口，加接卡式磁带录音机、软磁盘驱动器、打印机、实时音频频谱分析仪、绘图仪、（6、11、20兆）温氏磁盘、游戏操纵杆、IEEE-488 并行接口、RS-232C 串行接口、EPROM 写入器、通信接口、A/D 与 D/A 转换器等各种外部设备以及加接语言扩

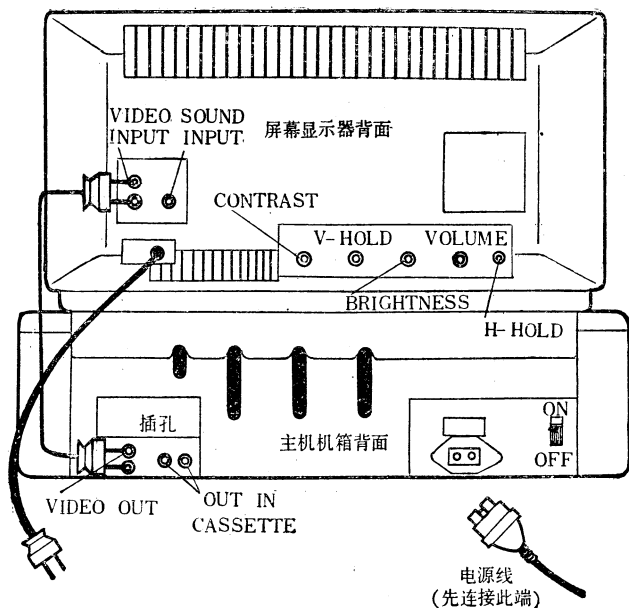


图1·2—19 屏幕显示器与主机连接

充卡、Z-80卡和汉字卡等等，来扩展 APPLE-Ⅱ 的功能。

下面，我们扼要介绍经常使用的卡式磁带录音机、软磁盘驱动器和打印机的操作方法以及它们与主机的连接。

卡式磁带录音机

卡式磁带录音机可以作为 APPLE-Ⅱ 微电脑的一种外存贮器。卡式磁带录音机能将信息存于磁带上，一旦需要可将磁带内的信息“读”到微电脑的内存中去。

卡式磁带录音机与主机的连接

在 APPLE-Ⅱ 主机机箱背面有 OUT (电脑输出) 与 IN (电脑输入) 两个插孔。我们可以用两根专用的单芯屏

蔽导线（导线两端均装有插头），将“OUT”与卡式磁带录音机的 MIC(MICROPHONE) 插孔相连；将“IN”与卡式磁带录音机的 EAR(EARPHONE) 或 MON(MONITOR) 插孔相连，就能够使 APPLE-Ⅱ 微电脑的内存贮器与卡式磁带机进行信息交换了（见图1·2—20）。

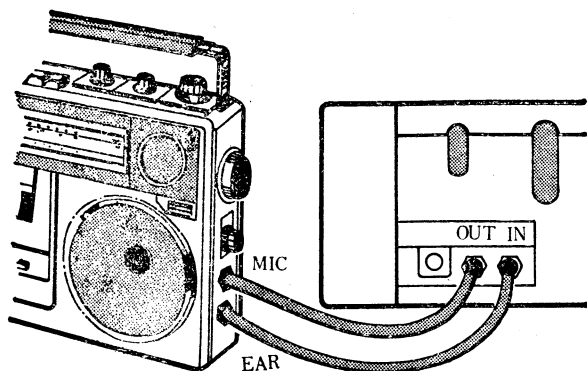


图1·2—20 APPLE-Ⅱ微电脑与卡式磁带录音机信息交换

卡式磁带录音机与主机交换信息的速度较慢，并且卡式磁带录音机的音量控制较为麻烦。

下面我们简单介绍一下卡式磁带录音机的音量控制开关的调节方法。

要使磁带中录的信息输入到主机的内存中去，必须使录音机具有适当的输出信号。假如音量控制开关调得过小，APPLE-Ⅱ微电脑将由于收不到信息而给出错误信息；反之，音量控制开关调得过大，也会破坏APPLE-Ⅱ微电脑的正常工作。为选择合适的信号输出，一般是通过多次调节和修正音量控制开关大小来实现的（即将音量控制开关由最小逐渐增大）。具体操作步骤为：

(1) 取出一盘录有电脑信息的磁带, 譬如标有“COLOR DEMOSOFT”的磁带(该磁带一般在添置APPLE-II系统时附带着), 装入卡式磁带录音机中, 并将磁带绕至开始贮存信息的地方, 并将录音机音量控制开关调至较小位置。

(2) 在APPLE-II微电脑键盘上键入LOAD命令。

(3) 按下卡式磁带录音机的PLAY键, 使磁带开始转动。

(4) 紧接着在APPLE-II微电脑的键盘上按下RETURN键。

这时, 显示屏上系统标示和光标立即消失, 大约过15秒钟以后, 会产生下述三种可能情况:

(a) 显示屏上出现“? SYNTAX ERROR”;

(b) 没有任何反应或出现“ERR”(“ERRERR”)信息;

(c) 听到“嘟”一声, 显示屏上没有显示任何信息。

对于情况(a), 不必调整卡式磁带录音机音量开关, 按上述(1)~(4)步骤重复一次, 若继续产生类似的错误信息, 试将卡式磁带录音机的读写磁头清洁一下, 或者换一盘磁带。

对于情况(b), 必须键入CTRL RESET键, 同时调整卡式磁带录音机音量, 再按照上述(1)~(4)步骤重复一遍。

对于情况(c), 表明一切正常, APPLE-II已经找到了程序的始端, 并正在将磁带上的信息输进内存。一般再过

15秒左右时间（具体时间长短取决于磁带上程序的长短），又会听到“嘟”的一声，显示屏上再度出现系统标示和光标。这时，就应按下卡式磁带录音机上的STOP键，停止磁带转动，并在音量控制开关上作一标记，以备今后再度进行信息交换时使用。这个音量控制开关的位置就是我们所寻找的合适的信号输出位置。

这时，如果在键盘上键入LIST RETURN命令，那么从磁带上输入到主机内存的信息将在显示器上显示出来。

使用磁带作APPLE-Ⅱ的外存，要特别注意保护好磁带，以防磁带中信息丢失。具体作法是：

（1）保持磁带表面清洁，不要用手指碰触磁带，也不要让磁带表面遭受其他污染，磁带用完后，应该及时装入专用的磁带盒内保存。

（2）磁带宜放置在阴凉通风处，切不可直接受到阳光照射。

（3）磁带贮藏的地方必须远离磁性物体（譬如不可放在电视接收机旁）。

（4）每盘磁带上应详细标明所贮存的信息的内容，以便于查找和调用。

软磁盘及软磁盘驱动器

软磁盘及其驱动器是APPLE-Ⅱ微电脑的外存贮器。它可以长期地保存电脑中暂时不用的信息，也可方便地将所贮存的信息传递给电脑。软磁盘信息贮存量大，传输速度快，操作十分简便，成为APPLE-Ⅱ微电脑系统中最重要 的外部设备之一。因此，几乎所有的APPLE-Ⅱ微电脑系统都

配上了软磁盘及其驱动器。

软磁盘驱动器根据所驱动的磁盘直径是 8 英寸 (1 英寸

$=2.54$ 厘米) 还是 $5\frac{1}{4}$

英寸而分成两种, 在 APPLE-II PLUS 型微电脑系统中, 一般都

配上 $5\frac{1}{4}$ 英寸软磁盘驱

动器, 称作 DISK II,

人们称它 5 英寸驱动器。

图 1.2—21 是 $5\frac{1}{4}$

英寸软磁盘驱动器的外

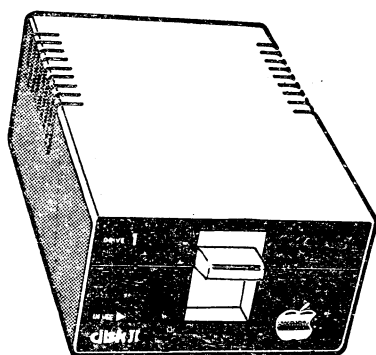


图 1.2—21 $5\frac{1}{4}$ 英寸软磁盘驱动器

型图。

一台 APPLE-II 微电脑系统至多可外接六台软磁盘驱动器。根据前文对主机板的介绍可知, 一块软磁盘驱动器的接口电路卡可以连接两台软磁盘驱动器。在仅用一台或两台驱动器的情况下, 接口电路卡习惯上插入主机板上的 6* 输入/输出插槽上; 如果用上两块或三块软磁盘驱动器接口电路卡的话 (即用三至六台驱动器), 那么, 这些接口电路卡依次插入主机板的 6*、5*、4* 输入/输出插槽上。我们称插在接口电路卡上方标有 “DRIVE-1” 插座上的那台软磁盘驱动器为第 1 号驱动器; 称插在接口电路卡下方标有 “DRIVE-2” 插座上的那台软磁盘驱动器为第 2 号驱动器 (见图 1.2—22)。对于使用多台软磁盘驱动器的情况, 每台驱动器的编号及与主机的连接法见下表:

驱动器台号	接口电路卡上的插座号	接口电路卡插入主机板上输入/输出插槽号
1	DRIVE—1	6 *插槽
2	DRIVE—2	
3	DRIVE—1	5 *插槽
4	DRIVE—2	
5	DRIVE—1	4 *插槽
6	DRIVE—2	

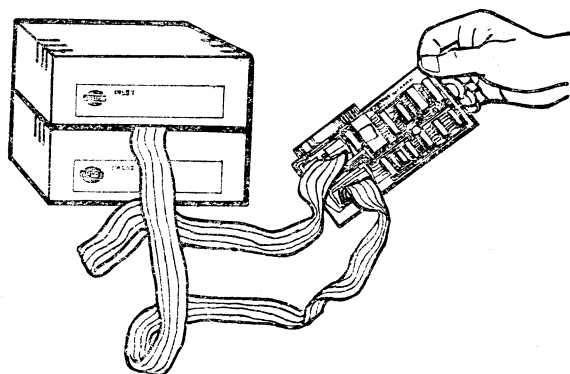


图1·2—22 软磁盘驱动器与接口电路卡

操作者最好在每台软磁盘驱动器的前端标上该驱动器在系统中的台号。

作上述规定是考虑到许多 APPLE-Ⅱ 软件都基于这个标准。

软磁盘驱动器与主机的连接

将软磁盘驱动器所引出的带状电缆插入接口电路卡的插

座上（若为1号驱动器插入“DRIVE-1”插座，2号驱动器插入“DRIVE-2”插座），在判断带状电缆将软磁盘驱动器与接口电路卡正确无误地连接好之后，遵照下述步骤将接口电路卡插入主机板的输入/输出插槽：

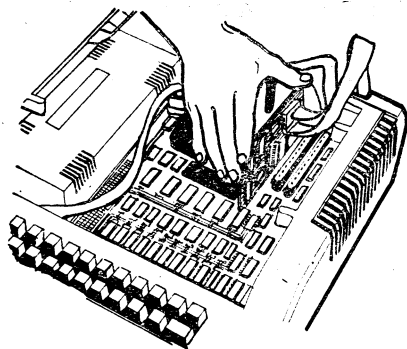


图1·2—23 将软磁盘驱动器接口电路卡插入主机输入/输出插槽

（1）关断主机壳背面的电源开关（注意：无论将什么外部设备的接口电路卡从主机输入/输出插槽中取出或插入，都必须首先切断电源，否则将损坏电脑系统）。

（2）取下主机的顶盖板。

（3）将连有带状电缆的接口电路卡上的梳状金属部分插入相应的插槽中（见图1·2—23），由于静摩擦力的缘故，接口电路卡会相当稳固地竖立在主机板上。

由于这些梳状金属部分是电路的连接点，因此，千万不要用手去摸触，以免将手上的油污或汗迹沾上去，造成接触不良。无论什么接口电路卡在插入插槽之前，都应用沾有酒精的棉球擦净这些梳状金属接点部分，如图1·2—24所示。

（4）当接口电路卡插入输入/输出插槽后，带状电缆应尽量往接口电路卡一端舒展，并使带状电缆面水平地通过主机机壳背部开口的平坦部分。一旦顶盖板盖上后，电缆被夹住了，再也不怕机壳外部分电缆的拉力了。

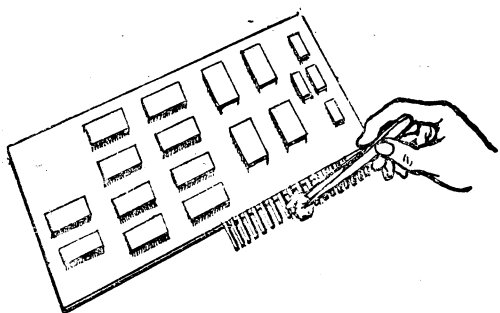


图1·2—24 用酒精棉球擦净接口电路卡的梳状金属接点

(5) 合上主机顶盖板, 将软磁盘驱动器安置妥当, 一般将它们安放在主机板的上面或近旁便于操作的地方, 如图1·2—25所示。

软磁盘驱动器不象APPLE-Ⅱ主机。它是一种机械装置, 装有电机等一些可移动的机械部分, 因此较易损坏。使用时必须安放妥当, 任何重压或碰击, 都可能使驱动器产生故障而不能正常工作。

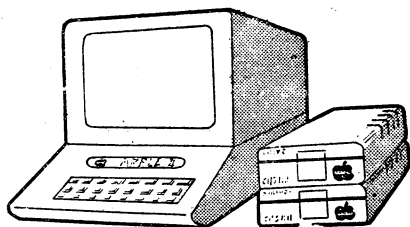


图1·2—25 将软磁盘驱动器放在适当位置

软磁盘驱动器很怕尘埃, 因此, 保持工作环境清洁是保护驱动器的重要措施。有时可在驱动器外面做个套子。

软磁盘驱动器不宜放在具有较强磁场的地方, 以免外部强磁场干扰和影响驱动器正常工作(尤其是对磁头产生不良影响)。

软磁盘驱动器能成为 APPLE-Ⅱ 微电脑的存贮器，其工作原理与卡式磁带录音机相似。驱动器内部的步进电机推动磁头运动，而磁头上的线圈的电流方向随着电脑中“0”、“1”脉冲不断变化，使磁力线方向不断改变(图1·2—26)，从而将磁信息记录在覆有磁性物质的圆形盘片上。

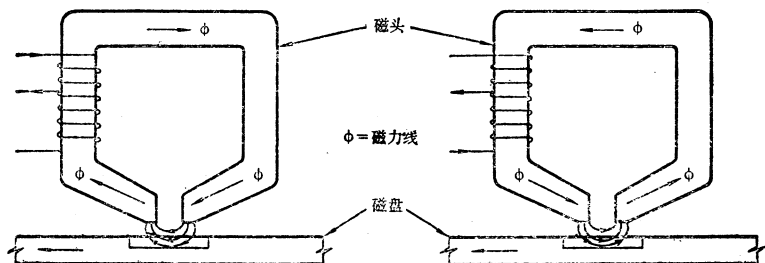


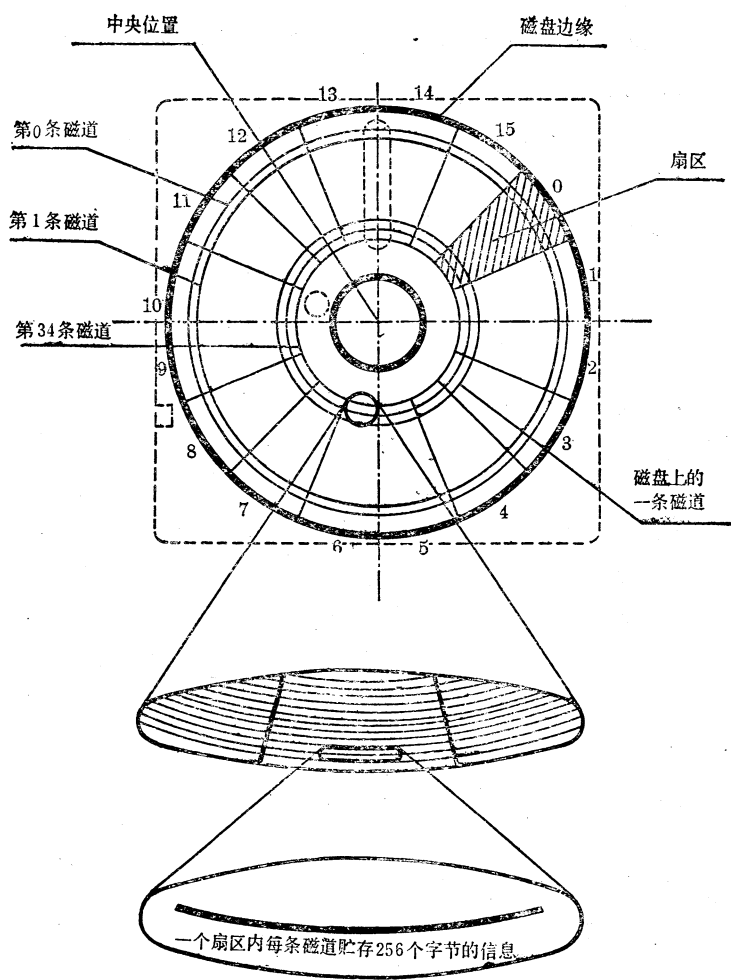
图1·2—26 软磁盘贮存信息的原理

APPLE-Ⅱ PLUS 型微电脑的软磁盘驱动器所使用的直径为 $5\frac{1}{4}$ 英寸的单面单密度盘片，每张可贮存143360个字节（即140K）的信息，几乎相当于主机内存（48KRAM）的三倍。

如同塑料薄膜唱片的声迹一样，软磁盘贮存信息的形式也是将整张盘片分划成一道道的同心圆磁道，但是磁盘上所贮存的并非是声音信息，而是由“0”与“1”组成的二进制数信息。

图1·2—27是根据 DOS3·3〔注〕版本描述的磁盘中信息

〔注〕：在配有软磁盘驱动器的 APPLE-Ⅱ 微电脑系统中，有一种特殊的程序，起着控制所有与磁盘驱动器相关的动作，这种程序称作磁盘操作系统 (DISK OPERATING SYSTEM)，简称 DOS。由于 DOS 有几种不同的版本（3·3、3·2及3·2·1等），因此，相应的操作步骤、磁盘中信息贮存格式等也就不同。本册子讨论的是 DOS3·3 版本，即比较新的版本。



格式化后盘片上磁道和扇区的划分
(.DOS 3.3版本)

图1·2—27 软磁盘中信息贮存的格式

贮存的格式。

整张盘片分为35条磁道（即35个同心圆），编号为0、1、2、……34；每条磁道分成16个扇区，编号为0、1、2、……15，每个扇区内存放256个字节的信息（即2048个“0”与“1”），因此，整张盘片共有 $16 \times 35 = 560$ 个扇区，总共可以贮存 $256 \times 560 = 143360$ 个字节的信息。由于每张盘片上将3条磁道留给DOS，1条磁道留给目录表，因此，实际提供操作者使用的仅31个磁道（即496个扇区）。

DOS管理着信息以一个扇区（即256个字节）为单位，一次存入软磁盘或从软磁盘取出。相应地，在主机的内存中也就开辟了一个256个字节的缓冲区。DOS还起着自动安排信息在磁盘中的存放区域（即放在哪一条磁道的哪一个扇区）的作用。操作者若要从主机内存中将信息存入软磁盘，或者从软磁盘取出信息，只要学会正确使用DOS命令就行了，不必人为地担心信息传递的方式及磁道与扇区的安排。

对于一张新买来的空磁盘，在第一次贮存信息时，必须按DOS3.3版本来规划它，即所谓格式化。有关如何格式化一张软磁盘的方法请参见附录Ⅲ。

软磁盘是一张外面覆有磁性材料的塑料圆盘片。为了保护盘片免受尘埃、油污等侵入，软磁盘被永久地封装在一个黑色正方形护套里，如图1·2—28所示。每张软磁盘护套边上都有一个写保护缺口，如果用胶纸将该缺口封起来（见图1·2—29），那么这张软磁盘就再也不能“写”进新的信息了，而只能将已存入的信息“读”出来，除非将胶纸撕去，缺口再度暴露，方能“写”入新的信息。

一张软磁盘能够贮存 $143360 \times 8 = 1146880$ 位二进制的

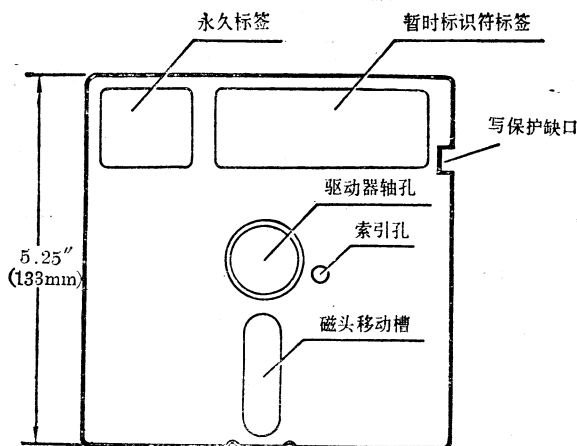


图1·2—28 5 $\frac{1}{4}$ 英寸软磁盘外型

信息，可见，贮存密度相当高。因此，一个完整的信息在盘片中所占的位置相当小。几粒灰尘、一个手指印都可能使资料发生错误。正由于软磁盘如此娇嫩，每个操作者都应注意保护好软磁盘（见图1·2—30）：

（1）握盘片时，手指只能接触黑色护套，不能接触到“磁头移动槽”上磁盘露出的部分。

（2）切不可将盘片放置在肮脏、多尘或油腻的地方，也不要让盘片表面积上灰尘，盘片用毕后，应及时插进专用纸套。

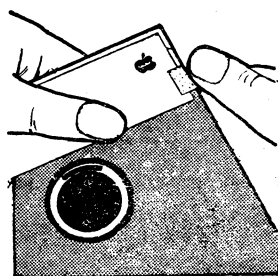


图1·2—29 用胶纸将写保护缺口封起来

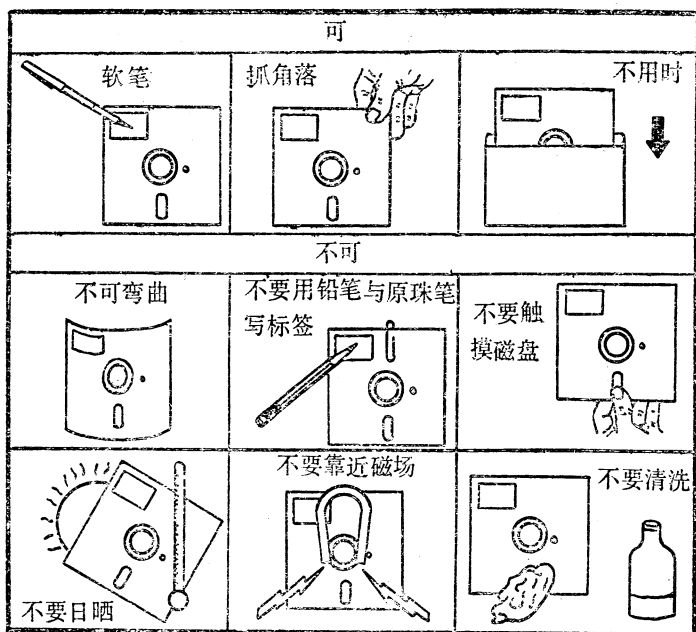


图1·2—30 保护软磁盘

(3) 尽管软磁盘的片基是塑料的，但过度弯曲仍会损伤盘片上的磁性覆盖层甚至损伤片基。盘片平时应保存在笔记薄型的盒子里，并且将盘片竖直放置（见图1·2—31）。

(4) 在盘片上写编号时，切不可用太尖、太硬的笔。一般可以将编号写在标签纸上，然后再贴到黑色护套上。

(5) 不能将盘片放在磁场较强的地方（例如电视接收机近旁），以免外界磁场破坏盘片内的信息。

(6) 软磁盘对温度过热或过冷都很敏感。因此，不能让盘片放在太阳下或其他热源附近，以免受热弯曲，使盘片受损。一般软磁盘的操作温度宜控制在10~52℃范围。

(7) 切忌用水或其他液体擦洗软磁盘。

可见, 保护软磁盘确实需要花点功夫, 但考虑到软磁盘具有信息贮存量大、使用起来远比卡式磁带方便、信息传输速度快等优点, 花点功夫保护盘片还是值得的。只要遵照上述各

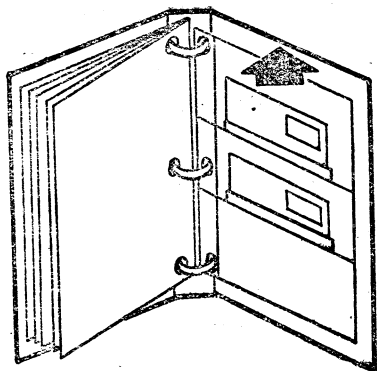
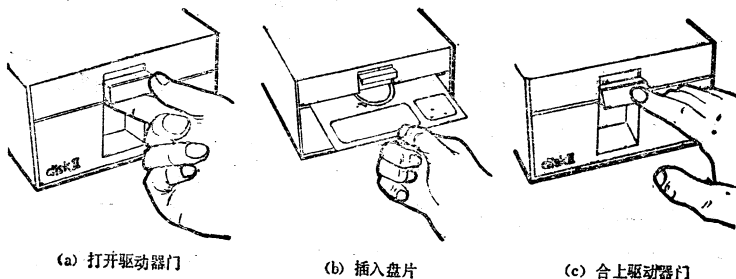


图1·2-31 将盘片插入薄型贮存盒中

项去做, 一般地说, 一张软磁盘平均上机使用时间可达40多个小时。也许读者会说: 寿命这么短啊! 其实, 每次从盘片内取出或存入信息一般仅操作几秒钟就行了, 40多小时的使用寿命, 相当于进行近万次存取操作, 寿命应该说是够长的了。

另外, 将软磁盘插入驱动器也必须十分小心。先将驱动器的门打开, 见图1·2—32(a), 然后将盘片水平地轻轻推入, 注意将有标签的那个面朝上。握盘片时, 右手拇指放



(a) 打开驱动器门

(b) 插入盘片

(c) 合上驱动器门

图1·2-32 将软磁盘装进驱动器

在标签纸上,让有椭圆形“磁头移动槽”的那一端先插入驱动器,见图1·2—32(b)。插入盘片时,切勿用力过猛,也不可使盘片发生扭曲。最后将驱动器的门合上,见图1·2—32(c)。

从驱动器内取出磁盘也必须十分小心。首先,打开驱动器的门,然后,小心翼翼地将盘片平稳地抽出。当驱动器的门被打开的同时,“读”、“写”磁头也就抬起来,离开盘片。因此,假如一张盘片插入驱动器一直未用,最好将驱动器的门打开,这样,可避免“读”、“写”磁头长期与磁盘接触。

应特别注意的是:当驱动器左下角“使用中”(IN USE)警告灯亮着的时候,千万不能将盘片抽出。否则,会使盘片内的资料遭到破坏,甚至使盘片本身受损。

打 印 机

打印机是一种常用的输出设备。它可将屏幕显示器所显示的各种字符、图形打印在纸上,供长期保存和使用。APPLE-Ⅱ微电脑系统所配备的打印输出接口电路卡可以配接三种型号的点阵式打印机:

(1) MX-80 Ⅲ 或 FX-80 (行宽80,每行由 480×8 或 960×8 点子组成,每秒打印80字符)。

(2) MX-100 Ⅲ 或 FX-100 (行宽136,每行由 816×8 或 1632×8 点子组成。每秒打印100字符)。

(3) CP-80 (与MX-80 Ⅲ F/T兼容)。

这三种打印机都采用双向打印方式,能接受96个ASCII码,能打印英、法、德等多种文字,也可打印汉字。

图1·2—33是一台典型的80列打印机的外型图。

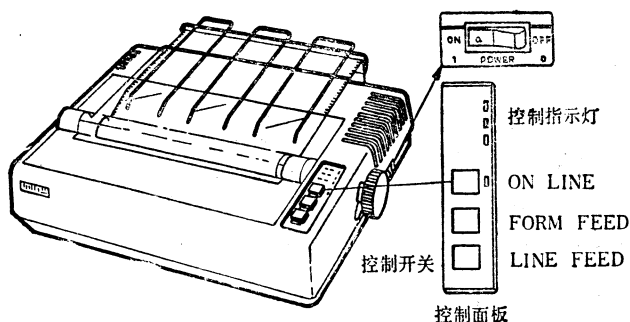


图1·2—33 80列点阵式打印机

由图可见，该打印机控制面板上有三个指示灯和三个控制开关，它们的功能分别为：

(1) “ON LINE”指示灯——灯亮，表明电源接通，数据可以传输至打印机。

(2) “READY”指示灯——灯亮，表明打印机准备接收信息。

(3) “PAPER OUT”指示灯——灯亮，表明打印纸即将用完。

(4) “ON LINE”开关——按下一次，打印机脱离“在线”作业，不能从主机接收信息；再按一次，打印就恢复“在线”作业，电源一接通，打印机就进入“在线”作业状态。

(5) “FORM FEED”开关——电源一接通，按下该开关，打印机便自动转到适当的印字位置。

(6) “LINE FEED”开关——打印机脱离“在线”作业时，按下该开关，打印纸会向前跳一行。

打印机与主机的连接

打印机与主机的连接方法同软磁盘驱动器与主机连接方法相似，也须先切断主机及打印机的电源，将电缆线一端与打印机相连，另一端插入接口电路卡，再将接口电路卡插入主机板的1*输入/输出插槽之中（图1.2—34）。

打印纸的安装

（1）取下打印机的前盖，使打印刻度尺转向操作者，并将打印纸导引轮移至轮轴中间的位置（图1.2—35）。

（2）翻起两边的导引轮轨的盖子（图1.2—36）。

（3）将打印纸穿过打印纸架与塑料滚动轮之间的窄缝，并插入打印机空隙，让打印纸绕过导引轮伸至打印机的前方（图1.2—37）。

（4）将打印纸两边的两排导引小孔套进导引轮轨的圆齿上，并压下两边的导引轮轨的盖子，然后转动手轮使打印纸卷上来。

（5）合上打印机前盖。由于打印纸往往采用折叠式

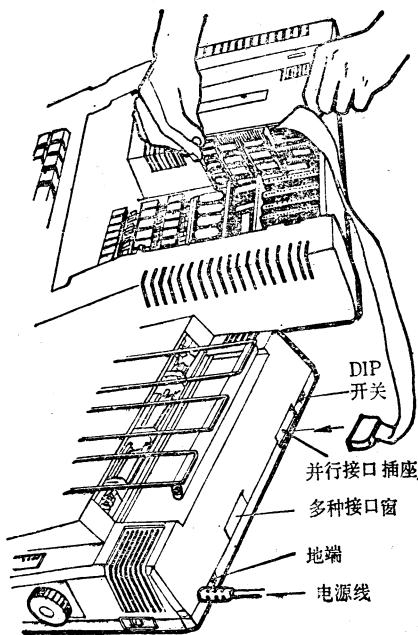


图1.2—34 打印机与主机的连接

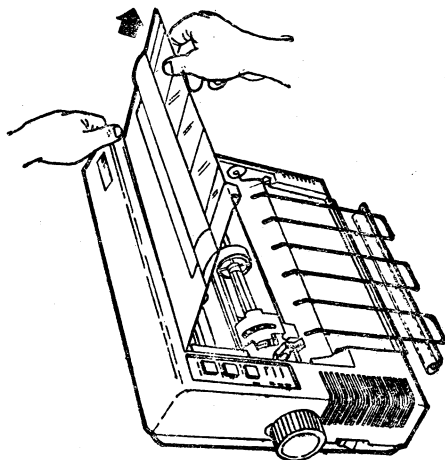


图1·2—35 取下打印机前盖
打印纸导引轮

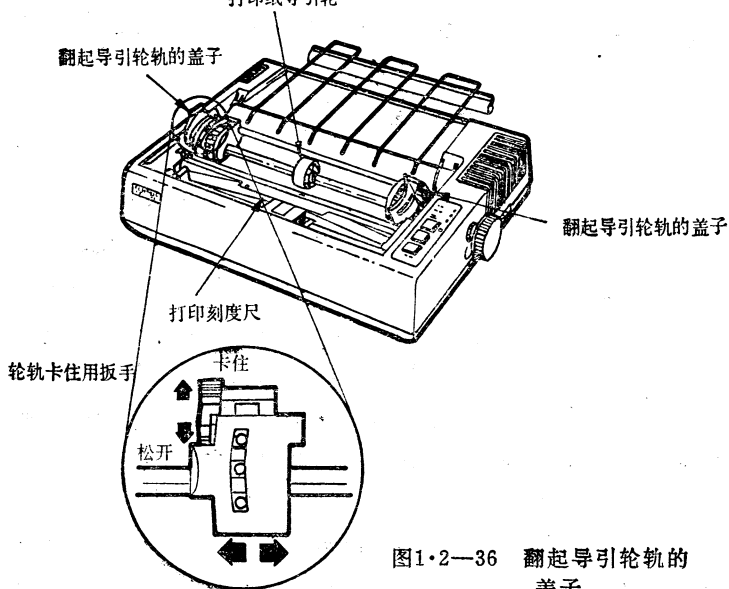


图1·2—36 翻起导引轮轨的
盖子

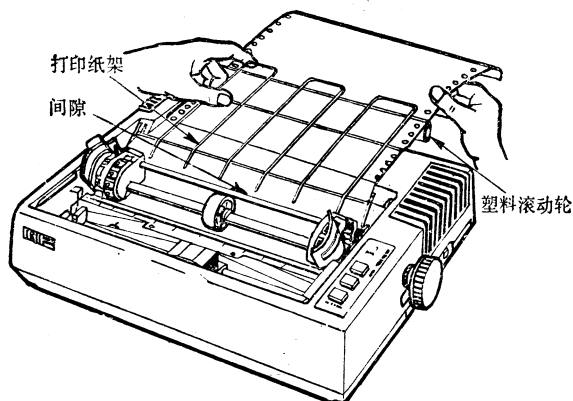


图1·2—37 装上打印纸

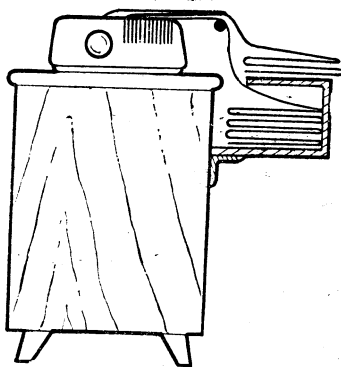


图1·2—38 打印机架

的，因此，最好做个打印机架（见图1.2—38），使未使用过的打印纸与已打印好的打印纸分别折叠好。

打印机的文本打印格式一般地说与显示屏上的显示格式相同，也就是说，命令打印机将输入的程序列出清单的话，打印出来的字符也同显示屏一样，每行至多40

个字符。要使打印格式作出多种变化，可以通过软件的方法来实现。下面简单介绍几种常用的打印格式。

打印行宽的改变

假如要打印机打印出来的程序每行超过40个字符，可以先从键盘上键入 PR*1，将打印机接口电路卡接通，然后将

希望设置的行宽数(即每行允许打印的字符数)存入“1656”加上“插槽号”(一般为“1”)的十进制地址单元之中。例如:设置行宽为60(即每行最多允许打印60个字符),可以这样编制程序:

```
10 SLOT=1
20 REM LIST-MODEL FOR 60 COLUMNS
30 POKE 1656+SLOT, 60
```

这样,打印机开列程序清单时,行宽就变成60了。但是这时候由于行宽超过40,屏幕显示器将不予显示了。

打印小写字母

假如要在语句中打印一些小写字母,那么可将这些小写字母夹在两个控制字符(CTRL W)之间而写入程序(这种控制字符在程序中采用扩展 BASIC语言中的CHR\$(23)函数来表达)。

例如要打印“Small letters PRINTING”,可以这样编写程序:

```
10 PR#1
20 PRINT "S"; CHR$(23); "MALL LETTERS";
CHR$(23); "PRINTING"
30 PR#0
40 END
RUN
Small letters PRINTING
```

放大或缩小打印字符

假如要将打印的字符放大或缩小,也可通过打印机的控制命令来实现,即在程序中采用扩展 BASIC语言中的CHR\$(n)函数。

设置放大字符的打印, 采用 CHR\$(14) 语句 (相当于 CTRL-N);

清除放大字符的打印, 采用 CHR\$(20) 语句 (相当于 CTRL-T);

设置缩小字符的打印, 采用 CHR\$(15) 语句 (相当于 CTRL-O);

清除缩小字符的打印, 采用 CHR\$(18) 语句 (相当于 CTRL-R)。

假如要打印放大的字符 “NORMAL” 和缩小的字符 “CONDENSED”, 可以这样编制程序:

```
10 PR#1
20 PRINT "NORMAL"
30 PRINT CHR$(14); "  &"
40 PRINT CHR$(15); TAB(25); "CONDENSED"
50 PRINT CHR$(18)
60 PR#0
70 END
RUN
NORMAL  &  CONDENSED
```

改变行间距打印

打印机打印的点与点之间的距离为1/72英寸, 标准行间距为1/6或1/8英寸 (取决于打印机中 DIP 开关的设置)。若要改变行间距打印, 也可通过打印机的控制命令来实现。

假如要求打印机行间距变成 $m/72$ 英寸 ($1 \leq m \leq 85$), 可以使用 “CHR\$(27); CHR\$(65); CHR\$(m)” 的语句。

例如要打印机打印 5 行(变行距) “VARIABLE LINE

SPACING”，可以这样编制程序：

```
10 PR#1
20 FOR I=4 TO 12 STEP 2
30 PRINT CHR$(27); CHR$(65); CHR$(I);
40 PRINT "VARIABLE LINE SPACING"
50 NEXT I
60 PR#0
70 END
```

2. 如何上机操作

第一次坐在电脑前，不免会有些不知所措之感。该从哪儿动手呢？

下面就上机操作的主要步骤作一简单介绍，给读者引个路。必须指出的是：学习电脑技术，一定要动手实践，多上机操作，并仔细留意操作中遇到的各种现象；每给出一条命令后，要观察和分析电脑的反应；要善于总结经验，勇于实践探索。这样，用不了多久，你一定能够熟练掌握电脑操作技术的。

正确作好主机与各种外部设备间的连接

有关主机与各种外部设备间的连接，前文在叙述基本系统与扩展时已分别予以一一介绍了。当外部设备与主机正确无误地连接以后，你所操作的 APPLE-Ⅱ 微电脑系统就构成了。图1·2—39所表示的就是一个常见的 APPLE-Ⅱ 微电脑系统的连接示意图。

将主机以及所有需要独立供电的外部设备的电源插头插

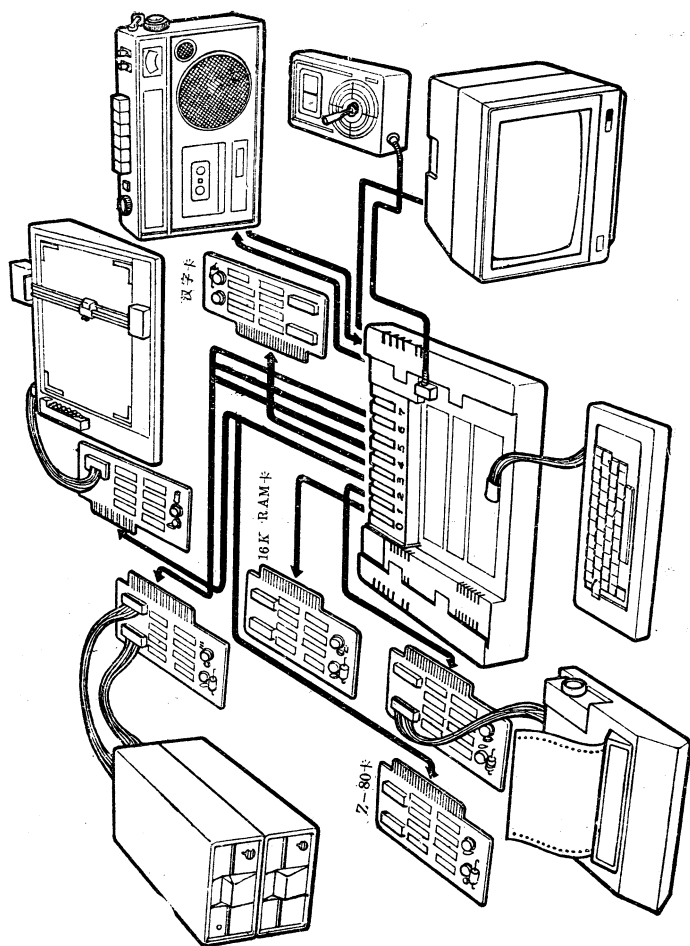


图1·2—39 常见 APPLE-I 微电脑系统连接示意图

入电源插座上。

接通电源

仔细检查了各个外部设备与主机之间的连接以后，先打开屏幕显示器的电源开关（见图1·2—40），大约相隔半分钟，再打开主机的电源开关（见图1·2—41）。

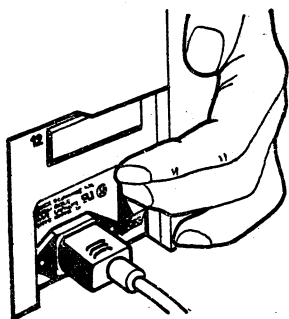
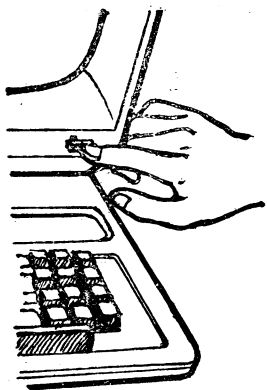


图1·2—40 打开屏幕显示器开关 图1·2—41 打开主机电源开关

对于 APPLE-Ⅱ PLUS 型微电脑来说，由于具有自动启动监控程序，因此，主机电源一接通，操作者就会听见“嘟”一声，键盘左下角的 **POWER** 指示键就点亮了，显示屏上会出现如图 1·2—42 所示的情况。“J”符号提示操作者，目前电脑正处于浮点 BASIC 语言状态，假如要电脑工作的话，请操作者输入用浮点 BASIC 语言写的程序或指令；“█”闪烁光标表示起始显示位置就是目前光标所处的位置。

但是，有时候接通主机电源后，会出现一些异常现象，这就要求我们采取一些相应的措施（见下表）。

开机后可能发生的情况	采取相应措施
1. 没有听到“嘟”声音, POWER 指示灯不亮 (也许主机未通电)	关断电源开关后, 重新再打开
2. 再次打开电源开关后, POWER 指示灯不闪动 (可能电源插座内无电或与外部设备连接有问题)	拔下电源插头, 检查插座是否通电。进一步检查各种外部设备与主机连接是否正确, 接头处接触是否良好
3. 排除主机未通电的可能后, APPLE-II 仍未动作 (可能主机本身有故障)	请专业人员检查、修理
4. 屏幕上无显示 (可能是主机与屏幕显示器连接不良)	检查屏幕显示器与主机间连接是否正确, 接触是否良好
5. 屏幕上无光标, 驱动器发出噪声, “INUSE”灯点亮	键入 CTRL RESET 后, 会出现 “  ”
6. 若主机3 *输入/输出插槽内插入了80列字符显示接口电路卡, 屏幕上无显示。	键入 P R ^{SHIFT} ₃ RETURN 后, 会出现 “  ”
7. 屏幕上出现 “*  ”, 表明主机处于编辑监控状态	键入 CTRL B RETURN 后, 会出现 “  ”

启动各种外部设备

在 **APPLE-II** 微电脑系统带上各种外部设备的情况下, 启动这些设备, 必须通过一些特殊的命令, 下面分别予以叙述:

将录在磁带上的信息输入主机内存中

按上文介绍那样, 准确地将卡式磁带录音机和主机连接好, 并调节音量控制开关至适当位置, 然后进行以下步骤:

(1) 将需要存入主机内存的磁带装入卡式磁带录音

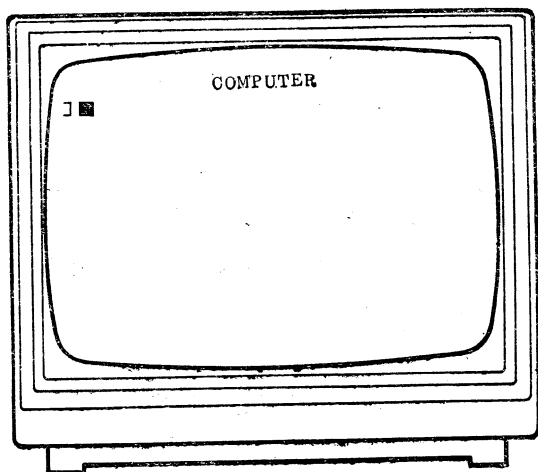


图1.2—42 刚开机时显示屏情况

机，并反绕至程序起始的位置；

(2) 在键盘上键入 **LOAD** 命令；

(3) 按下卡式磁带录音机上的 **PLAY** 键，使磁带开始转动；

(4) 立即在键盘上按下 **RETURN** 键。

这时，光标立即消失，几秒钟后，APPLE-Ⅱ发出“嘟”的一声，表明磁带信息已开始输进主机内存。经过一段时间（时间长短取决于磁带所存信息的多寡），APPLE-Ⅱ再次发出“嘟”的声音，显示屏上就出现“**LOAD**”，磁带信息传输结束；

(5) 按下卡式磁带录音机上的 **STOP** 键，使磁带停止转动；

(6) 在键盘上键入 **RUN** 命令和按下 **RETURN** 键后，

从磁带传送入 APPLE-Ⅱ 的程序便开始执行。

将内存信息存入磁带中

(1) 将需要存放信息的磁带装入卡式磁带录音机中, 并使磁带绕至起始端;

(2) 同时按下卡式磁带录音机上的 **PLAY** 键和 **RECORD** 键;

(3) 在 APPLE-Ⅱ 键盘上键入 **SAVE** 命令;

(4) 当击 **RETURN** 键后, APPLE-Ⅱ 发出“嘟”的一声, 表明信息已开始存入磁带, 待内存信息全部存入磁带后, 又会发出“嘟”的一声, 以通知操作者, 过程结束;

(5) 按下卡式磁带录音机上的 **STOP** 键, 磁带便停止转动。

将 DOS 存入主机内存中

DOS, 即磁盘操作系统, 是一种具有记录文件, 保存文件, 检索文件以及能完成许多其他功能的程序。APPLE-Ⅱ 微电脑要使用软磁盘作为外存贮器, 就必须将 DOS 的全部功能加到浮点 BASIC 语言中去。这种将 DOS 从磁盘中转移到主机内存去的过程, 我们称为“引导 DOS”。DOS 一旦引入后, 浮点 BASIC 语言就处在 DOS 支持下工作, 从而大大地扩充了 APPLE-Ⅱ 的功能。

APPLE-Ⅱ 微电脑处于不同的情况下, 有各种不同的引导 DOS 的方法:

(1) 在主机未通电之前, 将存有 APPLE DOS 的系统主磁盘 (SYSTEM MASTER DISK) 插入软磁盘驱动器中, 但所插入的那台软磁盘驱动器必须是位于主机板的输入/输出插槽号最大, 而系统中台号最小 (即 DRIVE 1) 的。

假如系统配有两台软磁盘驱动器，接口电路卡必须插在6*插槽内，并将系统主磁盘插入1号软磁盘驱动器内。

关上软磁盘驱动器门后，打开主机电源开关。这时，软磁盘驱动器的“IN USE”红灯亮起来了，并发出“呼呼”的转动噪声。过一会，软磁盘停止转动，“IN USE”红灯熄灭，而在屏幕显示器上出现了“J”，表明DOS引导工作完成，电脑处于DOS支持下的浮点BASIC语言状态，等候操作者给予命令。

(2) 当APPLE-II处于浮点BASIC语言(或整数BASIC语言，提示符为“>”)状态下，不必切断电源，也可引导DOS。只要从键盘上键入PR*6或IN*6命令(当然必须将系统主磁盘插入6*输入/输出插槽中接口电路卡的1号驱动器内)，再按RETURN键。软磁盘驱动器也会产生上述相同的过程，而完成DOS的引导工作。

(3) 当APPLE-II微电脑处于监控系统时(即显示屏上出现提示符“*”)，不用切断电源就能引导DOS，只要从键盘上键入C600G命令(当然系统主磁盘是插入6*输入/输出插槽的接口电路卡的1号驱动器内)，再击RETURN键。磁盘驱动器也会产生上述相同的过程，从而完成DOS的引导工作。我们还可以采用击CTRL键的同时击K键或P键，再击RETURN键来完成DOS的引导工作。

待DOS引导工作完成后，假如操作者想要了解一下这张系统主磁盘内究竟存放了一些什么程序，可以使用CATALOG RETURN命令，显示屏上便会显示出文件的名称。

一般操作者都将系统主磁盘进行复制保存，以防万一系

统主磁盘丢失或损坏。关于复制磁盘以及内存与软磁盘间的信息交换方法，请参阅有关 APPLE-Ⅱ DOS 文件管理手册。

打开和关闭打印机

当打印机与打印接口电路卡正确连接，并将该接口电路卡插入主机 1 *输入/输出插槽后，接通打印机电源。但这时打印机并不能使用，还需要键入命令加以控制。打开和关闭打印机的具体步骤为：

(1) 首先判断一下想要打印的程 序 是 否 确实存在于 APPLE-Ⅱ 内存中；

(2) 当光标移到一个空白行的最左端位置时，从键盘上键入 PR*1；

(3) 接着键入 LIST 命令（该命令将不被显示于屏幕上，却会打印在纸上）；

(4) 当按 RETURN 键以后，打印机就被启动了，整个内容也就被打印出来了。

但从节约打印纸张和加快电脑运行速度考虑，并不需要打印机一直工作，那么应及时关闭打印机。

可采用从键盘上键入 PR*0 命令，从打印机上收回输出权，并还给屏幕显示器（该命令同样被打印在打印纸上，而不显示于显示屏上）。

总之，在主机板的八个输入/输出插槽内接有各种外部设备的情况下，操作者凡要将输入或输出权分别从键盘或显示屏转移到其他外部设备时，可使用 PR* 或 IN* 命令。

PR*n（n 为 1 ~ 7 的正整数，即 7 个插槽口编号）表示在该命令以后，输出信息须从与插在 n* 输入/输出插槽上

接口电路卡相连的外部设备上送出。

PR*0 表示在该命令以后，输出信息仍由显示屏显示，即显示屏从外部设备上收回输出权。

IN*n (n 为 1~7 的正整数，即 7 个插槽的编号) 表示在该命令后，输入信息须从与插在 n* 输入/输出插槽上接口电路卡相连的输入设备上取得。

IN*0 表示在该命令以后，输入信息仍通过键盘键入，即键盘从外部设备中收回输入权。

如果在主机板的 n* 输入/输出插槽上并没有插入任何接口电路卡，那么键入命令 PR*n 或 IN*n 会使整个计算机系统停止工作。要使电脑恢复工作，可以按下 CTRL 和 RESET 键。

键盘输入

键盘是微电脑最基本的，也是最重要的输入设备。操作者给电脑的各种命令，需要电脑运行的各种程序主要地通过键盘输入。因此，正确、熟练地进行键盘输入操作，是一项十分重要的技能。

键盘操作必须注意这样几点：

(1) 在输入一个新的程序之前，必须从键盘上输入“NEW”并按 RETURN 键。该命令指示电脑自动清除内存中已有的所有程序（但屏幕上所显示的内容仍暂存），从而为即将输入的新程序腾出内存，不至于使新输入的程序与内存中原有的程序相互干扰。为了查证内存是否清除干净，可再从键盘上输入“LIST RETURN”命令。若内存已清除干净，屏幕上应无程序显示。这时，我们便可放心地输入新的程序了。

(2) 从键盘上输入程序的格式: 先输入语句标号, 再输入语句内容。每一语句输入完后, 必须按下回车键 **RETURN**。

输入的源程序, 除字符串之外, 凡是字母都必须大写。

输入程序时, 字符间一般可连续输入, 不用空格 (除引号内字符串需适当空格), 并且语句的标号也可以不按大小顺序排列后再输入, 但在列程序清单时, 电脑会自动按照语句标号从小到大加以排列, 该空格的地方, 字符间会留出适当的间隙。

(3) 若程序出错, 电脑会打印出错信息 (有关常见的出错信息请参见附录 IV), 根据出错信息指出的错误类型, 应及时予以修正。

(4) 当电脑进入死循环, 即无止境地运行时, 可依次按下 **CTRL C RETURN** 键或 **CTRL RESET** 键, 迫使电脑停止运行。当然也可在调试程序时, 在程序中安插 “**TRACE**” 和 “**NOTRACE**” 语句 (请参见本册子第二部分基本 BASIC 语言中关于键盘命令一节)。

电脑操作完毕后, 要关断电源。一般总是先切断主机电源, 然后再切断屏幕显示器和其他外部设备的电源。

3. 键盘输入的技能

图1.2—43上、下分别是 **APPLE-II** 微电脑的键盘与一般英文打字机的键盘。由图可见, **APPLE-II** 微电脑的键盘与一般英文打字机的键盘的键数及键位安排布局基本一致, 只是 **APPLE-II** 微电脑的键盘上多了几个键: 左边多了 **ESC**、**CTRL** 键; 右边多了 **RESET**、**REPEAT**、**RETURN** 键。

APPLE-Ⅱ 微电脑键盘上的 **SHIFT**、**←**、**→** 等键分别与英文打字机键盘上的大写键、倒格键等键相似。

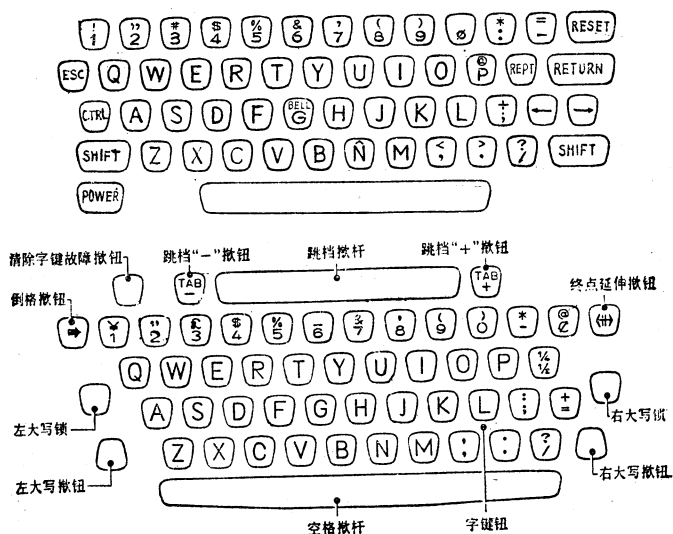


图1·2—43 APPLE-Ⅱ 键盘与英文打字机键盘的对照

因此，我们认为：打字机的基本操作方法同样适用于电脑的键盘操作。

为了正确地、快速地进行键盘输入，必须从一开始上机操作就十分注意训练自己以具有正确的操作姿势、击键法、指法、原稿阅读法等。

操作姿势

操作姿势是关系到能否正确、高速地进行键盘输入的一个重要因素。

一般地说，键盘操作者应坐姿端正；键盘位于胸部正前方大约20厘米；上臂自然下垂，肘部既不要紧贴上身，也不

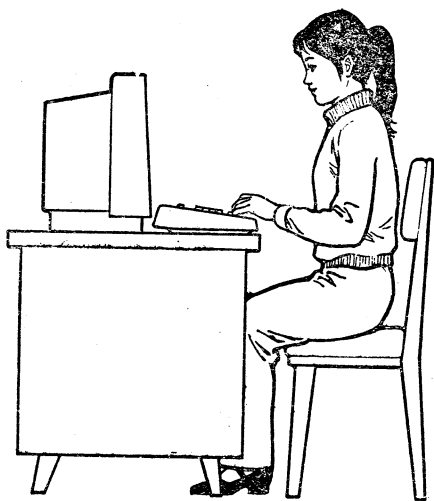


图1·2—44 键盘输入时的操作姿势

要过分向两侧张开，肘部与上体间距以能夹一个拳头为宜；双手的手指自然弯曲，轻轻地放于第三排基本键位上；头部稍向前倾，双眼注视键盘左侧桌面上的程序原稿；双膝平放，双脚着地；座位高低要适中，两肘部与键盘放置的桌面平齐。总之，整个

身子要求很平稳，很舒坦（见图1·2—44）。

切忌由于高度紧张，使双臂夹紧上身，双肩耸起，双眼死死盯住键盘进行操作。

击键法

所谓键盘输入，主要是用手指击键。我们这里所说的“击”键，并非是用手指“掀”键或“凿”键。正确的方法是：用手指前端迅速“击”键，犹如指端碰触到针尖一般，立即弹起，手指弹起的高度应离键盘2~3厘米，整个动作要求敏捷、迅速、果断。击键时不要过分用力，切忌一按到底，以免损坏键盘。初上机而又没有英文打字基础的读者，往往会犹豫不决，既想击键，又不敢击键，因此，常常造成击键不准或使指端击入两键之间。我们认为，一开始从事键盘操

作，就应养成大胆、果断的击键习惯。

击键时应该以手指带动手腕灵活操作。切忌手腕僵硬，靠肘部使力，这样必然动作迟钝，并且每击一个键，肩部也会协同动作，待一个程序输入完毕，手臂和肩部都会发酸，如同干了重活一般。

另外，手腕不能塌下，切忌将手腕搁于键盘框架上，使操作不灵便。

指法

APPLE-Ⅱ微电脑的键盘分成四排，由上往下，第一排基本上为数字字符；第二、三、四排基本上为字母字符。我们将第三排键作为基本键排，以A、S、D、F键及J、K、L、⁺键为基本键钮。它们分别由双手的八个手指分管着操作，手指操作分配见图1·2—45。位于键盘两边的键作这样分配：

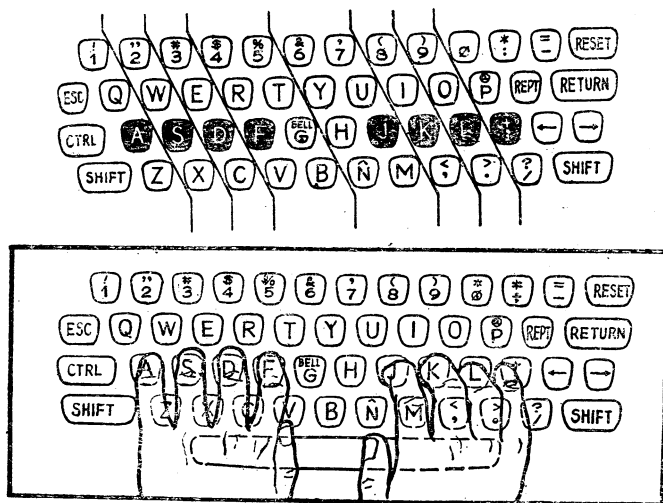


图1·2—45 键盘操作的指法

凡在左边的，由左手小指分管；在右边的，由右手小指分管。长条空格键由大拇指分管。可见，每个食指要各自分管两列键（即各管8个键），而每个小指所管得键最多，因此，必须加强小指灵活性和力度的训练。

对于初学者来说，上机时，应将左手四个指头（小指、无名指、中指、食指）自然弯曲后分放在 **A、S、D、F** 键上；右手四个指头（食指、中指、无名指、小指）自然弯曲后，放在 **J、K、L、 $\frac{1}{2}$** 键上，并且每击一键后，立即回复到基本键位上。一旦能熟练地击键以后，手指就不必每击一键再回复到基本键位上，而应采取双手不停的轮番击键指法。

击键时，一定要按基本指法，双手操作。从初学键盘输入时，就要养成这个习惯，切忌单手操作。有些操作者习惯于光用右手食指，见图1·2—46(a)，到处跳跃，紧张忙乱，这是绝对应该避免的。单指击键，必然影响输入速度，并容易击错键位。

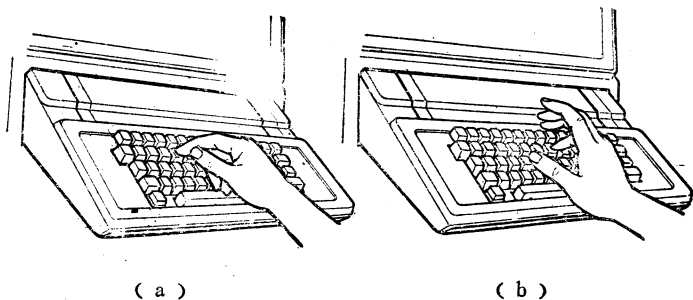


图1·2—46 错误的指法

在某一个手指击键时，其他手指不要高高翘起，见图1·2—46(b)。

原稿阅读方法

键盘输入过程中，一定要思想集中，双眼注视所编制的程序原稿（一般放在键盘左侧的桌面上），决不应窥视键盘。也不要看一句程序，双眼即转向键盘以指挥手指击键。

原稿阅读的方法一般有这两种：

（1）一个字符一个字符不停地往下看，手指也不停地根据所看到的一个字符一个字符有节奏地连续击键。而不应该看一个字符，击一次键，再看一个字符，再击一次键，这样，势必影响键盘输入速度的提高。

（2）对于所编制的程序较熟悉，并且英语基础较好的操作者来说，可以看一段（一句或多句）程序，输入一段，对于简单的程序尽可能默记在心，手指不停地击键，双眼可以扫视一下显示屏，以监视键盘输入的正确性，避免出错。

掌握了正确的阅读原稿的方法，才能击键连贯，有节奏感，准确而少犯错误。

当然，要做到双眼不窥视键盘而能准确击键决不是一天两天就能办到的，必须要求操作者对整个键盘上各个键位安排了如指掌，并熟记各个手指所分管的键位。做到手指击键时，根本无需思考，习惯地去击各个需要击的键。因此，对于初学者来说，不要图输入速度，而要反复熟悉键位，训练基本指法，做到击键果断、正确。一开始从事键盘输入就养成不用眼睛帮着手指找键位的习惯。由于整个键盘仅有52个键，只需用心，不用多久就能记住并掌握的。

另外，读原稿时，切忌用嘴念，也切忌头部来回转动，左顾右盼。这类多余动作都将影响键盘输入速度的提高。

为了帮助读者进行键盘输入技巧及熟练度的训练，下面

我们提供一个键盘输入练习的程序。

该程序开初（第115句）先告诉练习者在输入字符间和一群字符之后不要留空格；接着讯问练习者需要哪种难度的键盘输入练习，第120句允许练习者在1~7的难易度标志号中进行选择；对于初学者来说，难度1着重基本键位的击键训练，即双手的四个指头分别置于A、S、D、F和J、K、L、⁺键上，两个拇指置于长条空格键上，而其余难度则着重于训练对基本键位附近其他键位的击键。当然，击键时，双眼决不能帮着手指寻找键位，而应注视显示屏。第450~720句就是关于不同难度的子程序。第150~260句为子程序（其中夹着一个随机函数），对字符串变量进行赋值。第270句显示要打印的字符串。第290句练习者输入字符。第300句将6个字符串连成一个字符串，以便于比较。第305句提醒程序设计者在第330句加上音响效果。第310句是输入正确与否的判断，若完全正确则得分100%，并显示在屏幕上；若键盘输入错误，则产生“嘟”、“嘟”声（第330句），并将正确的字符显示出来（第350句），由于练习者键盘输入发生错误，因此，第380句要求在此难度上再练习一次。第390~400句允许键盘输入正确者再在同样难度或其他难度上进行练习。因为同样难度的反复练习，对提高键盘输入技巧是很有帮助的。

读者不妨上机试一下，也许会觉得这个程序既容易又饶有趣味，对提高键盘输入技巧及熟练度确有帮助。

```
100 HOME : PRINT " THIS IS TYPING  
PRACTICE."
```

```
110 PRINT : PRINT "TYPE THE LETTERS  
IN ONE LINE WITH "
```

```

115 PRINT "NO SPACE BETWEEN LETTERS.",
    PRINT
120 PRINT "WHAT LEVEL OF DIFFICULTY?
    (1-7)", : GET DF: PRINT DF
130 ON DF GOSUB 450, 490, 530, 570, 610, 650,
    690
140 PRINT
150 LET X=INT(5*RND(1)+1)
160 LET X$=MID$(A$, X, 3)
170 LET Y=INT(5*RND(1)+1)
180 LET Y$=MID$(B$, Y, 3)
190 LET Z=INT(RND(1)*5)+1
200 LET Z$=MID$(C$, Z, 3)
210 LET Q=INT(5*RND(1)+1)
220 LET Q$=MID$(G$, Q, 3)
230 LET K=INT(5*RND(1)+1)
240 LET K$=MID$(E$, K, 3)
250 LET V=INT(5*RND(1)+1)
260 LET V$=MID$(H$, V, 3)
270 PRINT X$, Y$, K$, Q$, Z$, V$,
280 PRINT : PRINT
290 INPUT T$
300 WR$=X$+Y$+K$+Q$+Z$+V$
305 REM ADD BELL ON TOP OF EXCL
    AMATION MARKS IN LINE 310
310 IF T$=WR$ THEN PRINT : PRINT
    "YOU SCORED 100%, VERY GOOD!!!",
    GOTO 390

```

```

320 IF T$ <> WR$ THEN 330
330 PRINT : PRINT "OOPS! ! "; S = -16336;
    FOR JJ=1 TO 55: B=PEEK(S);
    SOUND=B+B; NEXT JJ; PRINT
340 FOR X=1 TO 18 STEP 3
350 INVERSE : IF MID$(T$, X, 3) <>
    MID$(WR$, X, 3) THEN PRINT MID$
    (T$, X, 3); : MORMAL : PRINT "SHOULD
    BE "; : INVERSE : PRINT MID$(WR$,
    X, 3); MORMAL : PRINT" ", PRINT
360 NORMAL
370 NEXT X
380 PRINT : PRINT "TRV THES LETTERS
    AGAIN."; FOR I=1 TO 1000; NEXT I;
    GOTO 140
390 PRINT : PRINT "WOULD YOU LIKE THE
    SAME LETTERS AGAIN? "; : GET MM$;
    PRINT : IF MM$="Y" THEN 140
400 IF MM$="N" THEN 410
410 PRINT : PRINT "WOULD YOU LIKE
    DIFFERENT LETTERS? "; : GET TY$;
    PRINT
420 IF TY$="Y" THEN PRINT : GOTO
    120
430 IF TY$="N" THEN END
440 END
450 PRINT
460 A$="ASDFI"; B$="EJKL, "; C$=

```



```

"SEFIA"
470 E$="EK, DI"; G$="SEJLI"; H$ =
    "FSDAJ"
480 RETURN
490 PRINT
500 A$="RIJUA"; B$="DEKAI"; C$ =
    "FREUS"
510 E$="LUARD"; G$="SRFUI"; H$ =
    "JIL, E"
520 RETURN
530 PRINT
540 A$="SIJUH"; B$="HYKEG"; C$ =
    "URLIT"
550 E$="FGTUD"; G$="GEYRI"; H$ =
    "HTREY"
560 RETURN
570 PRINT
580 A$="YAHRW"; B$="QSIIP"; C$ =
    "WOKUQ"
590 E$="OWPFL"; G$="PEGTO"; H$ =
    "DORWQ"
600 RETURN
610 PRINT
620 A$="JVAPU"; B$="TBINQ"; C$ =
    "YMDML"
630 E$="OVFBN"; G$="WBGME"; H$ =
    "ONHVR"
640 RETURN

```

```

650 PRINT
660 A$="ZU/PX": B$="KTSVC": C$="
    "XYDB."
670 E$="EQFN/": G$="IWCNZ": H$="
    "XOHZC"
680 RETURN
690 PRINT
700 A$="2S28K": B$="KB3D3": C$="
    "D3J7J"
710 E$="7JF4F": G$="9L9F5": H$="
    "J6J, O"
720 RETURN
730 REM

```

```

*****
*      TYPING  PRACTICE      *
*****
*      R.M.  WONDERFUL      *
*                               *
*****

```

4. 程序的编辑与修改

键盘输入时，很可能会产生一些错误。避免错误的根本方法是：键盘输入时要仔细认真，平时加强键盘操作技能的训练；同时对已经出现的错误应能及时发现，及时予以改正。这就需要掌握在键盘上进行程序的编辑与修改方法，也就是学会操作 APPLE-Ⅱ 微电脑的键盘编辑工作。

APPLE-Ⅱ 的键盘编辑与修改功能主要是通过左位移键←、右位移键→、重复键 **REPT** 和换码键 **ESC** 等键来实现的。前三个键的功能与操作方法在键盘介绍时已详细叙述了，下面进一步谈谈换码键的功能与操作。

ESC 键是专门用于编辑的，其主要功能为：

(1) 当击 **ESC** 键后，再击 **A** 键，光标就会在所在行上向右移动一格，简记为 **ESC-A**；

当击 **ESC** 键后，再击 **B** 键，光标就会在所在行上向左移动一格，简记为 **ESC-B**；

当击 **ESC** 键后，再击 **C** 键，光标就会在所在列上向下移动一格，简记为 **ESC-C**；

当击 **ESC** 键后，再击 **D** 键，光标就会在所在列上向上移动一格，简记为 **ESC-D**。

在这样四条指令下，光标可向任意位置移动（见图 1·2—47）。采用 **ESC-A**、**ESC-B**、**ESC-C**、**ESC-D** 四条

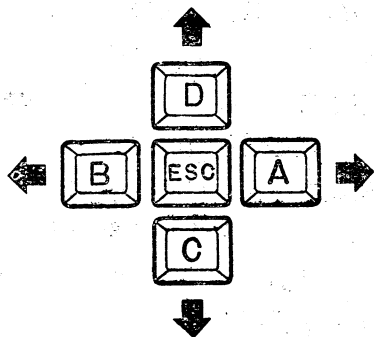


图 1·2—47 **ESC** 键的编辑功能

编辑命令只能使光标作左、右、下、上移动，而并没有改变显示屏上和主机内存中原有的内容。

(2) 当击 **ESC** 键后，再击 **E** 键，就能将光标从现在所在的位置起一直到显示屏最末一个字符全部删除。

(3) 当击 **ESC** 键后，在击 **SHIFT** 键的同时击 **P** 键，就能将整个显示屏清除干净，而光标又跑到显示屏左上角去了。这个功能类似于 **HOME** 命令。

(4) 由于 APPLE-II PLUS 型微电脑具有自动启动 ROM (即自动启动监控程序), 因此, 只需击一次 **ESC** 键, 电脑就处于编辑状态了。这时, 只要击 **I** 键, 光标就在目前所在的行上向上移动一行; 击 **J** 键, 光标就在目前所在的列上向左移动一位; 击 **K** 键, 光标就在目前所在的列上向右移动一位; 击 **M** 键, 光标就在目前所在的行上向下移动一行。这四种情况分列类似于上述 **ESC-D**、**ESC-B**、**ESC-A** 和 **ESC-C** (见图 1.2-48), 但光标每次移动不必重复击 **ESC** 键, 这是因为击 **ESC** 键一次后, 电脑已自动处于编辑状态了。

当然, 这四条编辑指令也不会改变显示屏以及主机内存中原有的内容。

假如要使微电脑脱离编辑状态, 那么只要按一下除 **I**、**J**、**K**、**M**、**REPT** 及 **SHIFT** 键之外的任何一个键就行了 (一般是按空格键)。

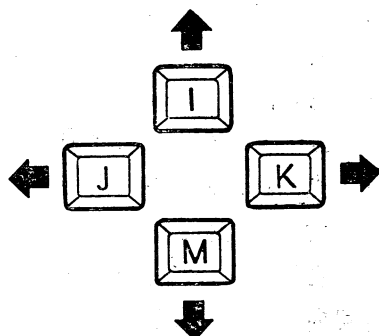


图 1.2-48 **ESC** 键的编辑功能

编辑与修改程序的具体作法是这样的:

首先将要修改的程序或语句用 **LIST** 命令显示在显示屏上。然后通过键入 **←**、**→**、**REPT**、**EST** 等特殊功能键, 移动光标, 修改程序。最后再使用 **LIST** 命令让修改后的程序列在显示屏上, 以便检查。

下面分别介绍几种修改程序的情况。

语句中错了几个字符怎么办

例如: **LIST 50 RETURN**

50 NFXT I

⋮
]■

上述第50句将“NEXT”错打成“NFXT”了，改错的操作步骤为：

(1) 按一下 **ESC** 键，使电脑进入编辑状态；按 **↑** 键若干次，直到光标上移到第50句所在行，并覆盖住字符“0”；再按 **←** 键，使光标左移，覆盖住字符“5”，即出现

■0 NFXT I

(2) 按下除 **I**、**J**、**K**、**M**、**RERT** 及 **SHIFJ** 键之外的任意一个键，使电脑脱离编辑而进入修改状态；接着按若干次 **→** 键，使光标覆盖住字符“F”（这时，被光标扫过的“50 N”等字符已复制到内存去了）；按一下 **E** 键，便将“F”改为“E”了。

(3) 继续按 **→** 键，直至光标扫过该语句的所有字符后，按下 **RETRN** 键（若该语句很长，可在按下 **→** 键的同时，按下 **REPT** 键，以加速光标移动的速度），这个错误就得改正了。

假如在键盘输入时，马上发现错误，如：

50 NFXT I■

这时可直接按 **←** 键四次，使光标左移至字符“F”上，即

50 N■XT I

再按下 **E** 键，接着按三次 **→** 键，使光标右移至语句体外面，这样就可继续进行键盘输入了。

语句中多输入了字符怎么办

例如：80 GOOTO 40

⋮
]■

上述第80句中“GOTO”语句多输入一个“O”字符，去除该字符的操作步骤为：

(1) 按一下 **ESC** 键，使电脑进入编辑状态；按 **I** 键若干次，直到光标上移至第80句所在行，并覆盖住字符“O”；再按 **J** 键，使光标左移，覆盖住字符“8”，即出现

```
00 GOOTO 40
```

(2) 按下任意一个键（除 **I**、**J**、**K**、**M**、**REPT** 及 **SHIFT** 键外），使电脑脱离编辑而进入修改状态；按若干次 **→** 键（让光标扫过“80 GO”；将这些字符复制入内存），直至光标覆盖上另一个“O”字符，即出现：

```
80 GO TO 40
```

这时，按一下空格键，以空格来代替多余的字符（这在浮点 BASIC 语法中是允许的）。

(3) 继续按 **→** 键，直至光标过整个语句的所有字符后，按下 **RETURN** 键，这个错误就被改正了。

语句中漏了几个字符怎么办

例如：LIST 50

```
50 PRIT "A^2="; A^2, "A^3="; A^3  
:  
]
```

上述第50句的“PRINT”语句中漏了字符“N”，增补字符的操作步骤为：

(1) 按一下 **ESC** 键，使电脑进入编辑状态；接着按若干次 **I** 键，直到光标上移至第50句所在行，并覆盖住“O”字符；再按 **J** 键，使光标左移，覆盖住字符“5”，即出现：

```
00 PRIT "A^2="; A^2, "A^3="; A^3
```

(2) 按下任意一个键 (除 I、J、K、M、REPT 及 SHIFT 键外), 使电脑脱离编辑状态, 按若干次 → 键 (使光标扫过 “50 PRI” 字符), 直至光标覆盖住 “T” 字符, 即出现:

```
50 PRI "A^2="; A^2, "A^3="; A^3
```

(3) 再按一下 ESC 键, 使电脑再度进入编辑状态; 接着按一下 I 键, 将光标上移至字符 “T” 上面, 即出现:

```
50 PRIT "A^2="; A^2, "A^3="; A^3
```

(4) 按一下除 I、J、K、M、REPT 及 SHIFT 键之外的任意一个键, 使电脑再次脱离编辑状态; 再按一下 N 键 (先前漏输入的字符), 这时 “N” 也就 “写入” 内存了, 显示屏上出现:

```
50 PRIT "A^2="; A^2, "A^3="; A^3
```

(5) 按一下 ESC 键, 使电脑第三次进入编辑状态; 接着按一下 J 键, 再按一下 M 键, 也就是使光标先左移后下移, 覆盖住字符 “T”, 显示屏上出现:

```
50 PRI "A^2="; A^2, "A^3="; A^3
```

(6) 按一下除 I、J、K、M、REPT 及 SHIFT 键之外的任意一个键, 使电脑第三次脱离编辑状态; 然后在按住 → 键的同时, 按下 REPT 键, 使从字符 “T” 开始的所有字符均复制到内存去; 最后按一下 RETURN 键, 便将漏输入的字符补上了。

为了检查上述修改是否正确, 可从键盘输入命令:

```
LIST 50 RETURN
```

显示屏上立即会出现:

```
50 PRINT "A^2="; A^2, "A^3="; A^3
```

从上述改错操作来看, 手续相当麻烦, 需要电脑多次进

入编辑和修改状态，并且我们假定所操纵的电脑是带有自动启动监控程序的，否则，要使光标作上、下、左、右移动，非按下 **ESC-A**、**ESC-B**、**ESC-C**、**ESC-D** 键不可，操作够复杂的了，尤其是碰上语句中有两个以上错误有待修改时，操作步骤就更复杂了。

因此，我们建议：凡遇到语句中有多处错误要修改的情况，干脆将该语句重新从键盘上输入一次，让电脑列程序清单时，自动剔除错误语句，而将后输入的被改正了错误的语句，按语句标号大小排入相应的位置。

要增补几句语句怎么办

一般在程序设计时，各语句标号间总留有一定的间隔。这时，间隔的作用就体现出来了。我们可以将要增补的语句按照应该在程序中出现的位置，另外编上一个标号（不许与原有语句标号重复，而且一定要介于所插入处上、下两语句标号之间），从键盘输入。

待 **LIST** 命令后，电脑会将整个程序的所有语句按标号大小自行整齐地排列在显示屏上。

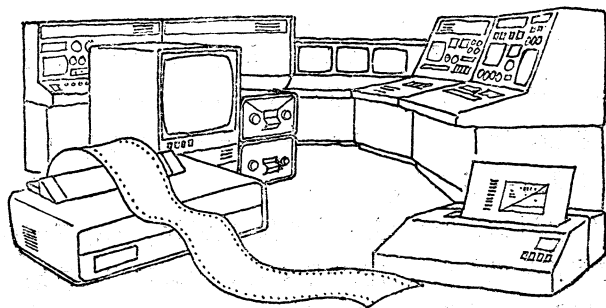
要删除几句语句怎么办

有时为了简化程序，需要删除几句语句，这时，只要从键盘输入所要删除的语句的标号，并紧接着每一标号后按一下 **RETURN** 键就可以了。待 **LIST** 命令后，电脑列出的程序清单中会自动删除这几行语句。

要删除一段程序怎么办

要删除一段程序，只需要运用 **DEL** 语句就可以了，即在 **DEL** 后列出所要删除的程序段前后两语句的标号（标号间用逗号分开）。具体情况详见 **DEL** 命令。

BASIC 语言



I. BASIC 语言的基本概念

1. BASIC 语言的基本特点

BASIC 语言是国际通用的算法语言之一。BASIC 一词是英语 BEGINNER'S ALL-PURPOSE SYMBOLIC INSTRUCTION CODE (初学者通用符号指令代码) 的缩写。

BASIC 语言是一种比较简单的算法语言。它虽然简单但并不贫乏, 适宜解决科学计算和事务管理中的中、小型题目。

BASIC 语言基本语句少(仅十几句), 且语句中所使用的词、运算符与英语单词、数学运算符基本一致。因此, 具有浅显易懂、容易掌握的特点, 也就有利于青少年和初学者学习, 并便于普及推广。

BASIC 语言是一种会话式的语言。操作者通过显示屏幕 (或电传打字机) 实现与微电脑对话, 即微电脑与使用者互相“交谈”, 彼此提出问题和回答问题。这种“会话”功能对于初学者特别有益。当源程序送入微电脑后, 若有错误, 操作者可立即通过键盘修改程序, 边算边改, 直到得出满意的结果。

BASIC 语言正因为具有上述特点, 因此受到人们普遍欢迎, 成为国际上通用的微电脑算法语言之一。目前几乎所有

微电脑都毫无例外地配有 BASIC 解释程序。但是,各种微电脑所使用的 BASIC 语言又有某些细微的区别。因此,使用 BASIC 语言前,需要查阅一下你所操作的微电脑系统的 BASIC 说明书。本章所介绍的 BASIC 语言是适用于 APPLE-Ⅱ 微型电脑的浮点 BASIC〔注〕。

2. BASIC 语言的基本符号

BASIC 语言是由一些特定的基本符号(字母和字符)和若干条语法规则所构成的。这些基本符号按特定的规则进行了二进制编码,而这种编码方式微电脑是能够领会的,除了这些基本符号外,其他符号微电脑便不能识别了。目前,微电脑普遍采用的编码方式是 ASCII 码(American Standard Code for Information Interchange, 美国标准信息交换码)。浮点 BASIC 语言中 ASCII 码及相应的显示符号、键盘表示列于附录Ⅱ中。

ASCII 码通常用七位二进制数表示,因此共有 128 个字符。浮点 BASIC 语言中经常采用的基本符号为:

英文字母(26个)

A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z

阿拉伯数字(10个)

〔注〕APPLE-Ⅱ 的 BASIC 语言分为整数(INTEGER)BASIC 和浮点(APPLESOFT)BASIC 两种。两者的区别见附录 V。考虑到本书的读者对象和受到篇幅限制,我们只介绍基础的浮点 BASIC 语言,不涉及到扩展 BASIC。

1 2 3 4 5 6 7 8 9 0

定义符

(1) 算术运算符: + (加) - (减) * (乘)
/ (除) ^ (乘幂)

(2) 关系运算符: < (小于) = (等于) > (大于)
<= (小于等于) >= (大于等于) < > (不等于)

(3) 特殊符号: ! " # \$ % & , ()
: @ ; , . ?

3. BASIC 语句的基本结构

我们通过下述例题来说明 BASIC 语句的基本结构。

已知: 圆形体育场的半径为75米。

求: 该体育场占地多少平方米?

运用 BASIC 语言我们编制了求解上述问题的程序:

```
10 LET R=75 ✓  
20 LET S=3.1416*R*R ✓  
30 PRINT "R="; R, "S="; S ✓  
40 END ✓
```

这样的指令组就是程序, 称作 BASIC 语言程序, 又称 BASIC 源程序 (Source Program)。若将该程序贮存到微电脑的内存中去, 就构成了“程序文件” (Program File)。

可见, BASIC 语言程序是由若干行 BASIC 语言组成的。每一行语言构成了一句 BASIC 语句。上述程序共有 4 句语句: 第10句是将半径值赋给变量 R; 第20句是将面积表达式

算得的值赋给变量S；第30句是打印输出语句；第40句告诉微电脑程序执行到此结束。也就是说，每一句语句分别指示微电脑完成一定的操作和运算。

一般的 BASIC 语句通常有下述四个部分构成：

语句标号 位于每句语句之前的数字编号称为语句标号。如上例中的“10”、“20”、“30”、“40”等。微电脑执行程序一般是循着语句标号的大小顺序进行的，但书写语句时可以不按大小顺序，一旦程序送入微电脑后，BASIC 解释系统会自动将源程序按标号从小到大的顺序排列起来，并依次执行。语句标号必须为正整数。语句标号与标号之间最好留有适当的间隔（上述程序中，每句间隔为10），便于程序修改时增补语句。若使用同一语句标号输入了不同的语句，那么前面输入的那一句将被抹去，电脑总是执行后输入的那一句。在 APPLE-II 机中，语句标号可以从 1 到 63999。

语句专用定义符 又称语句保留字（如“LET”、“PRINT”等）。它表明了语句的性质，定义符均用英语来表示（常常是英语单词或词组的缩写形式）。浮点 BASIC 保留字及其代表数字见附录 VI。

语句体 紧接于定义符后，是指示微电脑具体执行内容的实际工作部分。

回车记号 标志该语句结束。键盘输入时，在每句语句末尾须按一下 **RETURN** 键，这相当于我们平时写文章，一句话结束时用个句号。初学者可以在程序中每句语句后标上“↵”，以提醒自己在键盘输入时，每句语句末尾都按下“**RETURN**”键。

以上四个部分所组成的语句的集合就构成了程序。

4. BASIC 语言中的数

数在任何运算中都是不可缺少的要素, 在 BASIC 语言中, 它也往往是语句的重要组成部分。

(1) 在 BASIC 语句中, 一律采用十进制数, 但在书写或键盘输入一个数时, 不能使用逗号分位。例如:

56789000 不可写成 56,789,000

(2) 在 BASIC 语句中, 数的表示方法有两种——定点法与浮点法。

例如: +17, -893, 360000, 0,

+3.14159265, 0.9993, -72.896, -0.33333

这种数我们称作定点数, 即小数点的位置是固定的。

另外有一些较大或较小的数, 根据微电脑的输出功能或为了书写方便, 常采用带指数的书写形式 (即通常说的科学表达方法)。

例如: +8 E -08 (表示 $8 \times 10^{-8} = 0.00000008$),

-1.56 E +06 (表示 $-1.56 \times 10^6 = -1560000$),

0.91 E +04 (表示 $9.1 \times 10^3 = 9100$)

这种数我们称作浮点数, 即小数点的位置是浮动的。

可见, 浮点数是在定点数的基础上加上指数部分 ($E \pm n$) 后形成的。我们称 E 前面的数 (如 8, -1.56, 0.91) 为尾数, E 后面的数 (如 -08, +06, +04) 为指数或阶码。

(3) APPLE-II 内部存贮的数据精度大于九位, 但输出打印时, 只有九位 (凡超过九位时, 自动进行四舍五入)。

实数 (浮点法表示) 表示范围为: $-1 \times 10^{38} \sim 1 \times 10^{38}$ 。

凡超出该范围,微电脑将给出溢出信息。但在作加、减法时,数值可以达到 1.7×10^{38} 。当实数的绝对值小于 3×10^{-33} 时,微电脑就将它作“0”处理。

整数的表示范围为: $-32767 \sim +32767$ 。

数据打印(或显示)时,遵守以下规则:

(a) 若为负数,打印出“-”号。

(b) 若数的绝对值为整数,则从0~999999999(九位)就按整数打印。

(c) 若数的绝对值大于或等于0.01,而小于999999999.2,就按定点表示法(即不带指数)打印。

(d) 凡超出(b)、(c)两项规则的数值,就采用浮点表示法(即科学表示法)打印,其表示法为:

S	X.XXXXXXXX	ES	TT
一位符号位	九位尾数	一位符号位	两位指数

尾数前的符号若为“+”,则可省略,微电脑也不打印出来;尾数可用0~9的九位整数来表示,其中小数点前打印一位非零整数,小数点后打印八位非零整数;E后面的指数符号,不论“+”“-”均不能省略(如:0.94E+11不能写成0.94E11);指数部分只能由两位整数构成(如:0.91E-04),且指数部分不能单独构成一个数(如:10⁻⁵不能写成E-05,必须写成1E-05)

现举例如下:

要输出的数	实际输出的格式
4.932E+06	4932000
-1.234567899	-1.2345679
4000000000	4E+09

5. BASIC 语言中的变量

BASIC 语言中的变量概念与数学中的变量概念基本一致,它表示在程序执行过程中可以取不同值的量。程序中的变量,都取了一个名称,在微电脑的内存贮器中分配了相应的地址单元。

程序中的变量有三种形式:简单变量、下标变量和字符串变量。

简单变量

所谓简单变量,就是指它的取值是一个数。凡存放整数的简单变量,称作整形变量,变量名后带有“%”符号;凡存放实数的简单变量称作实形变量。

在浮点 BASIC 中规定:实形变量必须以一个英文字母开头,后跟一个(或多个)字母或数字,如:AB, A7, U, G 等。变量名最长可达238个字符,但微电脑只能识别最前面的两个字符(而打印时能将整个变量名都打印出来),如 BDE 和 BDXPE,计算机认为这两个变量是相同的。由此可见,实形变量最多可取 $26 \times (26 + 11) = 962$ 种不同的名称,但是,凡浮点 BASIC 语言中使用的保留字及常见的希腊字母 α 、 β 、 γ 等均不可作为变量名称。例如:GOTO、SQR、 $\alpha 1$ 等都不能作为变量名。

下标变量

所谓下标变量是指带有下标的变量,它们代表一组数。数

学中的向量 $X = (X_1, X_2, \dots, X_n)$ 及矩阵 $A = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \dots & \dots & \dots & \dots \\ A_{m1} & A_{m2} & \dots & A_{mn} \end{pmatrix}$,

在算法语言中通常分别将它们作为一个整体来处理，并称上述向量 X 和矩阵 A 为数组， X 、 A 称为数组名，在程序中写成： $X(N)$ 和 $A(M, N)$ ，其中圆括弧内的 M 和 N 称作下标，这就是下标变量。

凡只有一个下标的元素所组成的数组，称为一维数组，如 $X(N)$ ；凡有两个下标的元素所组成的数组，称为二维数组，如 $A(M, N)$ ；……

APPLE-Ⅱ 中允许有 88 维数组，但最常用的仍为一维数组和二维数组。数组的下标必须是大于等于零的整数，下标须放在圆括弧内，下标间用逗号分开。例如： $A(-1, 3)$ 、 $X(-3)$ 、 $M1, 4$ 等表示法都是不允许的。

字符串变量

这是 BASIC 语言独有的一种变量形式，它不是一个数而是表示一串字符。在实形变量名后加符号 $\$$ 就构成了字符串变量名。

字符串是由基本符号组成的一种排列，并用双引号引起来。字符串变量用以表示某个字符串，例如：

$A\$ = "THIS IS A BOY"$

$B\$ = "79231"$ （这里的 79231 不再表示数字，而是一个字符串）

浮点 BASIC 语言规定：字符串变量中的字符个数不得超过 256 个。

6. BASIC 语言中的标准函数

标准函数又称为内部函数。为方便使用者,在浮点 BASIC 语言中提供了十一种常用的函数,并将这些函数的运算编成一段段子程序,存放在机器内,故称为内部函数。程序中使用到这十一种函数时,只要给出它的自变量就可以得到函数值了。下面我们列表介绍这十一种标准函数。

标准函数	数 学 含 义	注 释
SIN (X)	$\sin x$ (正弦函数)	自变量 X 的单位为弧度
COS(X)	$\cos x$ (余弦函数)	自变量 X 的单位为弧度
TAN(X)	$\operatorname{tg} x$ (正切函数)	$\left\{ \begin{array}{l} \text{自变量 X 的单位为弧度, 若} \\ \text{X 值接近 } n \cdot \frac{\pi}{2} (n=0, 1, 2 \cdots) \\ \text{则 } \operatorname{tg} x \text{ 值可能会超过计算机} \\ \text{允许数值范围而溢出} \end{array} \right.$
ATN(X)	$\operatorname{arctg} x$ (反正切函数)	$\left\{ \begin{array}{l} \text{函数值单位为弧度, 且 } -\frac{\pi}{2} \\ < \operatorname{ATN}(X) < \frac{\pi}{2} \end{array} \right.$
SQR(X)	$+\sqrt{x}$ (算术平方根)	$X \geq 0$
EXP(X)	e^x (指数函数)	$e \approx 2.71828$
LOG(X)	$\ln x$ (e 为底的对数函数)	$X > 0$
ABS(X)	$ x $ (绝对值函数)	
INT(X)	求不大于 X 的最大整数 (取整函数)	$\left\{ \begin{array}{l} \text{例如: } \operatorname{INT}(-3.1) = -4, \\ \operatorname{INT}(3.1) = 3 \end{array} \right.$
SGN(X)	(符号函数)	$\operatorname{SGN}(X) = \begin{cases} 1 & (X > 0) \\ 0 & (X = 0) \\ -1 & (X < 0) \end{cases}$
RND(X)	产生 (0, 1) 区间内一个随机数 (随机函数)	$\left\{ \begin{array}{l} X > 0 \text{ 则 } \operatorname{RND}(X) \text{ 每次获得一个} \\ \text{新的随机数, 一般 } X \text{ 取 } 1 \\ X = 0 \text{ 则 } \operatorname{RND}(X) \text{ 得到的是最近一次} \\ \text{随机函数} \\ X < 0 \text{ 则 } \operatorname{RND}(X) \text{ 值将固定于微电脑内, 不会被改变} \end{array} \right.$

标准函数中只出现了四个三角函数，其他三角函数都可通过这四个函数得到，例如：

$$\operatorname{arc} \operatorname{ctg} x = \begin{cases} \operatorname{arc} \operatorname{tg} \frac{1}{x} & (x > 0) \\ \frac{\pi}{2} - \operatorname{arc} \operatorname{tg} x & (x \leq 0) \end{cases}$$

$$\operatorname{arc} \sin x = \operatorname{arc} \operatorname{tg} \frac{x}{\sqrt{1-x^2}}$$

$$\operatorname{arc} \cos x = \begin{cases} \operatorname{arc} \operatorname{tg} \frac{\sqrt{1-x^2}}{x} & (x > 0) \\ \pi + \operatorname{arc} \operatorname{tg} \frac{\sqrt{1-x^2}}{x} & (x < 0) \end{cases}$$

上述三角函数中的变量 X 单位都为弧度，因此，凡遇到用角度表示的 X 均须先化为弧度，即 $X \times 3.14159/180$ ；同样反三角函数计算出来的函数值也都为弧度，若需要用角度表示，均须乘以因子 $(180/\pi)$ 。

标准函数中的指数函数都是以 e 为底的，若要计算以 10 为底的常用对数，只要通过换底公式：

$$\log_{10} x = \frac{\ln x}{\ln 10} = 0.434294 \ln x \quad \text{就能求得。}$$

7. BASIC 语言中的算术表达式

所谓 BASIC 语言中的算术表达式，就是把数、变量（简单变量或下标变量）、函数（包括标准函数）用算术运算符（+、-、*、/、^）与成对的圆括弧连接成的具有通常数学意义的式子，单个的数和变量是算术表达式的特殊形

式。算术表达式运算的结果是一个数值。例如：

代 数 式	BASIC 表 达 式
(1) $\frac{Q}{a+\frac{b}{c+d}}$	Q/(A+B/(C+D))
(2) $\ln\sqrt{ x } \quad (x \neq 0)$	LOG(SQR(ABS(X)))
(3) $\sqrt{s(s-a)(s-b)(s-c)}$	SQR(S*(S-A) *(S-B)*(S-C))
(4) a^b	A^(B)
(5) $[(a+b)-c(c-d)](e+f)$	((A+B)-C*(C-D)) *(E+F)

在 BASIC 算术表达式中，每一个字符必须写在同一行上，例 (1) 决不能写成 $\frac{Q}{A+\frac{B}{C+D}}$ 。所有运算符都不能

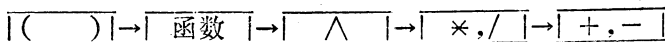
省略。乘号一定要写成“*”，不能写成“×”或“·”。在 BASIC 算术表达式中仅有一种圆括弧，没有普通代数式中的中括弧“[]”与大括弧“{ }”，且圆括弧一定要成对出现。两个运算符号不能写在一起，例如： A^{-B} ，不可以写成： $A\wedge-B$ ，而应写成： $A\wedge(-B)$ 。

对 BASIC 算术表达式的运算次序，微电脑作了如下规定：

- 第一级：先进行括弧内运算，层层括弧，由里向外算。
- 第二级：函数运算。
- 第三级：幂运算。
- 第四级：乘除运算。

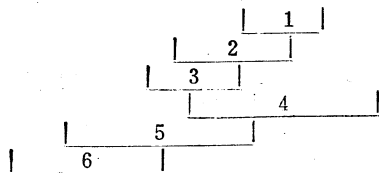
第五级：加、减运算。

级高的先运算，同级的按先左后右顺序运算，用图可表示成：



例如：计算， $E + P \sin^3 [6(S - A)]$ ，运算顺序如下：

$$E + P \times \text{SIN}(6 \times (S - A))^{\wedge} 3$$



（其中 1、2、3、4、5、6 表示运算的先后次序）

乘幂 A^B 运算时，若 B 不是整数，则要求 A 大于零，如： $(-8)^{\wedge}(1/3)$ 是不允许的。

8. 编制简易程序的一般方法

利用微电脑解决实际问题，就要给电脑输入一整套解题的步骤，这种步骤通常就是以程序的方式给出的。因此，编制程序就成为一个十分重要的问题了。

选择计算方案

需要利用电脑解决的问题，有时是直接以数学形式给出的，但在很多情况下，并没有提供数学模型。这就要求我们把实际问题化为数学问题，即构造数学模型，建立计算公式，并将计算公式归结为便于电脑处理的算术形式。

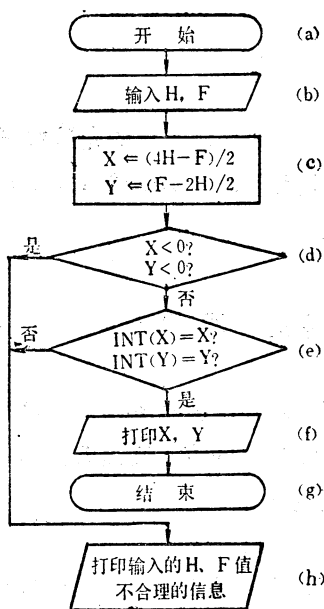
例如，有这样一个问题：“鸡兔同笼，从上边数有头35

个，从下边数有脚94只，问笼内鸡、兔各有多少？”〔注〕我们必须先构数学模型。

设：鸡有 x 只，兔有 y 只。那么根据一只鸡有二只脚，一只兔有四只脚，列出下列方程组：

$$\begin{cases} x + y = h & (1) \quad (h \text{ 为头数}) \\ 2x + 4y = f & (2) \quad (f \text{ 为脚数}) \end{cases}$$

然后将上述公式化简成便于电脑处理的简单算术运算式。



“鸡兔同笼”问题的流程图

(1) 式变形为

$$x = h - y \quad (3)$$

(3) 代入(2)，得

$$2(h - y) + 4y = f$$

则 $y = (f - 2h) / 2$ (4)

(4) 代入(3) 得

$$x = (4h - f) / 2 \quad (5)$$

画流程图

选择好了计算方案，接下来要考虑的问题是：整个计算先后顺序怎样？怎样的顺序最合理？这就是建立程序的思想。我们可通过画简单的流程图来实现。流程图对于程序编写就如同平时

〔注〕本题选自《孙子算经》：“今有雉兔同笼，上有三十五头，下有九十四足，雉兔各几何。”

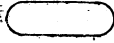
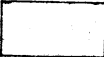
写文章前拟定写作提纲一样，提纲拟得好，自然顺理成章。

上述“鸡兔同笼”问题的流程图为：

程序开始(a)，先将 h 、 f （头数与脚数）告诉电脑(b)，然后再将 $(4h - f)/2$ 与 $(f - 2h)/2$ 分别赋给变量 x 、 y （即鸡与兔的只数），并要电脑根据已给的 h 、 f 值进行运算(c)，接着让电脑进行判断： x 、 y 是小于零吗？若小于零，显然所赋 h 、 f 值不合理，则程序转到(h)，若 x 、 y 均不小于零，则进一步判断： x 、 y 是整数吗？若不为整数，则程序也转到(h)，若 x 、 y 均为整数，就打印所得到的 x 、 y 值(f)，程序结束(g)。

可见，只要画好了流程图，编写程序就显得很方便了。流程图使程序设计思想明确，流程清晰、具体直观，且便于检查和修改程序。

流程图典型符号列表如下：

符 号	名 称	功 能
	端 点	表示开始、结束、停止、延迟、中断等流程图的端点
	处 理	进行框内所示符号或文字说明的处理。一个入口、一个出口
	判 断	为了选出几条路线中的一条而进行的判断。一个入口，两个出口
	输 入、 输 出	表示输入、输出处理。一个入口、一个出口
	流 程 线	表示信息流动的通道

编写程序

根据流程图，选择合理的语句来编写程序。

“鸡兔同笼”问题的程序为：

```
10 INPUT H, F
20 LET X=(4*H-F)/2
30 LEF Y=(F-2*H)/2
40 IF X<0 THEN 90
50 IF Y<0 THEN 90
60 IF INT(X)<>X THEN 90
70 IF INT(Y)<>Y THEN 90
80 PRINT "X=" ; X, "Y=" ; Y
85 GOTO 100
90 PRINT "IT IS UNREASONABLE" [注]
100 END
```

检查程序

由于编制程序是一项十分细致的工作，人为出错的可能性很大，因此，上机调试前，一般须对所编程序进行严格的检查并作适当的修改。查一下：所选择的计算方案是否反映了实际要解决的问题；流程图是否全面正确地体现了计算方案；编写的程序与流程图的对应关系怎样，有无遗漏和语法毛病；程序结构是否达到简单、合理；是否具用通用性与可扩充性等等。这一步检查完了，方才可以上机进行调试，进一步对程序进行检查。

[注]表示输入的数据不合理。

II. BASIC 语言的基本语句

1. LET 语句

结构 <标号> LET <变量> = <数值或算术表达式>

举例 初三(1)班数学考试成绩如下表,求全班平均分
数。

分 数	5	4	3	2
人 数	20	15	9	1

```
10 LET A=20
20 LET B=15
30 LET C=9
40 LET D=1
50 LET U=A+B+C+D
60 LET E=(5*A+4*B+3*C+2*D)/U
70 PRINT E
80 END
```

功能 LET 语句是赋值语句。它将“等号”右边的数值或算术表达式送给其左边的变量。我们可以形象地将 LET 语句中每一个变量看作为是一只只“匣子”的代号(如: A匣、B匣、C匣、D匣等),每只“匣子”内可存放一个数值或一个算术表达式的值。上例中的 LET 语句可用下图表示:

A	B	C	D	U	E
20	15	9	1	45	4.2

说明 (1) LET 语句中的“=”号并不具有数学中等号的意义, 它仅表明了一种“赋值”关系。上述程序中第60和70句我们可以改成:

```
60 LET A=(5×A+4×B+3×C+2×D)/U
70 PRINT A
```

可见, 在一个程序中, 变量A的内容(即A“匣子”中所装的内容)可不断改变, 但后进入的值或表达式可以“冲掉”先进入的值或表达式, 并且在第50和60句中, A、B、C、D的值被两次取出(只要程序需要, 可将变量中的值多次取用), 这就表明LET语句中的值是“取不尽, 冲得掉”的。第50和60句还表明LET语句具有运算功能。

(2) LET语句中“等号”左边不能为算术表达式, 只能是一个变量, 而右边出现算术表达式时, 一定要预先对算术表达式中每个变量赋予确定的值。

(3) LET语句能将字符串赋予字符串变量, 但字符串须用双引号引起来。例如:

```
10 LET X$="APPLE"
```

若将字符串赋予算术变量名, 电脑会给出错误信息:

```
? TYPE MISMATCH ERROR
```

(4) APPLE-Ⅱ电脑中, “LET”可省略, 只要写成:
 <标号> <变量> = <数值或算术表达式> 即可。一般几个不同的语句(包括LET语句)可写在一行中, 但语句间须用“:”号加以分隔。例如:

```
10 LET R=3: PRINT 2×R×3.1416
```

2.INPUT 语句

结构 <标号>INPUT <变量>

举例 同上例。

```
10 INPUT A, B, C, D
20 LET U=A+B+C+D
30 LET E=(5×A+4×B+3×C+2×D)/U
40 PRINT E
50 END
```

电脑运行时，执行到第10句，显示屏上会出现“？”，操作者应立即键入第一个数据20，然后再按 RETURN 键（即将数值20赋于变量A）。接着又出现“？”，操作者必须将第二个数据15赋于B……一直到将INPUT语句中每一个变量都赋值后，电脑才往下继续运行。

功能 INPUT语句是键盘输入语句。它也是一种给变量赋值的语句。比起LET语句，用INPUT语句为变量赋值增加了灵活性，增强了实时性，使程序变得简炼，同时实现了人机对话。INPUT语句常用于对参变数进行赋值，可以起到边输入，边计算，比较计算结果，及时修改参数。

说明 (1) INPUT语句从键盘输入的量只能是一个具体的数值（可以是整数或小数）或字符串，不允许是分数、函数或表达式。INPUT语句也可同时输入一组变量，但变量与变量间须用逗号分隔（见例），这里的变量允许是简单变量、下标变量或字符串变量。

(2) APPLE-II机中，进行多变量赋值时，要求给第一个变量赋值，显示屏上出现“？”，而要求给第二、三…

…个变量赋值时，显示屏上均出现“??”。

(3) 键入数据可有两种方式：(a) 每输入一个数据，按一下 **RETURN** 键；(b) 每输入一个数据，用“.”分隔，直至输入最后一个数据后才按 **RETURN** 键，即一次从键盘上输入若干个值，相互间用逗号隔开，如：

? 2, 3, 4, 5 **RETURN**

若键入的数据多于 **INPUT** 语句中变量的个数，电脑将不理睬多余数据，但给出信息：

EXTRA IGNORED

(4) **INPUT** 语句给字符串变量赋值时，输入的字符串可用双引号引起来，也可不用（但这时字符串中不可带逗号）。

在一句 **INPUT** 语句中，允许同时给数值变量和字符串变量赋值。例如：

5 **INPUT** X, D\$

另外，键盘输入语句还允许采用以下结构：

<标号> **INPUT** <字符串>, <一组变量>

其中 <字符串> 为操作者希望在显示屏上输出的字符串。例如：

20 **INPUT** "X, Y=", X, Y

执行该语句时，显示屏上输出字符串“X, Y=”，然后等待操作者从键盘上依次输入 X, Y 的值。对于 **APPLE-II** 微电脑来说，不允许出现下述形式的语句：

10 **INPUT** "X=", X, "Y=", Y, "Z=", Z

3.READ 和 DATA 语 句

结构 <标号> READ <变量>

⋮
<标号> DATA <数值>

举例 同上例。

```
10 READ A, B, C, D
20 LET U=A+B+C+D
30 LET E=(5*A+4*B+3*C+2*D)/U
40 PRINT E
50 DATA 20, 15, 9, 1
60 END
```

功能 READ/DATA 语句是读数/置数语句。它们也构成一种给变量赋值的语句。其中 DATA 语句在执行 RUN 命令时,并不能使电脑作任何操作,它只是起到给 READ 语句准备所需的数据的作用,而 READ 语句才是执行语句。当源程序输入时, BASIC 解释程序将 DATA 语句提供的数据依次存放于电脑的内存数据区中,并在第 1 个数据前设

↓ 数据指针

20	15	9	1	数据区

置了“数据指针”,当执行 READ 语句时,每读完一个变量,数据指针就自动向后移一格,指向后一个数据。READ/DATA 语句适用于数据较多的情况。

说明 (1) READ 和 DATA 语句必须配合使用,在一个程序中必须同时成对出现。

(2) DATA 语句中仅存放常数,而不能存放变量和表

达式（在扩展 BASIC 中，另有规定）。DATA 语句可放在程序的任何位置（通常放在程序的开头或末尾）。

（3）READ 和 DATA 语句可分成几句书写，只要变量和数据的对应顺序不变。

在源程序中，所有 READ 语句里的变量个数要小于或等于 DATA 语句中的数据个数。当 DATA 语句中的数据个数少于 READ 语句中的变量个数时，会输出下述信息：

? OUT OF DATA ERROR IN <(READ语句) 标号>

4.RESTORE 语 句

结构 <标号> RESTORE

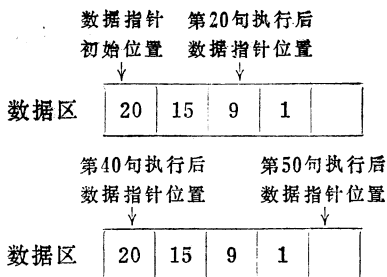
举例

```
10 DATA 20, 15, 9, 1
20 READ A, B
30 PRINT A, B
40 RESTORE
50 READ C, D, E, F
60 PRINT C, D, E, F
70 END
```

电脑执行上述程序后，打印出：

```
20 15
20 15 9 1
```

功能 RESTORE 语句是恢复数据区语句。当 DATA 语句所提供的数据要被反复读数时，为了简化程序，节省电脑内存，可利用 RESTORE 语句。



在上述例子中，第20句执行以后，“数据指针”指向“9”，第40句恢复数据区语句执行后，“数据指针返回第1个数据“20”。

说明（1）RESTORE 语句使“数据指针”恢复到DATA 语句中的第一个数据位置，而不是使程序回到第一句READ 语句。

（2）有时仅需重复读DATA 语句的部分数据（即跳过几个数据），这可通过在READ 语句中插入几个“假变量”，以进行假读。如上例中，我们除需要读20、15、9、1 外，还要读一次20和1，则可插入假变量T，程序如下：

```

10 DATA 20, 15, 9, 1
20 READ A, B, C, D
30 RESTORE
40 READ E, T, T, F
50 PRINT A, B, C, D, E, F
60 END

```

5.GOTO 语 句

结构 <标号> GOTO <标号>

举例 某厂1980年的产值为100万元,若每年平均增长率分别为4%、4.5%、5%、5.5%、6%、6.5%、7%、7.5%、8%、8.5%、9%、9.5%、10%,那么,到2000年该厂的产值为多少万元?

```
10 READ E
20 LET X=100*(1+E)^20
30 PRINT "E="; E, "X="; X
40 GOTO 10
50 DATA 0.04, 0.045, 0.05, 0.055, 0.06, 0.065,
        0.07, 0.075, 0.08, 0.085, 0.09, 0.095, 0.10
60 END
```

电脑反复执行10~40句程序段,一次又一次循环。当数据区中给定的13个值取完后,再次循环,无数可取,便显示出错误信息:

```
? OUT OF DATA ERROR IN 10
```

功能 GOTO 语句是无条件转向语句。用于改变程序的执行顺序。当执行到 GOTO 语句时,便无条件地转去执行该语句指出的那个标号的语句。

GOTO 语句往往用于需要循环执行某一段程序的情况。

说明 (1) GOTO 语句中必须指明要转向的目标语句的标号,而该目标语句确实要存在于程序中,否则将输出错误信息:

```
? UNDEF'D STATEMENT ERROR IN <(GOTO 语句的)标号>
```

(2) GOTO 语句往往与“IF”语句配合使用,即

<标号> IF <关系式> GOTO <标号>

这样,程序就具有循环和判断能力了。

6. IF-THEN 词 句(α)

结构 <标号> IF <关系式> THEN <标号>
<标号> IF <关系式> THEN GOTO <标号>
<标号> IF <关系式> GOTO <标号>

举例 同上例。

```
10 READ E
15 IF E=-1 THEN 60
20 LET X=100*(1+E)^20
30 PRINT "E="; E, "X="; X
40 GOTO 10
50 DATA 0.04, 0.045, 0.05, 0.055, 0.06, 0.065,
      0.07, 0.075, 0.08, 0.085, 0.09, 0.095, 0.01, -1
60 END
```

该程序只是在上述程序的基础上插入了第15句，并在第50句（置数语句）中添加了终止循环的控制数据（-1），使电脑读到（-1）时，无条件转到第60句结束。当然，我们也可通过设置计数器来辅助电脑作退出循环的判断。若设置N计数器，则只要在前述程序中添加下列两句语句就行了，而不必象本程序那样添上第15句，并在第50句中添加控制数（-1）了。

```
5 LET N=1
35 LET N=N+1; IF N>13 THEN 60
```

功能 IF-THEN 语句是条件转向语句。当条件语句中给定的条件满足时，程序将转去执行 THEN 后所指出的标号的语句。

IF-THEN 语句体现了电脑“判断”和控制转向功能。

说明 (1) IF 语句的关系式中通常使用下述比较算符
(在其他语句中是不允许使用比较算符的)。

BASIC比较算符	比较算符含义	相当于算术符号
=	等于	=
<	小于	<
>	大于	>
<=	小于或等于	≤
>=	大于或等于	≥
<>	不等于	≠

(2) IF 语句的形式:

(a) 一个变量与一个数值比较, 如:

```
10 IF A>=8 THEN 60
```

(b) 两个变量比较, 如:

```
10 IF M<>N THEN 50
```

(c) 一个变量与一个表达式比较, 如:

```
10 IF Q>= E^5 THEN 40
```

(d) 两个表达式比较, 如:

```
10 IF M^3=I^2+P*Q THEN 30
```

(3) IF 与 THEN (或 GOTO) 必须在语句中成对出现, 否则会输出错误信息:

```
? SYNTAX ERROR
```

IF-THEN 语句 (b)

结构 <标号> IF <关系式> THEN <语句> (或<语句; 语句>)

举例

```
10 FOR K=1 TO 10
20 LET R=RND (1)
30 PRINT "G="; R+0.5
40 IF G>1.49 THEN PRINT "THIS TIME I
   WIN"; GOTO 60
50 NEXT K
60 END
```

功能 这种结构的 IF-THEN 语句也是一种条件转向语句，只不过在“THEN”之后，所跟随的不是语句标号，而是一句不带标号的语句（若为两句以上语句，则语句间必须用“:”号分隔）。上述程序中的第40句，如果G大于1.49（即 $R > 0.99$ ），那么就打印出“这次我赢了”，从而转向第60句，程序结束；如果G不大于1.49，那么就转向第50句，作循环操作。如果10次循环中没有一次满足“ $G > 1.49$ ”的条件，程序将自行结束。

说明 （1）这种结构的条件转向语句在浮点BASIC语言中，只有当 IF 语句中的条件得到满足，才会转向执行 THEN 语句的内容。

（2）IF 语句中的条件判断既可以是关系式，也可以是逻辑表达式。例如：

```
10 X=RND(1)*10
20 IF X>=0 OR X<1 THEN PRINT "Y=",
   X^2; GOTO 10
30 PRINT "Y="; X^2-3*X+1
40 END
```


转入循环体。语句执行过程见图 2。

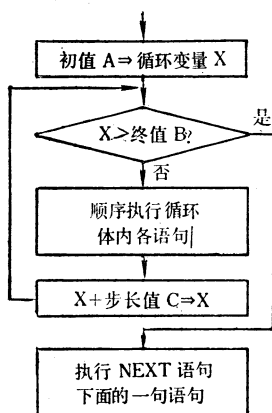


图 1

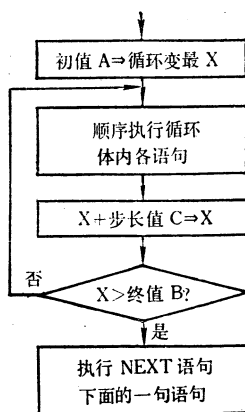


图 2

FOR-NEXT 语句一般多用于需要多次重复运算的程序之中。

说明 (1) FOR...TO...STEP 称为循环的起始语句, NEXT...是循环的终端语句, 需要循环执行的语句称为循环体。这三句相互独立, 同时又不可分割的语句构成了循环语句。循环起始语句和终端语句的循环变量名称必须一致(呼应)。循环变量可以用任一简单变量表示。

(2) 循环变量的初值、终值和步长值可以是数值, 也可以是被赋值的变量或者表达式。其中步长值可正可负。例如上述程序中的第20句可改为:

20 FOR N=999 TO 1 STEP -2

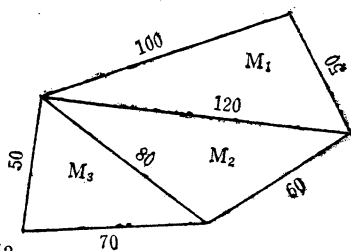
当步长值为 1 时, 循环起始语句中的“STEP 1”可以省略不写。

(3) FOR-NEXT 语句可以嵌套在另一个 FOR-


```

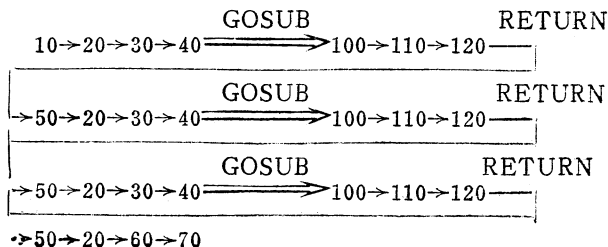
40 GOSUB 100
50 NEXT I
60 PRINT "M=", M
70 END
80 DATA 100,50,120,
      120,60,80,80,70,50
100 LET S=(X+Y+Z)/2
110 LET M=M+SQR(S*(S-X)+S*(S-Y)+
      S*(S-Z))
120 RETURN

```

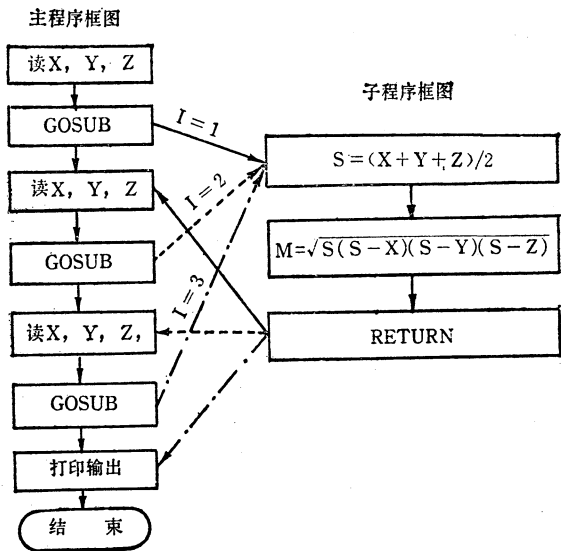


功能 GOSUB 语句和 RETURN 语句分别为转子语句和返回语句。在有的程序中，一些独立程序段往往要多次反复使用，这样的程序段我们称作子程序（上例中的第 100~120 句）。为了简化程序并避免出错，通常将某些需要反复计算的内容编成独立的子程序，而在主程序中（上例中的第 10~80 句）使用 GOSUB 语句去调用子程序，并在子程序末尾按上 RETURN 语句及时返回主程序，完成一次又一次的循环。上例中，若不采用 GOSUB 和 RETURN 语句，则第 100~120 句内容在程序中要反复书写三次，这样程序显得不精简且易出错。

上述程序的执行过程为：



若用程序框图则表示为：



说明 (1) 在子程序的最后一句必须使用 RETURN 语句，告诉电脑子程序执行结束，立即返回主程序。返回的准确地址是 GOSUB 语句紧接着的下一句语句。

(2) 子程序中的运算变量应在主程序中预先赋值。

(3) 子程序可以嵌套，例如：

```

10 PRINT " $ $ $ $ 1 "
20 GOSUB 100
30 PRINT "AAAA 4"
40 END
100 PRINT " & & & 2 "
110 GOSUB 200
120 RETURN
  
```



```
200 PRINT "***** 3"
```

```
210 RETURN
```

```
RUN
```

```
$$$$ 1
```

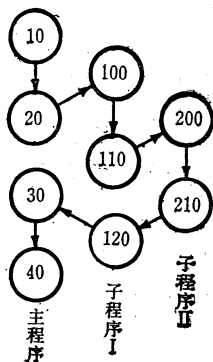
```
&&&& 2
```

```
***** 3
```

```
AAAA 4
```

该子程序嵌套为2级。若子程序嵌套超过25级深度，则输出错误信息

```
? OUT OF MEMORY ERROR
```



9. REM 语 句

结构 <标号> REM <注释文字的内容>

举例 求方程 $AX^2+BX+C=0$ ($B^2-4AC \geq 0$) 的根。

```
10 REM TO FIND QUADRATIC EQUATION'S
    ROOTS
```

```
20 PRINT "THE VALUES OF A,B,C"
```

```
30 INPUT A,B,C
```

```
40 LET R=SQR(B*B-4*A*C)
```

```
50 LET X1=(-B+R)/2*A:LET X2=(-B-R)/2*A
```

```
60 PRINT "X1="; X1, "X2="; X2
```

```
70 END
```

功能 REM 语句是注释语句 (REM 是 REMARK “注释” 一词的缩写)，其作用是对程序作注解和说明，便于别人和自己阅读。如上例中的第10句，说明本程序是“求解二

次方程的根”。

说明 (1) REM 语句如同 DATA 语句一样,也是非执行语句,可置于程序的任意位置(一般放在被说明的程序之前),但在列程序清单(即键入 LIST 命令)时,REM 语句的字符全部照原样打印出来。

(2) REM 语句通常使用文字语言,一行写不下(即超过 239 个字符)时,可另起一行,但必须在第二行前面再使用 REM 语句。

10. DEF 语 句

结构 <标号> DEF <函数名> (X) = <表达式>

举例

```
5 LET C=3.1416/180
10 DEF FNS(X)=SIN(X*C)
20 DEF FNT(X)=TAN(X*C)
30 PRINT "X", "SINX", "TANX"
40 FOR D=1 TO 90
50 PRINT D, FNS(D), FNT(D)
60 NEXT D
70 END
```

功能 DEF 语句是自定义函数语句(DEF 是 DEFINE “定义”的缩写)。在计算时,可以自己定义一些函数,以便于程序中反复使用。鉴于标准函数 SIN(X) 和 TAN(X) 中的自变量取值为“弧度”,感到较多不便,上例中我们就自己定义了 FNS(X) 和 FNT(X),其中自变量 X 就可以用角度来表示了。

说明 (1) 自定义函数语句中的函数名有两个部分组成：前面必须是 FN(FUNCTION之缩写), 后面跟一个(或两个)字母, 也可跟一个字母与一个数字。例如: FNAB、FNL6、FNQZE 等。

(2) 自定义函数的自变量是一个用圆括弧括起来的简单变量(允许一个或多个简单变量)。它是个虚设变量, 但须与等号右边的表达式中变量一致。

(3) 自定义函数语句只能在一个语句行内写完(即 239 个字符)。

(4) 在程序运行中, 凡使用自定义函数, 必须将括弧中的虚设变量换成已赋值的实际变量或常数或表达式。

(5) 自定义函数内可套用自定义函数。例如:

```
10 DEF FNS(X)=X-1
20 LET B=300
30 LET A=FNS(FNS(FNS(FNS(FNS(B)))))
40 PRINT A
50 END
```

11. DIM 语句

结构 <标号> DIM <数组名> (数组容量值)

举例 六名学生的数学、物理、化学考分如下表。试计算并打印他们每人的总分与平均分数。

	0	1	2	3	4
学生编号	数学	物理	化学	总分	平均分数
0	1	93	86	83	
1	2	87	92	79	
2	3	89	98	82	
3	4	91	75	93	
4	5	76	86	90	
5	6	84	93	71	

```

10 DIM C(5, 4)
20 FOR I= 0 TO 5
30 FOR J= 0 TO 2
40 READ A(I, J)
50 NEXT J
60 NEXT I
70 FOR I= 0 TO 5
80 LET A(I, 3)=0
90 FOR J= 0 TO 2
100 LET A(I, 3)=A(I, 3)+A(I, J)
110 NEXT J
120 A(I, 4)=A(I, 3)/3
130 NEXT I
140 DATA 93, 86, 83, 87, 92, 79, 89, 98
      82, 91, 75, 98, 76, 86, 90, 84, 93, 71
150 END

```

求每人总分
循环六次

功能 DIM 语句是数组定义语句,用以定义下标变量的个数。在编制程序时,人们往往将一些具有相同性质的数,组合在一起(构成数组),并将这样一大批数据作为一个整

体有顺序地存放在电脑内存之中。而 DIM 语句就是用于定义所处理数据的容量的。如上述程序中定义的数组 C 是一个二维数组，有 6 行（0~5）、5 列（0~4），共有 $6 \times 5 = 30$ 个数据。当然也可用 DIM 语句来定义一维数组或多维数组（APPLE-Ⅱ 最多允许定义 88 维数组）。

说明 （1）数组名可用一个（或两个）字母或一个字母与一个数字来表示。如：A、CB、T9 等。数组容量值一定要放在括弧内，并受到电脑内存容量的限制，不能无限扩大。

（2）若数组中数据个数不超过 10 个，可消去 DIM 语句。

（3）在一句 DIM 语句中，可定义若干个数组，如：

```
10 DIM NM(12), PQ(50), A(16)
```

（4）数组一旦定义后，在同一个程序中不允许重复定义，否则要输出错误信息：

```
? REDIM'D ARRAY ERROR
```

12. PRINT 语句和打印格式

结构 < 标号 > PRINT < 数值、表达式或字符串 >

举例

```
10 PRINT "THIS IS A SIMPLE PROGRAM"
```

```
20 LET X=10 : LET Y=30
```

```
30 PRINT "X+Y="; X+Y, "X*Y=", X*Y, "X-Y=";  
X-Y
```

```
40 END
```

```
RUN
```

THIS IS A SIMPLE PROGRAM

$X+Y=40$ $X \times Y=300$ $X-Y=-20$

功能 PRINT 语句是输出语句，即将电脑的信息（变量或表达式的值、字符串等）显示在显示屏上，若启动打印机的话，也可打印出来。

说明 (1) PRINT 语句具有运算功能，即在 PRINT 语句中若有表达式（表达式中变量已被赋值），则先将表达式进行计算，然后再输出计算结果。

(2) 凡要原样显示或打印的字符串必须外加引号后，写入 PRINT 语句中。引号起到确定字符串起、止界限的作用。因此，引号本身不能作为字符串中的一个字符。

(3) PRINT 语句后若没有任何表达式，其作用相当于输出一空行。

(4) 在 APPLE-II 机中，PRINT 语句中的“PRINT”一词可以用“?”代替。

(5) PRINT 语句有多种输出格式：

(a) 标准格式

PRINT 语句中需打印的各项以逗号“.”作为分隔标志，在屏幕上显示时每行只能显示三项，其余各项只能分别显示于第二、三……行上。其中第一项占有第1~16字符位置；第二项占有第17~32字符位置；第三项占有第33~40字符位置，若第三次内容超过8个字符，则多余部分占去下一行的第一个显示位置。例如：

```
10 LET X=2:LET Y=3
```

```
20 PRINT X, Y, X-Y, X+Y, X*Y, Y/X
```

```
30 END
```

RUN

2	3	-1
5	6	1.5
↑	↑	↑
①	①⑦	③③

对于打印格式，则要根据打印机的型号以及每行允许打印的字数而定。

(b) 紧凑格式

PRINT 语句中需打数的各项以分号“;”作为分隔标志，则相邻两项紧挨着显示或打印。例如：

10 LET X=2:LET Y=3

20 PRINT "X="; X, "Y="; Y, "X-Y"; X-Y,
"X+Y="; X+Y, "X×Y="; X×Y,
"Y/X="; Y/X

30 END

RUN

X=2	Y=3	X-Y=-1
X+Y=5	X×Y=6	Y/X=1.5
↑ ↑ ↑	↑ ↑ ↑	↑ ↑ ↑
① ③ ⑤	①⑦ ①⑨ ②①	③③ ③⑤ ③⑦

上面说到的 PRINT 语句，都是指的语句最后不带标点符号的形式。但是，PRINT 语句后面往往会带上标点符号。如果语句最后带有逗号，那么在程序中，凡是再遇到 PRINT 语句，这后一 PRINT 语句的内容将紧接着显示于（上一 PRINT 语句所显示的）同一行中的下一个显字位置；如果语句最后带有分号，那么凡是再出现的 PRINT 语句的内容将紧接着（不空格地）显示于（上一 PRINT 语句所显示的）同一行中的内容之后。例如：

10 PRINT "THIS IS A SIMPLE PROGRAM"

```

20 PRINT
30 INPUT "X, Y="; X, Y
35 IF X+Y=100 THEN PRINT "X+Y="; X+Y,
    GOTO 45
40 PRINT "***": GOTO 30
45 PRINT "X*Y="; X*Y,
50 PRINT "Y/X="; Y/X
60 END
RUN

```

THIS IS A SIMPLE PROGRAM

X, Y=39 45

X, Y=40 60

X+Y=100 X*Y=2400 Y/X=1.5

↑
①

↑
①7

↑
③3

从上述程序可知：第35句中一部分和第45句都是打印语句，语句最后均带逗号，其结果打印在同一行上。

(c) 定位格式

有时要将某一字符串或数值显示在屏幕（或打印在纸上）中央或某个特定的位置，就需要进行格式定位。

格式定位有下列几种方法：

通过击空格键进行定位

例如要将“THIS IS A SIMPLE PROGRAM”句显示在屏幕中央，根据显示屏一行允许排列40个字符的规定，上述语句共占24个字符位置，因此，应该从第 $\text{INT}((40-24)/2)+1=9$ 个字符开始显示这句话。我们可这样来输入：

```
10 PRINT "          THIS IS A SIMPLE
```



```

PROGRAM"
20 END
RUN
    _ _ _ _ _ _ _ _ THIS IS A SIMPLE PROGRAM

```

(其中“_”为空格)

通过打印格式函数 TAB(X) 进行定位

结构 <标号> PRINT TAB (指定的打印位置数 X):

<打印内容>

举例 上述程序可写作

```

10 PRINT TAB(9), "THIS IS A SIMPLE PROGRAM"
20 END

```

功能 TAB(X) 是打印格式函数, 起到打印或显示的定位作用, 它能使光标右移, 通常用于打印表格和图案的语句中。

说明 (1) TAB(X) 函数只能用于 PRINT 语句中, 不可单独作为一个语句使用。自变数 X 的取值范围: $0 \leq X \leq 256$ 。电脑将括号中的数值自动取整后, 确立字符显示的位置。

数值在 1~40 范围, 显示在第一行。

数值在 41~80 范围, 显示在第二行。

.....

数值在 241~256 范围, 显示在第七行。

TAB(0) 使光标移动到第 256 个字符的位置 (因为一句语句允许有 256 个字符), X 超过该范围则输出错误信息:

? ILLEGAL QUANTITY ERROR

(2) 打印格式函数中, 凡遇到打印字符串必须用引号引起来。

(3) 在两个或多个 TAB(X) 函数间要用分号或逗号

隔开，在同一语句中，若有多个 TAB(X) 函数，则它们在程序中的次序应根据每个 TAB(X) 函数括弧中 X 值，从小到大排列。

通过空格函数 SPC(X) 进行定位

结构 <标号> PRINT <打印内容>; SPC (空格数 X); <打印内容>

举例

```
10 FOR I=0 TO 3
20 PRINT "1 2 3 4 5"; SPC(I); "ABCDE"
30 NEXT I
40 END
```

RUN

```
1 2 3 4 5 A B C D E
1 2 3 4 5 □ A B C D E
1 2 3 4 5 □ □ A B C D E
1 2 3 4 5 □ □ □ A B C D E
```

功能 SPC(X) 是空格函数，起到将 PRINT 语句中两打印输出项间隔开 X 个空位，一般多用于打印表格或图案的语句中。

说明 (1) SPC(X) 函数只能用于 PRINT 语句中，不可单独作为一句使用。自变数 X 的取值范围： $0 \leq X \leq 255$ 。若空格数超出该范围，就会输出错误信息

? ILLEGAL QUANTITY ERROR

(2) SPC(X) 出现于 PRINT 语句中时（前后的分号可省略）可直接写成：

<标号> PRINT <打印内容> SPC(X) <打印内容>

通过 VTAB n 语句进行行定位

结构 <标号>VTAB<起始打印行号 n>

举例

10 HOME (消除屏幕上原有信息)

20 PRINT"*****"

30 VTAB 3

40 PRINT"\$ \$ \$ \$ \$"

50 END

RUN

***** ←第1行

←第2行(空行)

\$ \$ \$ \$ \$ ←第3行

功能 VTAB n 语句是行定位语句,表示 PRINT 语句中输出的信息须从指定的行号起打印,即将光标移至指定的行号处。

说明 VTAB n 是一个独立语句,但必须与输出语句配合使用,以指定输出打印的起始行号。n 为显示屏上的绝对行号。n 的取值范围为: $1 \leq n \leq 24$ 。若行号超出该范围,会输出错误信息:

? ILLEGAL QUANTITY ERROR

通过 HTAB n 语句进行列定位

结构 <标号>HTAB<起始打印的列号 n>

举例

10 HOME

20 FOR I=1 TO 5

30 HTAB I+2

40 PRINT"*****"

50 NEXT I

60 END

RUN

```
*****
*****
*****
*****
*****
*****
  ↑   ↑   ↑   ↑
 ①   ③   ⑥   ⑦
```

功能 HTAB n 语句是列定位语句，表示 PRINT 语句中输出的信息须从指定的第 n 列起打印，即将光标移至第 n 列。

说明 HTAB n 是一个独立的语句，但须与输出语句配合使用，起指定输出语句起始列号的作用。TAB(X) 函数与它有相近的功能，但 TAB(X) 只能用于 PRINT 语句中，其中 n 为显示屏上的绝对列号，其取值范围为： $1 \leq n \leq 255$ 。

当 n 在 1~40 的范围内，则光标位于本行上；而 n 在 41~80 内，则光标移到下一行……。若 n 超出上述范围，会输出错误信息：

? ILLEGAL QUANTITY ERROR

Ⅲ. 键盘命令与程序调试语句

1. 常用键盘命令

NEW 命令

NEW 命令〔注〕用于清除所有已经输入的语句和变量，通常在一个全新程序输入之前，给出 NEW 命令。NEW 命令发出后，尽管显示屏上原先的程序或运算结果仍显示着，但内存中已被清除了，从而为新程序的输入作好准备。若没有给出该道命令，则新输入的各语句将代替上一程序已经输入的具有相同标号的语句，而原先已经输入的其他标号语句并没有被破坏，依然存在，它会嵌在新的程序有关语句间，将程序搞乱。所以，若不键入 NEW 命令而输入一个新的程序，这只能起到使上一程序增添或删除若干语句的作用。

LIST 命令

LIST 命令的作用是在屏幕上列出当前已经输入的程序清单。LIST 命令可根据需要开列全部程序或局部程序的清单，它有下列几种形式：

(1) LIST RETURN: 列出全部程序清单(在全部程序输入完毕后发出该命令)。

(2) LIST <标号> RETURN: 列出某一句程序清单。
例如: LIST 30 RETURN(表示列出第30句程序清单)。

〔注〕凡属键盘命令前面不能有语句标号。

(3) LIST < 标号 >, < 标号 > RETURN: 列出某一段程序清单。例如: LIST 30, 60 RETURN(表示列出30~60句程序清单)。

(4) LIST—< 标号 > RETURN: 列出从开始到某一句程序的清单。例如: LIST—30 RETURN(表示列出从头到第30句程序清单)。

(5) LIST < 标号 >—RETURN: 列出从某一语句到程序结束的清单。例如: LIST 30—RETURN(表示列出从第30句到结束的程序清单)。

从键盘输入语句时,一般可连续输入(不空格)。一旦发出 LIST 命令后,显示屏上开列的程序清单会自动“重新构造”,将程序与语句间按一定的规格加进一些空格,使程序很齐整。

DEL 命令

DEL 命令用于删除程序中某些语句。其形式为: DEL N1, N2 RETURN。例如要删除第 100到150 句,可发出命令: DEL 100, 150 RETURN。

RUN 命令

RUN 命令将所有变量、指示器和堆栈都清除,并命令电脑执行已输入的程序。RUN 命令有两种形式:

(1) 初始执行命令形式:

RUN RETURN

即命令电脑从已给出程序的最小语句标号起开始执行。

(2) 继续执行命令形式:

RUN < 标号 > RETURN

即命令电脑从指定标号的语句执行程序。

如果程序中找不到所指定的语句标号（或标号值为负值），则输出错误信息：

? UNDEF'D STATEMENT ERROR

若所指定的语句标号大于63999，也会输出错误信息：

? SYNTAX ERROR

2. 常用程序调试语句

SPEED 语句

结构 <标号> SPEED= <数值X>

举例

```
10 SPEED=100
20 FOR N=-5 TO 5 STEP 0.25
30 PRINT TAB(N*N+10), "$"
40 NEXT N
50 END
```

功能 SPEED 语句是确定显示屏上字符显示速度的语句（当然也同时确定了打印机打印速度）。在数据较多、程序较长的情况下，当操作者要仔细审校程序或数据时，必须放慢显示屏上显示内容向上翻转的速度，这就可以使用 SPEED 语句进行调节。

说明 SPEED 语句是用于程序调节的语句，一般放在 PRINT 语句之前。显示字符的速度范围： $0 \leq X \leq 255$ ，即最低显示速度为 0，而最高显示速度为 255（一打开电脑，显示速度就设置在 255，即正常显示速度），若 X 超出该范围，会输出错误信息：

? ILLEGAL QUANTITY ERROR

上例中，若要恢复正常速度显示，可以在程序中删除第10句或输入下列语句：

```
10 SPEED=255
```

CLEAR 语句

结构 <标号> CLEAR

举例

```
10 DATA 2, 4, 6,  
20 FOR X=1 TO 3  
30 READ A(X)  
40 PRINT A(X)  
50 NEXT X  
60 CLEAR  
70 PRINT A(1), A(2), A(3)  
80 READ A(1), A(2)  
90 PRINT A(1), A(2), A(3)  
100 END  
RUN  
2 4 6  
0 0 0  
2 4 0
```

功能 CLEAR 语句是清除内存的语句。电脑执行到该语句时，将以前所有定义过的变量值都清除，同时将“数据指针”回复到 DATA 语句中第 1 个数据。上述程序中由于出现了第 60 句，因此，第 70 句打印出“0 0 0”，80 句重新从 DATA 语句中读出了前两个数据（第三个数据没有读），故 90 句只能打印出“2 4 0”。

说明 CLEAR 语句是用于程序调试的语句，同时也可

以命令形式出现，在执行程序之前，使用 CLEAR 命令以清除内存。

HOME 语句

结构 < 标号 > HOME

举例

```

10 HOME
20 FOR I=1 TO 11
30 PRINT TAB(12-I),
40 FOR A=1 TO 2*I-1
50 PRINT "$",
60 NEXT A
70 PRINT
80 NEXT I
90 END
RUN

```

功能 HOME 语句是清除显示屏语句。如果为了显示需要的程序或结果而清除屏幕上一切不需要的字符，就可使用HOME 语句。

说明 HOME 语句也是用于程序调试的语句,也可以命令的形式出现,以清除显示屏。HOME 语句与 CLEAR 语

句不同,HOME 语句不清除内存,仅清除屏幕,当然也就不会改变当前的程序和计算结果。

TRACE 和 NOTRACE 语句

结构 <标号> TRACE

<标号> NO TRACE

举例

```
10 TRACE
20 LET P=0 ; LET S=0
30 LET P=P+1
40 LET S=S+P
50 IF P<100 THEN 30
60 PRINT "S="; S ; END
RUN
#20 #30 #40 #50 #60 S=5050
```

功能 TRACE 语句是跟踪语句。它能显示当前执行到的语句标号,用以检查程序中是否有错误(如“死循环”等),以监督程序运行情况,若有问题可及时纠正。NOTRACE 语句是解除跟踪语句。若对某段程序感到较有把握,可插入 NOTRACE 语句,以解除跟踪。

说明 TRACE 和 NOTRACE 语句也都是程序调试语句。插入 TRACE 和 NOTRACE 语句并不影响程序执行和运算结果,待整个程序调试好后,可将它们从源程序中除去。

STOP 语句

结构 <标号> STOP

举例

```
10 INPUT A, B, C
20 PRINT A+B-C, B+C-A, C+A-B
```

```

30 STOP
40 LET S=(A+B+C)/2
50 LET M=SQR(S*(S-A)*(S-B)*(S-C))
60 PRINT M
70 END

```

功能 STOP 语句是程序暂停语句。在程序运行过程中，有时需要及时了解运行情况，以便对程序进行修改，往往在程序中插入 STOP 语句。当电脑运行到该语句时就暂时停止，在显示屏上显示出暂停语句的标号。在上述已知三角形三边 A、B、C，求三角形面积的程序中，电脑执行到第30句时，立即停止程序执行，并显示：

```
BREAK IN 30
```

表示程序暂停于第30句。由操作者根据第20句所显示出的三个数值来判别“三角形任意两边之和大于第三边”的条件是否满足。如果条件不满足，则重新输入 A、B、C 值（即转到第10句）；如果条件满足（即三个数均大于零），则键入命令：

```
GOTO 40 （或者 CONT）
```

电脑一接到命令，立即继续往下执行。

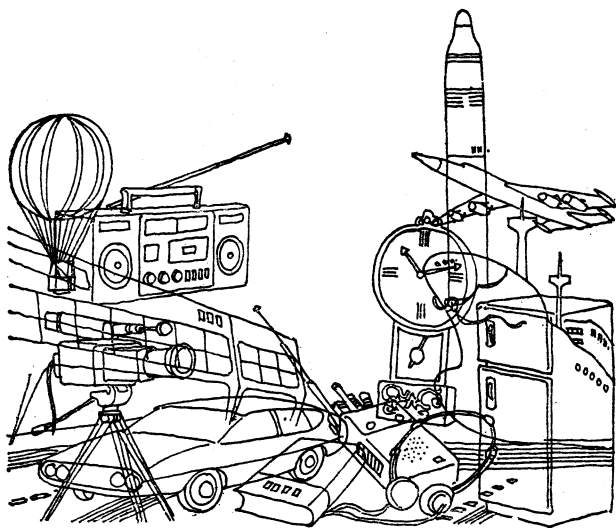
说明 STOP 语句也是用于程序调试的语句。它与 END 语句不同，后者表示程序执行结束，而 STOP 语句仅表示程序执行暂停，一俟程序修改完毕，又可继续往下执行。

上面我们所介绍的是最基本的浮点 BASIC 语言。在此基础上，我们可以进一步学习扩展的浮点 BASIC 语言（包括字符串变量的输入和运算、扩展的 IF-THEN 语句、ON 语句、GET 语句、POP 语句、POKE 语句、WAIT 语句、CALL

语句、PDL(X)函数、PEEK 函数、USR(X)函数等，以及一系列绘图命令和文件管理语句)。有关扩展的浮点BASIC语句请参见电脑语言方面的专著。整个浮点BASIC语言表，我们已列在附录Ⅶ中。

目前，电脑语言这个大家庭中，成员日益增多，新成员不断诞生。但是，BASIC语言仍是这个大家庭中一支很有活力、很有前途的家族。这个家族本身成员也较多，它有好几十种版本，然而最基本的BASIC语句却是大同小异的。本书所介绍的基本BASIC语言对于一般微电脑来说，基本上是通用的。因此，这一部分介绍的内容可以作为电脑语言初学者和大广青少年学习BASIC语言的入门。

趣味程序



冬冬抛球

下课铃一响，调皮的冬冬就跑到教学大楼的阳台上，从10米高处，向操场上抛下了一只小皮球。这只皮球弹性特别好，第一次弹起了5米高，再落下，第二次又弹起了2.5米高……就这样，小皮球落下又弹起，但每次弹起的高度总为落下时高度的一半。小皮球第一百次落地后，就再也弹不起来了。请问，这个小皮球在空中总共走过了多少距离？



好，第一次弹起了5米高，再落下，第二次又弹起了2.5米高……就这样，小皮球落下又弹起，但每次弹起的高度总为落下时高度的一半。小皮球第一百次落地后，就再也弹不起来了。请问，这个小皮球在空中总共走过了多少距离？

我们不妨分析一下：这个小皮球从10米高处抛下，第100次落地后

后再也弹不起来了。这就表明有99次落地后是弹起来的，并且每次弹起的高度为前一次落下时高度的一半，因此，我们可以列出算式：

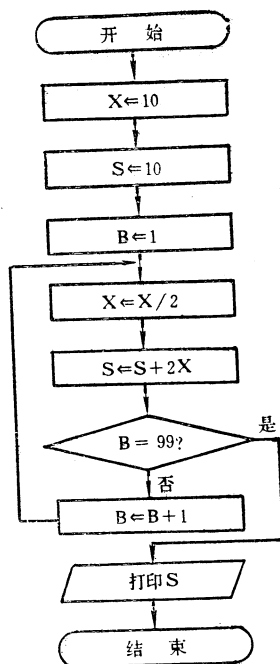
$$10 + (2 \times 5) + (2 \times 2.5) + (2 \times 1.25) + \dots$$

99个乘积项之和

这是一个求100项和的运算，且后一项乘积是前一项乘积

的一半，要是用笔列算式来算，那是够麻烦的。但是请微电脑来帮忙，算起来可就快多了。不过得把冬冬抛球的情况详详细细告诉电脑，这就需要编写程序。

我们用变量X表示小皮球每次落地后弹起的高度；用S表示小皮球在空中走过的路程（先将初始抛下高度告诉电脑），它实际上是个累加器；并请电脑每做一次加法，自动记数一次，记数到第99次（即加到第100项）后，就停止运行，这个计数器我们设它为B。这样设定后，我们就可以画出流程框图，并写出程序了。



```

10 REM DROPPING THE BALL
20 LET X=10
30 LET S=10
40 FOR B=1 TO 99
50 LET X= X/2
60 LET S=S+2*X
70 NEXT B
80 PRINT "TOTAL DISTANCE THE BALL"
90 PRINT "TRAVELS IS□□□", S; "□□□
METERS"
95 END
  
```

]RUN

TOTAL DISTANCE THE BALL

TRAVELS IS 30 METERS

假如冬冬是从15米高处将小皮球抛下，但这只小皮球每次弹起的高度是落下时的 $1/3$ ，并且第25次落地后小皮球再也弹不起来了。你能否编个程序，请电脑帮忙算一下小皮球在空中走过的距离吗？

上述问题的数学模型也可归结为：

$$S = 10 + 10 \left[1 + \frac{1}{2} + \cdots + \left(\frac{1}{2} \right)^{25} \right]$$
$$= 20 \left(1 - \frac{1}{2^{25}} \right) + 10$$

你能根据这个数学模型，重新设计流程图并编制一个简单的程序吗？

扔给猴子的胡桃

著名物理学家、诺贝尔奖金获得者李政道博士访问安徽合肥中国科技大学少年班时，曾提到一个世界著名的趣味数学题：

遇难船上的五名水手来到一个荒岛上。他们在那儿收集了一些胡桃，准备作为后几天的



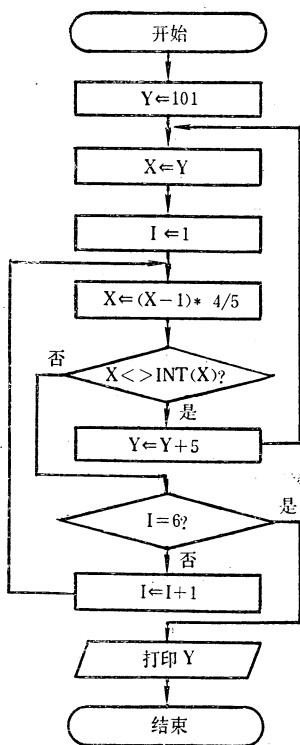
口粮。水手们在驻地中间把胡桃堆成一堆，夜间轮流值班看守。当第一个水手值班时，他担心以后分配胡桃时会被哄骗，就独自把胡桃平分成五份，并拿去了自己的一份。分完后还剩下一个胡桃，他顺手将这个胡桃扔给呆在旁边的一只猴子。轮到第二个水手守护时，他也产生了同样的疑心，便把剩余的胡桃也平分成五份，自己拿走一份。分好后又剩一个，他也把它扔给了那只猴子。第三、四、五个水手都如此办理。第二天早晨，五个水手一起又把最后剩下的胡桃平均分成五份，最后剩下一个胡桃还是给了那只猴子。

请算一下，白天水手们至少收集了多少个胡桃？

根据上述题意，我们可先作如下分析：设每个水手来值勤之前的胡桃数为 y ，而这个水手走了之后剩下的胡桃为 x ，则可写成如下的关系式：

$$x = \frac{4}{5}(y - 1)$$

根据这个关系式编程时，我们可采用迭代的方法，先给 Y 一个初值，然后将此式迭代 6 次。每次迭代完毕，还须考查结果是不是整数？如果不是，说明初值不合理，要改变初值，重新开始分。



这里，初值和循环步长的选择是很关键的，根据题意，我们选101为初值，5为步长。下面是有关的流程图和程序：

```
10 Y=101
20 X=Y
30 FOR I=1 TO 6
40 X=(X-1)*4/5
50 IF X<>INT(X) THEN 90
60 NEXT I
70 PRINT "Y=", Y
80 END
90 Y=Y+5
100 GOTO 20
```

]RUN

Y=15621

卡尔解题的巧妙方法

法国数学家卡尔·弗里得里希·高斯（1777~1855）从小就表现出非凡的数学才华。他十岁那年，老师上算术课时给大家出了一个题目： $1+2+3+\cdots+99+100=?$ 老师刚一写完题目，小卡尔立即举起了手：



“老师，这一百个数的和是5050”。同学们都用惊奇的目光注视着他，卡尔是怎么算出来的呢？

卡尔的算法很简单，思路却很巧妙。原来他发现这100个数有个奇妙的特性，就是如果依次将头尾两个数加起来，其和都是101，

$$\begin{array}{rcccccc}
 & 1 & 2 & 3 & \cdots & 49 & 50 \\
 + & 100 & 99 & 98 & \cdots & 52 & 51 \\
 \hline
 & 101 & 101 & 101 & \cdots & 101 & 101
 \end{array}$$

这样的数刚好50对，于是总和就是 $101 \times 50 = 5050$ 。

小卡尔算得确实很快，但是与电脑的计算速度相比，那却算不了什么了。

下面是用微电脑算这个题目的简单流程图和程序：

```

10 LET P=0
20 LET S=0
30 LET P=P+1
40 LET S=S+P
50 IF P<100 THEN 30
60 PRINT "S="; S
70 END

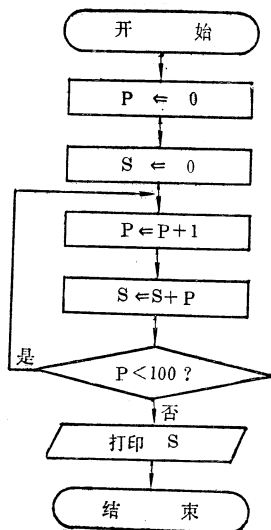
```

```

]RUN
S=5050

```

这个程序设计的基本思想是这样的：考虑到电脑运算速度相当快，因此，采用逐项累加的方法，而不是采用小卡尔的巧妙算法（当然，也可以采用小卡尔的



方法)。这样，程序相当简单，整个程序中仅设置了两个变量：即计数变量（参与累加的各个数）P 和累加变量（又称累加器）S。

第10、20两句，分别将P和S置于“0”。第30句使计数变量P由0增值为1。第40句是累加语句，将累加器初值S与计数值P相加，并将其和送给累加变量S。第50句是条件转向语句，判断计数值P是否小于100。条件成立的话，程序转至第30句，使计数值P（即参与累加的数字）加上1后，再重复执行第40、50句。即使在 $P=99$ 时，经判断P仍小于100，也同样转至第30句，使 $P=99+1=100$ ，并将这个100也加到累加器S之中。这时，再判断P值，已经不再小于100了。因此，程序就转到第60句，打印出累加值S。第70句是程序结束语句。

仿照上述程序的设计思想，你能否自己编制几个简单的程序并计算一下：

- (1) 求从101到200连续一百个自然数之和。
- (2) 求1000之内所有偶数的和。
- (3) 求1000之内所有5的倍数的数之和。

神机妙算变“方阵”

春秋战国时期，燕国举行庆典或军事演习，往往将士兵排列成实心的“方阵”。

一天，燕昭王举行阅兵。乐毅大将把训练有素的几千名精锐将士排列成62个方阵，每个方阵的人数全都一样。燕昭王在检阅台上看得十分高兴。

这时，乐毅大将亲自加入到将士们的行列中，只见战旗一挥，阵势大变，全体将士（包括乐毅大将本人在内）正好排成了一个新的大方阵。

你能否算一下，这些将士共有多少人？

我们不妨这样分析：

假设原来的62个小方阵其每边人数为 y ，乐毅大将加入阵列后，所排成的新大方阵每边人数为 x ，那么上述题意，我们可以列出方程：

$$62y^2 + 1 = x^2$$

显然，这是一个不定方程（即未知数的个数多于方程的个数），在一个方程中包含了两个未知数 x 与 y ，并且这两个未知数的次数都为2。要求解这样一个两次不定方程，确实是一件很不容易的事。

这时，我们又可以请电脑来帮一下忙了。电脑一定能迅速地通过一次次判断，找出满足上述条件的 x^2 值（即包括乐毅大将在内的所有将士人数）。根据以上分析，我们画出了流程图，并遵循流程图的逻辑关系，用BASIC语言编制出如下程序。

```
10 FOR X=7 TO 100
20 Y=1
30 IF 62*Y*Y+1<X*X THEN Y=Y+1
```



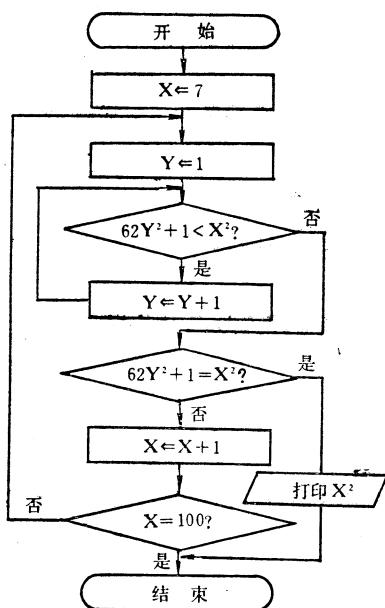
```

      GOTO 30
40  IF 62*Y*Y+1=X*X THEN 60
50  NEXT X
60  PRINT "X*X="; X*X
70  END

```

]RUN

X*X=3969



下面对程序设计作简单分析：

由于题目告诉我们，这些将士是“几千人”，因此，所排成的大方阵的每边人数不可能超过100人。如果方阵每边为1人，那么方阵一共只有1个人；如果每边为2人，那么方阵总人数就是4个……。由于原来的小方阵至少为1个人，共有62个小方阵，则大方阵的全体将士至少为63人。为此，

我们可以设置循环从 $x=7$ 开始，一直循环至 $x=100$ 为止（第10句）。然后，设原先的小方阵每边人数为1（第20句）。接着进行判断： $(62y^2+1)$ 是否小于 x^2 ？结论成立的话，就将 y 增值1，再进行上述判断（第30句）；若结论不

成立（即 $(62y^2+1)$ 不小于 x^2 ）就进一步判断： $(62y^2+1)$ 是否等于 x^2 （第40句）。如果结论不成立，就将 x 增值1后进入循环（第50句），即进入第10句，只要 x 的值不大于100，就允许一次又一次循环，其间总能找到一个 x 值满足条件 $62y^2+1=x^2$ ；如果结论成立（即条件满足），就打印出 x^2 值（第60句），程序的运行也就结束了（第70句）。

运行结果表明： x^2 的值为 3969，也就是说，所排成的新方阵的每边人数为63人，原先小方阵每边人数为 8 人。

“韩信点兵”问题

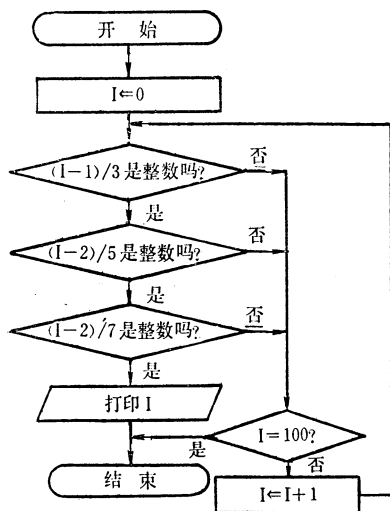
“韩信点兵”术是我国古代流传着的一种有趣的算法。韩信是我国汉代的一位大将。传说他计算士兵的方法很奇特。他不是一五一十地数，而是让士兵列队操练。分别以三人一排，五人一排和七人一排列队，然后将各种不同人数列队时余下的士兵数记下来，他就能知道士兵的总数了。当然这种算法的名称不止一个，宋朝的周密叫它“鬼谷算”，杨辉叫它“剪管术”，外国人称之为“中国余数定理”。最初记述这类算法的是一本名叫“孙子



算经”的书，还有一首顺口溜流传至今：

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。

这就是韩信点兵的计算方法。



为了说明这类问题，可以举一个简单的例子：有一篮红枣（数目在100以内），如将它们分成3颗一组，余1颗；分成5颗一组，余2颗；分成7颗一组，余2颗。问这篮红枣到底有多少？

今天我们如用微电脑来解这个问题，当然是很方便的。

下面是流程图和程

序：

```

5  FOR I=0 TO 100
10  IF (I-1)/3<>INT((I-1)/3) THEN 50
20  IF (I-2)/5<>INT((I-2)/5) THEN 50
30  IF (I-2)/7<>INT((I-2)/7) THEN 50
40  PRINT I
50  NEXT I
  
```


]RUN

37

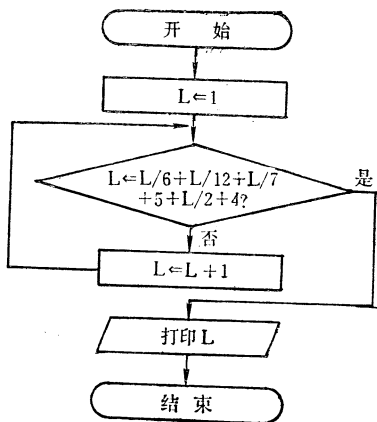
刁藩都的年龄之谜

关于古希腊数学家刁藩都的年龄，任何历史书上都没有明确的记载，他的生平更是无人知晓。然而，在他的墓志铭中却留下了关于他的生平的资料。这个墓志铭非常奇特，象是一个谜语，又象是一道数学题目，让我们来看看墓志铭的全文吧：



过路人！这里埋着刁藩都的骨灰。下面的数字可以告诉您，他一生寿命究竟有多长。他的生命的六分之一是幸福的童年。再活了他寿命的十二分之一，颊上长起了细细的胡须。刁藩都结了婚，可是还不曾有孩子，这样又度过了一生的七分之一。再过五年，他得了头生儿子，感到很幸福。可是命运给这孩子在这世界上的光辉灿烂的生命，只有他父亲的一半。自从儿子死了以后，这老头儿在深深的悲痛中活了四年，也结束了尘世生涯。

显然，如果只从这墓志铭的谜一样的语言中去猜测刁藩



都的年龄，那是很不容易的。然而，只要将这些话仔细分析，再设计一个简单的程序，在微电脑的帮助下，就能很快揭开这位数学家的年龄之谜。

下面就是解这个问题的流程图和 BASIC 程序：

```

10 L=1
20 IF L=L/6+L/12+L/7+5+L/2+4 THEN PRINT
   L: END
30 L=L+1: GOTO 20
  
```

JRUN

84

猪八戒分西瓜

猪八戒随唐僧、孙悟空和沙和尚一起到西天取经。有一天，天热极了，猪八戒找到了一块西瓜地，于是，他独自一人先畅开肚皮吃了个够，然后将剩下的西瓜中的一半和另外的半个西瓜留给师父唐僧；再将所剩下的一半又半个西瓜留给孙悟空；将最后剩下的西瓜的一半又半个给沙和尚。可是巧得很，分西瓜时八戒懒得动手，没有切开过任何一个西瓜，而分到最后又正好分完。你说猪八戒吃剩的西瓜共有多少？唐僧、孙悟空、沙和尚每人又各得到几个西瓜？

用微电脑来解这个问题时，我们可先作如下分析：分瓜时每人都多得到半个西瓜，但又没有用刀切开过，这说明大家得到的西瓜数都是整数。我们可建立下面的方程式：

$$x = y/2 + 1/2$$

其中， y 是每次剩下的西瓜总数。共 3 人，每人得到又是整数，可见剩下的西瓜数至少是 3 个。

设初值 $y_1=3$ ，然后用微电脑判别，每次分后所得的数 x 是否整数，如果不是整数，说明原有西瓜总数不对，令 $y_1+1=y_1$ ，通过运算再判别，一直到 3 次分瓜后 x 全是整数为止，并打印结果。

下面是这道题的流程图和 BASIC 程序：

```

5  DIM  A(3)
10  Y1=3
20  Y=Y1
40  FOR I=1 TO 3
50  X=Y/2+1/2
60  IF ABS(X-INT(X))<>0 THEN 120
70  A(I)=X
80  Y=Y-X
85  NEXT I

```



```

90 PRINT "Y="; A(1)+A(2)+A(3)
95 PRINT A(1)
100 PRINT A(2)
105 PRINT A(3)
110 END
120 Y1=Y1+1
130 GOTO 20

```

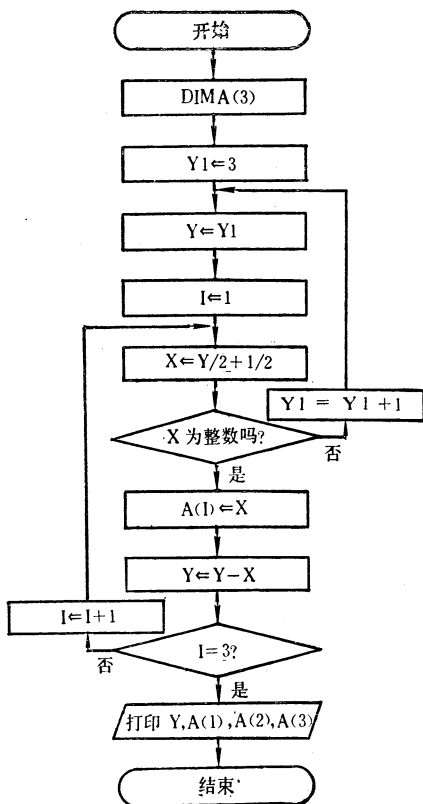
JRUN

Y=7

4

2

1



分 苹 果

托儿所里有一百多只苹果，要将它们分成若干组。如果分成5只一组，则最后余下3只；如果分成7只一组，则余下2只；如果分成9只一组，也多余2只。那么，这批苹果共有多少只呢？

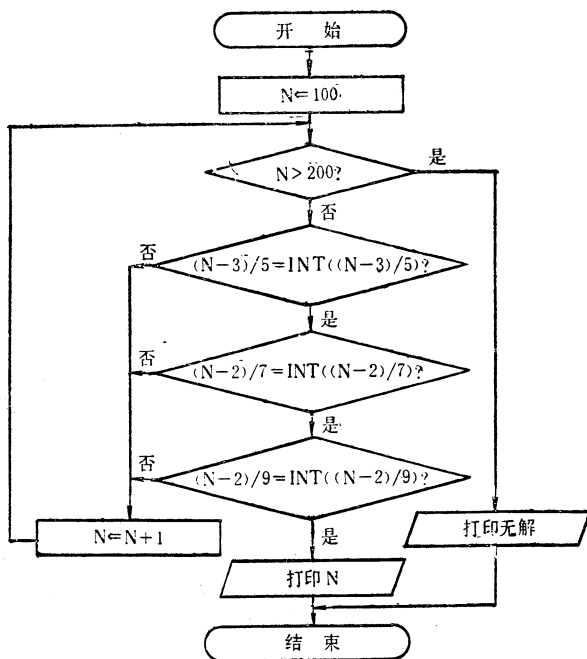


这批苹果的数目 N 是一个介于100到200间的数；而且应满足下述3个条件：即 $(N-3)/5$, $(N-2)/7$, $(N-2)/9$ 均为整数。根据这两点我们可以运用微电脑通过循环语句进行判断，满足以上条件的数就打印出来。下面是流程图和程序：

```
10 FOR N=100 TO 200
20 IF  $(N-3)/5 <> \text{INT}((N-3)/5)$  THEN 70
30 IF  $(N-2)/7 <> \text{INT}((N-2)/7)$  THEN 70
40 IF  $(N-2)/9 <> \text{INT}((N-2)/9)$  THEN 70
50 PRINT "NUMBER OF APPLES IS"; N
60 GOTO 90
70 NEXT N
80 PRINT "NO SOLUTION"
90 END
```

JRUN

NUMBER OF APPLES IS 128

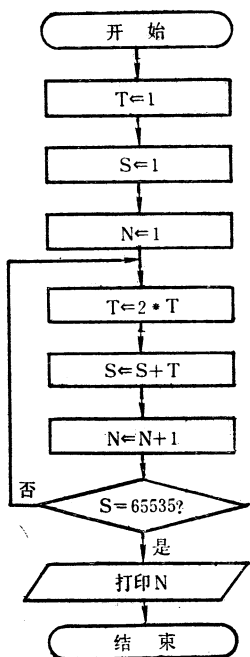


历尽创伤的勇士

传说古代某个国家有这么一个规定：出征的士兵第一次负伤发给抚恤金 1 个钱币，第二次受伤后发给 2 个钱币，第三次则发给 4 个钱币，依次类推，每负伤一次所得的钱币是

前一次的 2 倍。战争结束后有一个士兵，共得到抚恤金 65535 个钱币。试编写一个程序，用微电脑来算一下，这个勇士在沙场上一共负伤多少次。

以下是流程图和程序：



```

20  T=1:S=1:N=1
30  T=2*T
35  S=S+T
40  N=N+1
50  IF S=65535 THEN 70
60  GOTO 30
70  PRINT N
99  END
  
```

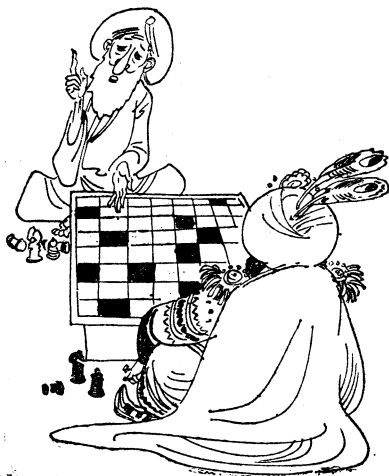
JRUN

16

无法赏赐的米

传说古印度有个国王，天性好玩。有一次，他下令在全国张榜招贤：谁能找到奇妙的游戏，将给予重赏。

有一个术士揭了榜。他发明的一种棋，使国王爱不释手。国王问术士：“你希望得到什么？”术士一笑答道：“我没有什么特殊的要求，只请大王在那棋盘的第一格子里放1粒米，在第二格子里放2粒米，在第三格子里放4粒米，然后在每一格子里都放进比前一



格子多一倍的米，64个格子放满了，我的要求也就满足了。”国王一听，满口答应。可是，旁边的算师的计算结果，却使国王大吃一惊，原来这些米可以覆盖整个地球，根本无法赏给术士。

下面我们请微电脑来算一下，这些米到底有多少。

设米粒数为 T ，体积为 V 立方米，由于1立方米的米粒约为 1.42×10^8 粒，则要计算出棋盘格上放置的米的体积，可通过下面的公式：

$$V = T / 1.42 \times 10^8 = (1 + 2^1 + 2^2 + 2^3 + \cdots + 2^{63}) / 1.42 \times 10^8$$

这样，我们就可以编写出如下的流程图和程序：

```

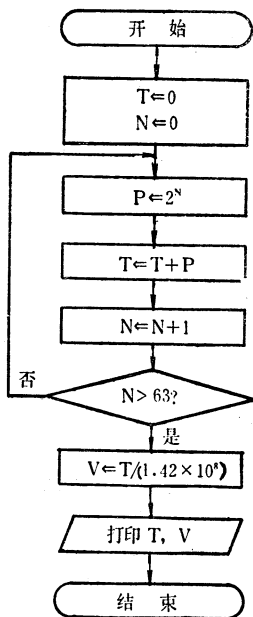
10 LET T=0
20 LET N=0
30 LET P=2^N
40 LET T=T+P
50 LET N=N+1
60 IF N<=63 THEN 30
70 LET V=T/(1.42E+8)
80 PRINT "T=", T
90 PRINT "V=", V
100 END

```

JRUN

T=1.84467441E+19

V=1.29906648E+11



可见，这个术士的胃口真大呀！

假如设地球的半径为 6.378×10^6 米，请读者计算一下，这些米能在地球表面铺上多厚的一层？

小猴与桃子

一群小猴采摘了一批桃子。第一天，它们吃掉这批桃子的一半又一只；第二天吃掉了余下的一半又一只；第三天再吃掉余下的一半又一只，它们每天按这个规律吃桃子，到第十天就只剩下一只了（即第九天余下的）。问这批桃子共有



多少只？

按题意仔细分析，我们
可得出小猴吃桃子的规律：

留给第二天（即第一天
吃剩）的桃子数为：

$$\begin{aligned} A_2 &= A_1 - \left(\frac{A_1}{2} + 1 \right) \\ &= \frac{A_1}{2} - 1 \end{aligned}$$

$$\text{即 } A_1 = 2(A_2 + 1)$$

留给第三天（第二天吃

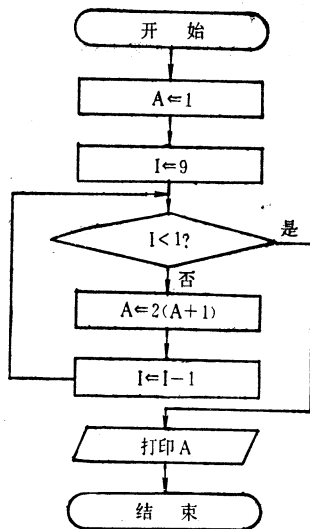
剩）的桃子数为：

$$\begin{aligned} A_3 &= A_2 - \left(\frac{A_2}{2} + 1 \right) \\ &= \frac{A_2}{2} - 1 \end{aligned}$$

$$\text{即 } A_2 = 2(A_3 + 1)$$

依次类推， $A_3 = 2(A_4 + 1)$ ， $A_4 = 2(A_5 + 1)$ ，…… $A_9 = 2(A_{10} + 1)$ 。到
第九天，只吃剩下一只留给
第十天吃，因此 $A_{10} = 1$ 。

根据这个规律，我们可
以画出流程图和编写出如下
程序，用微电脑来解：



```

10 LET A=1
20 FOR I=9 TO 1 STEP -1
30 LET A=2*(A+1)
  
```

```

40 NEXT I
50 PRINT "A="; A
60 END

```

]RUN

A=1543

马克思做过的数学题

在马克思的《数学手稿》中有这样一道题目：

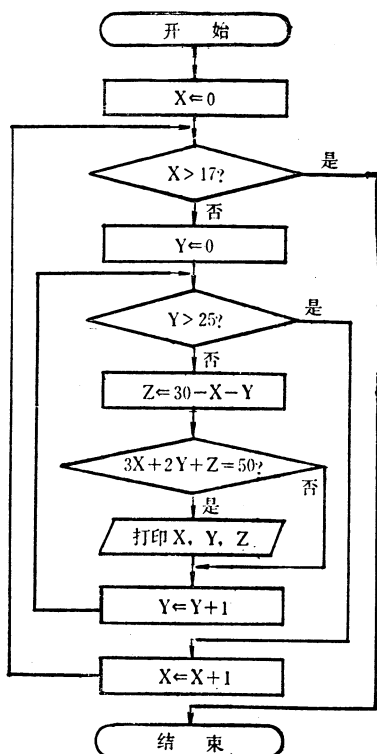
有30个人（其中包括成年男人、女人和小孩）到饭馆里用餐，每个男人花去3先令钱，每个女人花去2先令钱，每个小孩花去1先令钱。他们总共花去50先令。问：男人、女人、小孩各有多少人？

设有男人 x 人，女人 y 人，小孩 z 人，那么根据题意，我们可以列出如下方程组：

$$\begin{cases} x + y + z = 30 & (1) \\ 3x + 2y + z = 50 & (2) \end{cases}$$

显然，这是一个不定方程组，有三个变量但只有两个方程。根据方程 (2)，我们来确定 x, y, z 的取值范围。由于共花去50先令，因此，男人绝对不可能超过17人（ $0 \sim 17$ ）





人)，女人绝对不可能超过25人（0～25人），而小孩人数则为（ $30 - x - y$ ）人。这样，我们就可以让电脑作两次循环，在循环中判断所求出的男人、女人与小孩的组合是否符合方程（2）的条件，凡符合条件就打印结果。

下面是用电脑解答这个问题的流程图和程序：

```

5 PRINT "X", "Y", "Z"
10 FOR X=0 TO 17
20 FOR Y=0 TO 25
30 LET Z=30-X-Y
40 IF 3*X+2*Y+Z=50 THEN 70
50 GOTO 80
70 PRINT X, Y, Z
80 NEXT Y
90 NEXT X
  
```

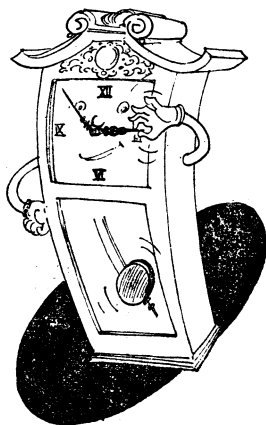
100 END

]RUN

X	Y	Z
0	20	10
1	18	11
2	16	12
3	14	13
4	12	14
5	10	15
6	8	16
7	6	17
8	4	18
9	2	19
10	0	20

爱因斯坦解难题

有一次，爱因斯坦病了，他的朋友——传记作家莫希柯夫斯基去看望他。两人谈了一阵之后，为了不使这位相对论创立者寂寞，莫希柯夫斯基就出了一道趣味数学题供他消遣，题目是这样的：“假如现在座钟的指针正指着12点整，要是将时针与分针对调一下，它们所指的时间还是合理的（仍是12点整）。但是



在其他时刻，譬如6点钟时，将时针与分针对调一下，就要闹笑话，因为这种位置是不可能存在的，即时针指着‘12’时，分针决不会指在‘6’上。于是，就产生了这样一个问题：钟的指针在什么位置上，时针与分针可以对调，并使对调后的指针仍能指示实际上可能的时刻？”

“是的，”爱因斯坦等莫希柯夫斯基一说完就回答道：“这个问题对躺在床上的病人确实是一个很好的问题，既有趣味而又不那么容易。不过，恐怕消磨不了多少时间，我已经知道应该怎样解了。”

你知道爱因斯坦是怎样快速解出这个难题的吗？

爱因斯坦将钟面整个圆周自“12”那点起分为60格（即60分钟），假如符合莫希柯夫斯基条件的时刻，时针走到了第 x 个格，分针走到了第 y 个格。由于时针走5格为1小时，分针走60格才为1小时，因此，现在时针走了 $x/5$ 个小时，而分针走了 $y/60$ 小时，也就是说，在 $(x/5 - y/60)$ 小时前，分针正指着“12”，时针也正指着钟面上某个数字上（即正点），表示自12点以后已经过了 m 个小时（ m 为从0到11的整数）。既然这个时刻满足莫希柯夫斯基条件，当然时针与分针可对换，类似地我们也可得到 $(y/5 - x/60)$ 小时前分针指着“12”，时针正指着钟面某个数字（也正好是正点），表示从12点以后已经过了几个小时（ n 为从0到11的整数）。

于是，我们可得到下述方程组：

$$\begin{cases} x/5 - y/60 = m \\ y/5 - x/60 = n \end{cases} \quad (m, n \text{ 是从 } 0 \text{ 到 } 11 \text{ 的整数})$$

从这个方程组得到：

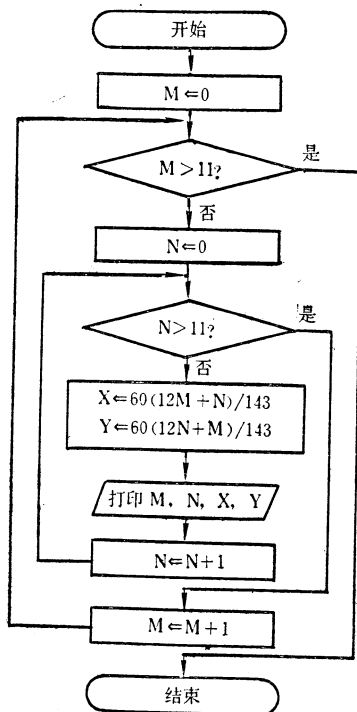
$$x = 60(12m + n)/143$$

$$(m, n = 0, 1, 2 \cdots 11)$$

$$y = 60(12n + m)/143$$

由于 m 的 12 个整数值中每一个都可与 n 的 12 个整数值中每一个相配。于是，我们可得到一系列符合莫希柯夫斯基条件的时针与分针可对换的时刻值。

根据以上分析所编制的流程图和程序如下：



```

10 PRINT TAB(1), "M", TAB(5), "N",
    TAB(10), "X", TAB(25), "Y"
  
```

```

20 PRINT
30 FOR M=0 TO 11
40 FOR N=0 TO 11
50 X=60*(12*M+N)/143
60 Y=60*(12*N+M)/143
70 PRINT TAB(1); M, TAB(5); N, TAB(10);
   X, TAB(25); Y
80 NEXT N
90 NEXT M
100 END

```

] RUN

M	N	X	Y
0	0	0	0
0	1	.41958042	5.03496504
0	2	.839160839	10.0699301
0	3	1.25874126	15.1048951
0	4	1.67832168	20.1398601
0	5	2.0979021	25.1748252
0	6	2.51748252	30.2097902
0	7	2.93706294	35.2447553
0	8	3.35664336	40.2797203
0	9	3.77622378	45.3146853
0	10	4.1958042	50.3496504
0	11	4.61538462	55.3846154
1	0	5.03496504	.41958042
:	:	:	:
11	0	55.3846154	4.61538462

11	1	55.8041958	9.65034965
11	2	56.2237762	14.6853147
11	3	56.6433566	19.7202797
11	4	57.0629371	24.7552448
11	5	57.4825175	29.7902098
11	6	57.9020979	34.8251748
11	7	58.3216783	39.8601399
11	8	58.7412588	44.8951049
11	9	59.1608392	49.9300699
11	10	59.5804196	54.965035
11	11	60	60

移 圆 盘

十九世纪有位法国数学家曾提出过这样一个问题：有 A、B、C 三根柱，A 柱上有三个按直径大小顺序排列的圆盘（即最大的在最下面而最小的在最上面）。现在要把所有圆盘移到 B 柱上去，其大小排列顺序依然不变，并规定：每次只能移动一个圆盘，且不论在哪一个柱上，小盘总是在大盘之上，但移动时允许利用 C 柱。为了完成这样的搬移，最少需要移动多少次？



我们先来分析最简单的一种情况，即假定 A 柱上有 3 个圆

盘，为了完成上述规定的搬移，从图示可以看出，最大的一个必须移动 1 次（即 2^0 ），中间的那个必须移动 2 次（即 2^1 ），而最小的那个必须移动 4 次（即 2^2 ），所以总共至少移动 $2^0 + 2^1 + 2^2 = 2^3 - 1 = 7$ （次）才能将 A 柱上 3 个盘移到 B 柱上。

同理，我们可求得 A 柱上若有 4 个圆盘要移动到 B 柱上去，至少要移动：

$$\begin{aligned} 2^0 + 2^1 + 2^2 + 2^3 \\ = 2^4 - 1 = 15 \text{ (次)} \end{aligned}$$

A 柱上有 5 个圆盘，至少移动：

$$2^5 - 1 = 31 \text{ (次)}$$

.....

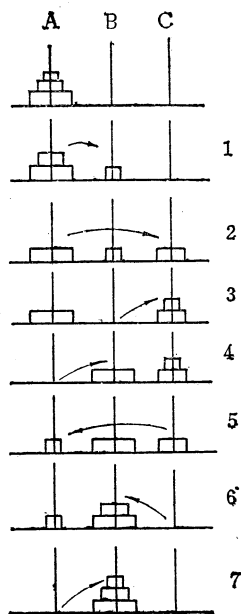
A 柱上如果有 n 个圆盘，则至少移动：

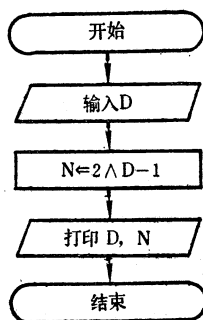
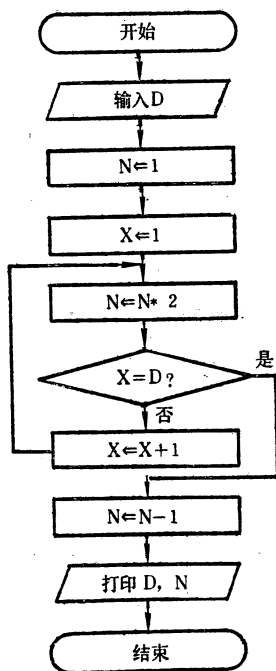
$$2^n - 1 \text{ (次)}$$

现在我们请微电脑来算一下，如果有 64 个圆盘，至少要移动多少次呢？以下是程序和流程框图。我们可以用下述两种方法来编制程序。请读者分析一下，哪种方法更合理些。

计算结果表明这个数字是相当大的！

假定每移动一次需要 1 秒钟，一年以 $365 \times 24 \times 3600 = 31536000$ 秒计算，则需要将近 5800 亿年的时间。显然，这样的搬移由人去做是根本不可能的。因为据科学家推测，地





球的年龄也不过只有几十亿年而已，而整个太阳系的寿命也不会超过 200 亿年。难怪有人说，当这 64 个圆盘全部搬移到另一根柱上时，“世界的末日”就要来到了。

```

10 PRINT "NUMBER OF DISCS",
20 INPUT D
25 N=1
30 FOR X=1 TO D
40 N=N*2
50 NEXT X
55 N=N-1
60 PRINT D, " DISCS REQUIRES "; N; "
   MOVES"
70 END
  
```

]RUN

NUMBER OF DISCS ?64

64 DISCS REQUIRES 1.84467441E+19 MOVES

10 PRINT "NUMBER OF DISCS",

20 INPUT D

30 $N=2^D-1$

60 PRINT D; " DISCS REQUIRES "; N;
"MOVES"

70 END

]RUN

NUMBER OF DISCS ?64

64 DISCS REQUIRES 1.84467441E+19 MOVES

买 水 果



用5元钱买100个水果，其中每个西瓜的价钱是5角，苹果每个1角，李子1分钱3个。试编一个程序，求出每种水果各买多少个。

解这个题目，我们可以设西瓜的个数为 x ，苹果的个数为 y ，李子的个数为 z 。如果5元钱全部用于买

西瓜，至多可买10个，故 x 的取值范围为 $1 \sim 10$ ；如果5元钱全部用于买苹果，至多可买50个，故 y 的取值范围为 $1 \sim 50$ ；而 $z = 100 - x - y$ 。所以，只要满足方程： $50x + 10y + (100 - x - y)/3 = 500$ 就行了。

为此，我们可以通过设置关于 X 和 Y 的两重循环，很快找出满足上述方程的 x 、 y 、 z 的解。

```

10 FOR X=1 TO 10
20 FOR Y=1 TO 50
30 Z=100-X-Y
40 IF 50*X+10*Y+Z/3=500 THEN 70
50 NEXT Y; NEXT X
60 END
70 PRINT "X="; X; "Y="; Y; "Z="; Z
80 GOTO 50

]RUN
X=2 Y=38 Z=60

```

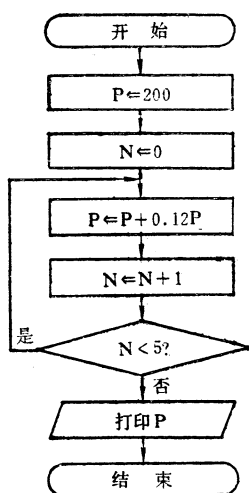
高 利 贷

解放前，地主用高利贷剥削农民的情况，今天的青少年可能不一定都了解。下面我们举一个例子，大家用微电脑算一下就可以知道，高利贷对农民来说是一种多么残酷的剥削。

一个农民向地主借了200元钱买种子，这笔高利贷月息为



12%，每月还须将利息归入本金，请算一下，五个月后农民要还给地主多少钱？下列流程图和 BASIC 程序能帮助你算出这笔账。

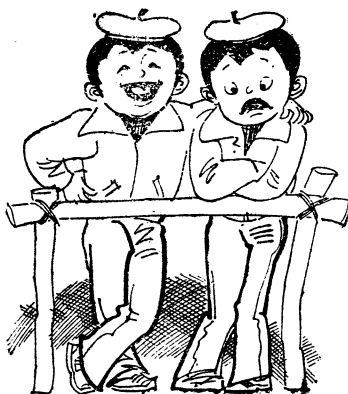


```

10 LET P=200
20 FOR N=1 TO 5
30 LET P=P+0.12×P
40 NEXT N
50 PRINT P
60 END
  
```

JRUN
352.468337

笑哈哈与愁煞人



下图是用微电脑打印出来的两个脸形，左面一个在哈哈大笑，右面一个却是愁容满面。

你知道微电脑是怎样将它们打印出来的吗？请看以下程序，其中第 100 句由你输入一个数值，再用 300 句

进行判断，如果输入的数是1，那么打印结果产生右面的愁眉苦脸；如果输入的是1以外的其他数，那么打印出的脸蛋总是笑呵呵的。

```

5  INPUT  S
10 PRINT  TAB(19)"*"
15 PRINT  TAB(16)"*****"
20 PRINT  TAB(15)"*****"
25 PRINT  TAB(14)"*";  TAB(24)"*"
30 PRINT  TAB(13)"*";  TAB(17)"-";
    TAB(21)"-";  TAB(25)"*"
35 IF S=1 THEN 70
40 PRINT  TAB(12)  "*" ;  TAB(17)" ^";
    TAB(21)"^";  TAB(26)"*"
45 PRINT  TAB(12)"*";  TAB(19)"#";
    TAB(26)"*"
50 PRINT  TAB(12)"*";  TAB(15)" * * *
    * *"; TAB(26)"*"
55 PRINT  TAB(13)"* *"; TAB(22)"* *"
60 PRINT  TAB(14)"* * * * * *"
65 PRINT  TAB(15)"*****"; GOTO
    105
70 PRINT  TAB(12)"*"; TAB(26)" *"
75 PRINT  TAB(12)"*"; TAB(17)" O"; TAB(21)
    "O";  TAB(26)"*"
80 PRINT  TAB(12)  "*" ; TAB (19)"#";
    TAB(26)"*"
85 PRINT  TAB(13)  "*" ; TAB(17)"*****";
    TAB(25)"*"

```


多少“英里”或“公里”。下述BASIC程序计算并打印出了一张从 0 到 100000 英尺的高度转换表(步长为 5000 英尺)，读者也可略微修改该程序，以编制一张用公里来表示高度的转换表(1 公里=0.6214英里=1093.6码=3280.8英尺)，当然也可以编出以“米”为单位的转换程序。

```

10 PRINT "ALTITUDE", "ALTITUDE"
20 PRINT "IN FEET", "IN MILES"
30 FOR F=0 TO 100000 STEP 5000
40 LET M=F/5230
50 PRINT F, M
60 NEXT F
70 END

```

IRUN

ALTITUDE IN FEET	ALTITUDE IN MILES
0	0
5000	.946969697
10000	1.89393939
15000	2.84090909
20000	3.78787879
25000	4.73484849
30000	5.68181818
35000	6.62878788
40000	7.57575758
45000	8.52272728
50000	9.46969697
55000	10.4166667

60000	11.3636364
65000	12.3106061
70000	13.2575758
75000	14.2045455
80000	15.1515152
85000	16.0984849
90000	17.0454546
95000	17.9924243
100000	18.9393939

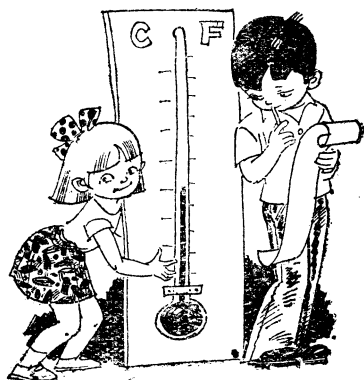
温 度 转 换

试将华氏温度F转换为摄氏温度C，并打印出F=32、34、36、……98、100度时相应的摄氏温度。

由于华氏温度与摄氏温度间存在下述关系：

$$C = (5/9) \times (F - 32)$$

因此，我们只需用循环语句（设步长为2）便能算出华氏32度起到100度时的相应摄氏温度。



以下是用 BASIC 语言编写的程序。

```

10 PRINT "F", "C"
20 FOR F=32 TO 100 STEP 2
30 LET C=(5/9)*(F-32)
40 PRINT F, INT (C)/100
50 NEXT F

```

60 END

]RUN

F	C
32	0
34	1.11
36	2.22
38	3.33
40	4.44
42	5.55
44	6.66
46	7.77
48	8.88
50	10
52	11.11
54	12.22
56	13.33
58	14.44
60	15.55
62	16.66
64	17.77
66	18.88
68	20
70	21.11
72	22.22
74	23.33
76	24.44
78	25.55

80	26.66
82	27.77
84	28.88
86	30
88	31.11
90	32.22
92	33.33
94	34.44
96	35.55
98	36.66
100	37.77

人 口 的 增 长

世界人口持续不断的增长将带来一系列严重的问题。采取有效措施控制人口的增长，已成为各国政府和人民迫切需要考虑的一个问题。



有人曾对南美的人口增长情况作过预测，因为它是世界上人口增长最快的地区之一。1976年南美的人口是2.12亿，当时它的增长速度是2.9%，假定这个增长速度保持不变，那么到21世纪

它的人口将比1976年多一倍。

下面是用电脑计算南美人口增长的流程图和程序：

10 $T=0$

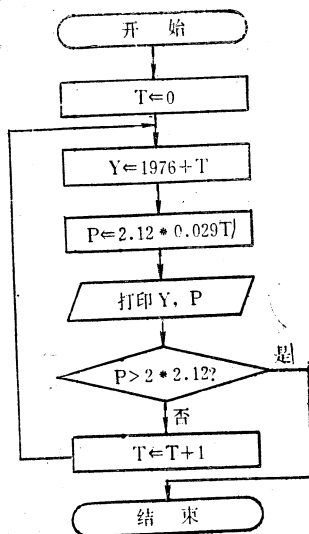
```

20 Y=1976+T
30 P=2.12*(1.029^T)
40 PRINT Y, P
50 IF P>2*2.12
    THEN 80
60 T=T+1
70 GOTO 20
80 END

```

JRUN

1976	2.12
1977	2.18148
1978	2.24474292
1979	2.30984047
1980	2.37682584
1981	2.44575379
1982	2.51668065
1983	2.58966439
1984	2.66476465
1985	2.74204283
1986	2.82156207
1987	2.90338737
1988	2.98758561
1989	3.07422559
1990	3.16337813
1991	3.2551161
1992	3.34951446
1993	3.44665038



1994	3.54660325
1995	3.64945474
1996	3.75528892
1997	3.8641923
1998	3.97625388
1999	4.09156525
2000	4.21022063
2001	4.33231703

求最大公因数

求360、810、1260、3580的最大公因数。

用 BASIC 语言编写的程序来求上述四数的最大公因数的方法，我们可以这样来设计：



(1) 将这四个数放在数组 A 中，并将 D 设为 1。

(2) 用 $B = 2$ 遍除 A 数组，若都能被 B 除尽，就将 D 设为 $1 \cdot 2$ ，且设 $B = 2^2 = 4$ 。再用 $B = 2^2$ 遍除 A 数组，若又能够被除尽，则设 D 为 $1 \cdot 2 \cdot 2$ ，再设 $B = 2^3 = 8$ ，若不能被除尽，就执行下一步。

(3) 用 $B = 3$ 遍除 A 数组，若都能被除尽，就将 D 设为 $D \times B$ ，且设 $B = 3^2$ ，再用新设的 B 值 9 遍除 A 数组，若能除尽，则再设 D 为 $D \times 3^2$ ，若不能被除尽，则执行下一步。

(4) 用 $B = 4$ 遍除 A 数组……一直重复进行，直到 B

=360 (因为360是上述四个数组中最小的一个数)。

```
10 DIM A(4)
20 LET D=1
30 FOR I=1 TO 4
40 READ A(I)
50 NEXT I
60 FOR B=2 TO 360
70 FOR I=1 TO 4
80 LET M=A(I)/B
90 IF M=INT (M) THEN 110
100 GOTO 150
110 NEXT I
120 LET D=D*B
130 LET B=B*B
140 GOTO 70
150 NEXT B
200 PRINT "( ", A(1), ", ", A(2), ", ", A(3),
    ", ", A(4), ")="; D
220 END
230 DATA 360, 810, 1260, 3580
```

]RUN

(360, 810, 1260, 3580)=10

求多个自然数的最大公约数

在数学中，常常需要求出多个自然数的最大公约数。遇到这类问题，往往要先将这些自然数一个个分别分解因数，然后加以观察，求出这许多数的最大公约数，假如所要求的



自然数个数较多，并且都较大，那么，求起来还挺费事。为此，我们请电脑来帮帮忙。下述程序能很快地求出任意多个自然数的最大公约数。程序中采用了欧几里德法（即辗转相除法），因此，算起来很快。

程序运行时，电脑先询问操作者，您需要求多少（N）个自然数的最大公约数。然后，操作者必须逐个将这N个自然数输入电脑，输入完毕后，电脑就开始“寻找”这N个自然数的最大公约数了。

```
10 INPUT N; DIM A(N)
20 B=1E+30
30 FOR I=1 TO N
40 PRINT "NO. ", I; , INPUT A(I); IF A(I)
   <B THEN
50 NEXT I
60 A(1)=A(2); A(2)=B
70 FOR I=1 TO N-1
80 P=A(I)-INT (A(I)/A(I+1))*A(I+1)
90 IF P<>0 THEN A(I)=A(I+1);
   A(I+1)=P; GOTO 80
100 NEXT I
110 PRINT A(N)
120 END
```

JRUN

74

NO.1?185

NO.2?640

NO.3?895

NO.4?1105

5

JRUN

73

NO.1?63

NO.2?252

NO.3?1008

63

整数的因子分解

试写出一个程序，能对已知的整数进行因子分解。

我们可以将待分解的整数通过 INPUT 语句输入给变量 N。N 先被 2 除，每除尽一次，打印一个 “ $\times 2$ ”，直到不能再被 2 除尽。然后，再把所得的商被 3 除，每除尽一次，打印一个 “ $\times 3$ ”，直到不能再被 3 除尽。再将所得的商被 4 除……，直到最后被 $\sqrt{N}$ 的整数部分除，

如果不能除尽，就转去打印 “ $1 \times N$ ”。



根据上述思想，编制出用 BASIC 语言写的“整数的因子分解”程序如下：

```
10 PRINT "INPUT NUMBER N=";  
20 INPUT N  
25 PRINT  
30 IF N=0 THEN 230  
40 IF SGN(N)<>-1 THEN 60  
50 PRINT N; "=-1"  
55 GOTO 70  
60 PRINT N; "=1",  
70 N=ABS(N)  
80 FOR I=2 TO SQR(N)  
100 IF N/I<>INT(N/I) THEN 150  
110 N=N/I  
130 PRINT "×", I;  
140 GOTO 100  
150 NEXT I  
160 IF N=1 THEN 190  
180 PRINT "×", N  
190 PRINT  
200 GOTO 10  
230 END
```

JRUN

INPUT NUMBER N=?253

253=1×11×23

INPUT NUMBER N=?1944

1944=1×2×2×2×3×3×3×3

INPUT NUMBER N=?23205

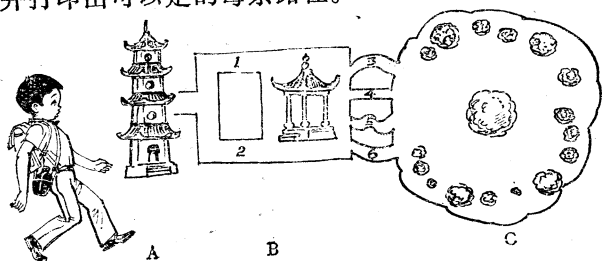
$23205=1 \times 3 \times 5 \times 7 \times 13 \times 17$

INPUT NUMBER N=?7687549

$7687549=1 \times 367 \times 20947$

到花园的路有几条

如图所示, 小明在公园里游玩, 想从古塔出发, 穿过亭子到达花园, 试编一个程序找出从古塔到花园共有多少条路, 并打印出可以走的每条路径。



下面, 就是寻找去花园的路径的流程图和程序。

```
10 FOR I=1 TO 2
20 FOR J=3 TO 6
30 PRINT "A-", I, "-B-", J, "-C"
40 NEXT J, I
50 END
```

[RUN

A-1-B-3-C

A-1-B-4-C

A-1-B-5-C

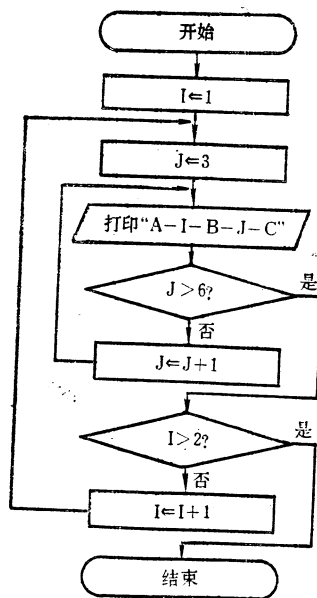
A-1-B-6-C

A-2-B-3-C

A-2-B-4-C

A-2-B-5-C

A-2-B-6-C

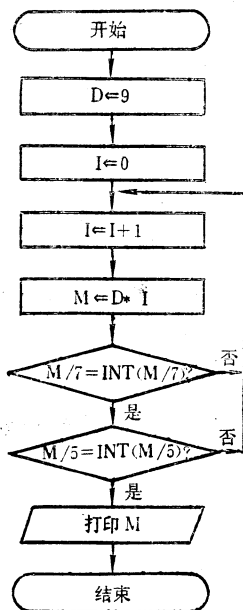


何 日 再 相 会

李明、王真和沈特在中学里是最要好的同学，毕业后分赴不同的岗位，很难有机会见面。李明在某次特快列车上当乘务员，随列车 5 天往返一趟；王真在大学里念书，一星期休息一天；沈特在近海客轮上当水手，9 天才回港口一次。

他们三人很想在一起叙叙，问至少需要多少天他们才能相逢一次？

这个问题实际上是个求最小公倍数问题，用微电脑来运算非常简便，下面是流程图和程序：



```

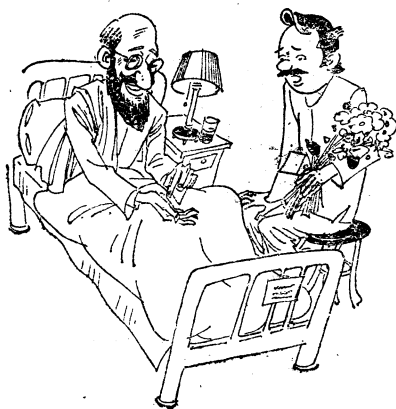
5 D=9
10 I=0
20 I=I+1
30 M=I×D
40 IF M/7<>INT
    (M/7) THEN 20
50 IF M/5<>INT
    (M/5) THEN 20
60 PRINT M
70 END
  
```

```

]RUN
315
  
```

乏味的车号

印度数学家拉曼纽扬在医院养病。他的朋友哈蒂来看望他，对他说：“我是坐出租汽车来的，那辆车的车号真乏味。”



拉曼纽扬问：“是多少？”

哈蒂回答：“1729。”

拉曼纽扬立刻微笑着说：“朋友，你错了，1729是个很有趣的数哩！你看，它可以写成 $1000+729$ ，而1000是10的立方，729又正好是9的立方。”

停了一会，他又说：“还不仅如此，1729又可以写成 $1728+1$ ，而1728是12的立方，1是1的立方。所以说，它是个可以用两种方法写成两个立方数之和的数。”

哈蒂说：“我想肯定还有别的自然数，也能用两种方法写成两个立方数之和的。”

拉曼纽扬点点头：“是的，有许多。不过这样的数当中，1729是最小的。”

这个故事提出了一个论断：1729是最小的能用两种方法写成两个自然数立方之和的数。我们可以编个程序，让电脑来证实这一点。

程序采用从自然数 1 开始，逐个试验的方式，凡能写成两个立方数之和的，就在所设置的“计数器” L 中加 1，如果 L 加到 2，就说明该数能用两种方法写成两立方数之和的形式，电脑随即打印出结论并停止运行。

程序运行的结果，证明了 1729 确实是最小的能用两种方法写成两个立方数之和形式的自然数。

```
10 FOR N=1 TO 1730
20 L=0
30 S=INT [N^(1/3)]
40 FOR A=1 TO S
50 FOR B=A TO S
60 IF A*A*A+B*B*B<N THEN 100
70 IF L=0 THEN A1=A, B1=B
80 L=L+1
90 IF L=2 THEN 120
100 NEXT B, NEXT A
110 NEXT N
120 PRINT "(", L-1, ")"; N; "=", A1, "^3+",
      B1, "^3"
130 PRINT "(", L, ")"; N; "=", A, "^3+", B,
      "^3"
140 END
```

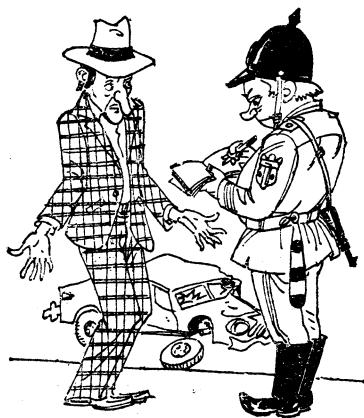
JRUN

(1) $1729=1^3+12^3$

(2) $1729=9^3+10^3$

逃 不 掉

某地发生了一起撞车事件。事件发生后，驾驶员驾车逃跑了。交通民警在调查此案时，询问了三名现场目击者。但遗憾的是，他们三人谁也记不清肇事车辆的车号了。第一个



目击者只记得有 4 位数字的车号的头两位数字是一样的；第二个目击者记得这个车号的后两位数字是相同的；第三个目击者只记得车号是某个自然数的平方。看来有关此案的问题还比较复杂。然而，一个精于数学的交通民警却根

据三位目击者所提供的情况精确地算出了肇事车辆的车牌号码。你能知道这个车牌号码是多少吗？

我们让微型电脑来推算。

假设 X 为车牌号码。因为 X 是一个四位数，要做到 $Y = \sqrt{X}$ ，则 Y 的取值范围必定是 $32 \sim 99$ 。我们可以利用循环语句，直接找到满足条件的 X 。

以下是流程图和程序：

```
5 FOR Y=32 TO 99
10 X=Y*Y
15 X1=INT(X/1000)
```



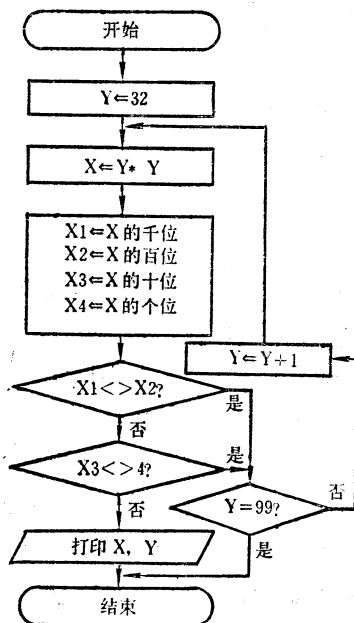
```

20  X2=INT((X-X1*1000)/100)
25  X3=INT((X-X1*1000-X2*100)/10)
30  X4=X-X1*1000-X2*100-X3*10
35  IF X1<>X2 THEN 50
40  IF X3<>X4 THEN 50
45  PRINT X
50  NEXT Y
55  END

```

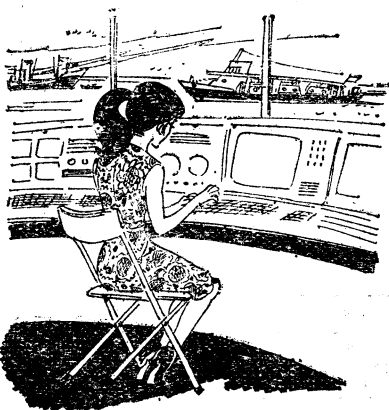
JRUN

7744



黄浦江上的渡口

上海黄浦江的某渡口，车辆来来往往，非常繁忙。尽管



汽车轮渡不停地往返接运，但有时岸边还是排

起了长长的汽车队伍。

为了解决这一问题，轮渡公司必须首先弄清通过该渡口的汽车流量规律和造成排队的原因。

虽然关于这个问题很难建立一个确定的数学模型，但我们可以利用随机数来模拟这一过程。

如以 5 分钟为一周期，在这期间，有 1 ~ 4 辆汽车开到渡口并等待摆渡过江，而汽车轮渡开一次只能载 3 辆车，假定轮渡每天开放 8 小时，试用微电脑来模拟这一过程并求出江边排队汽车的平均数目。

程序中利用随机整数表示每个周期内随机到达的车辆数，以渡口汽车是否满 3 辆作为排队的条件，并累加各周期排队的车辆数，将它除以周期数即为平均队伍内的汽车数。程序中 L 和 T 分别表示一个周期内队伍长度和累加的队伍长度。现模拟 5 天的情况，并用二重循环的形式，内循环为 8 小时，共 96 个周期，外循环为 5 天。运行结果表明了各天队伍的长度。

```

10 PRINT "DAY",
    "AVE. QUEUE
    LENGTH"
20 FOR D=1 TO 5
30 L=0
40 T=0
50 FOR C=1 TO 96
60 N=INT(4*
    RND(1))+1
70 L=L+N
80 IF L>3 THEN
    110
90 L=0
100 GOTO 130
110 L=L-3
120 T=T+L
130 NEXT C
140 A=T/96
150 PRINT D, A
160 NEXT D
170 END

```

IRUN

```

DAY   AVE.QUEUE
      LENGTH

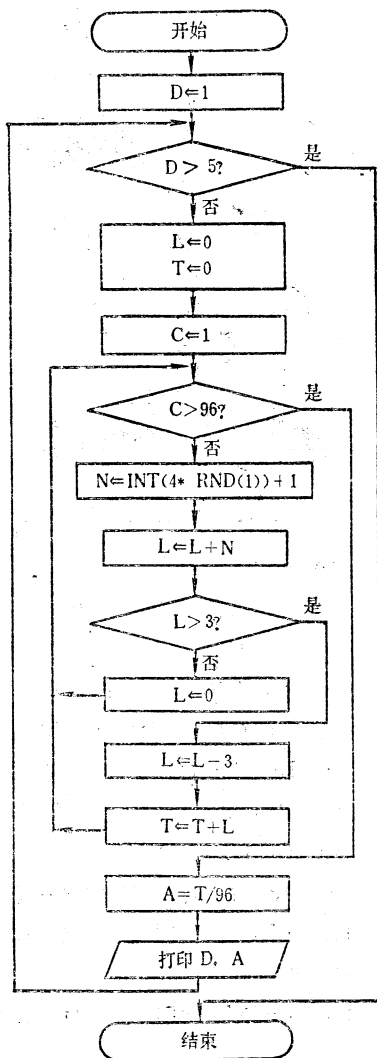
```

```

1      1.08333333
2      .364583333
3      .427083333
4      .916666667
5      .479166667

```

以下是流程图和程序：



猜 数



下列程序要求两位
游戏者各自在 1~100
范围内猜一个数，然
后由微电脑随机给出
一个数，看谁猜的数
与微电脑给出的数接
近。

```
10 PRINT "PLAYER1--GUESS IS";
20 INPUT P1
30 PRINT "PLAYER2--GUESS IS";
40 INPUT P2
50 C= INT (100*RND (1))
60 PRINT "COMPUTER SELECTED", C
70 IF ABS(C-P1)<>ABS(C-P2) THEN 100
80 PRINT "BOTH PLAYERS WERE EQUA L"
90 GOTO 140
100 IF ABS(C-P1)<ABS(C-P2) THEN 130
110 PRINT "PLAYER2 WAS CLOSEST"
120 GOTO 140
130 PRINT "PLAYER1 WAS CLOSEST"
140 END
```

]RUN

PLAYER1-GUESS IS? 3

PLAYER2-GUESS IS? 47

COMPUTER SELECTED10

PLAYER1 WAS CLOSEST

四个整数

你能寻找出四个10
以内的整数W、X、Y和
Z,使它们满足: $W^3 +$
 $X^3 + Y^3 = Z^3$ 吗?

以下是为寻找这四
个整数而设计的流程图
和用 BASIC 语言编写
的程序。



```
10 FOR W=1 TO 9
20 FOR X=1 TO 9
30 FOR Y=1 TO 9
40 FOR Z=1 TO 9
50 IF W*W*W+X*X*X+Y*Y*Y=Z*Z*Z
   THEN 110
60 NEXT Z
70 NEXT Y
80 NEXT X
90 NEXT W
```

```

100 GOTO 150
110 PRINT "W="; W
120 PRINT "X="; X
130 PRINT "Y="; Y
140 PRINT "Z="; Z
150 END

```

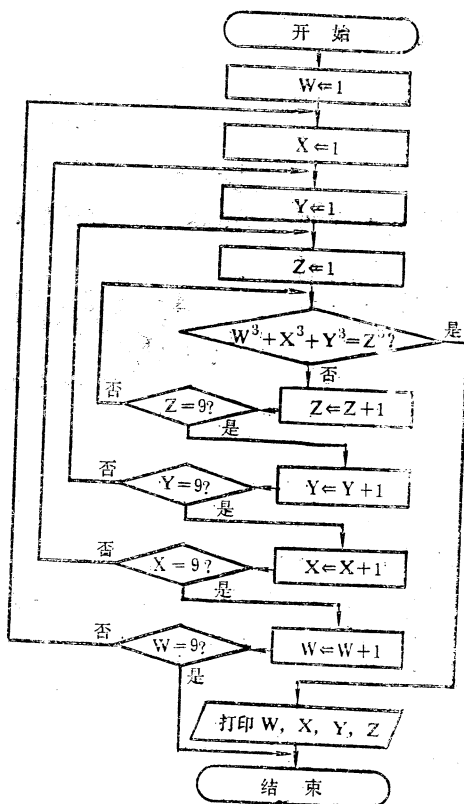
JRUN

W=1

X=6

Y=8

Z=9



十 个 数

有十个数，已知前五个数，而不知道后五个数是什么，但后五个数中每一个数均为它前面五个数的和。请用微机“思考”后，打印出这十个数。



```
10 INPUT A, B, C, D, E
20 PRINT A, B, C, D, E,
30 FOR I=6 TO 10
40 S=A+B+C+D+E
50 PRINT S,
60 A=B
70 B=C
80 C=D
90 D=E
100 E=S
110 NEXT I
120 END
```

1 RUN

72

??3

??7

??9

??1

2

3

7

9

1

22

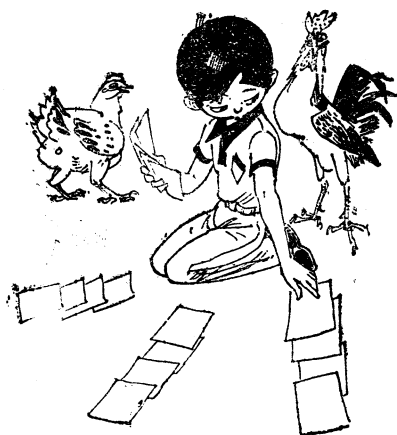
42

81

155

301

巧排数组



试将 1、2、3、

4、5 五个数字排成若干个数组，这些数组都是由上述五个数字中任取三个数字构成，但必须满足这样的条件：

(1) 三个数字互相不能重复；(2) 首尾两个数字之和等于中间的数字。我们请电脑将这些数组一一打印出来。

根据题意，我们知道，这些数组的首位与末尾数字都不可能为 5，为此，我们可设置二重循环：外循环变量为 I，从 1 至 4；内循环变量为 J，从 I+1 至 5。这样，所求的数组是：I、J、J-I。然后进行判断，假如 $J-I=I$ ，说明首尾数字相同，不符合条件 (1)，就再换一个数字。

下面是用 APPL E-II 微电脑巧排数组的流程图和程序。

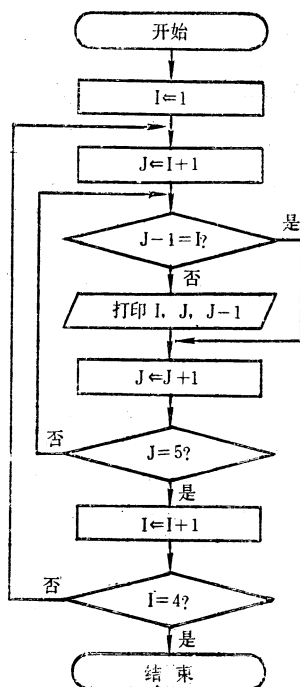
```

10 FOR I=1 TO 4
20 FOR J=I+1 TO 5
30 IF J-I=I THEN 50
40 PRINT I, J, J-I
50 NEXT J
60 NEXT I
70 END

```

]RUN

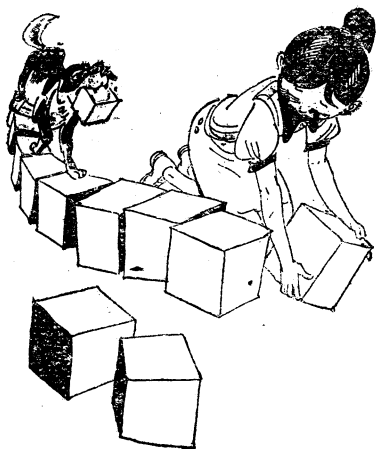
1	3	2
1	4	3
1	5	4
2	3	1
2	5	3
3	4	1
3	5	2
4	5	1



移 位

有一个自然数，它的头一位数字是 6，如果把这个 6 移到这个数的最后一位，那么新组成的数就是原来数值的 $1/4$ 。试问这个自然数是多少？

这个问题可先作如下分析：首位数字为 6 的自然数如果用代数式来表示的话，可以写成这样的形式： $6 \times 10^n + n$ ，将



6 移到末位，那么 该数数值就变成 $10n+6$ 。根据题意，即

$$\frac{1}{4}(6 \times 10^x + n) = 10n + 6,$$

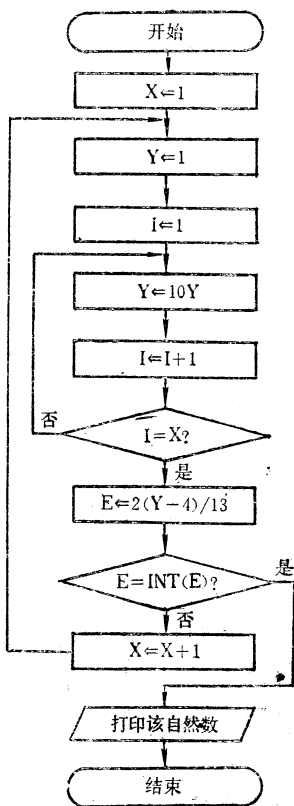
因而
$$n = \frac{2(10^x - 4)}{13}$$

这里， n 不可能是小数或分数，于是可将 1、2、3 等逐一代入 x ，找出适合的数字，根据这个数字求出 n 的数值，最后得出该自然数。

10 $X=1$

15 $Y=1$

20 FOR $I=1$ TO X



```

30 Y=10*Y
40 NEXT I
50 E=(Y-4)*2/13
60 IF INT(E)=E THEN PRINT 6*Y+E;
70 X=X+1; GOTO 15
80 END

```

JRUN

615384

N 阶 乘

从 1 到 n 间所有正整数的连乘称作 n 阶乘, 以 $n!$ 表示。
于是,

$$1! = 1$$

$$2! = 1 \times 2 = 2$$

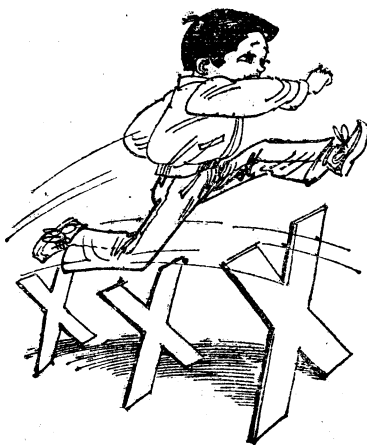
$$3! = 1 \times 2 \times 3 \\ = 6$$

$$4! = 1 \times 2 \times 3 \\ \times 4 = 24$$

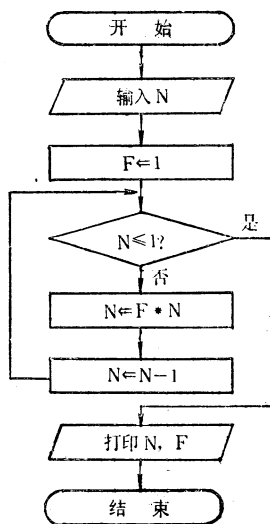
$$5! = 1 \times 2 \times 3 \\ \times 4 \times 5 = 120$$

以这种方式继续运
算, 我们能制成一张 $n!$

的表。但这种运算相当麻烦, 因为随着阶乘数的增大, 其积以难以想象的速率增大。仅以 $26!$ 来说, 其积将大于 4×10^{26} 。可是, 这件事让电脑来做, 却是既迅速又准确。下面是计算



$n!$ 的流程图和 BASIC 程序:



```

10 PRINT "TYPE N",
20 INPUT N
30 PRINT N,
40 LET F=1
50 IF N<=1 THEN 90
60 LET F=F*N
70 LET N=N-1
80 GOTO 50
90 PRINT "FACTORIAL=";
  F
100 END
  
```

```

]RUN
TYPE N?9
9FACTORIAL=362880
(FACTORIAL: 阶乘、阶乘积)
  
```

双 胎 胎 数 7744

一个四位数，它的千位数字和百位数字相同，十位数字和个位数字相同，并且这个自然数本身还是一个完全平方数。这样的数我们称它为双胞胎数。

要在1000到9999之中去搜寻这样的数，确实不容易，我们就请“聪明”的电脑来寻找双胞胎数吧！

为了让电脑快速搜寻出双胞胎数，我们可抓住双胞胎数是个完全平方数的特点，就只需从32到99找一个循环就行

了，不必从1000到9999
进行搜寻。

下述程序就是为寻
找双胞胎数而编写的。
搜寻结果，在四位自然
数的范围内只发现了一个
双胞胎数，那就是
7744。

程序中第30~70句
的功能是对这个平方数
的各位数字进行分离。

A(1)~A(4)分别代表个位数字到千位数字。

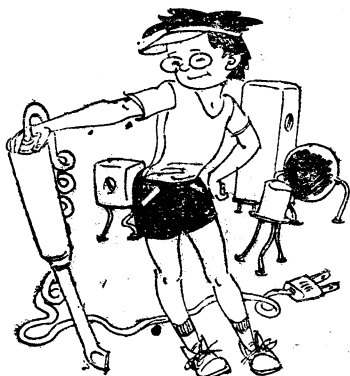
```
10 FOR I=32 TO 93
20 M=I*I
30 T=1
40 FOR J=1 TO 4
50 T=T*10
60 A(J)=INT(M*10/T)-INT(M/T)*10
70 NEXT J
80 IF A(1)<>A(2) OR A(3)<>A(4)
   THEN 100
90 PRINT M; " = "; I; " ^ 2"
100 NEXT I
110 END
```

JRUN

7744=88^2



电阻·电容·电压·时间



在RC串联电路上, 电容充电电压与时间 t 的关系为:

$$u(t) = u_0(1 - e^{-t/RC})$$

式中: $R=100\text{k}\Omega$, $C=1\mu\text{f}$, u_0 为直流电源电压(100V)。

我们只要编制一个程序, 利用微电脑可以很快地得出我们所需要的

的参数。

```
10 INPUT T
20 PRINT
30 READ R, C
40 U=100*(1-EXP(-T/(R*C)))
50 PRINT "R="; R, "C="; C, "U(", T, ")=";
60 DATA 100000, 0.000001
70 END
```

]RUN

?1.2

R=100000 C=1E-06 U(1.2)=99.9993856

电 阻 并 联

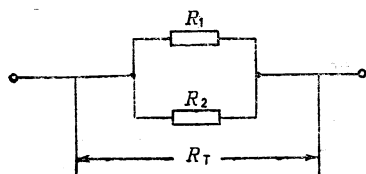
青少年无线电爱好者们在进行实验时，经常需要用到一些数值较特殊的电阻，如果手头找不到合适阻值的电阻，常用的方法是通过几个电阻串、并联来解决。



电阻串联的计算方法很简单，但并联就较复杂了。如下图所示的电路中， R_1 与 R_2 并联时，求总阻值 R_T 的计算公式如下：

$$R_T = \frac{R_1 \times R_2}{R_1 + R_2}$$

然而，在实际应用中，往往又不是已知 R_1 和 R_2 求 R_T ，却常常是已知电路上需要的 R_T ，和随手拿起的电阻 R_2 ，要求



出另一电阻 R_1 ，使之与 R_2 搭配得到需要的阻值 R_T 。这时，计算公式就变为

$$R_1 = \frac{R_T \times R_2}{R_2 - R_T}$$

下面，我们将上两式编进一个BASIC程序中，依靠微电脑，就能方便地求出所需的电阻值了。

```
5 REM---PARALLEL RESISTOR FOR AP
```

```
II--- --
```

```

10 HOME
15 PRINT "TOTAL RESISTOR (1) OR BRANCH
   (2)? "
17 GET A
18 HOME
20 POKE 34, 13
90 IF A=1 THEN 100
95 IF A=2 THEN 105
100 PRINT "                                R1"
102 GOTO 110
105 PRINT "                                R1=2 "
110 PRINT "          ---MMMMM---"
120 PRINT "          I                      I "
130 PRINT "          I                      I "
140 PRINT "----- I                      I-----"
150 PRINT "  I      I                      I  I"
160 PRINT "  I      I                      I  I"
170 PRINT "  I      ---MMMMM--- I"
180 PRINT "  I                      R2          I"
190 PRINT "  I                      I"
192 IF A=1 THEN 200
194 IF A=2 THEN 205
200 PRINT "  I----- RT=? -----I"
202 GOTO 250
205 PRINT "  I----- RT -----I"
250 PRINT : PRINT
280 IF A=1 THEN GOSUB 1000
290 IF A=2 THEN GOSUB 2000

```


BRANCH R1=153.818182

TOTAL RESISTOR (1) OR BRANCH (2)?

Diagram illustrating a rectangular layout with internal channels and labels:

- Top horizontal channel: MMMMMM
- Bottom horizontal channel: MMMMMM, R2
- Right side label: RT=?
- Left side label: I, I

TOTAL RESISTOR RT=6.66666667

程序中, 15句及90、95句通过设定变数A来选取计算并联电路的总电阻或分支电路的电阻值。100~205句是使显示器上出现并联电路图形, 电阻用“M”字符组成。1000、1100、2000、2100句是请你输入已知的电阻值。1200及2300句就是上述的两个计算公式。1300及2400句打印结果。1350及2480句是使程序构成回路, 使计算得以反复进行, 这对于在使用“凑”的方法寻找合适电阻时尤其有用。

2200句是一个判别语句,为防止当 R_T 大于或等于 R_2 时,出现电阻为负值或分母为零的情形。

程序运行结果,前者是计算分支电路电阻 R_1 的情形;后者是计算总阻值 R_T 的情形。

求串并联网络的等效电阻

电阻串、并联网络如图所示。如果共有10个 R_2 , 10个 R_1 , 并且 $R_1=200\Omega$, $R_2=300\Omega$, $R_0=100\Omega$, 试求A、B两端的等效电阻。

我们可以采用从右向左逐级递推的方法, 将整个网络用虚线分成若干级。第1级中串联电阻值为 R_1+R_0 , 将该阻值送到变量 R 中 (此时 $R=R_1+R_0$), 再将 R 与 R_2

并联后送 R 中, 即 $\frac{R \times R_2}{R + R_2}$

$\Rightarrow R$, (注意: 这里赋值号左边的 $R=R_1+R_0$, 赋值号

右边的 R 是个新值, 即 (R_1+R_0) 与 R_2 并联之值), 这就构成了第1级等效电阻。

重复上进步骤, 将第1级等效电阻 R 与 R_1 串联后再与 R_2 并联, 求出第2级等效电阻……如此下去, 便可一级一级地求出A、B端的等效电阻了。

流程图和程序如下:

```
5 INPUT R0, R1, R2
10 PRINT
15 PRINT "N", "R"
20 LET R=R0
30 FOR N=1 TO 10
```



```

40 LET R=R+R1
50 LET R=(R*R2)/(R+R2)
60 NEXT N
70 PRINT N, R
80 END

```

]RUN

? 100

? ? 200

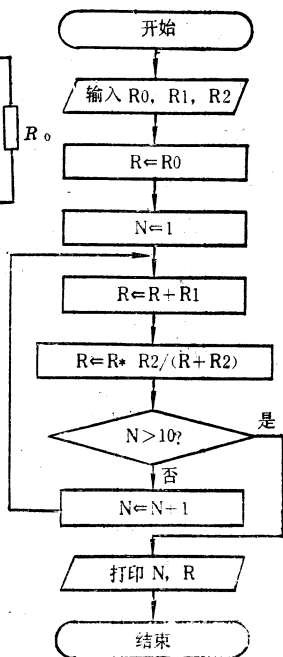
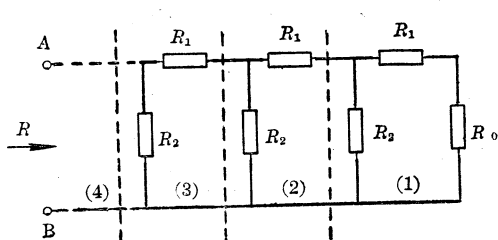
? ? 300

N

R

10

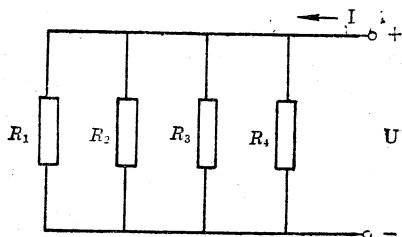
164.575122



求多电阻并联电路中的电流

图中是一个由 4 个电阻并联的电路，4 个电阻的阻值分别是：
 $R_1=100\Omega$, $R_2=150\Omega$,
 $R_3=240\Omega$, $R_4=360\Omega$ 。
若给电路两端加 24 V 的电压，试求电路中的电流。

以下是解这个问题的 BASIC 程序：



```
5 PRINT "R1, R2, R3, R4, U="
10 INPUT R1, R2, R3, R4, U
20 LET I=(1/R1+1/R2+1/R3+1/R4)*U
30 PRINT
40 PRINT "I=", I
50 END
```

JRUN

R1, R2, R3, R4, U=

? 100

? ? 150

? ? 240

? ? 360

? ? 24

I=.566666667

一传十 十传百



人们经常为一个奇闻的迅速传播而感到惊奇。现在假设，某城市的一个市民在半小时内把一则消息传给 3 个人，听到此消息的人在半小时之内又各自传给另外 3 个人，设每人只传播一次消息，到此为止，已经有 13 个市民知道这则消息了。

依此推算下去，要多长时间，这则消息可以使一个中等城市的 6000000 居民人人皆知？

在分析这个问题前，我们可先列一张表：

时间间隔 (小时)	得 知 消 息 人 数
$0 = 0 \times 0.5$	1
$0.5 = 1 \times 0.5$	$1 + 3$
$1 = 2 \times 0.5$	$1 + 3 + 3^2$
$1.5 = 3 \times 0.5$	$1 + 3 + 3^2 + 3^3$
$\vdots$	$\vdots$
$x = n \times 0.5$	$1 + 3 + 3^2 + 3^3 + \dots + 3^n$

从表中可以看出,得知消息的人数是以带有公比 $Q=3$ 的几何级数递增的。一共应该有6000000个人,得知消息,即:

$$1 + 3 + 3^2 + 3^3 + \dots + 3^n = 6000000$$

运用几何级数的计算公式:

$$1 + Q + Q^2 + \dots + Q^n = \frac{Q^{n+1} - 1}{Q - 1} = M$$

$$\text{所以 } n = \frac{\log(MQ - M + 1)}{\log Q} - 1$$

用上述数字代入的话,得到:

$$\frac{3^{n+1} - 1}{3 - 1} = 6000000$$

$$\text{即 } 3^{n+1} = 12000001$$

$$n \approx 13.838$$

$$13.838 \times 0.5 \approx 6.9 \text{ (小时)}$$

这就是说,从理论上计算表明,不到7个小时该城市所有居民都知道了这个奇闻。

下面我们用电脑来验证一下。

```

10 INPUT "I, Q, M="; I, Q, M
20 N=LOG(M*Q-M+1)/LOG(Q)-1
30 PRINT N*I
40 END

```

]RUN

I, Q, M=.5, 3, 6000000

6.91863961

百鸡问题



我国古代著名数学家张丘建在他编著的《算经》里提出这样一个问题：

“鸡翁一，值钱五，鸡母一，值钱三，鸡雏三，值钱一，百钱买百鸡。鸡翁、母、雏各几何”。

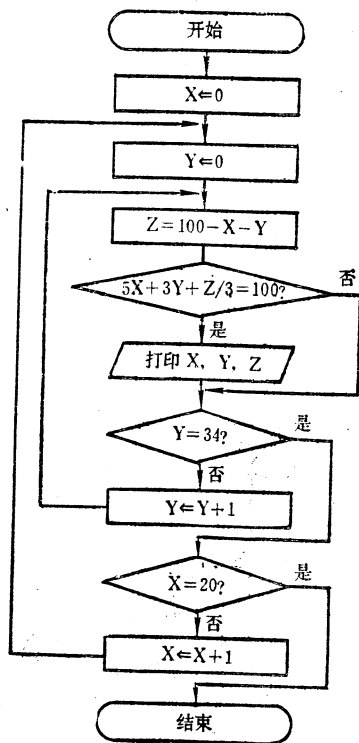
根据题意，我们假设公鸡为 x 只，母鸡为 y 只，小鸡为 z 只，则：

$$\begin{cases} x + y + z = 100 \end{cases} \quad (1)$$

$$\begin{cases} 5x + 3y + \frac{1}{3}z = 100 \end{cases} \quad (2)$$

显然，这是一个不定方程组。由于“鸡翁一，值钱

五”，因此，如果用百元钱买公鸡，也至多不超过20只。所以， x 不能大于20。同样道理，由于“鸡母一，值钱三”，所以百元钱全买母鸡也至多不超过34只。因此， y 不能大于34。解这个不定方程，也可以用循环语句。下面为流程图和程序：



```

10 FOR X=0 TO 20
20 FOR Y=0 TO 34
30 LET Z=100-X-Y
40 IF 5*X+3*Y+Z/3=100 THEN 60
50 GOTO 70
60 PRINT "X="; X, "Y="; Y, "Z="; Z
70 NEXT Y
80 NEXT X
90 END
  
```

JRUN

X=0

Y=25

Z=75

X=4

Y=18

Z=78

X=8

Y=11

Z=81

X=12

Y=4

Z=84

翻 两 番



某企业的产值每年平均增长 $C\%$ ，若今年产值为 P ，求当 C 分别为 6、8、10、12 时经过几年产值可以翻两番？

根据题意，如果每年平均增长 $C\%$ ，那么一年后为 $P \times (1 + C\%)$ 。所谓翻两番，也就是要求经过 n 年后，满足： $P \times (1 + C\%)^n =$

$4P_1$ 。这样，我们就可以设计如下的流程图和程序：

```
10 REM NUMBER OF YEARS TO FOORFOLD  
   THE PRODUCE  
20 PRINT "P=";  
30 INPUT P  
40 READ C  
50 IF C=-1 THEN 160
```

```

60 LET P1=P
70 LET Y=0
80 LET P1=P1*(1+C/100)
90 LET Y=Y+1
100 IF P1>=4*P THEN 130
110 GOTO 80
120 PRINT
130 PRINT "C=", C, "Y=", Y, "P1=", P1
140 GOTO 40
150 DATA 6, 8, 10, 12,
-1
160 END

```

JRUN

P=? 2

C=6 Y=24

P1=8.09786929

C=8 Y=19

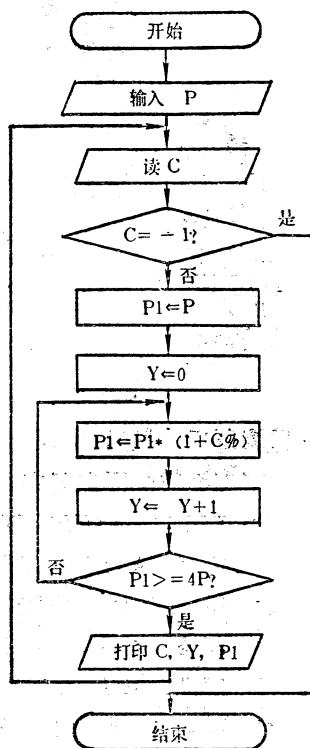
P1=8.63140212

C=10 Y=15

P1=8.35449634

C=12 Y=13

P1=8.72698623



算 术 测 验

(BASIC 语言)

这是一个专为小朋友写的程序。根据这个程序，微电脑能自动地给出一些两位数相加的算题，供小朋友练习，并对答案进行判别。如果答对了，电脑将通过显示屏显示出“RIGHT”（对）、“NEXT PLEASE”（请再答一题），



接着就给出一道新的算题。如果答错了，则显示“WRONG”（错）、“TRY AGAIN PLEASE”（请再试一下），并再次显示出原来的题目。连续答对十道算题，便显示出“VERY GOOD”（很好）；以资鼓励；如果中间曾答错过，则答完

十道算题后，显示“OK”，测验结束。

下面是流程图和程序：

```
5 PRINT "THIS PROGRAM GIVES A MATHEMATICAL TEST"
10 IF I > 10 THEN 62
15 LET A = INT(RND(1) * 90) + 10
20 LET B = INT(RND(1) * 90) + 10
25 PRINT A; "+"; B; "=",
30 INPUT C
```

```

35 IF A+B=C THEN 53
37 LET K=-1
40 PRINT "WRONG", "TRY AGAIN PLEASE"
45 GOTO 25
50 PRINT "RIGHT", "NEXT PLEASE"
55 LET I=I+1
60 GOTO 10
62 IF K<0 THEN
    PRINT "OK":
    GOTO 70
65 PRINT "VERY GOOD"
70 END

```

]RUN
 THIS PROGRAM
 GIVES A MATHE-
 MATICAL TEST

92+53=? 145

RIGHT

NEXT PLEASE

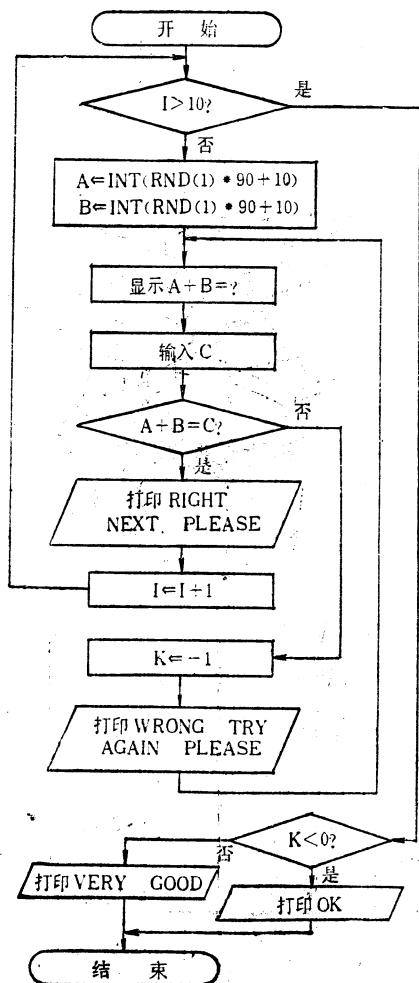
95+26=? 9

WRONG

TRY AGAIN

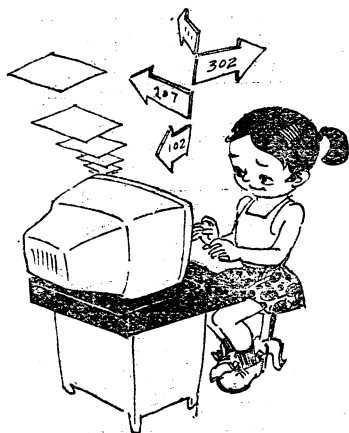
PLEASE

95+26=?



请读者将这个程序稍加修改，使电脑能进行三位数或更多位数的加法测验，或能进行减法、乘法和除法等运算的测验。

考场指南



某次竞赛考试凭准考证入场，请设计一个程序，在RUN命令执行以后，用INPUT语句输入下列表格数据，并用查表法找寻五个准考证号码（号码放在DATA语句中）应在的教室号码，打印输出准考证号码及所对应的教室号码。

准考证号码	2101~2147	1741~1802	0201~0287
教室号码	101	103	114
准考证号码	3333~3397	1803~1829	2511~2576
教室号码	209	305	306

```
10 DIM L(6), U(6), R(6)
```

```
20 FOR I=1 TO 6
```

```

30 INPUT L(I), U(I), R(I)
40 NEXT I
50 FOR I=1 TO 5
60 READ N
70 FOR J=1 TO 6
80 IF N <=U(J) AND N>=L(J) THEN 110
90 NEXT J
100 PRINT "NOT FOUND"; GOTO 120
110 PRINT N, " - - ", R(J)
120 NEXT J
130 END
200 DATA 2130, 1741, 255, 3339, 1800

```

]RUN

? 2101, 2147, 101

? 1741, 1802, 103

? 201, 287, 114

?3333, 3387, 209

?1803, 1829, 305

?2511, 2576, 306

2130 - - 101

1741 - - 103

255 - - 114

3339 - - 209

1800 - - 103

统计考试成绩

某班第1小组期中考试共考查三门功课，其中有两门为主课，现要求用微电脑挑选出符合下列条件之一的学生：



(1) 三门总分高于270分者；

(2) 两门主课在95分以上者；

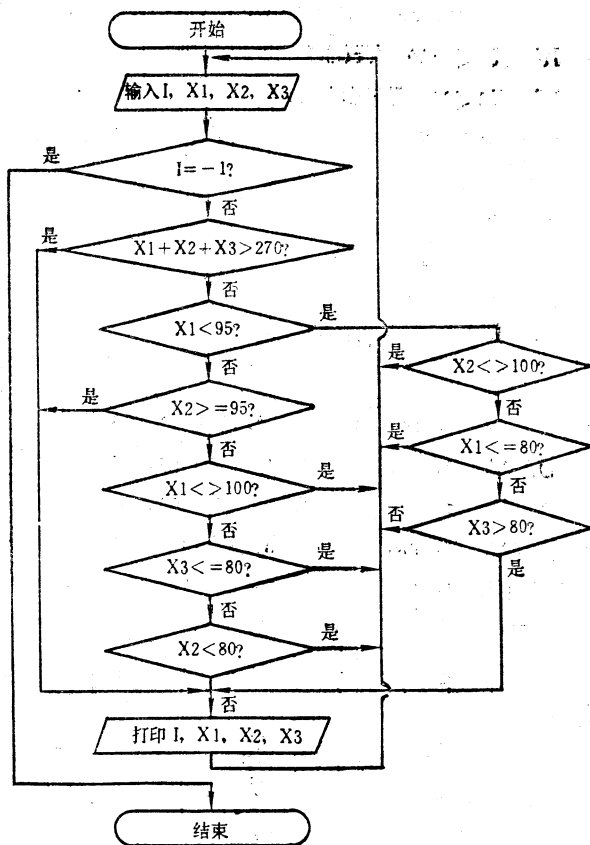
(3) 一门主课为100分，其他两门在80分以上者。

要求打印出这些学生的学号和各门功课的成绩。

该小组9名学生的学号和成绩分别如下：

学号(I)	语文(X 1)	数学(X 2)	外语(X 3)
1	100	96	75
2	91	92	90
3	90	81	85
4	100	85	76
5	95	96	74
6	81	100	83
7	98	80	75
8	95	93	80
9	90	95	93

以下是流程图和程序：



```

10 PRINT TAB (1); "I"; TAB (11); "X1"; TAB
   (21); "X2"; TAB(31); "X3"
30 READ I, X1, X2, X3
35 IF I=-1 THEN 210
40 IF X1+X2+X3 >270 THEN 100
50 IF X1 <95 THEN 120

```

```

60 IF X2>=95 THEN 100
70 IF X1<>100 THEN 30
80 IF X3<=80 THEN 30
90 IF X2<80 THEN 30
100 PRINT TAB(1), I, TAB(11), X1, TAB(21), X2,
    TAB(31), X3
110 GOTO 30
120 IF X2<>100 THEN 30
130 IF X1 <=80 THEN 30
140 IF X3 >80 THEN 100
150 GOTO 30
160 DATA 1, 100, 96, 75, 2, 91, 92, 90
170 DATA 3, 90, 81, 85, 4, 100, 85, 76
180 DATA 5, 95, 96, 74, 6, 81, 100, 83
190 DATA 7, 98, 80, 75, 8, 95, 93, 80
200 DATA 9, 90, 95, 93, -1, -1, -1, -1
210 END

```

JRUN

I	X1	X2	X3
1	100	96	75
2	91	92	90
5	95	96	74
6	81	100	88
9	90	95	93

择优录取

某中学打算成立数学兴趣小组。布告贴出后，前来报名的同学十分踊跃，大大超过了预先规定的人数。于是学校领导决定进行一次测验，按成绩择优录取，分数线定在75分，超过分数线的同学可以参加。



请你编一个程序，
将录取者的分数打印出来。

```
5  FOR I=1 TO 24
10 READ X
20 IF X< 75 THEN 10
40 PRINT X
45 NEXT I
50 END
60 DATA 80, 83, 72, 99, 95, 88, 79, 63, 100, 98
70 DATA 93, 85, 94, 81, 77, 71, 92, 74, 73, 78
80 DATA 75, 94, 66, 72
```

JRUN

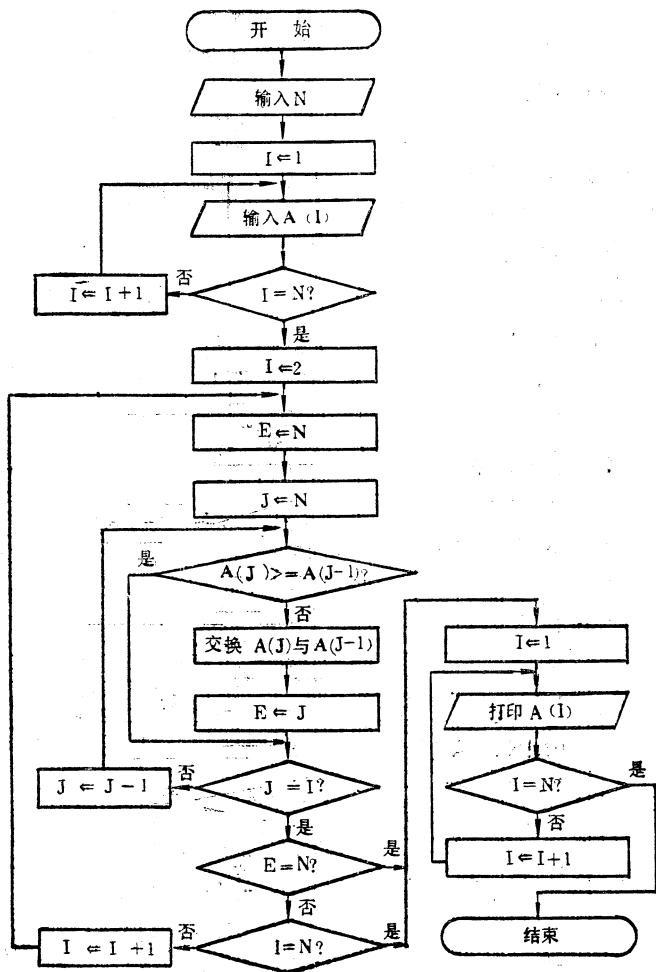
95	88	79
100	98	93
85	94	81
77	92	78
75	94	

假如数学兴趣小组只能录取前十名，你能否重新编个程序，让电脑挑选出前十名优胜者，并将他们的成绩打印出来呢？

谁的分数最高



上一问题中，如果要录取者的分数按大小顺序打印出，看谁的分数最高？这就比上一问题复杂一些。以下是按大小顺序打印录取者分数的程序和流程图：



5 PRINT
10 INPUT N
15 PRINT

```

20 DIM A(N)
25 FOR I=1 TO N
30 INPUT A(I)
35 PRINT " "
40 NEXT I
45 PRINT
50 PRINT
55 FOR I = 2 TO N
60 E = N
65 FOR J = N TO I STEP - 1
70 IF A(J) >= A(J-1) THEN 95
75 C=A(J)
80 A(J)=A(J-1)
85 A(J-1)=C
90 E=J
95 NEXT J
100 NEXT J
105 FOR I=1 TO N
110 PRINT TAB(4); A(I),
115 NEXT I
120 PRINT
125 END

```

JRUN

75	77	78
79	80	81
83	85	88
92	93	94

94 95 98

99 100

解一元二次方程

用微电脑解方程很方便，只要将方程的系数输入，程序运行后，电脑就能很快将解打印出来。下面我们来设计一个解一元二次方程的程序。

一元二次方程的标准形式是：

$$Ax^2 + Bx + C = 0$$

$$(A > 0)$$

由系数A、B、C的不同，方程的解具有下述几种情况：

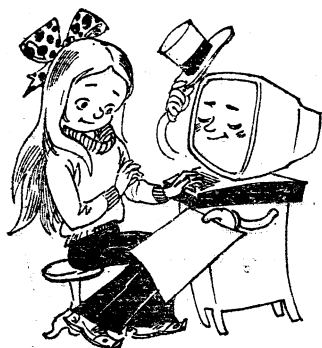
$$(1) \text{ 如果 } B^2 - 4AC > 0 \text{ 那么 } x_{1,2} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A},$$

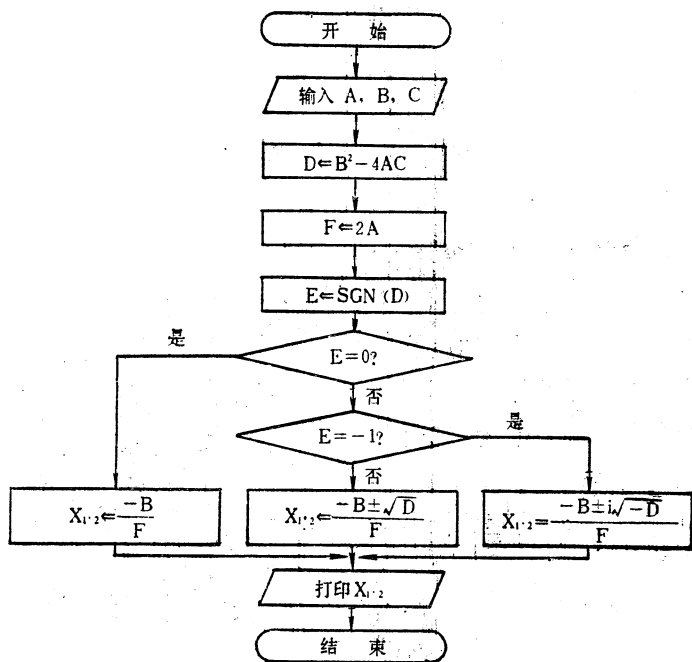
$$(2) \text{ 如果 } B^2 - 4AC = 0 \text{ 那么 } x_1 = x_2 = \frac{-B}{2A},$$

$$(3) \text{ 如果 } B^2 - 4AC < 0 \text{ 那么 } x_{1,2} = -\frac{B}{2A}$$

$$\pm i \frac{\sqrt{4AC - B^2}}{2A}.$$

根据上述结论，我们可以编制出以下流程图和程序：





```

10 PRINT "A="; :INPUT A
12 PRINT "B="; :INPUT B
14 PRINT "C="; :INPUT C
15 D=B*B-4*A*C
18 F=2*A
20 E=SGN(D)
22 IF E=0 THEN 34
24 IF E=-1 THEN 36
30 PRINT "X2="; -B/F+SQR(D)/F
32 PRINT "X2="; -B/F-SQR(D)/F: GOTO

```


40

```
34 PRINT "X1=X2="; -B/F: GOTO 40
36 PRINT "X1", -B/F, "+i", SQR(-D)/F
38 PRINT "X2=", -B/F, "-i", SQR(-D)/F
40 END
```

]RUN

A=? 6

B=? 9

C=? 11

X1=-.75+i1.12731244

X2=-.75-i1.12731244

]RUN

A=? 6

B=? 9

C=? 3

X1=-.5

X2=-1

]RUN

A=? 1

B=? 2

C=? 1

X1=X2=-1

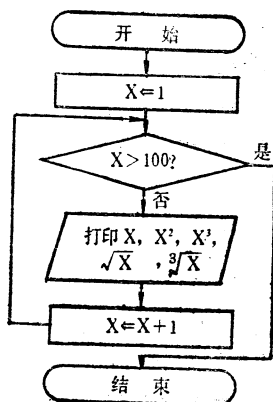
打印数学用表



一些常用的数学用表如果让一个中学生靠笔算来编制，那得花很长时间，而且难免出错，如果请微电脑来做这项工作，那就快速、方便、可靠，只要编制一个简单的程序，上机运行一下，不出几分钟，一张张数学用表就打印出来了。

下面是用 BASIC 语言编制的打印100以内平方、立方、平方根、立方根表的有关程序和流程图：

以下程序如果将20句进行修改，还可以得到其他的有关数表。



```
20 PRINT "X",
```

```
22 PRINT TAB(5), "X×X",
```

```

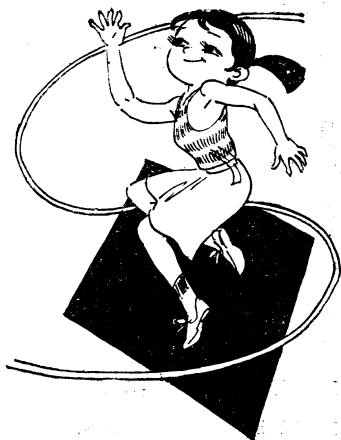
24 PRINT TAB(11), "X*X*X";
26 PRINT TAB(19), "X^(1/2)";
28 PRINT TAB(30), "X^(1/3)"
29 FOR X = 1 TO 100
30 PRINT X;
32 PRINT TAB( 5), X * X;
34 PRINT TAB(11), X * X * X;
36 PRINT TAB(19), X^(1/2);
38 PRINT TAB(30), X^(1/3);
40 NEXT X
45 END

```

JRUN

X	X*X	X*X*X	X^(1/2)	X^(1/3)
1	1	1	1	1
2	4	8	1.41421356	1.25992105
3	9	27	1.73205081	1.44224957
4	16	64	2	1.58740105
5	25	125	2.23606798	1.70997595
6	36	216	2.44948974	1.81712059
:	:	:	:	:
95	9025	857375	9.74679435	4.56290264
96	9216	884736	9.79795897	4.57885697
97	9409	912673	9.84885781	4.5947009
98	9604	941192	9.89949495	4.6104363
99	9801	970299	9.94987438	4.62606501
100	10000	1000000	10	4.64158884

打印正弦曲线



试打印从 $-\pi$ 到 $+\pi$ 间的正弦曲线，要求每隔10度打一个点。

要打印正弦曲线，只需用循环语句便可以了（设步长为 $3.1459/18$ ）。问题是曲线位置的设置，需要用到打印函数。如果要在显示屏上显示曲线，必须考虑显示屏仅能显示40列，使曲线的峰值尽可能落

在显示屏内。

以下是程序：

```
10 FOR X=-3.14159 TO 3.14159 STEP  
   3.14159/18  
20 PRINT INT (X*180/3.14159);  
30 PRINT TAB(22+17*SIN(X)); "*"   
40 NEXT X  
50 END
```

]RUN

-180	
-170	
-161	
-151	
-141	
-131	
-121	
-111	*
-101	*
-91	*
-81	*
-71	*
-61	*
-51	*
-41	*
-31	*
-21	*
-11	*
-1	*
9	*
19	*
29	*
39	*
49	*
59	*
69	*
79	*
89	*
99	*
109	*
119	*
129	*
139	*
149	*
159	*
169	*
179	*

如果要打印余弦、正切、余切或其他三角函数的曲线，只要将第30句语句稍加改动即可。

打印课程表



新学期开始了，开学那天，老师在黑板上写出了本学 期所有课程。方华同学灵机一动，编制了一个简单的程序，请电脑为全班每位同学各打印一张课程表。下面就是他编制的程序。请注意，程序中各门学科用简化的英语字母表示：

语 文 (CHINESE) 表示成 CHIN;

数 学 (MATHEMATICS) 表示成 MATH;

英 语 (ENGLISH) 表示成 ENGL;

政 治 (POLITICS) 表示成 POLI;

化 学 (CHEMISTRY) 表示成 CHEM;

物 理 (PHYSICS) 表示成 PHYS;

计算机 (COMPUTER) 表示成 COMP;

体 育 (SPORT) 表示成 SPOR;

自 修 (SELFSTUDY) 表示成 SELF.

```
10 DIM A$(12), B$(12), C$(12), D$(12),
```

```
    E$(12), F$(12)
```

```

20 PRINT TAB(1); "1"; TAB(7); "2"; TAB(13); "3";
   TAB(20); "4" TAB(26); "5"; TAB(32); "6"
30 READ A$, B$, C$, D$, E$, F$
40 PRINT TAB(1); A$; TAB(7); B$; TAB(13);
   C$; TAB(20); D$; TAB(26); E$; TAB(32);
   F$
50 GOTO 30
60 DATA "CHIN", "MATH", "ENGL"
70 DATA "CHEM", "CHIN", "POLI"
80 DATA "CHIN", "METH", "ENGL"
90 DATA "MATH", "PHYS", "ENGL"
100 DATA "SPOR", "CHEM", "COMP"
110 DATA "MATH", "PHYS", "PHYS"
120 DATA "ENGL", "PHYS", "CHIN"
130 DATA "SPOR", "CHEM", "COMP"
140 DATA "*", "*", "*", "*", "*", "*"
150 DATA "COMP", "POLI", "CHEM"
160 DATA "POLI", "ENGL", "SELF"
170 DATA "SELF", "POLI", "SELF"
180 DATA "SELF", "SELF", "SELF"
190 DATA "SELF", "SELF", "SELF"
200 DATA "SELF", "SELF", "SELF"
300 END

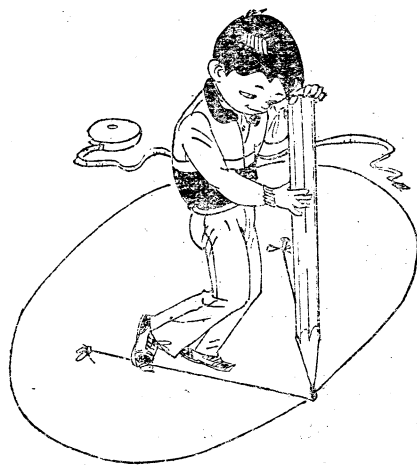
```

JRUN

1	2	3	4	5	6
CHIN	MATH	ENGL	CHEM	CHIN	POLI
CHIN	METH	ENGL	MATH	PHYS	ENGL

SPOR	CHEM	COMP	MATH	PHYS	PHYS
ENGL	PHIN	CHIN	SPOR	CHEM	COMP
✖	✖	✖	✖	✖	✖
COMP	POLI	CHEM	POLI	ENGL	SELF
SELF	POLI	SELF	SELF	SELF	SELF
SELF	SELF	SELF	SELF	SELF	SELF

求 椭 圆 周 长



在科学实验和生产实践中，常常遇到要计算椭圆周长的问题，一般用查椭圆积分表的办法。但椭圆积分表毕竟有限，不能把所有的原始数据都列入。当查不到所需数据时，还得进行插补运算，很是麻烦。

这里是一个用 BASIC 语言写的求椭圆周长的程序。只要把椭圆的长短半轴值从键盘输入，就可以求得椭圆周长，其精度在小数点后七位。

椭圆周长 L 由下式计算： $L=4aE(k)$ ，其中 a 为椭圆长半轴， $E(k)$ 为椭圆周率，是第二类完全椭圆积分。

$$E(k) = \int_0^{\pi/2} \sqrt{1 - k^2 \sin^2 \varphi} d\varphi, \quad k = \sqrt{1 - (b/a)^2} \text{ 称为偏}$$

心率, b 是椭圆的短半轴。

$E(k)$ 也可按下面求和公式计算:

$$E(k) = \frac{\pi}{2} \sum_{i=0}^{\infty} \frac{1}{1-2i} (i^{-\frac{1}{2}}) k^{2i} \text{ [注]}$$

式中,

$$\binom{x}{i} = \frac{x(x-1)\cdots(x-i+1)}{1 \cdot 2 \cdot 3 \cdots i} \quad \text{当 } i=0 \text{ 时, } \binom{x}{0} = 1$$

以下 BASIC 程序就是按此式编写的。

```

10 INPUT "B="; B
20 INPUT "A="; A
30 K=1-(B/A)^2
40 E=1:D=1:F=1:C=1
50 FOR I=1 TO 45
60 D=D/100*(0.5-I)
70 F=F/100*I
100 C=C*K
110 D2=D/F
120 E=E+D2/2*C/(1-2*I)
130 NEXT I
140 E=3.141592654/2*E
150 PRINT "E(K)=", E
160 PRINT "L=", 4*A*E
170 END

```

]RUN

[注]见《数学手册》(人民教育出版社)353页。

B=100

A=200

E(K)=1.21105603

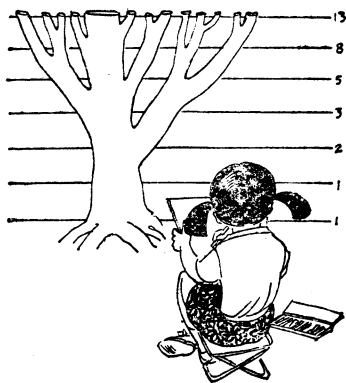
L=9687844823

菲波那奇数列

著名的菲波那奇数列是这样一列数，即从它的第三项起，每一项都是其前两项之和：

$$F_n = F_{n-1} + F_{n-2}$$

例如有一棵根据下列规律生长的树：它的每一根老枝（包括树干）每年长出一根新的分枝，这根新枝当年不分枝，第二年成为老枝后开始分枝。这种生长方式如图所示。几年以后，树枝的数目为 F_n 。各年的树枝数目组成了菲波那奇数列，它的前12个数为：



1、1、2、3、5、8、13、21、34、55、89、144。

下列程序能产生出菲波那奇数列的前30个数：

```
100 DIM F(30)
110 LET F(1)=1
120 LET F(2)=1
130 FOR N=1 TO 28
```

```

140 LET F(N+2)=F(N+1)+F(N)
150 NEXT N
160 FOR X=1 TO 30
170 PRINT F(X)
180 NEXT X
190 END

```

JRUN

1	1	2
3	5	8
13	21	34
55	89	144
233	377	610
987	1597	2584
4181	6765	10946
17711	28657	46368
75025	121393	196418
317811	514229	832040

请修改一下这个程序，以计算出菲波那奇数列的前50项。

阿姆斯特朗数

153是一个很有趣的数，仔细观察一下，我们能够发现：

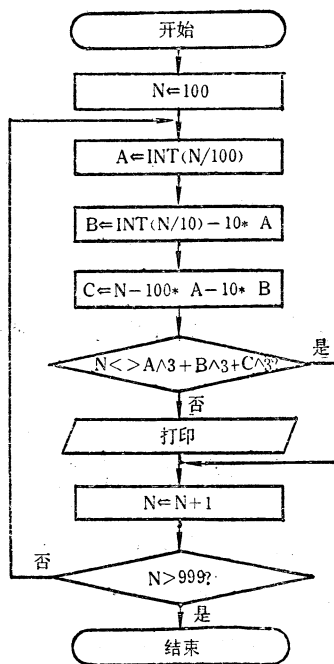
$$153=1^3+5^3+3^3$$

凡具有这种性质的数，我们称它为阿姆斯特朗数 (ARMSTRONG NUMBER)，也就是说，如果0~9的



某些数字的 3 次幂的和
等于这 N 个数字原始排
列成的数，那么，这 N
位数就是一个阿姆斯特朗
数。

下面所给的流程图
和程序能够找寻所有 3
位阿姆斯特朗数。



```

100 REM ARMSTRONG NUMBERS
110 FOR N=100 TO 999
120 LET A=INT(N/100)
130 LET B=INT(N/10)-10*A
140 LET C=N-100*A-10*B
150 IF N<>A*A*A+B*B*B+C*C*C
    THEN 190
160 PRINT "ARMSTRONG NUMBER", N
170 PRINT "EQUALS", A*A*A, "+", B*B*B,
    "+", C*C*C
180 PRINT
190 NEXT N
200 END

```

]RUN

ARMSTRONG NUMBER 153

EQUALS 1+125+27

ARMSTRONG NUMBER 370

EQUALS 27+343+0

ARMSTRONG NUMBER 371

EQUALS 27+343+1

ARMSTRONG NUMBER 407

EQUALS 64+0+343

毕达哥拉斯数

我国最早的一本数学书籍“周髀算经”中，记载着公元前几百年的两个人物——周公和商高的一段对话，其中提到了“勾三股四弦五”，这个思想以后就发展成为现在的“勾股定理”。这个定理说的是直角三角形的三条边长都满足 $a^2+b^2=c^2$ (a 、 b 为两直角边， c 为斜边) 的关系。古代埃及人也早已了解这种关系，他们在做竖琴时，就利用了边长为 3、4、5 的关系来得出直角。



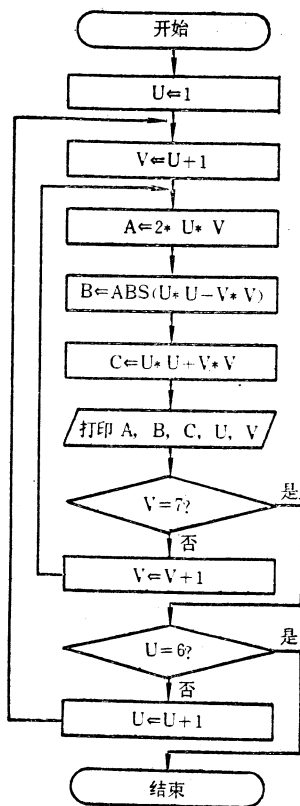
这个定理，在法国和比利时等国称为“驴桥定理”，而在大多数国家则叫做毕达哥拉斯定理。并将满足这种关系的数的组合，称作毕达哥拉斯数。

毕达哥拉斯数是无限多的，但是如何寻找毕达哥拉斯数呢？我们可以任意举出一个奇数来，把它的平方数分成相差 1 的两数，这三个数就是一组毕达哥拉斯数。

例如取奇数 11，那么 $11^2=121$ ，把 121 分成相差 1 的两数 60 和 61，则这三个数 11，60，61 就是一组毕达哥拉斯数。

还有一种很简单的方法：可以任意选定两个不相等的正整数 u 、 v ，那么利用公式 $a=2uv$ ， $b=|u^2-v^2|$ 和 $c=u^2+v^2$ ，

就可以找出由 a 、 b 、 c 组成的一组毕达哥拉斯数,让我们编一个程序,利用这个方法找出90以内所有的毕达哥拉斯数。由于三数不能超过90,所以取 u 最大为6, v 最大为7,且 $v=u+1$ 。以下是流程框图和程序:



```

10 PRINT TAB(1), "A", TAB(8), "B", TAB(16),
    "C", TAB(24), "U", TAB(32), "V"
20 FOR U=1 TO 6

```

```

30 FOR V=U+1 TO 7
40 A=2*U*V
50 B=ABS(U*U-V*V)
60 C=V*V+U*U
70 PRINT TAB(1), A, TAB(8), B, TAB(16), C,
    TAB(24), U, TAB(32), V
80 NEXT V
90 NEXT U
100 END

```

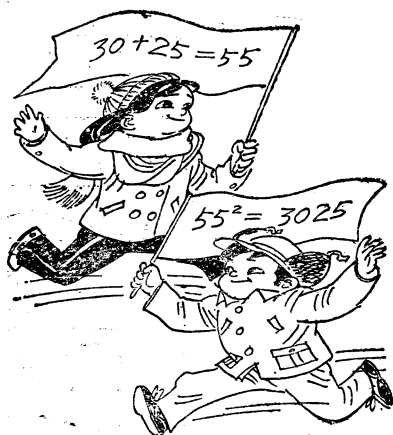
]RUN

A	B	C	U	V
4	3	5	1	2
6	8	10	1	3
8	15	17	1	4
10	24	26	1	5
12	35	37	1	6
14	48	50	1	7
12	5	13	2	3
16	12	20	2	4
20	21	29	2	5
24	32	40	2	6
28	45	53	2	7
24	7	25	3	4
30	16	34	3	5
36	27	45	3	6
42	40	58	3	7
40	9	41	4	5

48	20	52	4	6
56	33	65	4	7
60	11	61	5	6
70	24	74	5	7
84	13	85	6	7

特殊的数

3025是一个很特殊的四位数，我们如果将其拆成两部分，高二位（30）和低二位（25）之和为55，而55的平方正好又是3025。四位数中有这种特殊性质的数还有哪些呢？我们不妨通过下述程序来寻找一下。



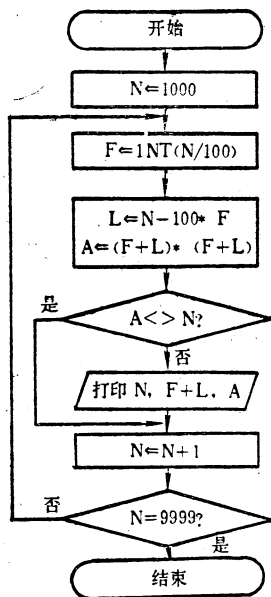
```

10 PRINT "N", "SUM", "SUM SQUARED"
20 PRINT
30 FOR N=1000 TO 9999
40 F=INT(N/100)
50 L=N-100*F : A=(F+L)*(F+L)
60 IF A<>N THEN 80
70 PRINT N, F+L, A
80 NEXT N
90 END

```

JRUN

N	SUM	SUM SQUARED
2025	45	2025
3025	55	3025
9801	99	9801



完 全 数

我们知道，6 的约数（除 6 本身之外）是 1、2 和 3，
如将这些约数相加，其和又组成了 6 本身，

$$6 = 1 + 2 + 3$$

我们称6为完全数。也就是说：一个完全数等于除了它本身以外的所有约数的代数和。

$2^{n-1}(2^n-1)$ 是求完全数的公式，其中因子 2^n-1 必须是一个素数。下列BASIC程序能帮你寻找并打印出100元内所有完全数。



程序设计思想是这样的：根据完全数的定义，在变数N从2到100之内逐个寻找。由于不考虑N本身作约数，故变数D的取值范围是从1到N/2。然后一次次运算，判断D是否为N的约数，如果是的话，存入累加器S之中。待所有约数都累加起来之后，将S与N进行比较，若S与N相等，就是一个完全数，就将该N打印出来；若S与N不相等，就将N加上1，然后，再重复上述过程。

```
10 FOR N=2 TO 100
20 LET S=0
30 FOR D=1 TO N/2
40 IF INT(N/D)<>N/D THEN 60
50 LET S=S+D
60 NEXT D
70 IF S<>N THEN 90
80 PRINT N, "IS A PERFECT NUMBER"
```

```

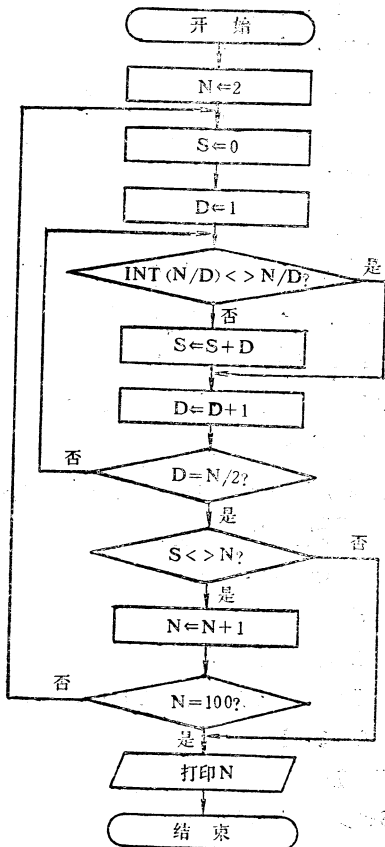
90 NEXT N
100 END

```

]RUN

6 IS A PERFECT NUMBER

28 IS A PERFECT NUMBER



你能否修改一下程序，找出第三个完全数？

有趣的幻数

有这样一类的幻数，把它各位上的数字相加，再将得出的结果的各位数字相加，如此不断加下去，最后结果总是1。例如，“298”就是这一类数：

$$2 + 9 + 8 = 19$$

$$1 + 9 = 10$$

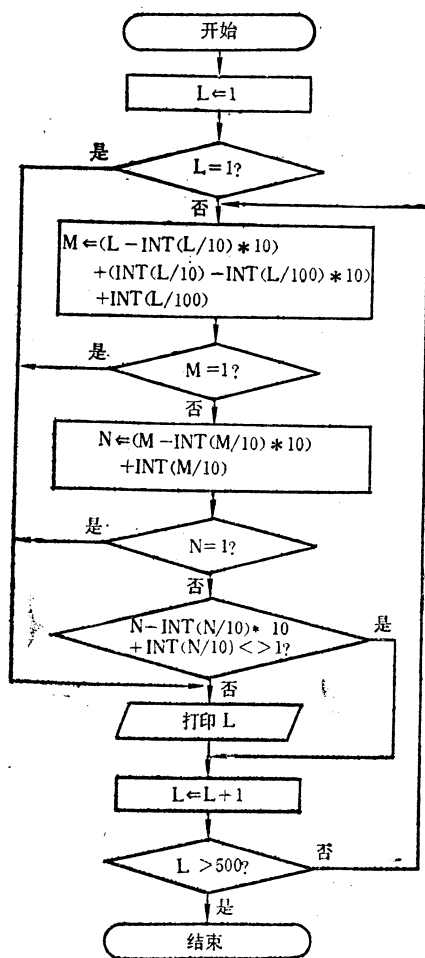
$$1 + 0 = 1$$

试编一程序找出500以内所有的这类有趣的幻数。

以下是流程图和程序：

```
10 FOR L=1 TO 500
20 IF L=1 THEN 80
30 M=(L-INT(L/10)*10)+(INT(L/10)-INT
   (L/100)*10)+INT(L/100)
40 IF M=1 THEN 80
50 N=(M-INT(M/10)*10)+INT(M/10)
60 IF N=1 THEN 80
70 IF N-INT(N/10)*10+INT(N/10)<>1
   THEN 90
80 PRINT L.
90 NEXTL
100 END
```





JRUN

1	10	19
28	37	46
55	64	73
82	91	100
109	118	127
136	145	154
163	172	181
190	199	208
217	226	235
244	253	262
271	280	289
298	307	316
325	334	343
352	361	370
379	388	397
406	415	424
433	442	451
460	469	478
487	496	

奇妙的六位数

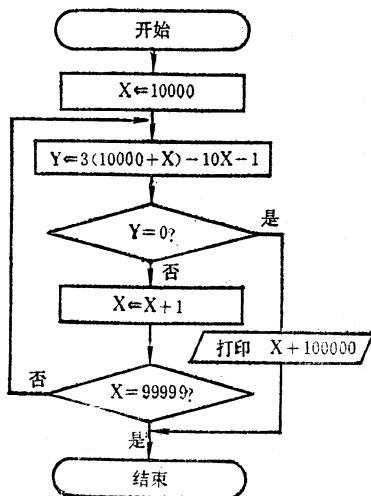
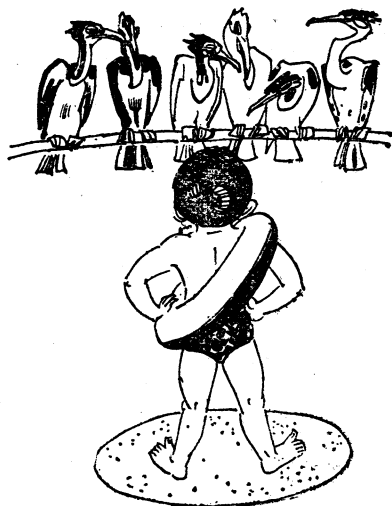
设有六位数 $1abcde$ ，乘以 3 以后变为 $abcde1$ ，试求这个六位数。

我们可以先作如下的分析，设 $x = abcde$ ，则：

$$\begin{aligned} 3(100000 + x) \\ = 10x + 1 \end{aligned}$$

解之得： $x = 42857$

这个问题可用下列程序由微电脑来解：



```

10 FOR X=10000 TO 99999
20 Y=3*(100000+X)-10*X-1
30 IF Y=0 THEN 50
40 NEXT X
50 PRINT X+100000
60 END

```

JRUN

142857

1 4 2 8 5 7 可真是个奇妙的六位数，它还具有以下的特点：

(1) 将142857分别乘以1、2、3、4、5、6，能得到下面六个数：

$$142857 \times 1 = 142857$$

$$142857 \times 2 = 285714$$

$$142857 \times 3 = 428571$$

$$142857 \times 4 = 571428$$

$$142857 \times 5 = 714285$$

$$142857 \times 6 = 857142$$

乘积中的每一个数，都是此六位数的各个数字按一定顺序变换而得到的。

$$(2) 142857 \times 7 = 999999$$

将1/7化成小数，得到的是一个无限循环小数：

$$1/7 = 0.142857142857\cdots\cdots\text{其循环节为}142857,$$

$$\text{即} 1/7 = 0.\dot{1}4285\dot{7} \quad \text{所以} \quad 0.\dot{1}4285\dot{7} \times 7 = 1$$

(3)

	{	8=1142856	(142856+1=142857)
		9=1285713	(285713+1=285714)
142857 × {		10=1428570	(428570+1=428571)
		11=1571427	(571427+1=571428)
			
		69=9857133	(857133+9=857142)

$$\begin{array}{r}
 57 \\
 + 75 \\
 \hline
 132 \\
 + 231 \\
 \hline
 363
 \end{array}$$

试写一个 BASIC 程序,找到一组用下列整数: 57、134、188、266、563作为首选整数而形成的此种性质的数。

以下是程序:

```

10 INPUT A
20 A$=STR$(A);I=LEN(A$)
30 FOR J=1 TO I/2
40 IF MID$(A$, J, 1)<>MID$(A$,
    I-J+1, 1) THEN E5
50 NEXT J
60 PRINT A; END
65 B$=""
70 FOR J=1 TO I
80 C$=MID$(A$, I-J+1, 1); B$=B$+C$
90 NEXT J
100 A=A+VAL(B$); GOTO 20

```

]RUN

?57

363

]RUN

?134

565

JRUN

?188

233332

JRUN

?563

88555588

JRUN

?266

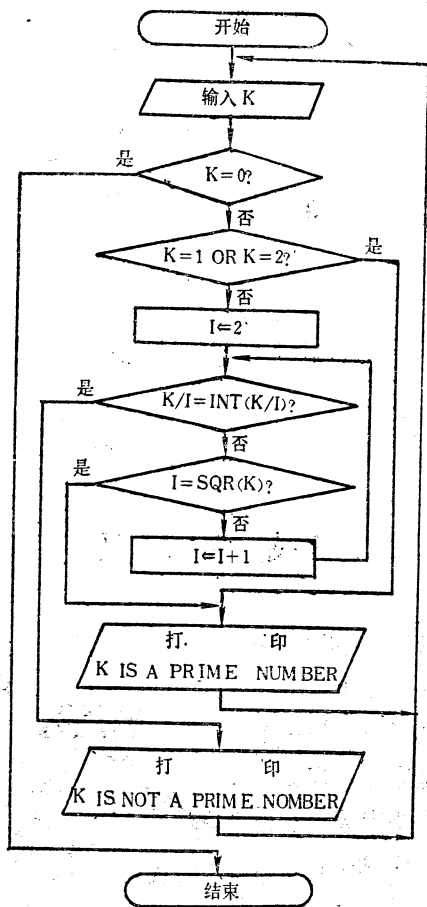
88555588

583 是 素 数 吗

两位同学为了判断素数，发生了一场争论。一个认为 583 是素数，另一个却说 583 不是素数，那么，583 究竟是不是素数呢？你能否能用微电脑来判别一下，哪位同学说得对？

下面是对任意已知数，判断其是否为素数的程序以及相应的程序框图。





```

5 PRINT "IS THE NUMBER PRIME?"
10 INPUT K
15 IF K=0 THEN 80
20 IF K=1 OR K=2 THEN 60
30 FOR I=2 TO SQR (K)

```

```

40 IF K/I=INT (K/I) THEN 70
50 NEXT I
60 PRINT K; " IS A PRIME NUMBER ";
   GOTO 10
70 PRINT K; " IS NOT A PRIME NUMBER";
   GOTO 10
80 END

]RUM
IS THE NUMBER PRIME?
?583
583 IS NOT A PRIME NUMBER
?571
571 IS A PRIME NUMBER
?123457
123457 IS A PRIME NUMBER
?342349
342349 IS NOT A PRIME NUMBER
?111
111 IS NOT A PRIME NUMBER

```

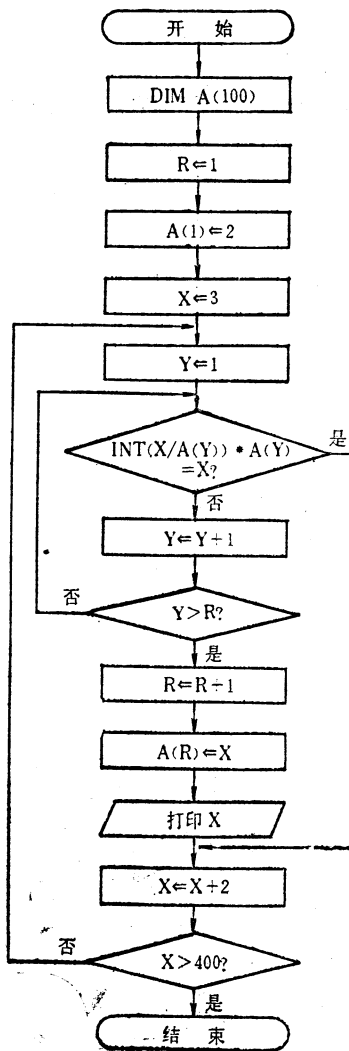
找 素 数

所谓“素数”，就是指这样一种正整数，它的约数只有1和它自身，它不能被分解为两个比自身小的因数的乘积。例如：7和13是素



数，但 4 和 10 就不是素数，因为 $4 = 2 \times 2$ ， $10 = 5 \times 2$ 。

下列程序能够帮助你找到 400 之内的所有素数。



```

5 DIM A(100)
10 PRINT "LIST OF
    PRIME NUMBERS"
15 PRINT
20 PRINT "2",
25 LET R=1
30 LET A(1)=2
35 FOR X=3 TO
    400 STEP 2
40 FOR Y=1 TO R
45 IF INT (X/A(Y))
    *A(Y)=X THEN
70
50 NEXT Y
55 LET R=R+1
60 LET A(R)=X
65 PRINT X,
70 NEXT X
75 END
  
```

]RUN
LIST OF PRIME
NUMBERS

2	3	5
7	11	13
17	19	23
29	31	37
41	43	47
53	59	61
67	71	73
79	83	89
97	101	103
107	109	113
127	131	137
139	149	151
157	163	167
173	179	181
191	193	197
199	211	223
227	229	233
239	241	251
257	263	269
271	277	281
283	293	307
311	313	317
331	337	347
349	353	359
367	373	379
383	389	397

孪生素数



两个相差 2 的素数称作孪生素数，你能将 1000 以内所有的孪生素数都写出来吗？下面的程序能通过微电脑将结果很快地打印出来：

```
10 DIM A(1000), B
   (400)
20 FOR X=2 TO
   1000
30 LET A(X)=0
40 NEXT X

50 LET C=0
60 LET S=SQR (1000)
70 FOR B=2 TO 1000
80 IF A(B)<0 THEN 150
90 LET C=C+1
100 LET B(C)=B
110 IF B>S THEN 150
120 FOR X=B TO 1000 STEP B
130 LET A(X)=-1
140 NEXT X
150 NEXT B
160 FOR X=2 TO C
170 IF B(X)-B(X-1)<>2 THEN 190
180 PRINT B(X-1), B(X)
```


190 NEXT X

200 END

7RUN

3	5
5	7
11	17
17	19
29	31
41	43
55	61
71	73
101	103
107	109
137	139
149	151
179	181
191	193
197	199
227	229
239	241
269	271
281	283
311	313
347	349
419	421
431	433
461	463

521	523
569	571
599	601
617	619
641	643
659	661
809	811
821	823
827	829
857	859
881	883

逐一“验证”哥德巴赫猜想的正确性

哥德巴赫猜想：任意一个比4大的偶数都可以写成两个质数之和。



如果你想验证哥德巴赫猜想，那就得提出一个个偶数，硬是将它们凑成两个质数之和。可是找出这两个质数，并非轻而易举，尤其是检验大偶数，困难更大。

为此，我们又想到了微电脑。

下面就是用微电脑逐一“验证”哥德巴赫猜想的程

序。

电脑要求你输入一个偶数N，它就能很快告诉你，这个偶数能写成哪两个质数之和。如果你输入的N不是偶数，电脑还会向你提出抗议：“THIS IS NOT AN EVEN NUMBER（这不是偶数）！”并要求你重新输入N。

```
10 INPUT "N="; N
20 IF INT(N/2)=N/2 THEN 40
30 PRINT "THIS IS NOT AN EVEN
   NUMBER! "; GOTO 10
40 FOR I=3 TO N/2 STEP 2
50 X=I; IF X<9 THEN 70
60 GOSUB 120; IF Y=0 THEN 100
70 X=N-I; IF X<9 THEN 90
80 GOSUB 120; IF Y=0 THEN 100
90 PRINT N, "=", I, "+", X; PRINT ; GOTO
   10
100 NEXT I
110 END
120 L=INT(SQR(X))
130 FOR J=3 TO L STEP 2
140 IF INT(X/J)=X/J THEN 170
150 NEXT J
160 Y=1; GOTO 180
170 Y=0
180 RETURN

J RUN
N=7
```

THIS IS NOT AN EVEN NUMBER;

$$N=20$$

$$20=3+17$$

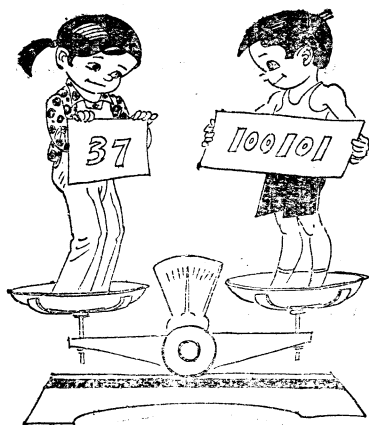
$$N=1024$$

$$1024=5+1021$$

$$N=500$$

$$500=13+487$$

数 制 转 换

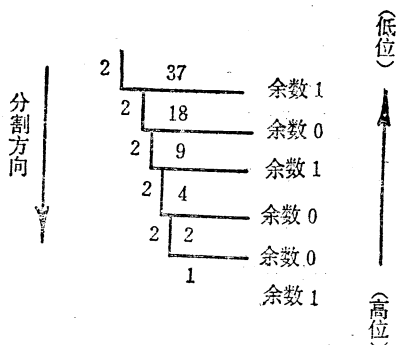


电脑是用电来驱动的，当然要服从电的规律。电的最基本的特性是电流的“通”和“断”，即“开”和“关”的状态；或者是电流流通的方向，即“正”和“负”的状态；以及电磁铁的磁化方向，即“N极”和“S极”的指向。电流和电磁铁总是处于两

种状态之一，并具有二进制的特性。所谓“二进制”，就是只有两个数码——“0”和“1”，并规定了“逢二进一”的计数规则。电脑中所用的数制就是二进制。但是，人们已

经习惯于使用十进制了。那么，十进制数与二进制数之间又如何进行转换呢？人们通常采用长除法，将十进制数转换为二进制数。譬如，要将十进制数37转换成二进制数（见左图）。我们先将37除以

2，余数“1”抄在右面（如果余数为“0”也应把它抄在右面）；再将商继续除以2，并抄下余数，直至商为1（将该“1”也抄在右面余数栏内）。然后，从下到上读余数，得到



了二进制数100101，这就是37的二进制数表示了。

如果要把一个很大的十进制数转换成二进制数表示，采用长除法来运算，是相当繁复的。为此，我们还是请电脑来帮一下忙吧！下面就是一个能将任意十进制数转换成二进制数的程序。

```

10 DIM D(20), N(20)
20 INPUT N(1)
30 P=0
40 IF 2^P>N(1) THEN 70
50 P=P+1
60 GOTO 40
70 FOR I=1 TO P
80 D(I)=INT(N(1)/2^(P-I))
90 N(I+1)=N(1)-(D(I)*2^(P-I))
100 NEXT I

```

```

120 PRINT N(1); "=",
130 FOR I=1 TO P
140 PRINT D(I);
150 NEXT I
160 PRINT: PRINT
170 GOTO 20
180 END

```

]RUN

? 37

37=100101

? 1024

1024=10000000000

两数平方和



有些正整数能表示成两个整数平方和的形式。例如：

$$17 = 1^2 + 4^2$$

$$41 = 4^2 + 5^2$$

试用 BASIC 语言写一个程序，找出50以内所有能表示成两个整数平方和的数。

由于要找的数在50以内，因此表示为两个

整数平方和形式的这两个整数，其最大的一个至多是 7（7 和平方已是 49）。我们可用循环语句来寻找。

下面是流程图和程序：

```

10 FOR N=1 TO
    50
20 FOR A=1 TO
    7
30 FOR B=1 TO
    7
40 IF  $A^2 + B^2 > N$ 
    2 < > N
    THEN 70
50 PRINT N; "=",
    A^2; "+", B^2
60 GOTO 90
70 NEXT B
80 NEXT A
90 NEXT N
100 END

```

↓ RUN

2=1+1

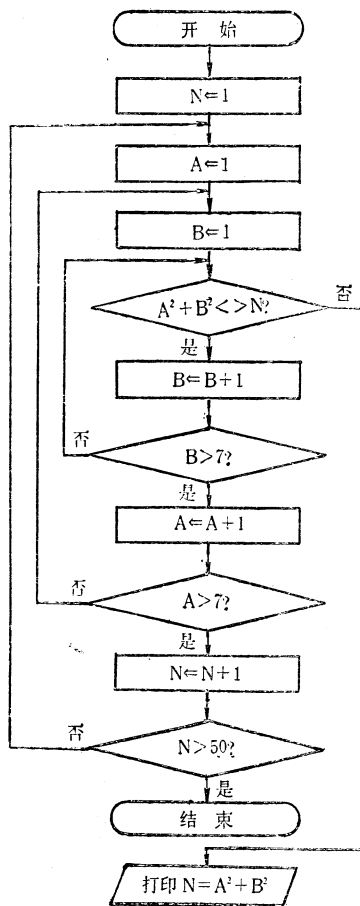
5=1+4

8=4+4

17=1+16

20=4+16

25=16+9



$$32=16+16$$

$$41=16+25$$

捣蛋鬼 3792

在某些科学实验项目中，往往会遇到随机数。我们这里所讲的“随机数”，是指你随便写出来的数，它也是一种随机数。每一位数可以都是0，也可以是1、2、3……也可

以是9，机会都是一样的。



你可能会说，随便写数谁不会。你可以试试，十有八九你“偏爱”某些数字，老是想它们；又“讨厌”某些数字，老是不写它们。因此，需要找一种“公平”的写“随机数”的方法。

科学家们曾经想出这样的方法：写下一个四位数，将它平方，剪取其十万、万、千、百这四位数字，抄下来；将这个四位数再平方，再以同样的办法，抄下四位数字。譬如，写下3274，将它平方，得10719076，把中间四位剪下，得7190，再平方……这样写下的一串数字从若干组开始，就可以认为是随机数了。如3274、7190、6961、4555、7480……

但是，科学家们遇到一个“捣蛋鬼”，它就是3792。3792

的平方是14379264，剪取中间的四位正好又是3792。于是便得到一连串3792,3792,3792……这怎么能叫做“随机数”呢？

好不容易找到的一种简便方法就这样被否定了。

现在要问：在四位数中，这种“捣蛋”数共有几个？它们是哪几个数？我们可以用下述简单的程序来寻找。

```

10 FOR A=1000 TO 9999
20 T=A*A-INT(A*A/1000000)*1000000
30 IF A<>INT(T/100) THEN 50
40 PRINT A
50 NEXT A
60 END

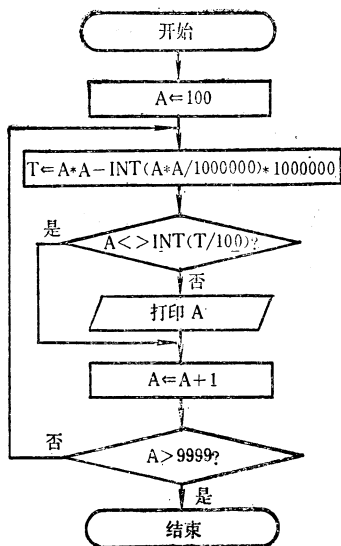
```

↓RUN

2500

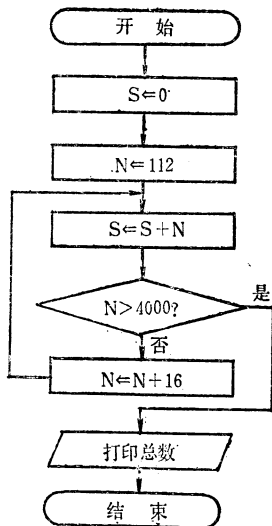
3792

7600



能被16除尽的数

112和128是能被16除尽的数。那么，在100和4000之间所有能被16除尽的数的总和是多少呢？下列程序能求出这个值。



```
10 REM NUMBERS DIVSIBLE BY 16
20 S=0
30 FOR N=112 TO 4000 STEP 16
40 S=S+N
50 NEXT N
```

```

60 PRINT "THE SUM OF ALL NUMBERS
    BETWEEN 112 AND 4000"
70 PRINT "THAT ARE DIVISIBLE BY 16 IS
    _ _ _ _ "; S
80 END

```

```

]RUN
THE SUM OF ALL NUMBERS BETWEEN 112
AND 4000
THAT ARE DIVISIBLE BY 16 IS      501664

```

整 数 A, B, C, D

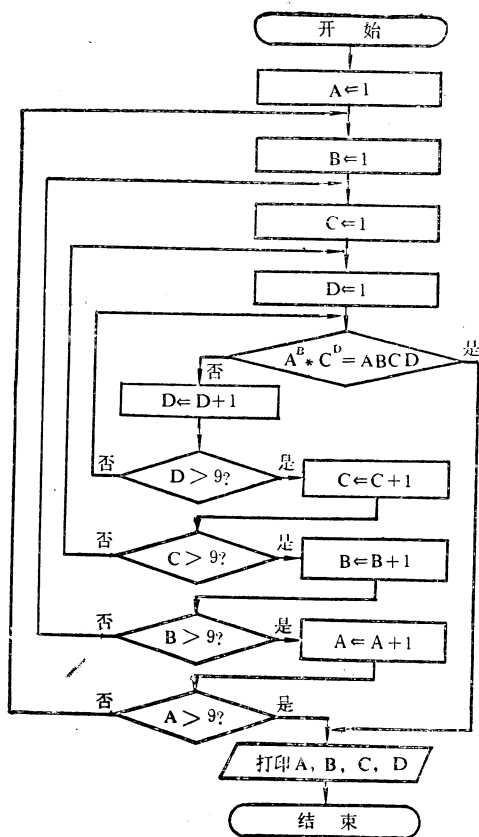
试找寻四个10以内
的正整数 A、B、C、D，
使它们满足下述关系：

$$A^B \times C^D \\ = ABCD$$

如果人工计算，一个数一个数地凑，相当麻烦。编个简单的搜索程序，让电脑来算一下，那就相当快了。



下面就是请电脑找寻整数 A、B、C、D 的流程图和程序。



```

5  REM INTERGER ABCD
10 FOR A=1 TO 9
20 FOR B=1 TO 9
30 FOR C=1 TO 9
40 FOR D=1 TO 9
50 IF A^B*C^D=1000*A+100*B+10*C+D

```

```

        THEN 110
60  NEXT D
70  NEXT C
80  NEXT B
90  NEXT A
100 GOTO 150
110 PRINT "A=", A
120 PRINT "B=", B
130 PRINT "C=", C
140 PRINT "D=", D
150 END

```

JRUN

A=2

B=5

C=9

D=2

整 数 和

你能够证明 N 个连续正整数的和的平方等于这 N 个正整数立方的和吗？例如：

$$(1+2+3+4)^2$$

$$=100$$

$$1^3+2^3+3^3+4^3$$

$$=100$$

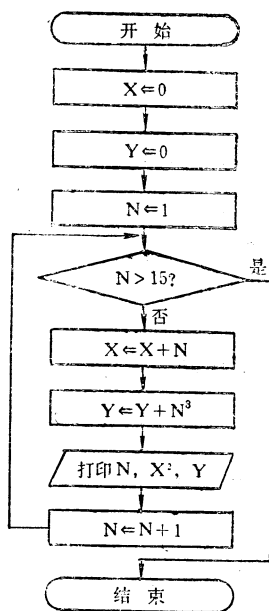


微电脑用穷举法能很快地验证整数的这个性质，这里所取的整数的个数 N 从1到15。

下面是流程图和用 BASIC 语言写的程序：

```

5 REM INTEGER SUMS
10 X=0; Y=0
20 PRINT "VALUE",
  "SUM OF", "SUM OF"
30 PRINT "OF N",
  "SQUARES", "CUBES"
40 PRINT
50 FOR N=1 TO 15
60 X=X+N
70 Y=Y+N*N*N
80 PRINT N, X*X, Y
90 NEXT N
100 END
  
```



↓RUN

VALUE OF N	SUM OF SQUARES	SUM OF CUBES
1	1	1
2	9	9
3	36	36
4	100	100
5	225	225
6	441	441

7	784	784
8	1293	1296
9	2025	2025
10	3025	3025
11	4356	4356
12	6084	6084
13	8281	8281
14	11025	11025
15	14400	14400

被墨迹覆盖了的数字

采购员老王买了一批水桶,回来向会计小韩报帐时,不小心将发票上沾上了墨水(见图),看不清单价了,总金额中头尾两个数也不见了。而老王又记不清水桶的单价,于是报帐发生了困难。你能通过微电脑的运算,将缺少的数字补全吗?

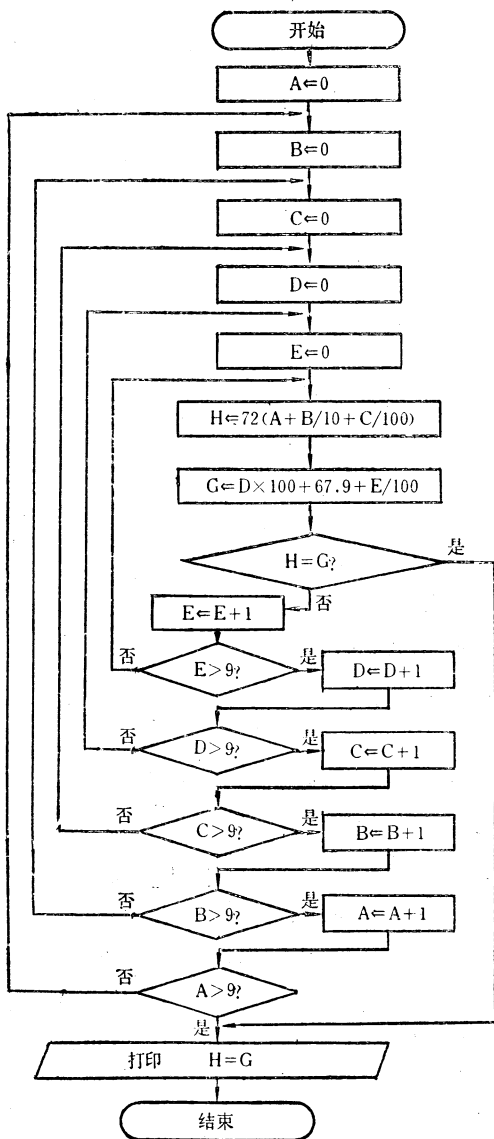


要给这个问题编制一个程序并不复杂,只需将下列

等式中 5 个未知量 A、B、C、D、E, 用 5 重循环来判断各个元素在它们的取值范围内的所有组合中,哪一种组合满足等式:

$$72 \times A \cdot BC = D67.9E \quad \text{即可。}$$

下面是程序和流程图,




```

5 FOR A=0 TO 9
10 FOR B=0 TO 9
20 FOR C=0 TO 9
30 FOR D=1 TO 9
40 FOR E=0 TO 9
50 H=72*(A+B/10+C/100)
60 G=D*100+67.9+E/100
70 IF H=G THEN 130
80 NEXT E
90 NEXT D
100 NEXT C
110 NEXT B
120 NEXT A
130 PRINT "72*", (A+B/10+C/100), "=", G
140 END

```

]RUN

72*5.11=367.92

玩 纸 牌

这是一个很简单的抽纸牌游戏，从52张牌中任意抽取2张，如果后一张是方块则算赢，如果2张都不是方块则输，假设玩这个游戏100、200、300、400、500…1000次，试确定每100次你能输、赢各多少次？



下列 BASIC 程序
能用微电脑来帮助你作
这个统计工作。程序是
用随机数来组成这副
牌的，其中：

方块 1~13

黑桃 14~26

草花 27~39

红桃 40~52

10 W=0

20 L=0

```

30 PRINT "GAMES", "WINS", "LOSSES"
40 PRINT
50 FOR G=1 TO 1000
60 X= INT(53*RND(1))
70 Y= INT(53*RND(1))
80 IF X>13 AND Y>13 THEN 110
90 W=W+1
100 GOTO 120
110 L=L+1
120 IF G/100<>INT(G/100) THEN 140
130 PRINT G, W, L
140 NEXT G
150 END

```

JRUN

GAMES

WINS

LOSSES

100	45	55
200	83	117
300	123	177
400	165	235
500	213	287
600	259	341
700	310	390
800	350	450
900	394	506
1000	438	562

奇怪的数学题

《奇怪的数学题》叙述了这样一件事：

四个人做游戏，每玩一次总有三个人赢，一个人输。输的人要按赢者手中的火柴根数进行赔偿，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，他们玩过四次之后，每人恰好各输一次，而且每人手中都正



好有16根火柴。请问,这四个人在做游戏前各有多少根火柴?

根据上述题意,似乎无法求解。但细想一下,发现这道《奇怪的数学题》应满足这样两个条件:(1)此题有唯一的结果,即每人最终都有16根火柴;(2)每次游戏后的火柴赔偿规则一律,即都是把三个人的火柴数加倍,而这些火柴全由第四者提供。基于这样两点,我们可以采用倒推法求解。

我们把最先输的人叫A,把第二次输的人叫B,把第三次输的人叫C,把最后一次输的人叫D。按照题意,四个人最后都剩下16根火柴,而且A、B和C赢了最后一次,所以在玩最后一次以前,A、B、C各有8根火柴,D有40根火柴。依次类推,我们很容易从赔偿规则推算出各人在游戏前具有的火柴根数。

这道《奇怪的数学题》是巧妙运用倒推法的经典例子,它是加倍博弈问题之一。下面我们利用微电脑来模拟人解题,推算出这四个人游戏前各人具有的火柴根数。

程序中第10句中的X为最后每人剩下的火柴根数,第15句中的N为参加游戏的人数。

```
10  X=16
15  N=4
20  PRINT "ORDER",
23  PRINT TAB(17)"A", TAB(23)"B", TAB(29)
    "C", TAB(35)"D"
25  FOR I=1 TO N
30  A(I)=X
35  NEXT I
```

```

40 PRINT "("; N; ")",
43 PRINT TAB(17)A(1); TAB(23)A(2); TAB
    (29)A(3); TAB(35)A(4)
45 FOR J=N TO 1 STEP -1
50 L=0
55 FOR I=1 TO N
60 IF J=I THEN 75
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 PRINT TAB(17)A(1); TAB(23)A(2); TAB
    (29)A(3); TAB(35)A(4)
90 NEXT J
95 END

```

JRUN

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

掷 硬 币



使用BASIC语言的随机函数来模拟掷硬币时出现正面朝上或反面朝上的情况（设出现正面朝上的几率小于0.5，出现反面朝上的几率大于等于0.5）。下列程序能模拟15个硬币的翻转，并打印出得到正

面朝上或反面朝上的次数。

```
10 T=0
20 H=0
30 FOR N=1 TO 15
40 R=RND(1)
50 IF R<0.5 THEN 90
60 PRINT "TAIL"
70 T=T+1
80 GOTO 110
90 PRINT "HEAD"
100 H=H+1
110 NEXT N
120 PRINT "NUMBER OF TAILS=", T
130 PRINT "NUMBER OF HEADS=", H
140 END
```

JRUN
 HEAD
 HEAD
 TAIL
 HEAD
 HEAD
 HEAD
 TAIL
 HEAD
 TAIL
 TAIL
 TAIL
 HEAD
 HEAD
 HEAD
 TAIL

NUMBER OF TAILS=6

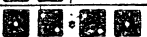
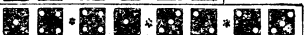
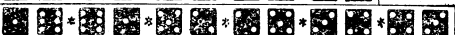
NUMBER OF HEADS=9

7 或 11

下图表示了掷 2 颗骰子可能出现的所有组合方式。仔细观察这个图，我们可发现有 6 种组合方式能得到“7”点，有 2 种组合方式能得到“11”点。下述程序模拟了一对骰子掷 1000 次所



可能出现的各种组合方式，并打印出这1000次中出现“7”点或“11”点的次数所占的百分比。运算结果告诉我们，出现“7”点或“11”点的总几率为16.5%，或者说1000次中有165次。

One Way	2		1/36
Two Ways	3		2/36
Three Ways	4		3/36
Four Ways	5		4/36
Five Ways	6		5/36
Six Ways	7		6/36
Five Ways	8		5/36
Four Ways	9		4/36
Three Ways	10		3/36
Two Ways	11		2/36
One Way	12		1/36

```

10 S=0
20 FOR T=1 TO 1000
30 D= INT(7*RND(1))+INT(7*RND(1))
40 IF (D-7)*(D-11)<>0 THEN 60
50 S=S+1
60 NEXT T
70 PRINT "PERCENTAGE OF ROLLS"
80 PRINT "WHICH ARE 7 OR 11 IS", S/10
90 END

```

]RUN

PERCENTAGE OF ROLLS

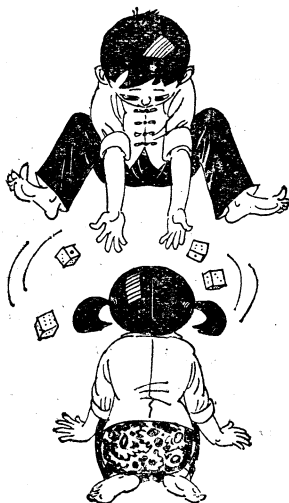
WHICH ARE 7 OR 11 IS 16.5

非六点的几率

概率论中的许多问题，我们都能用微电脑模拟来加以验证。

例如，有这样一个问题：掷四颗骰子，对于任何一颗骰子都不出现六点的几率是什么？理论计算表明，四颗骰子全部非六点的几率是0.48225。因为对一颗骰子来说，不出现六点的几率是 $5/6$ ，那么四颗完全相同的骰子全不出现六点的几率为：

$$\frac{5}{6} \times \frac{5}{6} \times \frac{5}{6} \times \frac{5}{6} = \frac{625}{1296} \\ = 0.48225$$



下列是用BASIC语言编写的程序，它能模拟四颗骰子掷1000次，最后得出全部非六点的几率。

```
10 REM NO SIXES
15 L=0
20 FOR X=1 TO 1000
25 D1= INT(7*(RND(1)))
30 D2= INT(7*(RND(1)))
35 D3= INT(7*(RND(1)))
40 D4= INT(7*(RND(1)))
45 IF D1=6 THEN 70
50 IF D2=6 THEN 70
```

```

55 IF D3=6 THEN 70
60 IF D4=6 THEN 70
65 GOTO 75
70 L=L+1
75 NEXT X
80 PRINT L, TAB(7); "TOSSES HADAT
    LEAST ONE SIX"
85 PRINT 1000-L, TAB(7); "TOSSES HAD
    NO SIXES"
90 PRINT (1000-L)/1000, TAB(7); "IS THE
    EXPERIMENTAL"
95 PRINT "PROBABILLITY THAT NO SIX
    SHOW ON"
100 PRINT "THE ROLL OF FOUR DICE"
105 END

```

JRUN

```

383 TOSSES HAD AT LEAST ONE SIX
617 TOSSES HAD NO SIXES
.617 IS THE EXPERIMENTAL
PROBABILLITY THAT NO SIX SHOW ON
THE ROLL OF FOUR DICE

```

掷 骰 子

下述程序模拟一对骰子的滚动，并输出25次滚动中每次翻出的点数。如将这个程序放进微电脑的存贮器内，还可模拟骰子作100次滚动。

```

10 INPUT "N=", N
20 PRINT "FIRST
   DIE ", "SECOND
   DIE", " TOTAL"
30 FOR A=1 TO N
40 D1= INT(7*
   RND(1))
50 D2= INT(7*
   RND(1))
60 T=D1+D2
70 PRINT D1, D2, T
80 NEXT A
90 END

```



]RUN

N=25

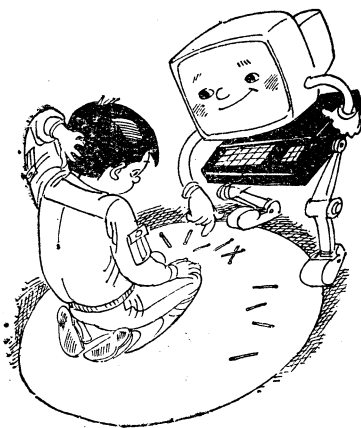
FIRST DIE	SECOND DIE	TOTAL
0	6	6
4	6	10
5	2	7
5	5	10
0	5	5
5	0	5
1	5	6
6	5	11
1	3	4
3	1	4
0	0	0

4
4
0
1
4
4
3
5
2
1
4
3
5
1

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

9
7
1
2
7
9
4
7
4
1
10
6
9
5

取 火 柴



“取火柴”是一个简单的游戏，很多人小时候也许玩过。现在请你和电脑来做这游戏。假如有 N 根火柴，你和电脑各为一方，双方轮流取。规定每次取的火柴根数不能超过 K 根，谁要是取了最后一根火柴，谁就输了。假如不动脑筋的话，你肯定会输给微电脑。不信的

话，你可以试试看。

下面，我们简单地分析一下，电脑是怎样和你对阵的。

首先，你必须指出总的火柴根数，你每次所取的火柴根数的最大值 K ，以及第一次你取的根数 X ，当然应满足： $N \geq K \geq X \geq 1$ 。电脑会算出在你第一次取走火柴后余下的根数，如果这时 $N=0$ ，那表示你已将最后一根火柴取走了，而输给了微电脑，电脑打印出“I WIN”（我胜了）；如果这时 $N>0$ ，电脑就要“考虑”选择什么数目，才能迫使你取最后一根火柴。它选择的数 $Y=(N-1)-\text{INT}((N-1)/(K+1)) \times (K+1)$ ，即 $(N-1)$ 除以 $(K+1)$ 所得的余数。这个公式的作用是：在最后余下 N 根的时候，电脑取走 $(N-1)$ 根，便把最后一根留给你。

以下是这个程序和流程图。玩了一局之后，继续想玩，只要从键盘上输入“1”；否则，输入“0”。如要想战胜电脑，就必须认真研究一下电脑所取火柴根数的规律，并以同样的方法对付它。既然程序是人设计的，人完全能够战胜电脑。请你鼓起勇气，试试看吧：

```
10 PRINT TAB(15) "MATCHES NUMBER"
20 PRINT : PRINT
30 PRINT "TYPE THE INITIAL MATCHES
   NUMBER YOU CHOOSE",
40 INPUT N
50 INPUT K
70 PRINT
80 IF K>N THEN 50
90 PRINT "YOUR MOVE IS",
100 INPUT X
```

```

110 IF X<1 THEN 220
120 IF X>K THEN 200
130 REM TEST FOR WIN
140 N=N-X
150 IF N>0 THEN 220
160 PRINT
170 PRINT "***** I WIN *****
      **"
180 PRINT
190 GOTO 370
200 PRINT "ILLEGAL MOVE"
210 GOTO 90
220 REM COMPUTER MOVE
230 Q= INT((N-1)/(K+1))
240 Y=N-1-Q*(K+1)
250 IF Y<>0 THEN 270
260 Y=1
270 N=N-Y
280 PRINT "MY MOVE IS ", Y
290 IF N=0 THEN 330
300 PRINT "MATCHES REMAING IS ", N
310 PRINT
320 GOTO 90
330 PRINT
340 REM IMPLEMENT LATER
350 PRINT "-----BEGINNER IS LUCK
      YOU BEAT ME -----"
360 PRINT

```

```

370 PRINT "IF P=1 TO PLAY AGAIN, IF
      P=0 OTHERWISE",
380 INPUT P
390 PRINT
400 IF P=1 THEN 20
410 PRINT
420 PRINT
430 END

```

JRUN (以下是这项游戏的实例)

MATCHES NUMBER

TYPE THE INITIAL MATCHES NUMBER YOU
CHOOSE? 187

(请你选定最初火柴根数。输入187)

? 34

(选定每次取火柴的最大根数。输入34)

YOUR MOVE IS ? 17

(你取走了多少根? 17根)

MY MOVE IS 29

(电脑取走 29根)

MATCHES REMAING IS 141

(剩下火柴 141根)

YOUR MOVE IS ? 33

(你取走了多少根? 33根)

MY MOVE IS 2

(电脑取走 2根)

MATCHES REMAING IS 106

(剩下火柴 106根)

YOUR MOVE IS ? 21

(你取走了多少根? 21根)

MY MOVE IS 14

(电脑取走 14根)

MATCHES REMAING IS 71

(剩下火柴 71根)

YOUR MOVE IS ? 23

(你取走了多少根? 23根)

MY MOVE IS 12

(电脑取走 12根)

MATCHES REMAING IS 36

(剩下火柴 36根)

YOUR MOVE IS ? 31

(你取走了多少根? 31根)

MY MOVE IS 4

(电脑取走 4根)

MATCHES REMAING IS 1

(剩下火柴 1根)

YOUR MOVE IS ? 1

(你取走了多少根? 1根)

***** I WIN *****

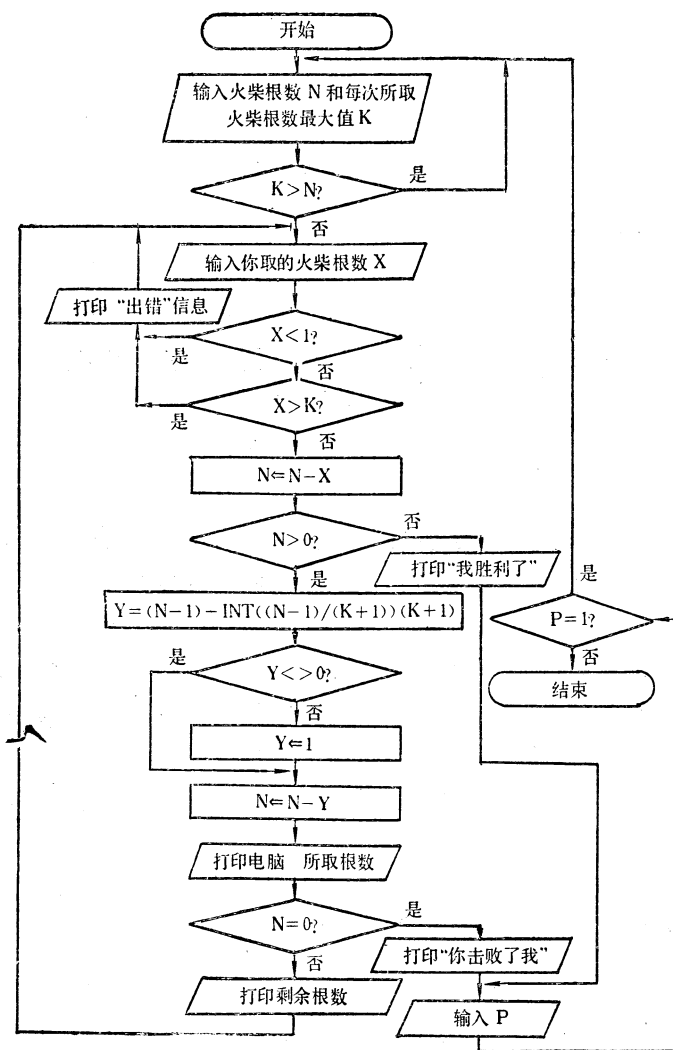
(电脑胜利了)

IF P=1 TO PLAY AGAIN, IF P=0

OTHERWISE

? 1

(如果你想继续玩, 输入1, 否则输入0)



打 靶



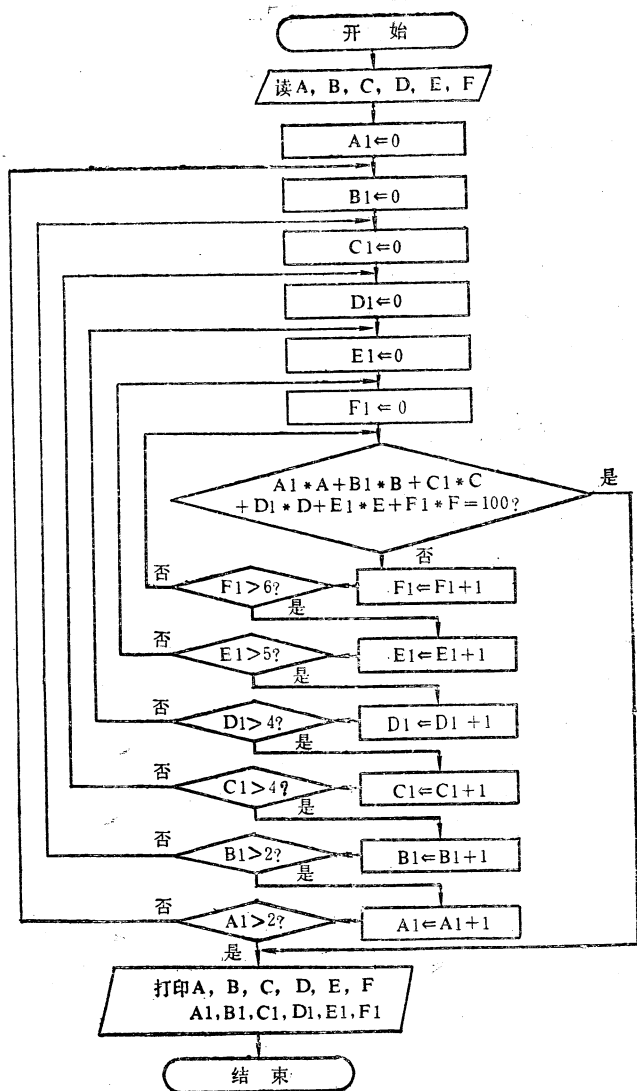
一名射手对着如右图所示的靶子练习射击。靶环上的数字很奇怪，毫无规律。但他连发 6 枪，总环数恰巧是 100。请问他命中了几环？

假如我们设 A 为 40 环；B 为 39 环；C 为 24 环；D 为 23 环；E 为 17 环；F 为 16 环。考虑到射手连发 6 枪，总环数为 100，因此，他打中 40 环至多为 2 次；打中 39 环也至多为 2 次；打中 24 环至多为 4 次；打中 23 环也至多为 4 次；打中 17 环至多为 5 次；而打中 16 环至多为 6 次。根据这样的分析。我们可以通过多次循环，进行搜索，找出满足：

$$A1 \times A + B1 \times B + C1 \times C + D1 \times D + E1 \times E + F1 \times F = 100$$

的各个 A1、B1、C1、D1、E1、F1 之值（即命中上述各环的对应次数）。

下面是解这个问题的流程图与用 BASIC 语言写的程序。



```

10 REM BULLSEYE
15 READ A, B, C, D, E, F
20 PRINT A, B, C, D, E, F
25 PRINT
30 FOR A1=0 TO 2
35 FOR B1=0 TO 2
40 FOR C1=0 TO 4
45 FOR D1=0 TO 4
50 FOR E1=0 TO 5
55 FOR F1=0 TO 6
60 IF A1*A+B1*B+C1*C+D1*D+E1*E
   +F1*F=100 THEN 95
65 NEXT F1
70 NEXT E1
75 NEXT D1
80 NEXT C1
85 NEXT B1
90 NEXT A1
95 PRINT A1, B1, C1, D1, E1, F1
100 DATA 40, 39, 24, 23, 17, 16
105 END

```

]RUN

40	39	24
23	17	16
0	0	0
0	4	2

巧 分 牛 奶

有三个大小不等的容器，其中最大的一个装满了牛奶，而两个小的却空着。现在请你只利用这三个容器分出你所想要的份数。怎么分法？请微机来帮助我们来分析其中某些特殊情况，从而找出所有可能的分法以及相应的步骤。



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

以下是解这个问题的 BASIC 程序：

```
5 INPUT A1, B1 C1
10 IF A1 >=B1 OR B1 >=C1 OR A1+B1<>
   C1 THEN 5
20 A=0:B=0:C=C1
30 C=C-A1
40 A=A1
50 PRINT "A="; A, "B="; B, "C="; C
60 IF B1 <=(B+A) THEN 110
70 B=B+A
80 A=0
```

```

90 PRINT "A="; A, "B="; B, "C="; C
100 GOTO 30
110 A=A-(B1-B)
120 B=B1
130 PRINT "A="; A, "B="; B, "C="; C
140 IF A=0 GOTO 190
150 C=C+B
160 B=0
170 PRINT "A="; A, "B="; B, "C="; C
180 GOTO 70
190 END

```

JRUN

? 7

? ? 8

? ? 15

A=7	B=0	C=8
A=0	B=7	C=8
A=7	B=7	C=1
A=6	B=8	C=1
A=6	B=0	C=9
A=0	B=6	C=9
A=7	B=6	C=2
A=5	B=8	C=2
A=5	B=0	C=10
A=0	B=5	C=10
A=7	B=5	C=3
A=4	B=8	C=3

$A=4$	$B=0$	$C=11$
$A=0$	$B=4$	$C=11$
$A=7$	$B=4$	$C=4$
$A=3$	$B=8$	$C=4$
$A=3$	$B=0$	$C=12$
$A=0$	$B=3$	$C=12$
$A=7$	$B=3$	$C=5$
$A=2$	$B=8$	$C=5$
$A=2$	$B=0$	$C=13$
$A=0$	$B=2$	$C=13$
$A=7$	$B=2$	$C=6$
$A=1$	$B=8$	$C=6$
$A=1$	$B=0$	$C=14$
$A=0$	$B=1$	$C=14$
$A=7$	$B=1$	$C=7$
$A=0$	$B=8$	$C=7$

钟面上的算术

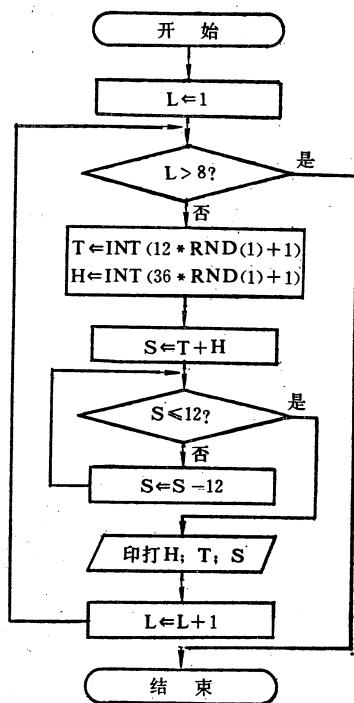
在钟面上，凡相差12小时的倍数的任意两个时刻，长、短针都指在相同的位置上。让我们用一只时钟来显示和说明时钟算术是怎么回事。“ $2+11$ ”表示时针以2点钟开始，



然后顺时针方向移动11个小时，指向1点钟，这样就得出：

$$2 + 11 = 1$$

下面的程序能给出从任意钟点（1~12）开始，加上任意的钟点数（实际上可以是任意范围，但本程序只允许从1~36）而得出的钟点数。



```
10 FOR L=1 TO 8
20 LET T=INT(12×RND(1)+1)
30 LET H=INT(36×RND(1)+1)
40 LET S=T+H
```



```

50 IF S <=12 THEN 80
60 LET S=S-12
70 GOTO 50
80 PRINT H, TAB(6), "HOURS FRM", TAB(20),
  T, TAB(24), "O'CLO CK"
90 PRINT "IT WILL BE", TAB(15);S;TAB(20);
  "O'CLOCK"
100 PRINT
110 NEXT L
120 END

```

JRUN

```

32 HOURS FROM 6 O'CLOCK
IT WILL BE 2 O'CLOCK

```

```

5 HOURS FROM 5 O'CLOCK
IT WILL BE 10 O'CLOCK

```

```

31 HOURS FROM 4 O'CLOCK
IT WILL BE 11 O'LOCK

```

```

30 HOURS FROM 2 O'LOCK
IT WILL BE 8 O'CLOCK

```

```

33 HOURS FROM 7 O'CLOCK
IT WILL BE 4 O'CLOCK

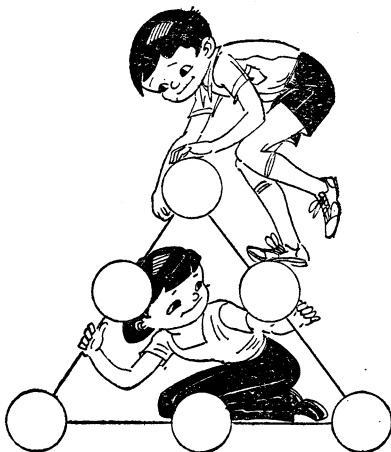
```

```

36 HOURS FROM 2 O'CLOCK
IT WILL BE 2 O'CLOCK

```

每边数字之和都相等的三角形



试编一个程序，完全由微电脑来完成，将 1、2、3、4、5、6 六个数字填入右边的各个圆圈中，使三角形每一条边上的三个数的和相等。但须注意圈中不能有重复的数字。输出格式：只要打印出三组数即可（三组数不必重排成三角形）。

以下是程序：

```

20 FOR A=3 TO 6
30 FOR B=1 TO 6
40 IF A=B THEN 180
50 FOR C=2 TO A-1
60 IF A=<C OR B=C THEN 170
70 FOR D=1 TO 6
80 IF A=D OR B=D OR C=D THEN 160
90 FOR E=1 TO C-1
100 IF A=E OR B=E OR C=<E OR D=E
    THEN 150
110 F=21-A-B-C-D-E
120 IF A+B<>D+E OR C+D<>F+A OR

```

F+E C THEN 150

120 PRINT A, " ", B, " ", C, C, " ", D, "
", E, E, " ", F, " ", A

150 NEXT E

160 NEXT D

170 NEXT C

180 NEXT B

190 NEXT A

200 END

JRUN

342

261

153

523

361

145

614

452

236

615

534

426

帕斯卡三角形

十七世纪法国著名的哲学家和数学家布莱斯·帕斯卡对数学游戏很感兴趣。这这兴趣使他发现了掷硬币时正、反面机遇的一个重要规律。他的发现能通过一个由数字构成的三角形来描述（也称杨辉三角形），它能方便地显示



在给定抛掷次数的情况下获得正、反两面结果的情况。

下列程序能打印出一个帕斯卡三角形：

```
10 DIM A(10, 10)
20 A(0,1)=1
30 FOR I=1 TO 10
40 FOR J=1 TO I
50 A(I, J)=A(I-1, J-1)+A(I-1, J)
60 NEXT J
70 NEXT I
80 FOR I=1 TO 10
90 FOR J=1 TO I
100 PRINT TAB(22-2*I+4*(J-1)); A(I, J),
110 NEXT J
120 PRINT
130 NEXT I
140 END
```

]RUN

```

      1
    1 1
  1 2 1
1 3 3 1
  1 4 6 4 1
    1 5 10 10 5 1
      1 6 15 20 15 6 1
        1 7 21 35 35 21 7 1
          1 8 28 56 70 56 28 8 1
            1 9 36 84 126 126 84 36 9 1
```

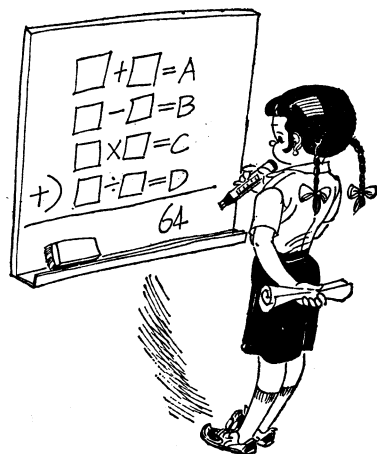
从以上三角形中可以看出，如果你掷一个硬币，那么正面向上的机会是每两次中有1次。如果你掷两个硬币，那

么得到两个正面向上的机会每 4 次中有 1 次；若要一正一反，则是每 4 次中有 2 次；若要 2 个反面同时出现，则也是每 4 次中有 1 次。如果你掷三个硬币，那么其机遇是：都正面向上的为 $1/8$ ，两正一反的为 $3/8$ 。都是反面的亦为 $1/8$ 。

巧 填 数 字

在下面算式中，
方框内只能填从 1 ~ 9
中的任意一个数，其结
果必须使运算式成立，
那么这个数是多少呢？

$$\begin{array}{rcl} \square + \square & = & A \\ \square - \square & = & B \\ \square \times \square & = & C \\ +) \square \div \square & = & D \\ \hline & & 64 \end{array}$$



我们可以设这个数为 I ，由于 I 不会小于 5，所以利用循环语句，使 I 从 5 到 9，步长取为 1，判断 I 是否满足算式的要求，如满足就退出循环，打印出 I 值。

以下是流程图和程序：

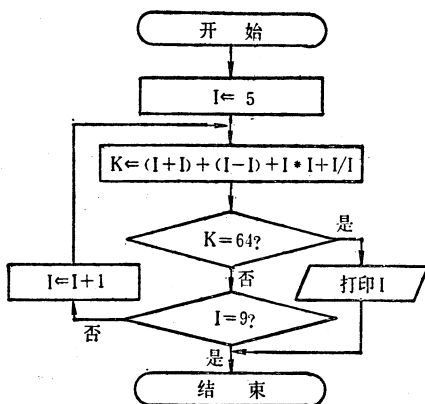
```
10 FOR I=5 TO 9
20 K=(I+I)+(I-I)+I*I+I/I
30 IF K=64 THEN 50
40 NEXT I
```

50 PRINT I

60 END

]RUN

7



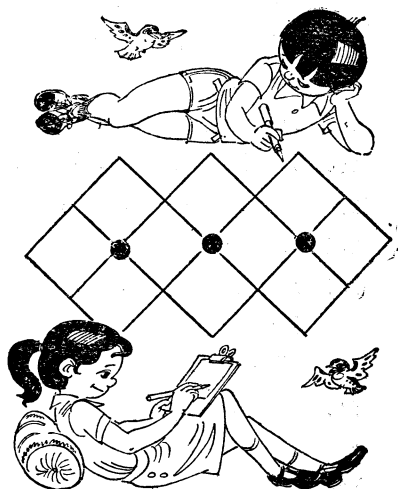
如果我们将上述算式的书写形式改变一下，可以写成：
 $(\square + \square) + (\square - \square) + \square \times \square + \square \div \square = 64$ ，该算式其实是求方程 $x^2 + 2x + 1 = 64$ 的根。请读者根据这个数学模型，重新编制程序，求得 $\square$ 中应填上什么数字。

填 方 格

把 1~10 十个整数填入方格，使每个黑点子周围的四格里的数字和相等。

我们可以这样来考虑：如果在三个黑点之间的两个正方形里填进了 1~10 之内的任意两个数 A 和 B，那么不管余下

的数怎么填，在环绕黑点的四个数相加时，A和B都必须参与两次。根据题意，有三组“四数之和”，并且和相等，因此，从1连加到10，再加上A和B，其和必能被3整除。可见，必须先确定A和B（可通过设置两重循环求得）。考虑到图形的对称性，并有效地排除重复，我们进一步设B比A大。这样一来计算速度就大大加快了。



下面就是填方格游戏的程序。

```

10 DIM P(10), Q(10)
20 FOR A=1 TO 9: FOR B=A+1 TO 10
30 IF (A+B+55)/3 <> INT((A+B+55)/3)
    THEN 70
40 FOR I=1 TO 10: P(I)=I: NEXT I
50 Q(4)=A: Q(7)=B: P(A)=0: P(B)=0
60 Z=Q(4): GOSUB 80
70 NEXT B, A: END
80 FOR S=1 TO 8
90 FOR R=S+1 TO 9
100 FOR T=R+1 TO 10
110 IF P(S)=0 THEN 150
120 IF P(R)=0 THEN 150

```

```

130 IF P(T)=0 THEN 150
140 IF P(S)+P(R)+P(T)=(A+B+55)/3-Z THEN
    GOUSB 240
150 IF S<>8 THEN 200
160 IF R<>9 THEN 200
170 IF T<>10 THEN 200
180 IF Z=Q(4) THEN 200
190 P(F1)=F1; P(F2)=F2; P(F3)=F3; S=F1; R=
    F2; T=F3; Z=Q(4)
200 NEXT T
210 NEXT R
220 NEXT S
230 RETURN
240 IF Z=Q(4) THEN 350
250 Q(8)=P(S); Q(9)=P(R); Q(10)=P(T); P(S)=0;
    P(R)=0; P(T)=0; N=4
260 FOR J=1 TO 10; IF P(J)=0 THEN 280
270 N=1+N; Q(N)=P(J); P(J)=0
280 NEXT J
290 PRINT "<", O+1; ">"
300 FOR G=1 TO 7 STEP 3
310 FOR H=G TO G+3
320 PRINT Q(H); " "; : NEXT H
330 PRINT "; "; : NEXT G; PRINT; O=O+1
340 FOR C=1 TO 10; P(C)=C; NEXT; P(A)=
    0; P(B)=0; S=F1; R=F2; T=F3; Z=Q(4);
    GOTO 360
350 Q(1)=P(S); Q(2)=P(R); Q(3)=P(T); P(S)=0

```


P(T)=0; P(R)=0; Z=Q(7); F1=S; F2=R; F3=
T; S=1; R=2; T=3

360 RETURN

JRUN

<1> 2 7 10 1; 1 6 9 4; 4 3 5 8;

<2> 2 8 9 1; 1 5 10 4; 4 3 6 7;

<3> 3 6 10 1; 1 7 8 4; 4 2 5 9;

<4> 3 7 9 1; 1 5 10 4; 4 2 6 8;

<5> 2 8 10 1; 1 4 9 7; 7 3 5 6;

<45> 3 6 8 7; 7 2 5 10; 10 1 4 9;

<46> 4 5 8 7; 7 1 6 10; 10 2 3 9;

<47> 1 5 10 8; 8 3 4 9; 9 2 6 7;

<48> 2 4 10 8; 8 1 6 9; 9 3 5 7;

<49> 3 6 7 8; 8 2 5 9; 9 1 4 10;

<50> 4 5 7 8; 8 1 6 9; 9 2 3 10;

螺旋方阵

有个 5×5 的方阵（见右图），
写着 1~25 这二十五个自然数，并且
从小到大排列，呈现从外到内的“螺
旋”形式。

试编制一个程序，在电脑指挥下
打印出这个螺旋方阵。当然，不允许
简单地采用字符串原样照打语句。

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

并进一步要求修改程序，使电脑在输入自然数 N 后，打
印出 $2N+1$ 见方的螺旋方阵。

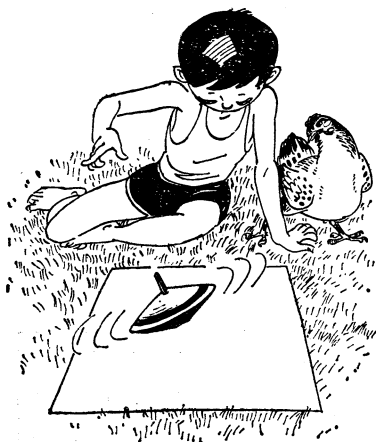
先根据输入的N，决定方阵的长与宽 ($M=2N+1$)，开数组 A(M, M)，并在方阵的中央填上 $M \times M$ 。

以 $N=2$ 为例，方阵长宽为 5 (20句)，开数组 A (5, 5) (30句)，并在中央填上 25 (40、50句)。

	↑	↑		3	→
		3	11	11	
5			25	2	4
		2	→	↓	
←		4			↓

然后，从中央出发，先向上走一格，向右走一格；向下走两格，向左走两格；向上走三格，向右走三格……每走一格填的数都少掉 1 (见左上图)。走步规律是 1、1、2、

2、3、3、…… (60句增加步数)，方向依次为上、右、下、左，反复循环 (70句改符号，与 90、120 句一起完成这些功能)。直到走到右上角 (这时所填的数正好减到 1)、(140、150句)。



当 25 个数在数组中填好后，用一个双重循

环把它们一起打印出来 (170~200句)。

下面就是打印螺旋方阵的程序。当 $N=2$ 时，打印出来的是 1~25 的螺旋方阵；当 $N=4$ 时，打印出来的是 1~81 的螺旋方阵。

```
10 INPUT N; N=N+1
```

```

20 M=2*N+1
30 DIM A(M, M)
40 P=M*M
50 X=N; Y=N; J=1; A(N, N)=P
60 K=K+1
70 J=(-1)*J
80 FOR L=1 TO K
90 Y=Y+J; GOUSUB 140
100 NEXT L
110 FOR L=1 TO K
120 X=X-J; GOSUB 140
130 NEXT L; GOTO 60
140 I=I+1
150 IF I > P THEN 170
160 A(X, Y)=P-1; RETURN
170 FOR I=1 TO M
180 FOR J=1 TO M
190 PRINT TAB(4*J); A(J, I);
200 NEXT J; PRINT; NEXT I
210 END

```

]RUN

? 2

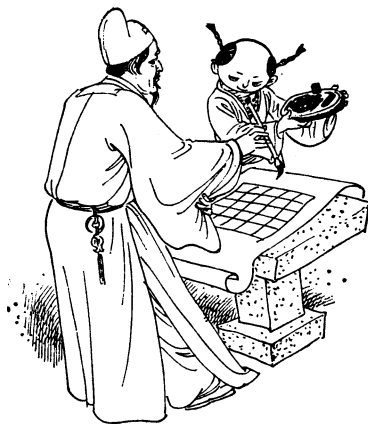
```

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

```

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

奇妙的幻方



幻方是指一类由数字组成的方阵，其每一列、每一行或每一对角线上各个数的和都相等。幻方最早出现在中国，据说早在南宋时期，著名的数学家杨辉就曾列出过10以内的多阶幻方。

最普通的幻方是由 $1 \sim n^2$ 个自然数组成的。例如：4阶幻方是由 $1 \sim 16$ 的自然数

组成。幻方每行、每列或每一对角线上各数之和可由一个简单的公式算出，即： $S = \frac{1}{2}n(n^2 + 1)$ 。根据这个公式，3阶幻

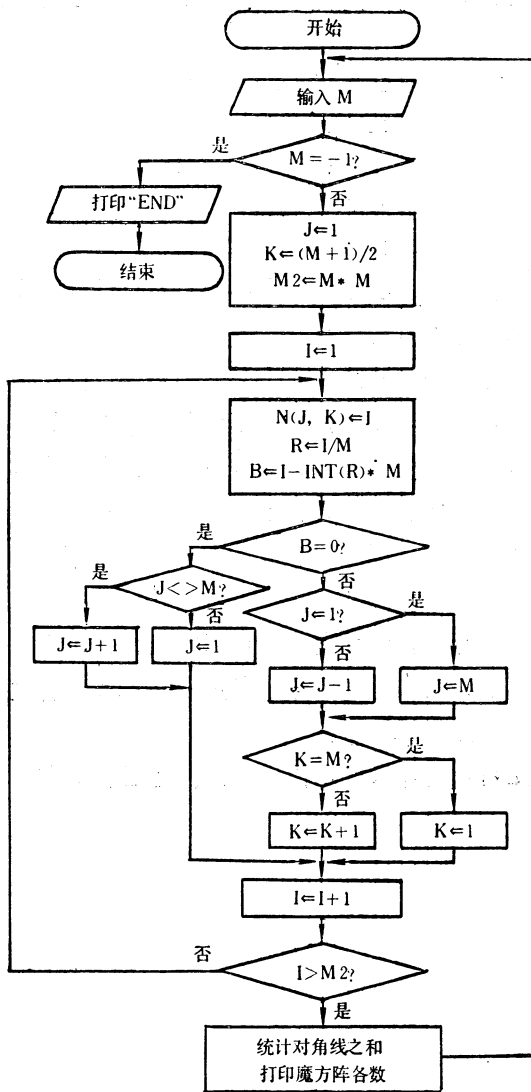
方的和为15，4阶幻方为34……。下面是一个5阶幻方的实例：

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

从以上例子可以看出，幻方中各个数字的排列有一定的规律：1的位置总是居于第一行的正中，以后按数字的大小顺序，每一个数的位置是上一个数的列数加1，行数减1。如“7”在第2行第3列，则“8”必在第1行第4列。如果上一个数在首行，则紧接着的数在末行；上一个数如在最后一列，那么紧接着的数必在首列。如某数按规律应该放的位置已经被占领了，则应将它放在上一个数的下一行同一列上。

编制一个程序，利用电脑就可以方便地打印出任意阶幻方及其每行、每列或每一对角线各数之和(SUM)，操作时只要事先将阶数输入即可。

下面是可以打印20阶以内幻方的流程图和程序：



```

10 DIM N(20, 20)
20 INPUT M
30 IF M=-1 THEN 900
40 LET J=1
50 LET K=(M+1)/2
60 LET M2=M*M
70 FOR I=1 TO M2
80 LET N(J, K)=I
90 LET R=I/M
100 LET B=I-INT (R)*M
110 IF B=0 THEN 210
120 IF J=1 THEN 150
130 LET J=J-1
140 GOTO 160
150 LET J=M
160 IF K=M THEN 190
170 LET K=K+1
180 GOTO 250
190 LET K=1
200 GOTO 250
210 IF J<>M THEN 240
220 LET J=1
230 GOTO 250
240 LET J=J+1
250 NEXT I
265 LET T=0
270 FOR I=1 TO M
275 LET T=T+N(I, I)

```

```

280 FOR J=1 TO M
290 PRINT TAB(J*4); N(I, J);
300 NEXT J
310 PRINT
320 NENT I
325 PRINT
330 PRINT TAB(4); "SUM=", T
335 PRINT
340 GOTO 20
900 PRINT "END"
999 END

```

]RUN

? 5

```

17 24 1 8 15
23 5 7 14 16
4 6 13 20 22
10 12 19 21 3
11 18 23 2 9
SUM=65

```

从任意数开始的幻方

上一例中的幻方是从数字 1 开始的，然而，幻方也能从任意数字开始，下面就是一个以“27”开始的 3 阶幻方：

```

34 27 32
29 31 33
30 35 28

```


这个幻方每行、每列及每一对角线上各数字之和为93，一般可由下式来计算：

$$(N^3+N)/2+N(P-1)=(3^3+3)/2+3(27-1)=93$$

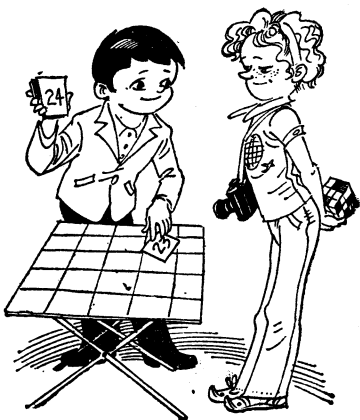
这里 N 是阶数，而 P 是开始的数。

下列程序能产生一个首数为任意整数的多阶（小于25阶）幻方。

```

10 DIM M(25, 25)
20 PRINT "TYPE SIZE OF SQUARE";
30 INPUT N
40 PRINT "TYPE STARTING NUMBER";
50 INPUT Y
55 PRINT
60 S=Y
70 K=1
80 I=1
90 J=(N+1)/2
100 M(I, J)=S
110 S=S+1
120 IF S>N*N+Y-1 THEN 260
130 IF K<N THEN 170
140 K=1
150 I=I+1

```



```

160 GOTO 100
170 I=I-1
180 J=J+1
190 K=K+1
200 IF I<>0 THEN 230
210 I=N
220 GOTO 100
230 IF J<=N THEN 100
240 J=1
250 GOTO 100
260 FOR I=1 TO N
270 FOR J=1 TO N
280 PRINT " ", M(I, J),
290 NEXT J
300 PRINT
310 PRINT
320 PRINT
330 NEXT I
340 END

```

JRUN

TYPE SIZE OF SQUARE ? 5
TYPE STARTING NUMBER ? 12

```

28 35 12 19 26
34 16 18 25 27
15 17 24 31 33
21 23 30 32 14
22 29 36 13 20

```

]RUN

TYPE SIZE OF SQUARE ? 7

TYPE STARTING NUMER ? 9

38 47 56 9 18 27 36

46 55 15 17 26 35 37

54 14 16 25 34 43 45

13 22 24 33 42 44 53

21 23 32 41 50 52 12

29 31 40 49 51 11 20

30 39 48 57 10 19 28

人们陆续发现的幻方还有“双料幻方”、“同心幻方”“分块幻方”、“质数幻方”、“马步幻方”等等，真是稀奇古怪。有人甚至将幻方喻为数学里的“百慕大三角”。

让我们以微电脑为工具，去探索幻方的奥秘吧！

血 型 检 索

在上一个世纪，人们对血液的认识还很肤浅。经常发生这样的事，一个病人在接受输血后不久就死去了；而有的时候，输血倒能挽救患者的生命。

到底是怎么回事呢？这个谜直到1900年才由生物学家卡尔·兰登斯汀纳揭开。原来，人类的血型有四种不



同的类型：即 O、A、B、AB 型，输血的时候必须遵循一定的规律才不会出问题。

通过进一步的研究又发现，知道了父母亲的血型，就可以分析子女的血型，但子女的血型往往有几种可能。

父母的血型	子女可能的血型
O×O	O
O×A	O、A
O×B	O、B
O×AB	A、B
A×A	O、A
A×B	O、A、B、AB
A×AB	A、B、AB
B×B	O、B
B×AB	A、B、AB
AB×AB	A、B、AB

下面是一个关于血型检索的电脑程序。只要把父亲和母亲的血型告诉电脑，它就能自动打印出子女可能的血型。程序中巧妙地运用了二进制数各位的值，即 1、2、4 和 8。以 1 对应 O 型；2 对应 A 型；4 对应 B 型；8 对应 AB 型，使用二进制数后，将父亲与母亲血型对应值相加时，同一个和数只能对应于一种情况。譬如，父母亲血型对应值相加是 5，就只可能是 1 + 4，即 O 型和 B 型。这个和数就是血型检索的根据。当和数大于 9 时，包括了三种情况，即 A 和 AB 型；B 和 AB 型；AB 和 AB 型，而子女可能的血型都为 A、B 和 AB 型，因此，也就可以列为同一情况(见 170 句)。

```

10 INPUT "MOTHER'S BLOOD TYPE: "; C$
20 INPUT "FATHER'S BLOOD TYPE: "; D$
31 IF C$ = "O" THEN X=1; GOTO 41
32 IF C$ = "A" THEN X=2; GOTO 41
33 IF C$ = "B" THEN X=4; GOTO 41
34 X=8
41 IF D$ = "O" THEN Y=1; GOTO 100
42 IF D$ = "A" THEN Y=2; GOTO 100
43 IF D$ = "B" THEN Y=4; GOTO 100
44 Y=8
100 S=X+Y
110 PRINT "CHILDREN'S BLOOD TYPES: ";
120 IF S=2 THEN PRINT "O"; GOTO 200
130 IF S=3 OR S=4 THEN PRINT "O, A";
    GOTO 200
140 IF S=5 OR S=8 THEN PRINT "O, B";
    GOTO 200
150 IF S=6 THEN PRINT "O, A, B, AB";
    GOTO 200
160 IF S=9 THEN PRINT "A, B"; GOTO 200
170 PRINT "A, B, AB"
200 END

```

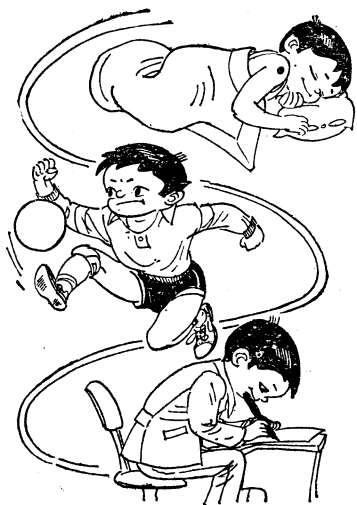
]RUN

MOTHER'S BLOOD TYPE: O

FATHER'S BLOOD TYPE: A

CHILDREN'S BLOOD TYPES: O, A

测定你的生物节律



在日常生活中，几乎每个人都有这样的体验：有时情绪饱满、精神焕发；而在另一些时候则相反。一些学者根据人体内部的周期盛衰性，认为每个人从诞生的那天起直到生命的终止，都存在着周期分别为23天、28天和33天的体力、情绪和智力的变化，组成了一首协调、美妙而又神秘的三重奏。那么，怎样测定自己的生物节律呢？

1. 首先计算从你出生的那天到你打算了解的那天的总天数。

2. 分别用23、28、33来除总天数，所得余数就是那天里三周期所处的位置。

以下是用微电脑计算各人生物节奏的 BASIC 程序。运行结果能打印出三条不同的曲线，分别表示体力、智力和情绪的高涨和低落情况，每条曲线的第一点就是这天在周期变化中的位置。

程序中：A = 出生年份。

B = 所需了解的那天的年份。

C = 出生的月份。

D = 所需了解的那天的月份。

E = 出生日。

F = 所需了解的那天的日期。

```
10 INPUT A, B, C, D, E, F
20 T=(B-A)*365
30 G=INT((B-A)/4)
40 T=T+G
50 Y=0
55 IF C=D THEN 250
60 IF C<D THEN 110
80 C1=D
90 C2=C
100 GOTO 130
110 C1=C
120 C2=D
130 FOR X=C1 TO C2-1
140 IF X=2 THEN 230
150 IF X>=8 THEN 210
160 IF X/2=INT(X/2) THEN 190
170 Y=Y+31
180 GOTO 240
190 Y=Y+30
200 GOTO 240
210 IF X/2=INT(X/2) THEN 190
220 GOTO 170
230 Y=Y+28
240 NEXT X
```

```

250 Y1=F-E
260 IF C>D THEN 290
270 Y2=T+Y+Y1
280 GOTO 300
290 Y2=T-Y+Y1
300 D1=(Y2/23-INT(Y2/23))*23
310 D2=(Y2/28-INT(Y2/28))*28
320 D3=(Y2/33-INT(Y2/33))*33
330 TAB(10); "PHYSICAL"; TAB(30); "WIT",
    TAB(50); "EMOTION"
350 TAB(10); D1; TAB(30); D3; TAB(50); D2
360 I1=23
370 I2=D1
380 GOSUB 500
390 I1=33
400 I2=D3
410 GOSUB 500
420 I1=28
430 I2=D2
440 GOSUB 500
450 END
500 FOR X=6.28318*I2/I1 TO 6.28318*(1+I2/I1)
    STEP 6.28318/I1
510 TAB(30+20*SIN(X)); "*"
520 NEXT X
530 RETURN

```


图 书 管 理

这是一个为“家庭图书馆”统计图书的简易程序。它最高可以存入一百本不同种类的书籍，统计栏目包括书名、作者名称、书价、出版社以及出版年份等。一开始打印出的“1”、“2”字分别代表“输入资料”或“输出资料”。如果输入了一系列资料后，电脑便会打印出是继续输入资料还是把资料提取出。所有的资料也可以只提取某一部分。这个程序简洁、明朗，在重点部分还加上了反显示（INVERSE）及声音。



```
10 HOME ; NORMAL
20 PRINT SPC(5) ; "THE PROGRAM FOR
   STATISTECS BOOKS"
30 PRINT "YOU CAN TYPE 1 TO INPUT
   DATA"
40 PRINT "YOU CAN TYPE 2 TO LIST THE
   DATA"
50 GET Z$ ; IF Z$="1" THEN 80
60 IF Z$<>"2" THEN 10
70 IF Z$="2" THEN 200
```

```

80 INPUT "HOW MANY BOOKS YOU INPUT",
  L
90 FOR F=1 TO L
100 HOME: PRINT "THE", F , "BOOK"
110 INPUT "THE NAME OF THE BOOK" ,
  A$(F)
120 INPUT "THE AUTHOR OF THE BOOK",
  B$(F)
125 INPUT "THE PRICE OF THE BOOK",
  C$(F)
130 INPUT "THE PUBLISHER" , D$(F)
140 INPUT "THE PUBLICATION YEAR" ,
  E$(F)
150 PRINT "DO YOU WANT TO CORR.(Y/N)
  ? " : GET T$
160 IF T$<>"Y" AND T$<>"N"
  THEN 150
170 IF T$="Y" THEN 100
180 NEXT F
190 HOME : INVERSE , PRINT "ALL THE
  BOOKR ARE MEMBERED"
200 PRINT "HOW YOU CAN TYPE THE
  FILE NUMBER TO LIST THE DATA"
210 INPUT I : NORAL
220 HOME : PRINT "THE" , I, "BOOK"
230 PRINT "THE NAME OF THE BOOK" ,
  A$(I)
240 PRINT "THE AUTHOR", B$(I)

```

```

250 PRINT "THE PRICE"; C$(I)
260 PRINT "THE PUBLISHER" ; D$(I)
270 PRINT "THE PUBLISHER YEAR" ;
    E$(I)
280 CALL -198 ; CALL -198
290 FLASH ; PRINT "PUSH ANY KEY TO
    AGAIN"
300 GET N$ ; GOTO 10

```

计 算 星 期 几

小朋友们经常会问：今年“六一”儿童节是星期几？明年国庆节是星期几？后年春节又是星期几？……如果手头没有一本万年历，推算起来是相当费事的。

然而，微电脑却可以很快地告诉你某天是星期几。

以下是有关这个问题的程序（从1750年开始），使用时，只须将“年、月、日”输入即可。

```

1 FOR P=1 TO 10
5 INPUT X, Y, Z
10 PRINT
15 PRINT " ", Z, "-"; Y, "-"; X, "; ";

```



```

17 C=0
18 IF Y=1 THEN 37
20 FOR I=1 TO Y-1
25 READ K
30 C=K+C
35 NEXT I
36 RESTORE
37 C=C+Z
40 IF Y<=2 THEN 70
50 IF X/4=INT(X/4) THEN 65
55 C=C+28
60 GOTO 70
65 C=C+29
70 S=X-1+(X-1)/4-(X-1)/100+(X-1)/400+C
75 M=S/7-INT(S/7)
80 N= INT(M*7)
85 IF N=0 THEN 125
90 IF N=1 THEN 135
95 IF N=2 THEN 145
100 IF N=3 THEN 155
105 IF N=4 THEN 165
110 IF N=5 THEN 175
115 IF N=6 THEN 185
120 NEXT P
122 END
125 PRINT "SUNDAY"
130 GOTO 120
135 PRINT "MONDAY"

```

```

140 GOTO 120
145 PRINT "TUESDAY"
150 GOTO 120
155 PRINT "WEDNESDAY"
160 GOTO 120
165 PRINT "THURSDAY"
170 GOTO 120
175 PRINT "FRIDAY"
180 GOTO 120
185 PRINT "SATURDAY"
190 GOTO 120
195 DATA 31, 0, 31, 30, 31, 30, 31, 31, 30, 31, 30

```

]RUN

? 1984

? ? 1

? ? 4

4-1-1984; WEDNESDAY

? 1985

? ? 1

? ? 1

1-1-1985; TUESDAY

? 1985

? ? 5

? ? 1

1-5-1985; WEDNESDAY

? 1985

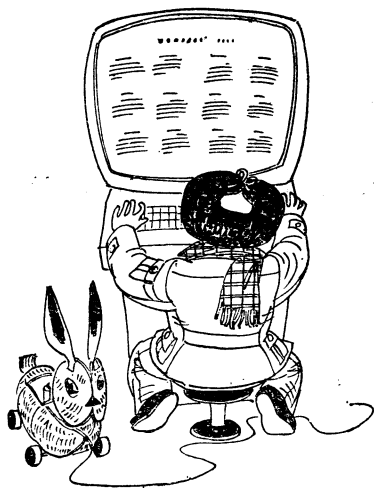
? ? 10

? ? 1

1-10-1985; TUESDAY

打印一张年历

辞旧岁，迎新春。新的一年来临的时候，家家都要换年历。如果用微电脑自己设计、打印一张新的年历，那将是另有一番情趣！



以下就是一个用微电脑打印出任何一年年历的程序。虽然程序比较复杂，但使用却很方便。其中，20句是要你输入需要打印年历的年份，接着启动打印机；70~90句是用来判断输入的年份是否闰年，因为它关系到这一年的二月份是28天还是29天。接着的程序是处理年、月、日、星期的排列位置，然后用170、180、330等句把它们打印出来。第140句是星期的英文简写，第370句是月份的简写，这些都可以根据个人的爱好作适当的改动，如改成用汉语拼音表示等。

须注意这个程序只适合在打印机上打印，它在显示器上显示的画面是混乱的。

```
10 HOHE
```

```
20 INPUT "WHAT IS THE YEAR YOU
```

```
WANT ? "; A; PR#1
```

```

30 W=A-1+INT((A-1)/4)-INT((A-1)/
    100)+INT((A-1)/400)+1
40 DIM MO(12), WE(12), AR(12), MO$(12),
    PA$(12): WE(1)=W-INT(W/7)*7
50 FOR X=1 TO 12: READ MO(X): NEXT
60 FOR X=1 TO 12: READ MO$(X): NEXT
70 IF INT(A/4)*4-A=0 THEN MO(2)=29
80 IF INT(A/100)*100-A=0 THEN MO(2)=28
90 IF INT(A/400)*400-A=0 THEN MO(2)=29
100 FOR X=2 TO 12
110 WE(X)=WE(X-1)+MO(X-1)
120 WE(X)=WE(X)-INT(WE(X)/7)*7: NENT
    X
130 PRINT SPD(32), "CALENDAR OF ", A
140 TI$= "SU MO TU WE TH FR SA",
    FOR X=1 TO 12: AR(X)=1: NEXT
150 FOR X=1 TO 4: BC=22
160 IF X=3 THEN BC=21
170 PRINT SPC(14), MO$((X-1)*3+1), SFC
    (BD), MO$((X-1)*3+2), SPC(22), MO$
    ((X-1)*3+3)
180 PRINT SPC(5)TI$, SPC(5), TI$, SPC(5),
    TI$
190 FOR WE=1 TO 6
200 IF X=1 THEN FOR A=1 TO 3
210 IF X=2 THEN FOR A=4 TO 6
220 IF X=3 THEN FOR A=7 TO 9
230 IF X=4 THEN FOR A=10 TO 12

```

```

240 PP=WE(A)
250 PA$(A)="": IF WE(A) THEN FOR CO=1
    TO WE(A)*3: FA$(A)=PA$(A)+" ":
    NEXT
260 NE$=STR$(AR(A)): DU$="": FOR
    CO=1 TO 3-LEN (NE$): DU$=DU$
    +" ": NEXT: NE$=DU$+NE$
270 IF NE$ =" 0" THEN NE$=" "
280 PA$(A)=PA$(A)+NE$
290 PP=PP+1: IF PP=7 THEN KK=1: AR(A)
    =AR(A)+1: GOSUB 390: WE(A)=0: NEXT
    A: GOTO 320
300 AR(A)=AR(A)+1: IF AR(A)=MO(A)+1 OR
    AR(A)=1 THEN AR(A)=0
310 GOTO 260
320 PA$=" "+PA$(A-3)+" "+PA$(
    (A-2)+" "+PA$(A-1)
330 PRINT PA$: NEXT WE'
340 PRINT " "
350 NEXT X
360 DATA 31, 28, 31, 30, 31, 30, 31, 31, 30, 31,
    30, 31
370 DATA "JAN", "FEB", "MAR", "APR", "MAY",
    "JUNE", "JULY", "AUG", "SEP", "OCT",
    "NOV", "DEC"
380 GOTO 440
390 IF AR(A)=1 THEN AR(A)=0
400 IF AR(A)=MO(A)+1 THEN AR(A)=0

```


410 RETURN

440 POKE 33, 40: PR#0: END

打印变量图表

在统计工作中，人们往往喜欢用图表来说明问题，因为这种方法可以显示各种变量的关系，使人一目了然。

以下就是一个可以显示15种变量的程序，当然这个程序比较复杂，但使用起来却很方便。其中第10~15句是输入变量的数目，第60句是输入变量的名称，如学生成绩、工厂的日产量、实验数据等，第140句是要你输入各种变量的具体数字，当所有数据输入完毕后，微电脑



使用高清晰度画面来显示图

表和打印图表，其中X轴表示各变量的名称，Y轴表示各变量的数量。

5 HOME

8 VTAB 12: HTAB 12

10 PRINT "HOW MANY OBJECTS? "

12 VTAB 14: HTAB 12

15 INPUT "(LESS THAN 15)"; A

20 IF A>15 THEN 10

25 FOR X=36864 TO 37005: READ XX;

```

        POKE X, XX:  NEXT X: ROT=1,
        SCALE=1
26  POKE 232, 0: POKE 233, 144
30  DIM OB$(A), AM(A)
40  T=33/A-1
45  HOME
50  FOR X=1 TO A
55  PRINT
60  PRINT "WHAT IS OBJECT ", X, "
        LABELLING? ";
65  PRINT
70  FOR Y=1 TO T
75  PRINT
80  GET A$
90  IF ASC(A$)=13 THEN Y=T: A$="",
        GOTO 120
100 IF ASC(A$)<>8 THEN 120
110 IF Y<>1 THEN Y=Y-2: GOTO 120
115 GOTO 80
120 PRINT A$, : OB$(X)=OB$(X)+A$:
        NEXT Y: PRINT : NEXT X
130 MA=0
140 FOR X=1 TO A: PRINT "WHAT IS THE
        AMOUNT IN OBJECT ", X, : INPUT
        AM(X): IF AM(X)>(MA) THEN MA=
        AM(X)
145 PRINT: PRINT
150 NEXT

```

```

200 K=MA/5; X=1; E=K
210 K= LOG(K)*.434294481; IF K>0 THEN
    Y= INT(K)-1; GOTO 250
220 K=MA/5; Y=1
230 K=K*10; Y=Y+1; IF INT(K)=0 THEN
    230
240 L=MA/5; L= INT(L*10^Y+.5)/(10^Y);
    GOTO 260
250 L=MA/5; L=L/(10^Y); L=INT(L+.5)*
    (10^Y)
260 B=L; X=B
270 HGR : FOR Z=4 TO 0 STEP -1; AA$
    =STR$(B); B=B+X; FOR ZZ=0 TO
    LEN(AA$)-1
271 IF MID$(AA$, ZZ+1, 1) <>"." THEN
    XDRAW ASC (MID$(AA$, ZZ+1, 1))-47
    AT ZZ*7, Z*32; GOTO 275
273 HPLOT ZZ*7+4, Z*32+7
275 NEXT ZZ, Z
277 DR=X
280 X=33/A; X=INT(X)+1; IF X*A>33
    THEN X=X-1
281 HOME : VTAB 21; HTAB 1; PRINT "0";
290 VTAB 21; FOR Z=0 TO A-1; HTAB
    (X*Z+6); PRINT OB$(Z+1); : NEXT
    : PR=7*X; D E=PR; IF PR>39 THEN
    PR=39
300 X=15; HPLOT 32, 0 TO 32, 159; HPLOT

```

```

      TO 254, 159
330  FOR X=1 TO A
340  T1=159-AM(X)/DR*32: IF T1<0 THEN T1
      =0
360  T=(X-1)*DE+32
370  FOR Y=T TO T+PR
380  HPLOT Y, T1 TO Y, 159: NEXT Y, X
390  GOTO 390
1500 DATA 10, 0, 22, 0, 39, 0, 47, 0, 50, 0, 70,
      0, 80, 0, 93, 0, 106, 0, 113, 0, 128, 0
1510 DATA 9, 45, 21, 54, 54, 30, 63, 7, 32, 36,
      36, 73, 50, 30, 23, 54, 0
1520 DATA 73, 62, 14, 54, 54, 111, 5, 0
1530 DATA 17, 5, 40, 173, 246, 23, 30, 23, 45, 45,
      5, 0
1540 DATA 41, 45, 245, 23, 14, 21, 246, 63, 7,
      32, 0
1550 DATA 73, 246, 246, 46, 45, 245, 4, 32, 36, 0
1555 DATA 73, 9, 63, 63, 54, 45, 45, 50, 246, 63,
      7, 32, 6
1560 DATA 73, 57, 191, 54, 54, 14, 45, 5, 32, 28,
      63, 7, 0
1570 DATA 41, 45, 245, 246, 246, 54, 0
1580 DATA 17, 118, 45, 21, 246, 63, 7, 32, 76, 9,
      32, 28, 63, 7, 6
1590 DATA 145, 146, 14, 45, 5, 32, 36, 226, 63,
      23, 118, 45, 5, 0

```

工 资 统 计

某工厂有工人若干，每人工资额又不完全相同。请你帮助设计一个程序，用微电脑来统计全厂工资总额，以及在发放工资时，能按张数最少而又便于分发的要求，来计算各种票面值的人民币，各应有多少张？



以下是用 BASIC 语言写的程序。

其中，Q：工资总额

A1：票面值为10元的张数

B5：票面值为 5 元的张数

B1：票面值为 1 元的张数

C5：票面值为 5 角的张数

C1：票面值为 1 角的张数

D5：票面值为 5 分的张数

D1：票面值为 1 分的张数

N：待输入的职工人数

S：待输入的各人的工资额

10 Q=0

20 A1=0

```

30 B5=0
40 B1=0
50 C5=0
55 C1=0
60 D5=0
70 D1=0
80 A0= INT (A)
90 A1=A1+A0
105 PRINT "NUMBER OF EMPLOYEES",
110 INPUT N
120 FOR I=1 TO N
125 PRINT "VALUE OF SALARY",
130 INPUT S
140 Q=Q+S
210 A0= INT (S/10)
220 A1=A1+A0
310 K=1
320 X=S
330 X0=A0*10
340 GOSUB 710
350 B5=B5+W5
360 B1=B1+W1
410 K=10
420 GOSUB 710
430 C5=C5+W5
440 C1=C1+W1
510 K=100
520 GOSUB 710

```

```

530 D5=D5+W5
540 D1=D1+W1
550 NEXT I
605 PRINT
606 PRINT
608 PRINT "SUM OF SALARY: ", Q
610 PRINT "NUMBER OF 10 YUAN: ", A1,
      "NUMBER OF 5 YUAN: ", B5,
611 PRINT "NUMBER OF 1 YUAN: ", B1
620 PRINT "NUMBER OF 5 JIAO: ", C5,
      "NUMBER OF 1 JIAO: ", C1
630 PRINT "NUMBER OF 5 FEN: ", D5,
      "NUMBER OF 1 FEN: ", D1
700 END
710 W1=0
720 W5=0
730 W=X-X0
740 W0=INT(W/K)
750 IF W0>=5 THEN 780
760 W1=W0
770 GOTO 800
780 W5=W5+1
790 W1=W0-5
800 X=W
810 X0=W0/K
820 RETURN

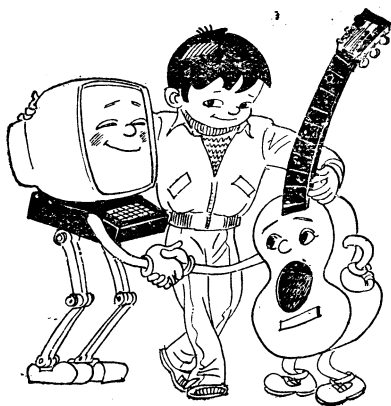
```

]RUN

NUMBER OF EMPLOYEES? 5
VALUE OF SALARY? 56.31
VALUE OF SALARY? 78.97
VALUE OF SALARY? 60.45
VALUE OF SALARY? 55.38
VALUE OF SALARY? 144.39

SUM OF SALARY: 395.5
NUMBER OF 10 YUAN: 37
NUMBER OF 5 JIAO: 1
NUMBER OF 5 FEN: 4
NUMBER OF 5 YUAN: 3
NUMBER OF 1 JIAO: 17
NUMBER OF 1 FEN: 6
NUMBER OF 1 YUAN: 8

让 APPLE 奏乐



这是一个可以产生声音的程序，我们可利用它来设计自己所需要的音乐。程序分两部分：一部分是用机器语言写的发生器程序；另一部分是用 BASIC 语言写的，用以确定歌曲中音阶和拍子的程序。

使用时，首先利用监控程序的指令，输入机器语言：

```
0308-FF FF AD 30 C0 B8
```

```
D0 05
```

```
0310-CE 09 03 F0 09 CA
```

```
D0 F5
```

```
0318-AE 08 03 4C 0A 03
```

```
60
```

然后，输入 BASIC 程序：

```
10 HOME : PRINT : PRINT : PRINT
```

```
20 INPUT "WHAT IS THE VALUE OF 'A'  
(FREQUENCY)? ", A
```

```
30 PRINT : PRINT
```

```
40 INPUT "WHAT IS THE VALUE OF 'B'  
(DURATION)? ", B
```

```
50 POKE 776, A: POKE 777, B: CALL 778
```

```
60 VTAB 14: HTAB 15: PRINT "A="; A
```

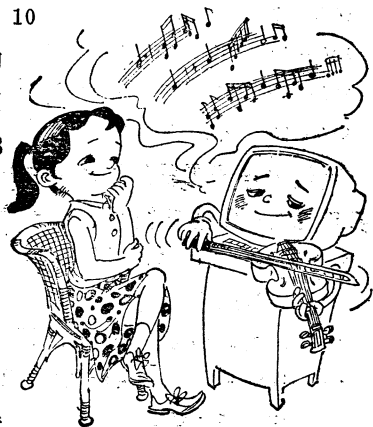
```
70 VTAB 16: HTAB 15: PRINT "B="; B
```

```
80 GET X$: GOTO 10
```

此时，只要输入不同的 A、B 值，就可以听到不同的调子，但须注意，A、B 值不能大于 255。

美妙的电子琴

演奏美妙动听的音乐，
是 APPLE-II 微电脑的特



殊功能,尤其是在玩游戏程序时,用音响配上图案更是引人入胜。遗憾的是仅用 BASIC 语言很难表达这种优美的音乐效果,只能产生单调的“哗哗”之声。然而如果使用一段机器码(如下表所示)情况就不同了。

※300L

300-AD	30	C0	LDA	\$C030
303-AD	20	C0	LDA	\$C020
306-D0	04		BNE	\$030C
308-C6	01		DEC	\$01
30A-F0	08		BEQ	\$0314
30C-CA			DEX	
30D-D0	F7		BNE	\$0306
30F-A6	00		LDX	\$00
311-4C	00	03	JMP	\$0300
314-60			RTS	

※300.317

300-AD	30	C0	AD	20	C0	D0	04
308-C6	01	F0	08	CA	D0	F7	A6
310-00	4C	00	03	60	FF	FF	FF

我们的电子琴程序,就是使用了这样一段机器码,以下就是用 BASIC 语言编写的电子琴程序:

```

10 FOR A=768 TO 788: READ B: POKE A,
   B: NEXT
20 B=32: SPEED=255: TEXT: NOTRACE
30 HOME: HTAB 11: PRINT "MUSIC
   PROGRAM"

```

```

40 HTAB 14: PRINT "1 2 3 4"
50 IF B=32 THEN POKE 1165, 113
60 IF B=80 THEN POKE 1167, 114
70 IF B=128 THEN POKE 1169, 115
80 IF B=255 THEN POKE 1171, 116
90 PRINT: PRINT: GET A$: B$=A$: C=VAL
   (B$)
100 ON C GOSUB 190, 200, 210, 220, 230, 240,
    250, 260
110 IF C>0 AND C<9 THEN 170
120 IF A$=" A " THEN B=32
130 IF A$=" S " THEN B=80
140 IF A$=" D " THEN B=128
150 IF A$=" F " THEN B=255
160 GOTO 30
170 VTAB 12: HTAB 15: PRINT "NOTE: ", C$
180 POKE 0, A: POKE 1, B: CALL 768:
    GOTO 30
190 A=134: C$=" C ": RETURN
200 A=127: C$=" D ": RETURN
210 A=121: C$=" E ": RETURN
220 A=116: C$=" F ": RETURN
230 A=111: C$=" G ": RETURN
240 A=107: C$=" A ": RETURN
250 A=103: C$=" B ": RETURN
260 A=99: C$=" C ": RETURN
270 DATA 173, 48, 192, 173, 32, 192, 208, 4,
    198, 1

```

280 DATA 240, 8, 202, 208, 247, 160, 0, 76, 0, 3, 96

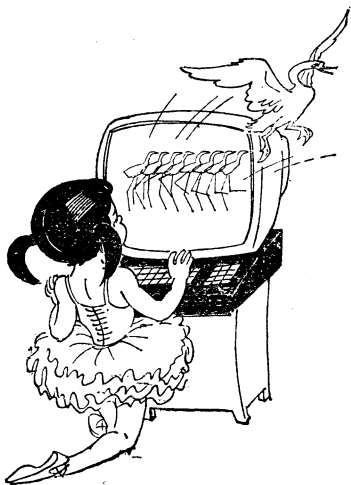
当将全部程序输入微电脑后，“电子琴”就能使用了（那段机器码已由该程序的 READ、DATA 及 POKE 放入适当的地方以供使用）。使程序运行（即打入 RUN）以后，微电脑首先将270行以及280行的 DATA 存入适当的位置，以构成那段发声的机器码，之后将显字速度定为最大（SPEED=255）及全文字显示，然后清洗画面，显示出“MUSIC PROGRAM”的字样。此时，如按下1~8键，便会发出八个基本的音符，分别为：1是Do，2是Rei，3是Me……等，此外，你会发现有“1、2、3、4”等字，而其中一个是在闪烁着的，这是拍子选择，1是最短，4是最长，分别用键 A、S、D、F 来加以控制（A是1，S是2，D是3，F是4），当然闪烁着的一个就是目前的拍子。

如果你要将演奏的音乐录到录音磁带上，只要在发出声音的时候，把录音机的 MIC 接到微电脑背后的 OUTPUT 插孔上即可。

让这个电子琴程序使你成为一位音乐大师吧！

快乐的小天鹅

在 APPLE-Ⅱ 微电脑的显示屏上，用 BASIC 语言来产生一幅活动的画面，那是十分有趣的事。



下面我们提供的高分辨率绘图程序,虽然语句多了一些,但它能产生一幅栩栩如生的图象,快活的小天鹅在显示屏上走动起来了,由远走近,由小变大,图象清晰,神态逼真。不信,您就试试吧!

```
0 HIMEM: 8046
1 INC=2
2 MAX=9
10 HGR : X=140: Y=85
11 S0=1: S1=20: S2=5: R0=0
13 POKE -16302, 0
14 GOTO 2000
50 SCALE=S1: ROT=R1: DRAW 1 AT X, Y:
  ROT=R2: DRAW 1: ROT=R3: SCALE=S2:
  DRAW 1: SCALE=S1: ROT=R4:
  DRAW 1 AT X, Y: ROT=R5: DRAW1:
  ROT=R6: SCALE=S2: DRAW 1
60 SCALE=S0: ROT=R0: DRAW 2 AT
  X, Y: RETURN
100 RESTORE
105 FOR I=1 TO MAX
120 READ R1, R2, R3, R4, R5, R6
130 HCOLOR=3: GOSUB 50
135 FOR D=1 TO PDL(0): NEXT D
145 HCOLOR=0: GOSUB 50
146 X=X+INC
150 NEXT I
155 RESTORE
160 IF X<228 THEN 100
```

```

165 S1=RND(1)*40+10; S2=S1/4; INC=
    S1/8
170 S0=S1/12; IF S0<1 THEN S0=1
175 X=20
180 GOTO 100
200 REM
201 REM
300 REM START OF LEG MOVEMENTS
311 DATA 32, 26, 16, 38, 30, 16
312 DATA 33, 26, 16, 42, 32, 18
313 DATA 33, 26, 16, 44, 33, 20
314 DATA 34, 27, 16, 45, 24, 18
315 DATA 34, 27, 16, 43, 22, 12
316 DATA 35, 28, 16, 38, 18, 13
317 DATA 35, 28, 16, 32, 15, 8
318 DATA 36, 29, 16, 30, 16, 12
319 DATA 36, 29, 16, 32, 20, 12
998 REM
999 REM A ONE VECTOR SHAPE
1000 POKE 232, 100; POKE 233, 31
1010 POKE 8046, 1; POKE 8047, 0; POKE 8048,
    4; POKE 8049, 0; POKE 8050, 4; POKE
    8051, 0
1030 GOTO 100
2000 REM LEG VECTOR AND BODY
2001 REM VECTOR STORED HERE
2005 POKE 232, 110; POKE 233, 31
2010 POKE 8046, 2; POKE 8047, 0; POKE 8048,

```

```

        6: POKE 8049, 0: POKE 8050, 8: POKE
        8051, 0: POKE 8052, 4: POKE 8053, 0
2015 FOR DUM=1 TO 54: READ I: NEXT
        DUM
2020 FOR LOC=8054 TO 8191
2025 READ ASEC, BSEC
2026 IF ASEC=9 OR BSEC=9 THEN 2060
2030 BYTE=(BSEC*8)+ASEC+36
2035 POKE LOC, BYTE
2045 NEXT LOC
2060 POKE LOC, 0: POKE LOC +1, 0: GOTO 100
2100 REM
2101 DATA 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0,
        1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1
2102 DATA 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,
        1, 1, 0, 1, 1, 1, 1, 1
2103 DATA 0, 3, 3, 3, 3, 0, 3, 3, 0, 3, 3, 3, 3,
        2, 3, 2, 3, 2, 2, 2, 2
2104 DATA 3, 2, 2, 3, 2, 2, 3, 2, 3, 3, 2, 3, 3,
        3, 3, 2, 3, 3, 3
2105 DATA 2, 3, 3, 3, 2, 3, 3, 2, 3, 2, 3, 2, 3,
        3, 2, 2, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 1
2106 DATA 9, 9

```

打 印 图 案

利用 APPLE-II 微电脑可以打印各种图案，而且程序设计也不太复杂。假如您想让妈妈在毛线衣或各种服装上绣

一些图案，不妨先在电脑显示屏上设计一下。



编制图案打印程序的一般方法是：先用一张小方格纸或坐标纸，将需要打印的图形用墨笔画好，然后由左至右将图形上每一行中的黑色部分起止的格数记下来，这就是微电脑打印时，所需要读取的数据。

下面，我们打印一个蝴蝶图案，程序十分简单。你不妨自己也画几个图，编制

简单程序来打印一下。

```
10 PRINT TAB(16); "BUTTERFLY"
20 PRINT
30 PRINT
40 PRINT
50 READ X
60 IF X=40 THEN 40
60 IF X=100 THEN 130
70 PRINT TAB(X);
80 READ Y
90 FOR I=X TO Y
100 PRINT "X";
110 NEXT I
120 GOTO 50
130 END
```


150 DATA 1, 4, 36, 39
 155 DATA 40, 2, 2, 4, 8, 32, 36, 38, 38
 160 DATA 40, 2, 10, 16, 17, 23, 24, 30, 38
 165 DATA 40, 2, 11, 15, 15, 18, 18, 22, 22, 25,
 25, 29, 38
 170 DATA 40, 3, 5, 10, 12, 18, 18, 22, 22, 28,
 30, 35, 37
 175 DATA 40, 3, 4, 7, 8, 11, 13, 19, 19, 21, 21,
 27, 29, 32, 33, 36, 37
 180 DATA 40, 3, 4, 6, 6, 9, 9, 11, 12, 14, 14,
 19, 19, 21, 21, 26, 26, 28, 29, 31, 31, 34, 34,
 36, 37
 185 DATA 40, 3, 4, 6, 6, 9, 9, 11, 12, 15, 15,
 19, 19, 21, 21, 25, 25, 28, 29, 31, 31, 34, 34,
 36, 37
 190 DATA 40, 3, 4, 7, 8, 11, 14, 16, 16, 20, 20,
 24, 24, 26, 29, 32, 33, 36, 37
 195 DATA 40, 4, 5, 10, 11, 13, 15, 17, 17, 20,
 20, 23, 23, 25, 27, 29, 30, 35, 36
 200 DATA 40, 4, 12, 15, 25, 28, 36
 205 DATA 40, 5, 7, 11, 15, 19, 21, 25, 29, 33, 35
 210 DATA 40, 6, 10, 16, 24, 30, 34
 215 DATA 40, 7, 16, 19, 21, 24, 33
 220 DATA 40, 9, 14, 17, 18, 20, 20, 22, 23, 26, 31
 225 DATA 40, 11, 17, 19, 21, 23, 29
 230 DATA 40, 10, 16, 18, 22, 24, 30
 235 DATA 40, 9, 15, 17, 17, 19, 21, 23, 23, 25, 31
 240 DATA 40, 9, 11, 15, 15, 17, 17, 19, 21, 23,

23, 25, 25, 29, 31

245 DATA 40, 9, 11, 16, 17, 19, 21, 23, 24, 29, 31

250 DATA 40, 9, 11, 16, 17, 20, 20, 23, 24, 29, 31

255 DATA 40, 9, 11, 15, 17, 20, 20, 23, 25, 29, 31

260 DATA 40, 9, 12, 14, 16, 24, 26, 28, 31

265 DATA 40, 10, 15, 25, 30

270 DATA 40, 10, 11, 13, 14, 26, 27, 29, 30

275 DATA 40, 11, 13, 27, 29, 40, 11, 12, 28, 29,
40, 11, 12, 28, 29, 100

]RUN

```

****
* ****
*****
***** ** ** *****
***** * * * * *****
*** *** * * *** ***
** ** *** * * *** **
** * * ** * * * * *
** * * ** * * * * *
** ** ***** * * *****
** ** *** * * * *** **
***** *****
*** *****
***** *****
***** ** * ** *****
***** *** *****
***** *****
***** * *** * *****
*** * * *** * * ***
*** ** *** ** ***
*** ** * ** ***
*** *** * *** ***
**** *** *** ****
***** *****
** ** ** **
*** ***
** **
** **

```

祝 您 成 功

我们热忱地希望这本小册子能使您对电脑产生兴趣，并在学习和运用电脑中给您带来乐趣。

我们衷心地祝愿您在掌握微电脑这一有用的工具中获得成功！在未来的事业上成功！

下面是用 TAB 语句编制的打印“祝您成功”的程序。



```

10 PRINT TAB(1)"*      ***"; TAB(14)"*
   *"; TAB(28)"*"; TAB(38)"*
20 PRINT TAB(2)"*      *      *"; TAB(14)
   " * *"; TAB(28)"**"; TAB(38)"*"
30 PRINT TAB(3)"*      *      *"; TAB(14)"*
   *      *"; TAB(23)"*      * *"; TAB
   (38)"*"
40 PRINT TAB(1)"***** *      *"; TAB(14)
   " * *****"; TAB(23)"*****";
   TAB(32)"***** *      *"
50 PRINT TAB(4)"*      *      *"; TAB(13)"*      *
   *      *"; TAB(23)"*      *"; TAB(33)"*
   *****"
    
```

```

60 PRINT TAB(4)"* * *", TAB(12)"*
   * * *", TAB(23)"* *", TAB(33)
   "*"
70 PRINT TAB(4)"* * *", TAB(11)"* *
   * * *", TAB(23)"* *", TAB(33)
   "*"
80 PRINT TAB(3)"* ***", TAB(13)"*
   * * *", TAB(23)"* *", TAB(33)
   "*"
90 PRINT TAB(2)"*** * *", TAB(13)"*
   * * *", TAB(23)"***** *", TAB
   (33)"* * *"
100 PRINT TAB(1)"* ** * *", TAB(13)
   "*" * * *", TAB(23)"* * * *",
   TAB(33)"* **"
110 PRINT TAB(3)"* ** *", TAB(13)"* *
   * *", TAB(23)"* * * *", TAB(33)
   "*"
120 PRINT TAB(3)"* * *", TAB(13)"*
   ** * *", TAB(23)"* * * *",
   TAB(33)"* * *"
130 PRINT TAB(3)"* * *", TAB(13)"*
   **", TAB(23)"* * * *", TAB(33)
   "*"
140 PRINT TAB(3)"* * *", TAB(13)"*
   *", TAB(23)"* * * *", TAB(33)
   "*"
150 PRINT TAB(3)"* * *", TAB(23)"*

```

```

      * ***, TAB(33)"* * *"
160 PRINT TAB(3)"* * *"; TAB(12)
    " * * *"; TAB(23)"* * *"
    TAB(33)"* * *"
170 PRINT TAB(3)"* * *"; TAB(12)"*
    * * *"; TAB(23)"* * *"; TBA
    (33)"* * *"
180 PRINT TAB(3)"* * *"; TAB(12)
    " * * *"; TAB(23)"* * *";
    TAB(33)"* * *"
190 PRINT TAB(3)"* * *"; TAB(12)"*
    * *"; TAB(22)"* * *";
    TAB(32)"* * *"
200 PRINT TAB(3)"* * *"; TAB(14)
    " * * *"; TAB(21)"* * *";
    TAB(31)"* * *"
210 END

```

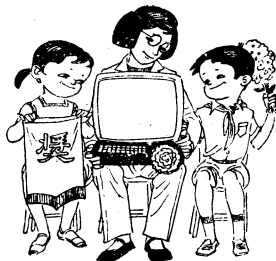
..RUN

```

*      ****      * *      *      *
*      * *      * *      **      *
*      * *      * *      * *      *
**** * *      * * * * * **** * *
* * *      * * * * * *      * * * *
* * *      * * * * * *      * * *
* * * * * * * * * * * * * *
* * * * *      * * * * * * * *
* * * * *      * * * * * * * *
* * *      * * * * * * * * * *
* * *      * * * * * * * * *
* * *      * * * * * * * * *
* * *      * * * * * * * * *
* * *      * * * * * * * * *
* * * * * * * * * * * * *
* * * * * * * * * * * * *
* * * * * * * * * * * * *
* * * * * * * * * * * * *
* * * * * * * * * * * * *
* * * * * * * * * * * * *
* * * * * * * * * * * * *
* * * * * * * * * * * * *

```

附



1984年全国青少年计算机程序设计竞赛
(基础知识赛) 试题

小 学 组

考试时间90分钟

一、以下的每段小程序，都有一个语法上的错误，请改正它：
(15分)

(注意：①不改原程序，只改语法错误。

②要求把改好的程序全部重抄在答卷纸上，不许在试题上修改。)

(1)

```
10 LET X=2
20 PRINT X=2
30 LET B=X
40 LET C=RND (1)
```

(2)

```
10 LET A=5
20 LET 3B=7
30 LET C2=6
40 LET D1=A
```

50 IF C2=0 THEN GOTO 10

(3)

10 INPUT A, B

20 IF A>0 THEN PRINT B

30 GOTO 10

(4)

10 LET A=1

20 LET Y=5

30 LET S=0

40 FOR X=A TO Y

50 LET S=S+1

60 NEXT A

70 PRINT S

(5)

10 LET W=0

20 LET U=1

30 LET W=W+1

40 LET U=U*W

50 IF W≤5 THEN GOTO 30

60 PRINT U

二、分别编写程序计算下列各个式子：(15分)

(1)

$$\frac{28+57 \times 2}{102-25 \div 6} + \frac{1007-638}{31+5 \times 63}$$

(2)

$$1 + \frac{27}{52} \div \left(1 + \frac{3 \times 137}{56} \right) - \frac{56}{28 \times 31}$$

(3)

$$178 + \frac{253}{1 + 381 \times 7.2 - 8.1 \div 3.2}$$

三、下列程序运行后，计算机将给出什么结果：(24分)

(注意：①结果应写在答卷纸上，不许在试题纸上回答。

②结果应按计算机所输出的格式书写。)

(1)

```
10 READ A, B, C
20 LET D=A+B
30 LET E=B+C
40 LET F=D+E
50 PRINT "D="; D, " ", "E=", E
60 PRINT
70 PRINT "F=", F
80 DATA 2, 3, 5
```

(2)

```
10 LET A=1000
20 FOR I=1 TO 10
30 READ B
40 IF A>B THEN LET A=B
50 NEXT I
60 PRINT A
70 END
80 DATA 901, 503, 747, 356, 108
90 DATA 237, 13, 23, 1000, 2000
```

(3)

```
5 DIM C(20)
10 FOR I=1 TO 20
```



```

20 READ C(I)
30 IF C(I)=0 THEN GOTO 60
40 PRINT C(I);
50 GOTO 70
60 PRINT "✖";
70 NEXT I
80 DATA 1, 0, 0, 2, 0, 0, 0, 3, 0, 0
90 DATA 0, 0, 4, 0, 0, 0, 0, 0, 5, 0

```

四、某车库内存放着自行车和三轮车共213辆，已知车轮总数是548只。现要求你编写一个程序（要求使用FOR…NEXT语句），算出自行车、三轮车各为多少辆。（10分）

五、编一个让计算机和小朋友玩“剪刀，钉锤，布”的游戏。用1代表剪刀，2代表钉锤，3代表布。计算机先从1、2、3中随便选一个数。然后印出‘？’，请小朋友也从1、2、3中选一数输入，根据“钉锤赢剪刀，剪刀赢布，布赢钉锤”的规则判定胜负，打印出

“NIN YING LE”	（您赢了）
“NIN SHU LE	（您输了）
“BU SHU BU YING”	（不输不赢）

如果小朋友输了，或者不输不赢，就再来一次，一直到小朋友赢了为止。

要是小朋友给出的不是1、2、3中的数，就提醒他“NIN CUO LE”（您错了），请他重新输入一个数。（13分）

六、编写一个程序计算：（10分）

$1 + 2 \times 2 + 3 \times 3 + 4 \times 4 + 5 \times 5 + \dots + 20 \times 20 + 21 + 3 \times 6 \times 9 \times 12 \times \dots \times 27$ 的值。

七、小华有6角9分钱，小龙有7角8分钱，他们用这些钱全部购买铅笔。商店里有8分、6分、5分、4分一支的铅笔四种。请编

写一个程序，在 8 分钱一支的铅笔必须购买一支的前提下，使买的铅笔数最多，而钱又恰好用完。程序运行后计算机应印出小华和小龙所购买的每种铅笔的数量。（13分）

1984年全国青少年计算机程序设计竞赛
（基础知识赛）试题

初 中 组

考试时间120分钟

一、下面每段小程序都有一个以上语法上的错误，请改正：（18分）

（注意：①不改原程序，只改语法错误。

②要求把改好的程序全部重抄在答卷纸上，不许在试题上修改。）

（1）

```
10 READ A, B
20 LET C=B
30 PRINT "C="; C, A
40 END
```

（2）

```
10 INPUT X, Y
20 IF X>=0 THEN GOTO 40
30 IF Y<=0 THEN GOTO 10
40 PRINT X, 3Y
```

（3）

```
10 LET A=1
20 FOR X=A TO 5
```

```

30 A=AX
40 NEXT A
50 IF A=0 THEN GOTO 70
60 PRINT A.X
70 END

```

二、以下程序运行后，计算机将给出什么结果：（32分）

（注意：①结果应写在答卷纸上，不许在试题纸上回答。

②结果应按计算机所输出的格式书写。）

（1）

```

10 FOR I=1 TO 4
20 FOR J=1 TO I
30 PRINT INT ((I+1)/2);
40 NEXT J
50 PRINT
60 NEXT I
70 END

```

（2）

```

10 READ A, B, C
20 IF A>0 THEN LET A=-1; GOSUB 60
30 IF A<0 THEN LET B=0; GOSUB 80
40 LET A=1
50 GOTO 20
60 LET C=1
70 RETURN
80 PRINT A, B, C; END
90 DATA 0, 1, 2

```

（3）

```

10 LET T=0

```

```

20 READ X
30 IF X=0 THEN GOTO 110
40 IF X<0 THEN GOTO 20
50 PRINT X
60 LET T=T+X
70 PRINT T
80 GOTO 20
90 DATA -1, 3, -15, 7, -12
100 DATA 10, -30, -40, 0
110 END

```

(4)

```

10 DIM A(2, 3), B(3, 2)
20 FOR I=1 TO 2
30 FOR J=1 TO 3
40 READ A(I, J)
50 PRINT A(I, J), " ",
60 NEXT J
70 PRINT
80 NEXT I
90 DATA 1, 2, 3, 4, 5, 6
100 FOR J=1 TO 3
110 FOR I=1 TO 2
120 B(J, I)=A(I, J)
130 PRINT B(J, I), " ",
140 NEXT I
150 PRINT
160 NEXT J
170 END

```

三、编写一个程序利用随机函数产生10个各不相同的两位正整数，并求其中的最大值。（10分）

四、编一个程序，它可从键盘上输入三个数，把它看成是三角形三边的长度，然后进行判断。如果这是一个等边三角形，就印出“3”；如果这是一个等腰三角形，就印出“2”；如果这是一个不等边三角形，就印出“1”；如果这三个数根本不能代表一个三角形的三边（譬如，其中有一个数是负数），就印出“X”。（10分）

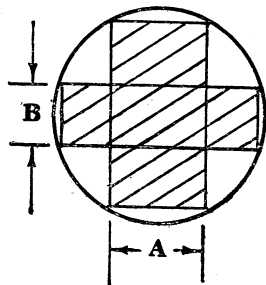
五、用 FOR……NEXT 语句，在下列范围

$$10 \geq X \geq -10, \quad 6 \geq Y \geq -6, \quad 1 \geq Z \geq -1$$

内求方程 $X^3 + Y^3 + Z^3 = 3$ 的整数解。（10分）

六、小明有一个最多只能装10斤重的网袋。现有白菜5斤、猪肉2斤、鱼3.5斤、酱油连瓶重1.7斤、白糖1斤、土豆5.1斤。设计一个程序使小明的网袋装的物品总重量最大。（10分）

七、在半径为R的圆内有两个互相正交的内接长方形，如附图所示，其中A和B可以随便选，但必须是整数。



编一个程序，用来选择A和B的值使阴影部分面积最大。（10分）

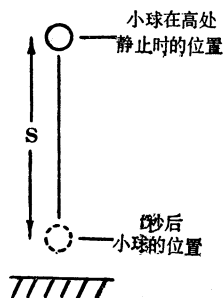
1984年全国青少年计算机程序设计竞赛 (基础知识赛) 试题

高 中 组

考试时间120分钟

〔题1〕 小球从高处由静止开始自由下落，用闪光相机每隔0.1秒拍照一次，得到如下表所示数据：

小球下落时间 t 秒	0	0.1	0.2	0.3	0.4	0.5
小球下落距离 s 厘米	0	4.9	19.5	44.0	78.5	122.3



已知重力加速度 $g = 980 \text{ 厘米/秒}^2$ ，考虑到由于测量不精确，存在一定的误差，如果每一组用 t 、 s 值算出的加速度 G ，都满足 $|G - g| < 10 \text{ 厘米/秒}^2$ ，则可认为小球作匀加速直线运动。

试排一程序来证明小球自由下落是匀加速直线运动。如是匀加速直线运动请打 SHI (是)，否则打印 FOU (否)

(15分)

〔题2〕有以 G_1 、 G_2 、 G_3 、 G_4 、 G_5 命名的五顶“金”冠，它们都是由金银两种贵金属制作的，这五顶“金”冠分别在水中和空气中称量时的重量列表如下：

重量(克) / “金”冠名 / 称量方式	G_1	G_2	G_3	G_4	G_5
在水中称	493	451	475	460	450
在空气中称	521	481	502	512	492

已知金的比重为 19.3 克/厘米^3 ，银的比重为 10.5 克/厘米^3 。

试排一程序将五顶“金”冠的名字，按含金比例从高到低，依次打印出来。

〔提示〕本题比重大的“金”冠含金比例高。(20分)

〔题3〕某班一学习小组共有9名同学，他们的学号依次为1，2，3，……，9。期末考试语文、代数、外语课及格者名单(以学号表示)如下：

	及格者的学号
语 文	1, 9, 6, 8, 4, 3, 7
代 数	5, 2, 9, 1, 3, 7
外 语	8, 1, 6, 7, 3, 5, 4, 9

试排一程序统计三门课都及格的有谁，打印出他们的学号。

要求用 READ DATA 语句依表格所列顺序读入数据。(20分)

〔题4〕下面这个程序的目的是要印出函数 $F(X)$ 在 X 为2，3，4时的值，指出该程序错在哪里，请在原程序基础上将它改正。在改动时，120句内容不允许变化。(5分)

```
10 FOR X=2 TO 4
20 GOSUB 100
30 PRINT F(X)=C
40 NEXT X
50 END
60 GOTO 20
100 READ A, B
110 LET C=A*X+B
120 DATA 2, 3
130 RETURN
```

〔题5〕编一个“念数字”的程序，它能让计算机完成以下动作。计算机先印出一个“？”，请你输入一个0到99之间的数。然后计算机就用汉字拼音印出这个数的念法来。

例如：? 3 5

SANSHIWU

? 0

LING

如果给出的数不在 0 和 99 之间，计算机就印出“CUO LE”（错了），请求重新输入。

注：为了使不熟悉汉语拼音的同学也能做这个题，把“〇，一，二，三，……，九，十”的拼法写在下面

〇 LING 一 YI 二 ER 三 SAN 四 SI 五 WU

六 LIU 七 QI 八 BA 九 JIU 十 SHI （10分）

〔题6〕 在 1 米见方的平面上，有一个由闭合曲线围成的区域（图 1），我们希望计算它的面积。

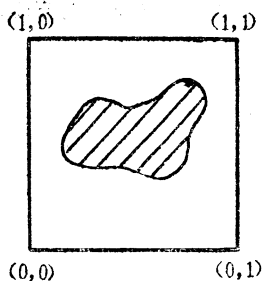


图 1

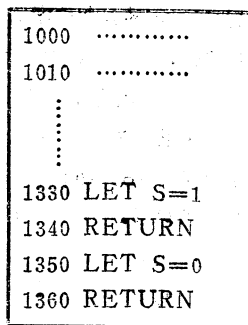


图 2

假设已经有了一个用 BASIC 语言写好的子程序（图 2），它可以判断平面上任一点 (X, Y) 是否在该曲线的内部，从 1000 号语句进入该子程序，若 (X, Y) 这一点位于曲线内部，它就令 $S=1$ ，然后返回。若 (X, Y) 这一点位于曲线之外，它就令 $S=0$ ，然后返回，现在打算用数学实验方法来近似地估算上述区域的面积。方法是：

随机地产生个N点的坐标 (X, Y), 其中X和Y均在0和1之间均匀分布, 对于每一点, 都可以用上述子程序来判断它是否位于曲线内部。

若N点中共有M点位于曲线之内, 当N很大时 (例如 $N > 1000$) 可以近似地认为

$$\text{面积 } A = \frac{M}{N}$$

试根据上述想法编出程序。

(10分)

[题7] 编写一个程序, 用INPUT语句读入一个数, 如该数是0, 则印出图1; 如该数是1, 则印出图2; 如该数是2, 则印出图3; 在其他情况下, 印出图4。

(20分)

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

图 1

		1		
	1	1		
		1		
		1		
		1		
		1		
1	1	1		

图 2

	2	2	2	
2				2
			2	
		2		
	2			
2				
2	2	2	2	2

图 3

	?	?	?	
?				?
			?	
		?		
		?		
		?		

图 4

小学组上机试题

使用机型	上机使用的时间	通过否	监考老师签名

(此表由老师填写)

小明想编写一个程序, 用计算机印出如下的小队旗。下面就是她所编写的程序, 请你上机调试好。

要求：1.将程序全部输入计算机。

2.改正程序中的错误，使程序运行后，印出小队旗。

```
1          10 FOR I=1 TO 18
1 1          20 IF I>9 THEN 90
1 1 1        30 FOR J=1 TO I
1 1 1 1      40 PRINT "1"
1 1 1 1 1    50 NEXT J,
1 1 1 1 1 1  60 PRINT
1 1 1 1 1 1 1 70 GOTO 100
1 1 1 1 1 1 1 1 80 PRINT "1"
1 1 1 1 1 1 1 1 1 90 NEXT I
1          100 END

1
1
1
1
1
1
1
1
1
1
```

初中组上机试题

小龙想编写一个程序，用计算机印出如下的图形。以下就是他的程序，请你上机调试好。

要求：1.将程序全部输入计算机。

2.改正程序中的错误，使程序运行后，印出所要的图形。

参 考 答 案

(全国)竞赛小学组

一、

(1)

10 LET X=2

20 PRINT X

30 LET B=X

40 LET C=RND (1)

(2)

10 LET A=5

20 LET B=7

30 LET C2=6

40 LET D1=A

50 IF C2=0 THEN GOTO 10

(3)

10 INPUT A, B

20 IF A>0 THEN PRINT B

30 QOTO 10

(4)

10 LET A=1

20 LET Y=5

30 LET S=0

40 FOR X=A TO Y

50 LET S=S+1

```

60 NEXT X
70 PRINT S
(5)
10 LET W=0
20 LET U=1
30 LET W=W+1
40 LET U=U*W
50 IF E<=5 THEN GOTO 30
60 PRINT U

```

二、

```

(1)
10 LET A=(28+57*2)/(102-25/6)
20 LET B=(1007-638)/(31+5*63)
30 PRINT A+B

```

(2)

```

10 LET A=56/(28*31)
20 LET B=(3*127)/(1-A)
30 LET C=52/(1+B)
40 PRINT 27/(1+C)

```

(3)

```

10 LET A=178
20 PRINT A+253/(1+381*7.2-8.1/3.2)

```

三、

(1)

JRUN

D=5 E=8

F=13

(2)

JRUN

13

(3)

JRUN

1*2**3***4***5*

四、

10 LET A=213

20 LET B=548

30 FOR I=1 TO INT (B/3)

40 IF B-3*I=2*(A-I) THEN GOTO 60

50 NEXT I

60 PRINT I, A-I

70 END

JRUN

122 91

五、

10 LET X=INT (RND (1)*3)+1

20 INPUT Y

30 PRINT X

40 IF X-Y=0 GOTO 150

50 IF X-Y=1 GOTO 130

60 IF X-Y=-2 GOTO 130

70 IF X-Y=2 GOTO 110

80 IF X-Y=-1 GOTO 110

90 PRINT "NIN CUO LE"

100 GOTO 10

110 PRINT "NIN YING LE"

120 STOP

```

130 PRINT "NIN SHU LE"
140 GOTO 10
150 PRINT "BU SHU BU YING"
160 GOTO 10
]RUN
? 1
2
NIN SHU LE
? 3
3
BU SHU BU YING
? 2
2
BU SHU BU YING
? 3
1
NIN SHU LE
? 1
1
BU SHU BU YING
? 2
3
NIN SHU LE
? 3
1
NIN SHU LE
? 1
3

```

NIN YING LE

BREAK IN 120

六、

```
10 LET A=1+21
20 FOR I=2 TO 20
30 LET A=A+I*I
40 NEXT I
50 LET B=1
60 FOR J=3 TO 27 STEP 3
70 LET B=B*I
80 NEXT J
90 PRINT A+B
100 END
```

七、

```
5 LET E=0; LET F=0
10 READ A
15 IF A=0 THEN GOTO 110
20 DATA 69, 78, 0
30 LET B=A-B
40 LET C=INT (B/4)
50 LET D=B-C*4
60 IF D=3 THEN LET C=C-2; LET E=1,
    LET F=1
70 IF D=2 THEN LET C=C-1; LET F=1
80 IF D=1 THEN LET C=C-1; LET E=1
90 PRINT "8-1", " ", "6-", F, "5-", E
    " ", "4-", C
100 GOTO 5
```


110 END

JRUN

8-1 6-0 5-1 4-14

8-1 6-1 5-0 4-16

(全国)竞赛初中组

一、

(1)

10 READ A, B

20 LET C=B

30 PRINT "C=", C, A

40 END

50 DATA 1, 2

(2)

10 INPUT X, Y

20 IF $X \geq 0$ THEN GOTO 40

30 IF $Y \leq 0$ THEN GOTO 40

40 PRINT X, $3 \times Y$

(3)

10 LET A=1

20 FOR X=A TO 5

30 $A = A \times X$

40 NEXT X

50 IF $A = 0$ THEN GOTO 70

60 PRINT A, X

70 END

二、

(1)	(2)
JRUN	JRUN
1	-101
11	
222	
2222	
(3)	(4)
JRUN	JRUN
3	123
3	456
7	14
10	25
10	36
20	

三、

```

10 DIM A(10)
20 FOR I=1 TO 10
30 A(I)=0
40 NEXT I
60 FOR I=1 TO 10
70 LET B=INT(90*RND(1)+10)
80 FOR J=1 TO I-1
90 IF B=A(J) THEN GOTO 70
100 NEXT J
110 LET A(I)=B
115 PRINT A(I)
120 NEXT I
125 M=5

```

```

130 FOR I=1 TO 10
140 IF A(I)>M THEN LET M=A(I)
150 NEXT I
155 PRINT
160 PRINT M
170 END
]RUN

```

92

98

15

39

65

99

56

69

95

26

99

四、

```

10 INPUT "A=", A
20 INPUT "B=", B
30 INPUT "C=", C
40 IF A<=0 GOTO 200
50 IF B<=0 GOTO 200
60 IF C<=0 GOTO 200
70 IF A+B<=C GOTO 200
80 IF B+C<=A GOTO 200
90 IF C+A<=B GOTO 200

```

```
100 IF A=B GOTO 150
110 IF A=C GOTO 160
120 IF B=C GOTO 160
130 PRINT 1
140 STOP
150 IF A=C GOTO 180
160 PRINT 2
170 STOP
180 PRINT 3
190 STOP
200 PRINT "X"
210 STOP
```

]RUN

A=26

B=33

C=25

1

BREAK IN 140

]RUN

A=33

B=33

C=22

2

BREAK IN 170

]RUN

A=2

B=2

C=3

X

BREAK IN 210

如果用扩展 BASIC 语言编写

10 INPUT "A, B, C="; A, B, C

20 IF A<0 OR B<0 OR C<0 THEN 90

30 IF A+B<=C OR B+C<=A OR A+C<=B
THEN 90

40 IF A=B AND B=C THEN 80

50 IF A=B OR B=C OR A=C THEN 70

60 PRINT "1"; END

70 PRINT "2"; END

80 PRINT "3"; END

90 PRINT "X"; END

]RUN

A, B, C=5, 6, 7

1

]RUN

A, B, C=5, 5, 10

X

]RUN

A, B, C=5, 5, 5

3

JRUN

A, B, C=5, 5, 6

2

五、

10 FOR X=-10 TO 10

20 FOR Y=-6 TO 6

30 FOR Z=-1 TO 1

40 IF $X^3 + Y^3 + Z^3 \neq 0$ THEN 60

50 PRINT "X=", X, "Y=", Y, "Z=", Z

60 NEXT Z

70 NEXT Y

80 NEXT X

90 END

JRUN

X=1

Y=1

Z=1

六、

5 G1=0

10 FOR A=0 TO 5 STEP 5

20 FOR B=0 TO 2 STEP 2

30 FOR C=0 TO 3.5 STEP 3.5

40 FOR D=0 TO 1.7 STEP 1.7

50 FOR E=0 TO 1

60 FOR F=0 TO 5.1 STEP 5.1

70 G=A+B+C+D+E+F

80 IF G>10 THEN 110

90 IF G1>=G THEN 110

100 G1=G; A1=A; B1=B; C1=C; D1=D; E1=E;

```

      F1=F
110  NEXT F
120  NEXT E
130  NEXT D
140  NEXT C
150  NEXT B
160  NEXT A
170  PRINT "A="; A1, "B="; B1, "C="; C1,
      "D="; D1, "E="; E1, "F="; F1
190  PRINT "G="; G1
200  END

```

]RUN

```

A=0          B=2          C=0
D=1.7        E=1          F=5.1
G=9.8

```

七、

```

10  INPUT "R="; R
20  LET D=2*R
30  LET M=0
40  FOR B=0 TO D
50  LET E=SQR (D*B-B*B)
60  FOR A=0 TO E
70  LET F=SQR (D*B-A*A)
80  LET S=F*A+E*B-A*B
90  IF S<M GOTO 130
100 LET M=S
110 LET U=A

```

```

120 LET V=B
130 NEXT A
140 NEXT B
150 PRINT "A", U
160 PRINT "B=", V

```

]RUN

R=5

A=5

B=5

]RUN

R=2

A=2

B=2

]RUN

R=3

A=3

B=3

(全国)竞赛高中组

【题1】

```
10 FOR T=0.1 TO 0.5 STEP 0.1
```

```
20 READ S
```

```
30 G=2*S/(T*T)
```

```
40 IF ABS(G-980)>10 THEN 80
```



```

50 NEXT T
60 PRINT "SHI"
70 GOTO 90
80 PRINT "FOU"
90 END
100 DATA 4.9, 19.5, 44.0, 78.5, 122.3

```

] RUN

SHI

[題 2]

```

10 DIM R(5)
20 FOR I=1 TO 5
30 READ G1, G2
40  $R(I) = G2 / (G2 - G1)$ 
50 NEXT I
60 FOR I=1 TO 5
70 M=R(I)
75 N=I
80 FOR J=1 TO 5
90 IF M>R(J) THEN 120
100 M=R(J)
110 N=J
120 NEXT J
130 PRINT "G", N
140 R(N)=0
150 NEXT I
160 END
200 DATA 493, 521, 451, 481, 475, 502, 460, 512,

```

] RUN

G1

G3

G2

G5

G4

[题 3]

10 DIM A(9)

20 READ S

30 IF S=-1 THEN 60

40 A(S)=A(S)+1

50 GOTO 20

60 FOR I=1 TO 9

70 IF A(I)<>3 THEN 90

80 PRINT I

90 NEXT I

100 END

110 DATA 1, 9, 6, 8, 4, 3, 7

120 DATA 5, 2, 9, 1, 3, 7

130 DATA 8, 1, 6, 7, 3, 5, 4, 9, -1

] RUN

1

3

7

9

[题 4]

```

10 FOR X=2 TO 4
20 GOSUB 100
30 PRINT "F(", X, ")="; C1
40 NEXT X
50 END
60 GOTO 20
100 READ A, B
110 LET C=A*X+B
115 RESTORE
120 DATA 2, 3
130 RETURN

```

] RUN

F(2)=7

F(3)=9

F(4)=11

[题 5]

```

10 INPUT X
20 IF X<0 GOTO 140
30 IF X>99 GOTO 140
32 IF X>0 GOTO 40
34 PRINT "LING";
36 STOP
40 IF X<10 GOTO 110
50 LET Y=INT (X/10)
60 GOSUB 160
70 PRINT "SHI";
80 LET Y=X-Y*10
90 GOSUB 160

```

```

100 STOP
110 LET Y=X
120 GOSUB 160
130 STOP
140 PRINT "CUD LE"
150 GOTO 10
160 IF Y>0 GOTO 190
180 RETURN
190 IF Y>1 GOTO 220
200 PRINT "YI";
210 RETURN
220 IF Y>2 GOTO 250
230 PRINT "ER";
240 RETURN
250 IF Y>3 GOTO 280
260 PRINT "SAN";
270 RETURN
280 IF Y>4 GOTO 310
290 PRINT "SI";
300 RETURN
310 IF Y>5 GOTO 340
320 PRINT "WU";
330 RETURN
340 IF Y>6 GOTO 370
350 PRINT "LIU";
360 RETURN
370 IF Y>7 GOTO 400
380 PRINT "QI";

```

```

390 RETURN
400 IF Y>8 GOTO 430
410 PRINT "BA",
420 RETURN
430 PRINT "JIU",
440 RETURN

```

改用扩展 BASIC 语言编写

```

10 DIM A$(11)
20 FOR I=0 TO 11: READ A$(I): NEXT I
30 INPUT N
40 IF N<0 OR N>99 OR INT(N)
   <>N THEN PRINT A$(11), GOTO 30
50 IF N=0 THEN PRINT A$(0)
60 A=INT(N/10)
70 IF A=0 THEN 90
80 PRINT A$(A), A$(10),
90 B=N-A*10
100 IF B=0 THEN 120
110 PRINT A$(B)
120 END
130 DATA "LING", "YI", "ER", "SAN", "SI",
        "WU", "LIU", "QI", "BA", "JIU", "SHI",
        "CHOU LE"

```

IRUN

? 23

ERSHISAN

JRUN

? 98

JIUSHIBA

JRUN

? 1

YI

JRUN

? 189

CHOU LE

[题 6]

5 INPUT N

10 M=0

20 FOR I=1 TO N

30 X=RND(1)

40 Y=RND(1)

50 GOSUB 1000

60 M=M+S

70 NEXT I

80 PRINT M/N

90 END

1000 REM SUB

1336 S=1

1340 RETURN

1350 S=0

1360 RETURN

1000和1336句之间的程序 与题目给出的具体 图形有关, 现在无

法写出。

[题 7]

```
10 DIM M(4*35)
20 FOR I=0 TO 4*35-1
30 READ M(I)
40 NEXT I
50 INPUT X
60 IF X<0 GOTO 90
70 IF X>2 GOTO 90
80 GOTO 100
90 LET X=3
100 FOR I=0 TO 6
110 FOR J=0 TO 4
120 LET K=X*35+I*5+J
130 IF M(K)=9 GOTO 160
140 IF M(K)<3 GOTO 180
150 IF M(K)=3 GOTO 200
160 PRINT " ", " "
170 GOTO 210
180 PRINT M(K),
190 GOTO 210
200 PRINT "? ",
210 NEXT J
220 PRINT
230 NEXT I
240 STOP
250 DATA 9, 0, 0, 0, 9, 0, 9, 9, 9, 0, 0, 9, 9,
      0, 0, 0, 9, 0, 9, 0, 0, 9, 9, 0, 0, 9, 9, 9,
```

```

0, 9, 0, 0, 0, 9,
260 DATA 9, 9, 1, 9, 9, 9, 1, 1, 9, 9, 9, 9, 1,
9, 9, 9, 9, 1, 9, 9, 9, 9, 1, 9, 9, 9, 1, 9,
9, 9, 1, 1, 1, 9
270 DATA 9, 2, 2, 2, 9, 2, 9, 9, 9, 2, 9, 9, 9,
2, 9, 9, 9, 2, 9, 9, 9, 2, 9, 9, 9, 2, 9, 9, 9,
9, 2, 2, 2, 2, 2
280 DATA 9, 3, 3, 3, 9, 3, 9, 9, 9, 3, 9, 9, 9,
3, 9, 9, 9, 3, 9, 9, 9, 9, 3, 9, 9, 9, 9, 9, 9,
9, 9, 9, 3, 9, 9

```

如果用扩展 BASIC 语言编写

```

10 DIM M(3, 35)
20 FOR I=0 TO 3: FOR J=1 TO 35
30 READ M(I, J)
40 NEXT J, I
50 INPUT N
55 C$=STR$ (N)
60 IF N<0 OR N>2 OR N<>
INT(N) THEN C$="?", N=3
70 FOR I=1 TO 35
90 IF M(N, I)=1 THEN PRINT C$, " GOTO
110
100 PRINT " ";
110 IF I/5=INT(I/5) THEN PRINT
120 NEXT I
130 END
140 DATA, 1, 1, 1, , 1, , , 1, 1, , , 1, 1, 1,
, 1, , 1, 1, 1, , , 1, 1, , , 1, , 1, 1, 1, 0

```


150 DATA, , 1, , , , 1, 1, , , , 1, , , , , 1,
 , , , , 1, , , , , 1, , , , 1, 1, 1, 0
 160 DATA, 1, 1, 1, , 1, , , , 1, , , , 1, , , ,
 1, , , , 1, , , , 1, , , , , 1, 1, 1, 1, 1
 170 DATA, 1, 1, 1, , 1, , , , 1, , , , 1, , , ,
 1, , , , , 1, , , , , , , , , , 1, , 0

IRUN	IRUN	IRUN
? 0	? 1	? 2
000	1	???
0 0	11	? ? ?
0 00	1	?
0 0 0	1	?
00 0	1	?
0 0	1	
000	111	?

(全国)竞赛小学组上机题

```
10 FOR I=1 TO 18
20 IF I>9 THEN GOTO 80
30 FOR J=1 TO I
40 PRINT "1";
50 NEXT J
60 PRINT
70 GOTO 90
80 PRINT "1";
90 NEXT I
100 END
```

(全国)竞赛初中组上机题

```
10 FOR X=1 TO 9
20 PRINT TAB(20-X)
30 FOR Y=1 TO 2*X-1
40 PRINT "1";
50 NEXT Y
60 PRINT
70 NEXT X
80 FOR I=8 TO 1 STEP -1
90 PRINT TAB(20-I)
100 FOR J=1 TO 2*I-1
110 PRINT "1";
120 NEXT J
130 PRINT
140 NEXT I
150 END
```

(全国)竞赛高中组上机题

原来的程序执行结果是:

JRUN

9
9
8
8
7
7
6
6
5
5
4
4
3
3
2
2
1
1
0

? NEXT WITHOUT FOR ERROR IN 90

下面的程序是正确的: (在APPLE机上执行)

```
10 FOR I=0 TO 18
20 PRINT TAB(ABS(9-I)+10);
30 LET B=2*I+1
40 IF I>9 THEN B=37-2*I
50 FOR J=1 TO B
60 PRINT INT((19-I)/2);
70 NEXT J
80 PRINT
90 NEXT I
100 END
```

附录 I

APPLE-II 微电脑主机内存贮器概况

存贮器 种类	页	数		内存单元地址		内存安排与使用情况
		十六进制	十进制	十六进制	十进制	
RAM	0	\$ 00	0~255	\$ 0000~\$ 00FF	系统程序	浮点 BASIC 系统 占用区 (用户 的程序和数据不 能使用本区域)
	1	\$ 01	256~511	\$ 0100~\$ 01FF	系统堆栈	
	2	\$ 02	512~767	\$ 0200~\$ 02FF	键盘输入一行的缓冲 区	
	3	\$ 03	768~1023	\$ 0300~\$ 03FF	监控程序向量位置	
LOMEM	4~7	\$ 04 \$ 07	1024~2047	\$ 0400~\$ 07FF	文字资料显示, 低 分辨率图形第一页	↓ 用户程序从这里 开始存放
	8~11	\$ 08~\$ 0B	2048~3071	\$ 0800~\$ 0BFF	文字资料显示, 低 分辨率图形第二页	
→ RAM	12~31	\$ 0C~\$ 1F	3072~8191	\$ 0C00~\$ 1FFF		用户使用区
	32~63	\$ 20~\$ 3F	8192~16383	\$ 2000~\$ 3FFF	高分辨率图形第一 页	
	64~95	\$ 40~\$ 5F	16384~24575	\$ 4000~\$ 5FFF	高分辨率图形第二 页	

HIMEM	96~149	\$ 60~\$ 95	24576~38399	\$ 6000~\$ 95FF	(DOS 存在时) ↑ ↑ 用户的数据、字 符串、数组从这 里向上存放
HIMEM	150~191	\$ 96~\$ BF	38400~49151	\$ 9600~\$ BFFF	↑ 用户的数据、字 符串、数组从这 里向上存放
ROM	192~199	\$ C0~\$ C7	49152~51199	\$ C900~\$ C7FF	I/O(2K) 输入/输出功能 函数
	200~207	\$ C8~\$ CF	51200~53247	\$ C800~\$ CFFF	I/O(2K) 输入/输入使用区
↓ 1M	208~255	\$ D0~\$ FF	53248~65535	\$ D000~\$ FFFF	ROM(12K)

用户不能使用本
区域

附录 II

ASCII 码 表

ASCII 码	显 示 字	键	ASCII 码	显 示 字	键
0		CTRL-@	10	(linefeed)	CTRL-J
1		CTRL-A	11		CTRL-K
2		CTRL-B	12		CTRL-L
3		CTRL-C	13	(carriage return)	CTRL-M
4		CTRL-D	14		CTRL-N
5		CTRL-E	15		CTRL-O
6		CTRL-F	16		CTRL-P
7	(bell)	CTRL-G	17		CTRL-Q
8	(backspace)	CTRL-H or ←	18		CTRL-R
9		CTRL-I	19		CTRL-S

20		CTRL-T	33	!	!
21	(forward space)	CTRL-uor→	34	"	"
22		CTRL-V	35	#	#
23		CTRL-W	36	\$	\$
24	(cancel line)	CTRL-X	37	%	%
25		CTRL-Y	38	&	&
26		CTRL-Z	39	'	'
27		ESC	40	(	(
28		n . z .	41	)	)
29		CTRL-SHIFT-M	42	*	*
30		CTRL-A	43	+	+
31		n . a .	44	,	,
32	Space	Spacebar	45	—	—

续表

ASCII码	显 示 字	键	ASCII码	显 示 字	键
46	•	•	58	:	:
47	/	/	59	;	;
48	0	0	60	<	<
49	1	1	61	=	=
50	2	2	62	>	>
51	3	3	63	?	?
52	4	4	64	@	@
53	5	5	65	A	A
54	6	6	66	B	B
55	7	7	67	C	C
56	8	8	68	D	D
57	9	9	69	E	E

70	F	F	83	S	S
71	G	G	84	T	T
72	H	H	85	U	U
73	I	I	86	V	V
74	J	J	87	W	W
75	K	K	88	X	X
76	L	L	89	Y	Y
77	M	M	90	Z	Z
78	N	N	91	[	n . a .
79	O	O	92	/	n . a .
80	P	P	93	]	SHIFT-M
81	Q	Q	94	^	^
82	R	R	95		n . a

注：如果可以由监督系统或整数 BASIC 产生，则加入 128 到 ASCII 码内，n.a. = 不能由 APPLE- I 键盘产生。

附录 III

软磁盘的格式化步骤

(1) 把 DOS 引导到 APPLE-Ⅱ 微电脑的内存。

(2) 在键盘上输入一段“招呼”程序。招呼程序的一般格式为：

NEW

10 REM INITIALIZED ON 1984. 6. 1

20 REM SYSTEM MEMORY 48K

30 REM DOS VERSION 3.3

40 PRINT "WR-2, VOL.3"

50 END

(说明：10句注明盘片格式化日期；20句注明格式化时内存 RAM 的容量；30句注明格式化时 DOS 的版本；40句给盘片取名编号。)

(3) 将需要格式化的盘片插入软磁盘驱动器(一般插入与6* 输入/ 输出插槽上的接口电路卡相连的 I 号驱动器内)。

(4) 从键盘上输入格式化命令：

INITHELLO, S6, D1, V3

(说明：“HELLO”是上述招呼程序的名字，作为磁盘文件名；“S6”表明6*插槽；“D1”表明第1台驱动器；“V3”表明盘片编号。)

这时，I 号磁盘驱动器“IN USE”灯亮，并发出盘片转动声，过一会盘片停止转动，“IN USE”灯灭，屏幕显示器出现浮点 BASIC 的提示符“】”，表示格式化工作完成。从此，这个软磁盘片就可用来存放各种类型的磁盘文件了。

附录 IV

浮点 BASIC 错误信息表

(1) ?BAD SUBSCRIPT ERROR IN< 语句标号 >

在使用下标变量时,下标数字超过了定维语句 (DIM) 定出的容量,或下标个数不一致。错误代码是107。

(2) ?CAN'T CONTINUE ERROR

如果你中断程序,再执行,可以使用 CONT 命令。除此以外,在其他情况下使用 CONT 命令就会发生错误。

(3) ?DIVISION BY ZERO ERROR IN< 语句标号 >

使用值为零的算术表达式做除数,错误代码133。

(4) ?FORMULA TOO COMPLEX ERROR IN

< 语句标号 >

公式太复杂。错误代码191。

(5) ?ILLEGAL DIRECT ERROR IN< 语句标号 >

用了 INPUT, DEF FN 或 GET 等语句的命令形式,而这些语句的命令形式是不允许的。

(6) ?ILLEGAL QUANTITY ERROR IN < 语句标号 >

在一个字符串函数、数值函数、描图叙述等语句中,所用到的数值超过了能接受的范围。错误代码53。

(7) ?NEXT WITHOUT FOR ERROR IN< 语句标号 >

NEXT 语句与 FOR 不匹配。错误代码 0。

(8) ? OUT OF DATA ERROR IN< 语句标号 >

要读入的数多于 DATA 语句中的数。错误代码42。

(9) ? OUT OF MEMORY ERROR IN < 语句标号 >

说明程序所需要用的内存单元超出了允许范围,譬如程序太

长、简单变量或下标变量太多、FOR/NEXT 嵌套循环超过 10 层、括号超过 36 层等。错误代码 77。

(10) ? OVERFLOW ERROR IN < 语句标号 >

计算出的数值大小超过了允许的范围，这个范围是 $-1.7 \text{ E} + 38$ 到 $1.7 \text{ E} + 38$ 。错误代码 69。

(11) ? REDIM'D ARRAY ERROR IN < 语句标号 >

如果在程序中对前面已经定义好维数的数组重新定义，就会发生错误。错误代码 12。

(12) ? RETURN WITHOUT GOSUB ERROR IN
< 语句标号 >

RETURN 语句缺少 GOSUB 与其对应。错误代码 22。

(13) ? STRING TOO LONG ERROR IN < 语句标号 >

字符串的字符个数超过 255。错误代码 176。

(14) ? SYNTAX ERROR IN < 语句标号 >

语句本身的各类错误，譬如：标点符号、括号次序、语句不完整等等。另外，其他种种错误信息涉及不到的错误也统统归入这一类错误之中。错误代码 16。

(15) ? TYPE MISMATCH ERROR IN < 语句标号 >

应该使用字符串的地方，用了算术表达式或者变量，或者应该使用数值的地方，却用了字符串。错误代码 163。

(16) ? UNDEF'D FUNCTION ERROR IN < 语句标号 >

所使用的函数不是标准函数，或者是未被定义过的函数。错误代码 224。

(17) ? UNDEF'D STATEMENT ERROR IN < 语句标号 >

转向语句转到一个程序中不存在的语句标号。错误代码 90。

附录 V

浮点 BASIC (APPLESOFT) 与 整数 BASIC (INTEGER BASIC) 区别

下述命令只能用于 APPLESOFT	下述命令只能用于 Integer BASIC 而不能用于 APPLESOFT
而不能用 Integer BASIC	
ATN	AUTO
CHR\$,COS,	DSP
DATA,DEF,FN,DRAW,	
EXP,	
FLASH,FN,FRE,	
GET,	
HCOLOR,HGR,HGR2,HLMEM:, HPLOT,	
INT, INVERSE,	
LEFTS,LOG,LOMEM: ,	
MID\$,	MAN,MOD,
NORMAL	
ON...GOSUB, ON...GOTO, ONE- RR...GOTO,	
POS,READ, RECALL, RESTO- RE,RESUME,	
RIGHT\$,ROT	
SCALE,SHLOAD,SLN,SPC,SPE- ED,SQR,STOP	
STORE,STR\$,	
TAN,	

续表

下述命令只能用于 APPLESOFT 而不能用于 Integer BASIC	下述命令只能用于 Integer BASIC 而不能用于 APPLESOFT
USR, VAL, WAIT XDRAW	

下述命令名称不同

Integer BASIC	APPLESOFT
LCK	CLEAR
CON	CONT
TAB	HTAB (在 APPLESOFT 中另有 TAB 命令)
GOTOX *10+100	ONXGOTO 100,110,120
GOSUBX * 100+1000	ONXGOSUB 1000,1100,1200
CALL-936	HOME(或CALL-936)
POKE 50,127	INVERSE
POKE 50,255	NORMAL
X	X % (%表示整形变量)
#	<>或><

附录 VI

浮点 BASIC 的保留字及其代表数字

代表 数字	保 留 字	代表 数字	保 留 字	代表 数字	保 留 字
128	END	152	ROT=	176	GOSUB
129	FOR	153	SCALE=	177	RETURN
130	NEXT	154	SHLOAD	178	REM
131	DATA	155	TRACE	179	STOP
132	INPUT	156	NOTRACE	180	ON
133	DEL	157	NORMAL	181	WAIT
134	DIM	158	INVERSE	182	LOAD
135	READ	159	FLASH	183	SAVE
136	GR	160	COLOR=	184	DEF
137	TEXT	161	POP	185	POKE
138	PR#	162	VTAB	186	PRINT
139	IN#	163	HIMEM,	187	CONT
140	CALL	164	LOMEM,	188	LIST
141	PLOT	165	ONERR	189	CLEAR
142	HLIN	166	RESUME	190	GET
143	VLIN	167	RECALL	191	NEW
144	HGR2	168	STORE	192	TAB(
145	HGR	169	SPEED=	193	TO
146	HCOLOR=	170	LET	194	FN
147	HPLOT	171	GOTO	195	SPC
148	DRAW	172	RUN	196	THEN
149	XORAW	173	IF	197	AT
150	HTAB	174	RESTORE	198	NOT
151	HOME	175	&	199	STEP

续表

代表 数字	保 留 字	代表 数字	保 留 字	代表 数字	保 留 字
200	+	212	ABS	224	TAN
201	-	213	USR	225	ATN
202	*	214	FRE	226	PEEK
203	/	215	SCRN(	227	LEN
204	^	216	PDL	228	STR\$
205	AND	217	POS	229	VAL
206	OR	218	SQR	230	ASC
207	>	219	RND	231	CHR\$
208	=	220	LOG	232	LEFT\$
209	<	221	EXP	233	RIGHT\$
210	SGN	222	COS	234	MIDS
211	INT	223	SIN		

附录 VII

浮点BASIC(APPLESOFT) 语言一览表

1. 简单变量

类 型	举 例	范 围
实 型	AB	+/-9.99999999 E +37
整 型	AB%	+/-32767
字 符 串 型	AB\$	0~255个字符

说明：这里 A 是字母，B 可以是字母也可以是数字。变量名可以由两个以上字符组成，但仅有前面两个字符是有意义的
例如：AB% 和 AB3QS% 是同一整型变量

2. 数组变量

类 型	举 例	说 明
实 型	AB(3,12,7)	数组的大小是受有效内存所限制的
整 型	AB%(3,12,7)	
字 符 串 型	AB\$(3,12,7)	

3. 代数运算符

符 号	意 义	说 明
=	赋值给变量	LET 是选用项
-	负 号	
^	乘 幂	不可写成“↑”
*	乘 号	不可写成“×”或“·”，也不可省略
/	除 号	不可写成“÷”
+	加 号	
-	减 号	

4. 关系和逻辑运算符

符 号	意 义	说 明
=	等 于	关系或逻辑表达式为真时,其值为 1;为假时,其值为 0,逻辑运算符也能够用于字符串的比较
<>	不 等 于	
<	小 于	
<=	小于或等于	
>	大 于	
>=	大于或等于	
NOT	逻辑“非”	
AND	逻辑“与”	
OR	逻辑“或”	

5. 系统和实用命令

命 令	功 能
LOAD	从磁盘或磁带上装入程序
SAVE	把程序存到磁带或磁盘上去
NEW	删除当前程序
RUN	从最小行号处开始运行程序
RUN477	从477行开始运行程序
STOP	停止执行并报告停止于那一行
END	停止执行，但不显示信息
CTRL C	用于立即执行方式，以停止程序或列表
RESET	无条件地转向监控程序。用 CTRL C 或 OG 来返回到 APPLESOFT
CONT	继续执行由 STOP, END 或 CTRL C 所终止的程序
TRACE	提供调试手段，列出它每次新执行到的每一行的行号
NOTRACE	关闭 TRACE
PEEK(X)	送出存储在单元 X 的内容
POKE X, 13	将内存单元 X 的内容修改成13
WAIT X, Y, Z	等待，直到 X 单元的内容与 Z 进行“异或”(XOR)运算，将中间结果再与 Y 进行“与”(AND)运算，所得到的值为非零时为止
CALL X	转向以内存单元 X 开始的机器语言子程序
USR(X)	把值 X 送给机器语言子程序
HIMEM:	置 APPLESOFT 程序可使用的最高的内存有效地址

续表

命 令	功 能
LOMEM:	置APPLESOFT 程序可使用的最低的内存有效地址

6. 编辑命令和与格式有关的命令

命 令	功 能
LIST	列出整个程序清单
LIST X-Y	列出从 X 行到 Y 行的程序段
DEL X,Y	删除从 X 行到 Y 行的程序段
REM XYZ	用来书写程序的注解, 程序将这部分忽略
VTAB Y	把光标移到 Y 行 (1~24行)
HTAB X	把光标移到 X 列位置上 (1~40列)
TAB(X)	仅用于 PRINT 语句, 它把光标移到 X 列位置上 (1~40列)
POS(O)	送出光标当前所在的水平位置的值 (0~39)
SPC(X)	仅用于 PRINT 语句, 在最后一次打印的项和要打印的下一项之间放上 X 个空格
HOME	清除显示屏, 将光标移到左上角
CLEAR	所有的变量重置为零
FRE(0)	送出用户还可使用的内存总数
FLASH	置电脑的输出为闪烁方式
INVERSE	置电脑的输出为白底黑字方式
NORMAL	关闭闪烁或白底黑字方式

续表

命 令	功 能
SPEED=X	置字符输出速率为X(从0~255)
ESC A	把光标向右移一格
ESC B	把光标向左移一格。
ESC C	把光标向下移一格
ESC D	把光标向上移一格
右箭头(→)	把光标下的字符输入内存, 并把光标向右移一格
左箭头(←)	删除当前行中刚刚键入的字符, 并把光标向左移一格
CTRL X	删除当前键入的行

7. 有关数组和字符串的命令

命 令	功 能
DIMA(X,Y,Z)	置A的最大下标, 保留 $X+1 \times Y+1 \times Z+1$ 个实型元素的存储空间, 由A(0,0,0)元素开始
DIMAS(X,Y)	置A\$的最大下标, 它可以包含 $X+1 \times Y+1 \times Z+1$ 个字符串元素, 每个元素最多可达255个字符
LEN(A\$)	送出A\$中所含的字符个数
STR\$(X)	将X的数值转换成对应的字符串送出
VAL(A\$)	以A\$中的第一个非数字字符为界, 将其前面的字符转换成数值送出
CHR\$(X)	送出代码为X的ASCII字符
ASC(A\$)	送出A\$中的第一个字符的ASCII代码

续表

命 令	功 能
LEFT\$(A\$,X)	送出A\$中的最左边的X个字符
RIGHT\$(A\$,X)	送出A\$中的最右边的X个字符
MID\$(A\$,X,Y)	送出A\$中的从字符X开始的Y个字符
+	用于连接字符串的运算符
STORE A	把数值数组A保存到磁带上,不可直接用来存储字符串数组
BECALL B	把数值从磁带上装回, B数组必须已经被正确地用 DIM 定义过

8. 有关输入/输出的命令(同时参考 LOAD 和 SAVE, STORE 和 RECALL 命令)

命 令	功 能
INPUT A\$	在显示屏上显示一个问号(?), 等待用户键入一组字符串值给A\$
INPUT"XYZ", A	在显示屏上打印出XYZ, 等待用户键入一个实数值给A
GET A\$	等待用户键入单个字符值给A\$, 它不需要按 RETURN 键
DATA X,"Y",Z	建立 READ 语句能使用的数据元素表
READ A\$	将下一个 DATA 的元素赋给A\$
RESTORE	重新从第一个 DATA 元素开始, 用 READ 语句来读
PRINT"X="; X	在显示屏上打印字符串"X="和变量X的值, 分号表示紧接着上一项打印, 逗号则表示各打印项之间留有三个制表域(tab field)的长度, 即一个打印三项, 每项起始位置分别为1, 17, 33列。符号"?"也表示 PRINT 语句

续表

命 令	功 能
IN#6	从#6插槽上的外部设备中进行预输入,而不是从键盘(IN#0)上进行
PR#6	置输出为#6插槽上的外部设备,而不是显示屏(PR#0)
LET X=Y	把Y的值赋给变量X,LET是选用项
DEF FN A(X)=X+23/X	定义一个函数FNA,在后面使用时,FNA的自变量由定义中的表达式X来代替。FNA(4)将送出值9.75

9. 有关流程控制的命令

命 令	功 能
GO TO n	转向第n行
IF X=n THEN m	如果断言X=n那么就转到m行; 如果X≠n,那么执行紧接着的下一行号语句
FOR X=n TO m STEP Q : NEXT X	执行FOR语句和相应的NEXT语句之间的所有语句。X从n开始,步长为Q,直至X大于m,才转向NEXT后面的语句 如果没有指明步长(STEP)的大小。则表明STEP为1
NEXT X	定义3 FOR...NEXT循环语句的底,X是选用项
GOSUB n	转向行号为n开始的写程序
RETURN	表明子程序的结束,返回到最后的GOSUB的下面一行语句处执行
POP	从RETURN地址栈中移出一个地址

命 令	功 能
ONX GOTO m, n, p	开关语句, 转向表中所列出的第 X 个行号所指出的语句, 如果 $X=2$, 就转向 n 行; 如果 $X=3$, 就转向 p 行
ON X GOSUB m, n, p	转向表中第 X 个行号的子程序
ONERR GOTO m	在后继语句执行中, 如果产生错误, 使程序转到错误处理程序(在 m 行)上执行, 而不是显示信息, 也不停止程序的运行
RFSUME	在错误处理程序中, 使得返回到发生错误的语句处执行

10. 有关作图和游戏控制命令

命 令		功 能
低 分 辨 率 作 图	GR	置低分辨率作图方式, 使显示屏上部40行40列用以显示图形, 底部留有4行显示文字资料
	COLOR=X	为画下一个点置颜色(0~15)
	PLOT X,Y	将有色的点显示于横坐标为X, 纵坐标为Y的位置上, X和Y在0到39范围内, (0,0)为屏幕左上角
	HLIN X1,X2 AT Y	从点(X1,Y)到点(X2,Y)画一条水平线
	VLIN Y1,Y2 AT X	从点(X,Y1)到点(X,Y2)画一条垂直线
	SCRN(X,Y)	送出位于显示屏上(X,Y)这一点的颜色数

续表

命令		功能
高分辨率作图	MGR	置高分辨率作图方式(第1页):使显于屏上部160行280列(即280×160个像素的点阵区域)用以显示图形,底部留出4行显示文字
	HGR2	置第2页高分辨率作图方式,将整个显示屏用以显示图形(即280×192个像素的点阵区域)
	HCOLOR=X	为画下一个点置颜色(0~7)
	H PLOT X, Y	把有色点置在横坐标为X,纵坐标为Y的位置上,X的变化范围从0到279,Y的变化范围从0到159(对HGR),或到191(对HGR2),(0,0)为显示屏的左上角
	H PLOT X1,Y1 TO X2,Y2	从点X1,Y1到点X2,Y2画一条线,命令还可以扩充,(诸如加上TOXN,YN等点)
	SHLOAD	从磁带上装入一个图形表
	DRAW 3 AT X,Y	按先前所装入的图形表中所定义的*3形状来绘图,起始位置为X,Y,颜色由HCOLOR所确定
	XDRAW 3 AT X,Y	按图形表中所确定的*3形状来绘图,所画的每一点颜色数是屏幕上那一点原有颜色数的补数
	ROT=X	为执行DRAW或XDRAW:可将形状进行旋转,ROT=0表示垂直,ROT=16表示顺时针方向右旋90°,ROT=32表示顺时针方向右旋180°等等
游戏	SCALE=X	置DRAW或XDRAW图形的比例因子(1到255)游戏控制
	PDL(X)	送出游戏控制器X(0~3)所置的值,PDL(X)的值可在0~255间进行变化
	PEEK(X-16287)	如果其值大于127,表示游戏控制器X(0到2)上的按钮已被按下
	PEEK(-16336)	使APPLE的喇叭发出“哔哔”声

11. 常用数学函数

函 数	说 明
SIN(X)	表示X弧度的正弦值
COS(X)	表示X弧度的余弦值
TAN(X)	表示X弧度的正切值
ATN(X)	表示X的反正切值，以弧度为单位
INT(X)	表示小于等于X的最大整数
RND(1)	每次使用时，总送出一个 0 ~ 0.999999999 的随机数
RND(0)	再度送出上一次的随机数
RND(-3)	送出值为4.48217179E--08，对于每一个不同的负自变量，总送出相应于不同变量的一个固定值。使用该随机数后，带有正自变量的 RND 函数，也产生一个固定序列
SGN(X)	若 $X > 0$ 函数值为 1；若 $X = 0$ ，函数值为 0；若 $X < 0$ ，函数值为 -1
ABS(X)	表示X的绝对值
SQR(X)	表示X的正平方根
EXP(X)	表示e的X次方的值， $e = 2.718289$
LOG(X)	表示X的自然对数的值

附录Ⅱ

微电脑常用术语

存取时间 (Access Time) 自调用数据至取出数据所需的时间，或自送入数据到完成存贮所需的时间。

累加器 (Accumulator) 用来存放算术和逻辑运算结果的中央处理单元 (cpu) 中的主要寄存器。有些微电脑具有两个累加器或把多个寄存器用作累加器。

字首组合同 (Acronym) 用名词、术语或短语的各个单词的第一个字母或前几个字母组合成的词。例如 BASIC 是 Beginners All-Purpose Symbolic Instruction Code 的字首组合同。

地址 (Address) 算术存贮单元或输入/输出设备口的二进制数字。通常用十六进制数表示。

算法 (Algorithm) 解决某一问题的一套规则 (通常指数学规则)。

算术及逻辑运算部件 (ALU) 能够对数据字完成算术和逻辑运算的逻辑电路。ALU 为微电脑提供了加、减、移位、加 1、减 1、“与”、“或”及“异或”的功能。

程序语言 (APL) 一种会话语言。它包含许多构造与处理数组的运算符，能简洁、紧凑地写出源程序。是较现代的数学定向程序语言之一。

数组 阵列 (Array) 按一定的格式排列起来的一列同一属性的项目。

美国信息交换标准码 (ASCII) ASCII 是用于字母和数字数据串行传输的 8 单位码。头 7 位表示 128 个标准 ASCII 的字符，第 8 位是用来检错的奇偶位。在 8 个数据位前有一个起始位，数据位后面有

一到两个终止位。

汇编程序 (Assembler) 一种符号翻译程序,它把助记符转换成机器码。

汇编语言 (Assembly Language) 一种介于机器语言与编译语言 (高级语言) 之间的电脑语言。它面向机器,是为特定的电脑系列设计的,采用助记符和标准缩略语的形式书写的低级程序语言。

辅助存贮器 (Auxiliary storage) 为弥补电脑内部主存贮器容量不足而使用的一种存贮器。

波特速率 (Baud Rate) 表示数据传输速率。对异步的字来说,这是一个字内每位数据的传输速率。

初学者通用符号指令码 (BASIC) 是一种用于编制程序的人机对话语言,最初由达特默思学院提出。这种语言允许使用简单的英文字和普通的数学符号来完成解题所需的算术和逻辑运算。

二进制数字系统 (Binary Number System) 用符号 1 和 0 表示量值的数字系统。数字逻辑就使用这种二进制信息。

比特 (Bit) 二进制数中的位。是数字系统或微电脑中最小的数据单位。

总线 (Bus) 微处理机和其他有关装置间进行数字通讯的通道。总线一般传输双向数据或单向地址信息。总线通常具有 4、8、12 或 16 条并行线,每条线传输 4、8、12 和 16 位字中的一位。

字节 (Byte) 作为一个单位来处理的一串二进制数,最常用的是 8 位二进制数。

卡片 (Card) 一种存贮介质,是可以在它上面垂直于纸面穿上一排排小孔的方法以记录文字、数据或符号的纸片。

进位 (Carry) 进行加法运算时产生的溢出。在微电脑中,大于所允许的字长时,就产生进位。

盒式磁带 (Cassette) 一种磁带存贮装置,磁带装在塑料盒内。

阴极射线管 (Cathod Ray Tube) 具有屏幕并可显示信息的真

空管。包含一个阴极射线管的键盘终端，通常称作 CRT 终端。

中央处理单元 (Central Processing Unit) 电脑系统的一个组成部分，它包含算术逻辑单元和控制线路，简称 CPU。

字符 (Character) 用于表示信息的字母、数字、标记或其他符号。人们已设计了一整套电脑的输入、输出、贮存和操作用的字符。

芯片 (Chip) 制有集成电路的微小硅片，其间含有大量单元电路。芯片这个词常可以和集成电路一词互换使用。

时钟 (Clock) 为微处理机、存储器 and 输入/输出设备提供公用时钟信号的脉冲发生器。时钟通常分为单相或双相 (1 Ø 或 2 Ø)。

代码 编码 (Code) 表示信息的一整套符号和规定。

面向商业的通用语言 (COBOL) 是 Common Business Oriented Language 的字首组合词，是一种主要用于商业及数据处理的类似英语的程序设计语言，是一种应用最普遍的编译程序语言。

编译程序 (Compiler) 一种电脑程序 (即软件)，它把诸如 BASIC、COBOL 或 FORTRAN 等用符号书写的面向过程的语言，翻译转换成面向电脑的代码语言 (即机器语言)。

编译程序语言 (Compiler Language) 是一种有助于电脑使用者为解决问题而书写的面向过程的语言。诸如 BASIC、COBOL、APL、FORTRAN 等。它比较接近日常用语。

电脑系统 (Computer System) 数字电脑及其有关的外围设备与软件。

数据 (Data) 能够输入到电脑，并由电脑进行处理的数字、字母和符号。

数据库 (Data Base) 从多方面收集到的数据的集合，以满足对信息的处理和检索的需要。

数据处理 (Data Processing) 是关于用自动设备 (它可以包括电脑在内，也可以不包括电脑) 进行操作的术语，是信息处理的同义词。

故障排除 (Debugging) 从流程图或电脑程序及硬件中消除错误和故障的过程。

诊断程序 (Diagnostic Programs) 用于校验微电脑系统某些部分的程序，用以保证微电脑可靠运行。

直接存取 随机存取 (Direct Access) 将数据存入存储器或从存储器取出，所需的时间与所在的物理位置无关。一台微电脑的主存储器就是随机存取存储器，而磁盘是另一种随机存取存储器。

磁盘存储器 (Disk Storage) 在旋转盘面上进行磁记录的一种存储装置。是一种随机存取存储器。

文件编制 (Documentation) 对一个具体问题组织信息，使其成为一个有用的文件的过程。通常包括问题的陈述、有关算法、流程图、穿孔纸带、程序编制和程序操作指令等项目。

编辑程序 (Editor) 具有编辑功能的程序。这种程序允许程序员使用微处理机和一个终端设备作为高级打字机来编写和修正源码。

取指令 (Fetch) 从存储器中读出下一步程序并把它存储在指令寄存器中。

文件 (File) 表示目的、形式和内容上彼此相似的信息的有机集合。

固件 (Firmware) 软件指令或数据用固定的或“坚实”的方法贮存起来，通常采用 ROM 形式，一般软件程序贮存在磁盘上，而它却装进微电脑主机存储器内。

触发器 (Flip-Flop) 具有两种稳定状态的逻辑电路。它可以根据命令置“1”或置“0”。

软磁盘 (Floppy Disk) 一种用于电脑的辅助存储装置。它装在一个塑料护套内，一般常用的有 8 英寸和与 $5\frac{1}{4}$ 英寸直径两种。

流程图 (Flowchart) 一种电脑解题的图象描述，作为程序设计的思路体现。包含有箭头、矩形、菱形、圆形、平行四边形等框图和符号，直观地表现用电脑解题的过程和形式。

公式翻译程序语言 (FORTRAN) 是 FOR mula TRAN sla-
tor 的字首组合词, 是用于科学计算的面向过程的电脑语言。

硬拷贝 (Hard Copy) 微电脑打印出来的输出。这是相对阴极射线管或光电二极管上的信息显示方式而言的。

硬件 (Hard ware) 电脑系统的物理设备 (即实际装置), 是相对于软件而言的。

十六进制 (Hexidecimal) 一个由 16 个字母组成的数字系统; 常用字母 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F。十六进制通常用来表示从 0000 到 1111 的四位二进制数字。

高级语言 (High Level Language) 一种使用英文字母、十进制算术和普通代数式的电脑语言。每一行或每句语句代表了大量的电脑运算。

信息 (Information) 由人们所指定的数据所包含的意义, 它不随载荷它的物理设备形式的改变而改变。

信息处理 (Information Processing) 数据所表示的信息的处理及决定所处理的数据的意义。

信息检索 (Information Retrieval) 对大量情报资料进行分类、存贮和查找, 得到所需信息的过程和技术。这是电脑科学的一个分支。

指令系统 (Instruction Set) 一台微电脑按给定二进制代码所能执行的全部操作, 即一台微电脑的全部指令。

接口 (Interface) 微电脑与某些给定的装置进行信息交换所需的电路。这些装置可以是存贮器或输入/输出设备。接口是电子装置之间实现互连的通用术语。

解释程序 (Interpreter) 按步解题的一种程序, 而不是在处理前把全部问题转换成机器语言。

中断 (Interrupt) 外部设备通过输入线向微处理机请求为它服务。中断程序先使微处理机完成正在执行的指令周期, 然后为外部

设备服务。

集成电路 (Integrated Circuit) 制造在一块芯片上的电子电路，芯片上集成着大量相互连接的晶体管等器件，以完成某种功能。集成电路有几种规模：小规模集成电路 (SSI)，小于20门；中规模集成电路 (MSI)，20~200 门；大规模集成电路 (LSI)，大于200门，目前已制成了超大规模集成电路 (VLSI)。微电脑就是利用大规模集成电路技术制造的。

转移 (Jump) 离开指令执行的正常序列，使程序计数器从新位置开始工作。转移可以有条件的或无条件的。

循环 (Loop) 一种编制程序的技术，使电脑重复执行同一指令，直到满足结束条件为止。

程序库 (Library) 可供今后引用的经过组织的电脑程序集合。这些程序通常贮存在软盘、磁带或纸带上。

机器语言 (Machine Language) 电脑能“懂得”，能够执行的语言或指令，是电脑的基本语言。

助记符 (Mnemonics) 把微电脑指令系统用英文缩写字母表示的一种形式。助记符可以由程序员转换成二进制指令，这叫做人工汇编。将助记符转换成机器码的程序叫做汇编程序。

磁芯 (Magnetic Core) 一种数据存贮器，是用磁性材料制成的环形体。当穿过它的导线或绕组通过电流时，它就磁化并形成脉冲，它可以取两种磁化状态中的一种，并保持它。

监督程序 (Monitor) 一种监督操作、处理和对其他程序进行分时程序。

操作码 (Opcode) 在指令字中，用以标明该指令执行什么样的基本功能的代码部分。

并行 (Parallel) 把数字的所有位同时从一处移到另一处。

编程序 (Programming) 编写解决特定问题或完成特定任务的一系列电脑指令。

可编程只读存储器 (PROM) 一种可由用户编制程序而不是由厂商来编制程序的只读存储器, 通过专门编程的设备进行编制。但只能编一次。

随机存取存储器 (RAM) 按并行二进制地址字随机存取而不是按时间依次存取的任何读/写存储器。

只读存储器 (ROM) 存有永久性信息的存储器。只能读出信息, 不能写入信息。

运行 (Run) 微电脑连续进行取指令和执行指令的状态。

串行 (Serial) 以每次一位的方式传输数据字, 而不是在多根线上同时传输整个字。

软件 (Software) 支持微电脑全部程序和编程的资料。

K 字母K被用来代表数 2^{10} , 即1024。电脑的存储容量常用多少K字或字节来表示。

源程序 (Source Program) 为了对一个问题进行逻辑解题所编写的一组指令。在电脑执行这一程序前, 必须由汇编程序、编译程序或解释程序等把它翻译成机器语言。

栈 (Stack) 一系列用于临时存放程序计数器内容的寄存器或存储单元。

子程序 (Subroutine) 用来完成一个特殊功能的短程序。经常被使用于主程序中的很多部分, 或为许多不同的主程序所公用。

真值表 (Truth Table) 用来表示一个逻辑电路全部可能的输入状态和输出结果的一种方式。

字 (Word) 微电脑所用的位数固定的二进制代码。此代码是作为一个整体来处理 and 运算的。一个字通常分为若干字节。