

娃娃用电脑

WAWA YONG DIANNIAO



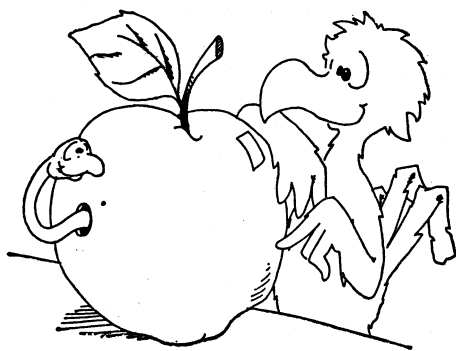
统一书号： R 13024·207

定 价： 1.20 元

娃娃用电脑

胡金初 王钧权 周宠安编译

徐培南 审校



少年儿童出版社

内 容 提 要

本书根据美国 EDWARD H. CARLSON 教授所著《KIDS AND THE APPLE》(娃娃和苹果机)一书编译而成。原著于 1982 年出版, 1983 年又再版了两次, 深受美国小读者的欢迎。它以 APPLE II 型微电脑为主, 讲述 BASIC 语言的程序设计, 在其他微电脑上有较大通用性。

本书是针对少年儿童的特点编写的: 叙述具体细致, 循序渐进, 便于读者理解和掌握; 例题丰富, 大都是游戏和图形, 趣味性强; 课文的编排也很有特色, 每课课文前设有指导性的“教学参考”和复习题, 课后有作业, 书末有答案。全书共 33 课, 图文并茂, 生动活泼, 循循善诱, 一学就懂, 引人入胜, 是一本学习使用微电脑的实用书。

本书适合高小、初中学生学习, 也可供一般初学电脑者作自学教材。

目 录

致小读者	(1)
致家长	(3)
致老师	(4)
有关程序编制问题	(5)
本书简介	(7)

第一部分 语句介绍

1. HOME(清屏幕)语句、PRINT(打印)语句、 NEW(清内存)命令和 RUN(执行)命令	(1)
2. 发声功能和 FLASH(闪烁)语句、INVERSE (白底黑字)语句、NORMAL(黑底白字)语句	(9)
3. LIST(列表)命令和 REM (注释)语句	(14)
4. 专用键	(22)
5. INPUT (键盘输入)语句	(27)
6. 打印方法与 SPEED(速度控制)语句	(31)
7. LET(赋值)语句	(36)
8. GOTO(转向)语句与RESET(复位)键	(41)
9. IF(条件)语句	(46)
10. 数字	(53)

- 11. TAB(打印格式)函数与延时循环 (60)
- 12. 带有数字的 IF (条件)语句 (66)
- 13. RND(随机数)函数和 INT(取整) 函数 (72)
- 14. 信息存入磁盘 (79)

第二部分 绘图、游戏及其他

- 15. 语句的简略形式 (86)
- 16. VTAB(纵向定位)语句和HTAB
(横向定位)语句 (93)
- 17. FOR-NEXT(循环) 语句 (98)
- 18. 编辑方式与执行方式 (104)
- 19. 用字符串移动图象 (109)
- 20. 变量名 (113)
- 21. LO-RES(低分辨率)图形 (117)
- 22. 用HLIN(水平线)语句和VLIN(垂直线)
语句绘图 (124)
- 23. 密写与GET(取数)语句 (130)
- 24. GOSUB(转子程序)语句、RETURN(返回)
语句和 END(结束)语句 (136)

第三部分 进一步编制程序的方法

- 25. 行编辑 (146)
- 26. 字符串函数 (151)
- 27. 字符串和数的变换 (158)
- 28. 游戏棒 (163)

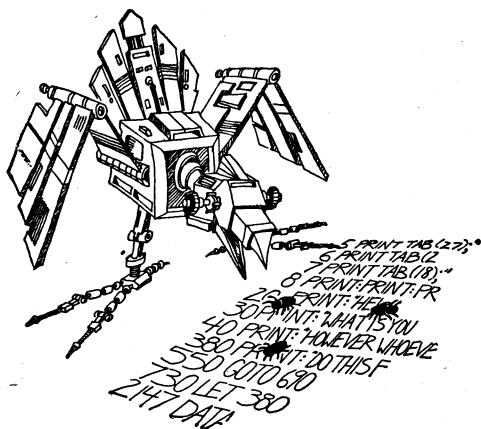
29. ASCII代码、键盘和 ON-GOTO(多路转向)	
语句	(171)
30. 数组和DIM(数组说明)语句	(181)
31. AND(与)、OR(或)、NOT(非)逻辑关系	(187)
32. 方便用户的程序	(196)
33. 程序的调试和 STOP(暂停)语句、CTRL-C(控制)	
键、CONT (继续执行)命令	(205)
附录一 磁盘使用法	(211)
附录二 将程序存入磁带	(214)
附录三 保留字表	(218)
附录四 作业题答案	(220)
附录五 错误信息表	(274)

致小读者

小读者们！这是一本教你们怎样给“苹果”电脑编写程序的书。从这书中，你们将学会编写一些游戏程序，如单人游戏、棋类游戏，以及文字游戏等等。这样，你们就可以用自己创作的游戏，邀请伙伴们进行比赛，一起愉快地度过一部分课余时间。

也许，通过编制一些特殊的专用程序，还可以帮你解决许多一时无法解决的问题。

你们还可以编写一些算术运算或文字拼写的练习程序去帮助你们的弟妹们。甚至还可以通过编写程序使自己对学习历史、



外语感到更加轻松。

那么，怎么使用这本书呢？一定要把所有的例题和作业都做一遍。如果发现有什么地方弄不懂，就应该把学过的课程从头开始，认真仔细地再读一遍，看看是不是有什么细节被忽略了，或没搞清楚。要是经过自己的努力还是没弄懂，那就该向你们的父母或老师请教。

书中每一课都附有复习题。只有能明确无误地回答出这些复习题，你才能说：这节课我已经学完了。

愿幸福的蓝鸟把你学习中的害虫全部啄食干净！

致 家 长

这本书是专为十至十四岁的孩子学习“苹果”电脑语言 (APPLESOFT BASIC) 而编写的。书内有学习指导、内容解释、练习题、复习题和测验题。书后附有作业答案。你们的孩子开始学习此书的时候,需要你们给他们以帮助;在他们遇到困难的时候,尤其需要你们给予热情的鼓励。

学习编写程序不是一件容易的事。因为它需要运用一些较为复杂的概念。它还要求一丝不苟和准确无误,而孩子们就恰恰缺乏这些素质。正因为如此,让少年儿童学习编制程序是极为有益的。孩子们倘能长期坚持学习,逐步掌握书中的知识,不断地编写出一些饶有趣味的程序来,就能从中得益非浅。

使用本书说明:全书共分 33 课,每课都附有家长必读的“教学参考”。它概括了本课要学的内容,给予若干有用的提示,并提供了一些(通常在电脑上做的)复习题,以便检查孩子们是不是已经掌握了这些技能和概念。

“教学参考”主要是为家长编写的,但年龄大一点的孩子读一读这些“教学参考”也会有所收获。年龄小的孩子就不一定要读它,因为他们自己能从所学的课文里获得所需的知识。至于对幼童来说,最好的办法是,家长跟他们一起朗读课文,再展开讨论,来加深他们对课文的理解,然后再让他们动手做。

致 老 师

这本书是为中、小学生编写的。可供学生在带磁盘或磁带的苹果电脑上学习 APPLESOFT BASIC 电脑语言。课文内有解释(包括插图),有例题,有作业题。每课书里附有为教师编写的“教学参考”,对教材内容作概括的介绍,并提供一些有用的提示和较为合适的复习题。

本书是供自学用的,但是也可作为学校的教科书。

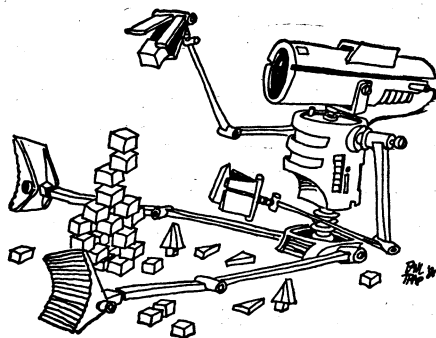
从广义来讲,这本书是使用 BASIC 语言来教授程序编制,而不是用来教 BASIC 语言的。编写程序可以开发人的智力,教会人们理解各种各样的新概念。其中有些概念就是各种程序。它们可以被命名,分成几个基本的部分,并进行调试。其他一些概念包括:把程序按智力能接受的大小分割成模块,在分级系统中,组织好模块的编排,构成循环,以便重复执行这些模块,并进行条件测试(用 IF.....THEN 语句)等等。

教师应将每一概念与学生的日常经历联系在一起,选择恰当的语言,合适的例题,形象的插图和生动的比喻,来帮助儿童熟悉这一门“陌生”的学科。

有关程序编制问题

通常有一种误解,认为编制电脑程序完全和做算术题一样,其实并不是这样。电脑的工作与孩子们活动中最为相似的要算是搭积木和写作文了。

BASIC语言就象一套积木,重复使用少量的语句(积木块),按照一定的规则组合,就能得到奇妙的结果。随着孩子们对电脑的不断了解,编制程序的技能也会日积月累,不断提高,使每个语句所起的作用越来越大。就象孩子一开始只会用“三角形积木”做屋顶,而过一些时候,他们就能用两块这样的积木搭出一棵挺拔的松树。



一个程序就象一篇文章，是满足某种需要的一件作品。孩子们要写一篇题为“我怎样度过暑假”的作文，就要采用某一种写作文体。对初学者来说，一开始可能会感到紧张，手忙脚乱，不知所措。但经过几年锻炼后，就可以从容写作了。他们已经不需要花多少时间就能写出较好的段落，在构思文章的结构方面所花的时间就更少；随着写作技能的不断提高，他们就能得心应手，运用自如了。现在只要先考虑一下体裁，然后注意一下选词和标点符号就行了。

正如孩子们所看到的，电脑的工作与做算术也确实存在某些相似之处，它们都需要学习一些严格的规则。在算术中要学习运算过程，而在编制程序中则要学习语法。即使是微不足道的疏忽都会使最后的结果完全搞错。在编制程序中，富有成效的方法是：把有错误的程序看成是有一部分是正确的，然后对它的错误部分进行修改，这是编制程序中常遇到的事。做算术题时，能发挥创造性的范围是很有限的，而在编写程序的过程中却特别强调富有创造性，两者形成了鲜明的对照。

编制程序的好处是为儿童提供了一种教育方法，而这在其他地方是不容易得到的。只要按一下键盘，在屏幕上就会自动产生或消失一些文字和各种图案，这就能使我们将精力集中在学习主要的东西上：分析电脑为什么不按我的要求做；怎样合成各部分的程序，包括循环程序、子程序以及各种语句。学习编写电脑程序能提高办事效率，电脑还能自动指出程序中存在的语法错误。

在程序编写中学到的分析和合成的能力，在其他许多领域中都能应用，这对培养儿童的思维能力和工作方法很有益处。

本书简介

本书共有三十三篇课文，每课课文前都设有为教师或家长写的具有指导性的“教学参考”和复习题，课后设置一些作业题，书末有答案。课文和指导性的“教学参考”风格不同，课文主要讲实际使用的方法，而“教学参考”则是对课文作比较详细的介绍或者解释。

对于教师或家长来说，如果对 BASIC 语言还不甚了解，或者也是一个初学者，那也不要紧，每课学生学的课文就是你学 BASIC 语言理想的入门向导。BASIC 语言是一种电脑上用的高级语言，由于这种语言比较简单，便于解释执行，所以绝大多数的微电脑上都配有 BASIC 语言。BASIC 是英文 *Beginner's All-purpose Symbolic Instruction Code* 的缩写，意思是初学者通用符号指令码。它并不神秘，也不难学。尽管各种不同的微电脑所使用的 BASIC 语言不完全相同，但其基本部分是一样的。因此本书介绍的内容，在各类微电脑上具有较大的通用性。

本书共分三个部分。第一部分介绍常用的语句，一开始先对程序的编制作一个概述，然后很快就转到编写兴趣盎然的程序上。第二部分介绍如何用一些功能较强的语句来作图和编写简单的游戏程序，通过生动有趣的图案来提高学生的兴趣，使他们能较快地掌握简单程序的编制方法。最后一部分进一步介绍

编制程序的方法，除了继续引入一些有用的语句外，还教你学会编写各种各样程序的技能，例如程序的编辑和纠错方面的技术，使学生通过本书的学习，能自己动手编制一些有用的程序。

每一课课文后面的作业，包括编写程序，通常是短小的程序。当然，作业的理想“答案”不是唯一的，书的末尾给大部分作业题的程序提供了一个参考答案，其中有一些答案还是由使用过本书的小读者自己做的呢。

书中的第十四课信息存入磁盘和第十八课编辑方式和执行方式这两课，由于内容上具有独立性，在学过第一课以后，随时都可以学习。

第一部分 语句介绍

第一课 HOME(清屏幕)语句、PRINT (打印)语句、NEW(清内存)命令和RUN (执行)命令

教学参考:

本课简要介绍如何使用电脑,其中包括如何开机等,这些内容都安排在下文的正文里,如有不明确的地方可由老师或家长来补充。

书后面的附录一是谈磁盘的保养与用法,里面附有磁盘使用说明以及教你如何输入有用的问候HELLO(您好)程序等。

本课介绍的语句和命令有:

“HOME”, “PRINT”, “NEW”和“RUN”。

课文内容还包括:

1. 如何开机。
2. 用键盘输入各种语句,然后按RETURN(回车)键。
3. 什么是程序以及标有序号的语句行。
4. 用HOME语句清除屏幕。
5. 用NEW命令清除存储器。
6. 说明屏幕上显示的内容与存储器所保存的内容是不同的(这个概念一开始可能对学生来说是难以理解的)。

7. RUN 命令能使电脑把存储器中的各行语句按行号的先后次序执行。

8. 语句行号选用时可以跳过几个号码,说明了为什么这样做。

复习题:

1. 写一个打印你姓名(用汉语拼音)的程序。

2. 把这个程序从屏幕上擦掉,但要保留在存储器内。

3. 执行这个程序。

4. 把这个程序从存储器内抹掉。

5. 写一个能清除屏幕,然后打印“HELLO”(您好)的新程序。

6. 执行新程序。

7. 将新程序从存储器中抹去,但要留在屏幕上。

课 文

如何开机

假如放在你面前的一台电脑是苹果 II 型机,那么:

1. 找一张供你专用的磁盘片或一张名为“DOS 3.3 系统盘”的盘片。

2. 拿住磁盘片,使标签朝上,大拇指按在标签上。

3. 小心地将盘片放入 1 号驱动器中,然后关上驱动器的门。

4. 合上电脑背面左边的电源开关。

5. 接通屏幕显示器的电源。如果是其他类型的电脑,请遵照各自的说明书操作。

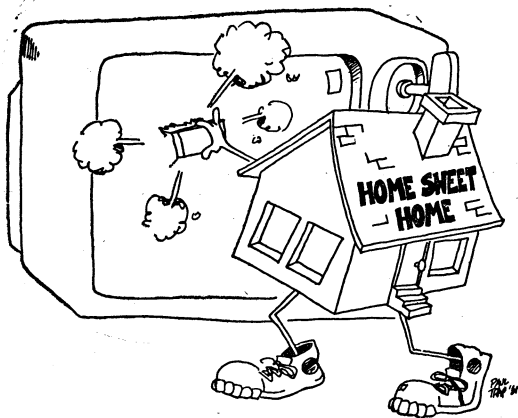
按键操作

开机后，可以随便从键盘上键入一些内容，比如：ABCDE，它们马上会显示在屏幕上。



命令电脑工作

试试看，用键盘输入：



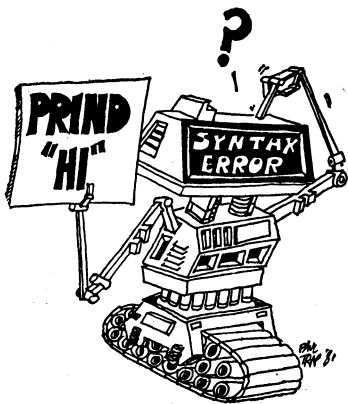
HOME

然后再按“RETURN”(回车)键。

“HOME”(清屏幕)语句告诉电脑将屏幕上的内容抹去,并将光标移至屏幕的左上角。

看一看,电脑屏幕所显示的是否符合上述要求。

如果电脑没有按照你的命令工作,而是发出“嘟”声,并显示“SYNTAX ERROR”(语法错误)的信息,说明你的操作有错误,这时得重新正确地键入HOME,然后按RETURN(回车)键。



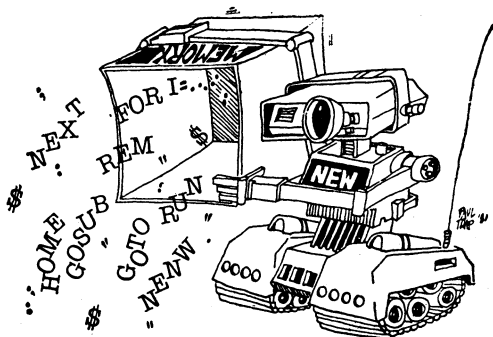
电脑大约只懂 100 个单词,本课学的只是其中的 4 个,所以你使用时,需要弄清楚电脑懂的是哪些词。

NEW(清内存)命令

由键盘输入:

NEW

然后按 RETURN(回车)键。



MEMORY——存储器

NEW(清内存)命令的意思是清除存储器内原来的内容,这样就可以把新的程序放进存储器。

怎样输入一行语句

当我们讲“输入”时,就意味着要做两件事:

1. 键入一行内容;
2. 然后按RETURN(回车)键。

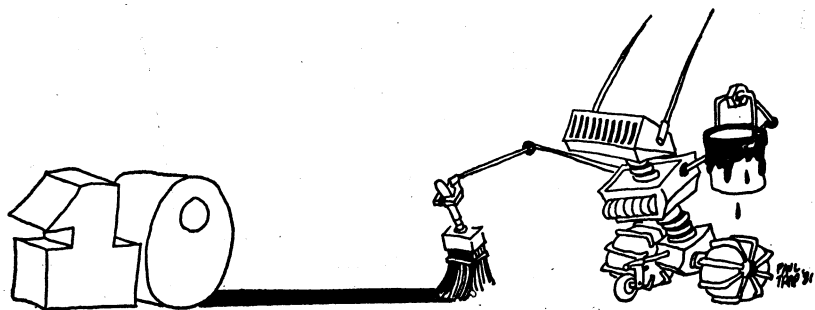
例如输入这样一行:

```
10 PRINT "HI"
```

注意,这里要输入”引号,方法是先按住键盘上的SHIFT——换档键,同时再按一下标有2和”符号的那个键就行了。

在这行输入结束时要按一下RETURN(回车)键,表示一行内容输入结束。这样标有10的那一行语句已进入电脑的存储器,并可以一直呆在那儿,直到输入NEW(清内存)命令或者把电脑的电源关掉为止。当然这是一个极短的程序。

数字0和字母“O”的表示



为了避免数字 0 和字母 O 的混淆，在电脑中总是把数字 0 写成这样：

数字 0:

而把字母 O 写成：

字母 O:

因此写这两个符号时一定要小心。

正确: 1 HOME

错误: 1 HOME

什么是程序

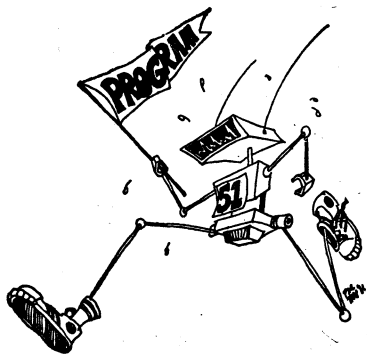
程序是人们要求电脑执行的一系列语句的组合，这些语句往往被列成行，每一行由一个数字编号开头(叫行号或标号)。上面所输入的程序仅有一行语句。

怎样执行一个程序

刚才已把下面这个程序放在存储器内了：



PROGRAM——程序



PROGRAM——程序

```
10 PRINT "HI"
```

现在输入：

```
RUN
```

记住，按RETURN——回车键了吗？

RUN(执行)命令告诉电脑，看看存储器中有没有程序，如果有，就按程序的行号大小顺序执行各个语句。

上面的10号语句，电脑有没有执行PRINT(打印)语句呢？PRINT语句是要电脑把引号里的内容打印在屏幕上，显示出来。电脑于是就打印出：

```
HI
```

怎样给程序中的一行行语句编号

请输入下列程序：

```
1 REM PROGRAM 1
```

```
2 HOME
```

3 PRINT "HI"

该程序有三行,每行有一个语句,其中HOME(清屏幕)语句和PRINT(打印)语句这两个语句我们已经学过了,REM(注释)语句在稍后就要学到。

让我们再看下面的程序:

```
10 REM PROGRAM 1
20 HOME
30 PRINT "HI"
```

这个程序除行号不同外,其他和上面的程序完全相同。行号可以取连续的数字,但在实际使用时常常跳过几个数字,这样在修改程序时,便于在原来的语句行之间插入新的语句行。在执行已经输入的程序时,电脑总是从标有最低行号的语句行开始,按行号的大小顺序依次执行。

作业1:

1. 使用 HOME 语句并解释其作用。
2. 使用 NEW 命令并说明它的功能。
3. 写一个使用一次 HOME 语句,二次 PRINT 语句的程序,然后用 RUN 命令执行该程序。

第二课 发声功能和 FLASH(闪烁)语句、INVERSE(白底黑字)语句、NORMAL(黑底白字)语句

教学参考:

本课一开始就介绍 PRINT CHR\$(7) 语句,该语句能使苹果机发出“啞”声。我们希望能多做一些发出声音的程序,从而使学生的程序变得丰富多采。

在课文中还要介绍“字符串常数”的概念。当数字以字符串的形式出现时,就不能把它当数字来直接参加运算,例如“19”是不能直接用在算术运算中的。



FLASH(闪烁)语句和 INVERSE(白底黑字)语句能使屏幕显示多样化,NORMAL(黑底白字)语句能将屏幕上的内容显示回到正常的显示方式。这些语句在编辑方式或一般程序中都能使用。

复习题:

1. 怎样使苹果机发出“嘟”声? 怎样抹掉屏幕上显示的内容?怎样清除存储器中的内容? 如何打印你的姓名?
2. 什么是“字符串”?什么是语句? 试举几例。
3. 要输入一行语句应按哪个特殊键?
4. 电脑打印出 SYNTAX ERROR 是什么意思?
5. 怎样用闪烁的字母打印“FIRE”(火)一词,并使电脑发出“嘟”声?

课 文

输入:

```
NEW
```

```
HOME
```

NEW(清内存)命令能把存储器的内容清除掉,HOME(清屏幕)语句能将屏幕上的内容抹去。掌握了这两句之后,我们就开始学第二课。

苹果机象小鸟一样发出“嘟”的叫声

输入下列程序:

```
10 HOME
```

```
20 PRINT CHR$(7)
```

```
RUN
```

电脑有没有发出“嘟”声？是 20 号语句使电脑发出“嘟”声的。注意在 () 内的数只能是 7。你在编制其他一些程序时，同样可用这方法使苹果机发出“嘟”声。

打印一行空行

运行下面的程序：

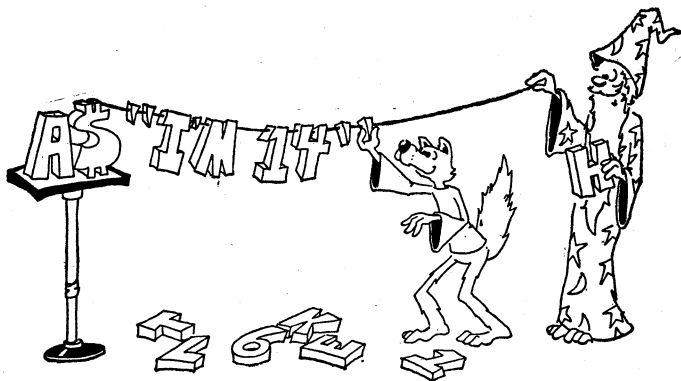
```
10 HOME  
20 PRINT "HERE IS A LINE"  
30 PRINT  
40 PRINT "ONE LINE WAS SKIPPED"
```

30号语句只打印一行空白，因为 PRINT 后面什么也没有。

字符串常数

看一下这些 PRINT(打印)语句：

```
10 PRINT "JOE"  
10 PRINT "#D47[[*[["  
10 PRINT "19"
```



```
10 PRINT "3.14159265"  
10 PRINT "I'M 14"  
10 PRINT " "
```

凡是在引号中的字母、数字及符号都称为字符，即使是空格也算作字符。例如：

```
10 PRINT " "
```

引号中是一个空格，但也叫字符。一般说来，排列成行的一串字符组成一个“字符串”。这些字符象一根绳子上的珠子一样串在一起。我们通常把引号内的字符串称为“字符串常数”，因为它是由一串字母、数字和符号组成的。所谓“常数”是因为它始终保持不变，不随程序的运行而变化。

闪烁显示

使用下面这些语句，可以使出现在屏幕上的内容显得更有趣味。

NORMAL 语句是指通常使用的黑底白字显示方式。

INVERSE 语句和上面的方式相反，是白底黑字显示方式。

FLASH 语句则为闪烁的显示方式。

切记！在程序中使用 FLASH 或 INVERSE 语句之后，在结尾须放上 NORMAL 语句，这样程序在运行结束时就可以显示出“OK”（一切正常）。

输入并执行下列程序：

```
10 REM INVERSE, FLASH AND NORMAL  
20 HOME  
30 PRINT "HI THERE"  
40 INVERSE
```



```
50 PRINT "INVERSE NOW"
```

```
60 NORMAL
```

```
70 PRINT "NORMAL AGAIN"
```

```
80 FLASH
```

```
90 PRINT "FLASHING NOW"
```

```
95 NORMAL
```

作业2:

1. 写一个程序,打印出你姓名的三个字(用汉语拼音),要求同时用三种不同的显示方式:姓用闪烁方式,名的两个字分别用白底黑字和黑底白字方式显示。

2. 在上面的程序中,再在每个字前面加一个“嘟”声。

第三课 LIST(列表)命令和REM(注释)语句

教学参考:

本课的内容包括:

1. 介绍 LIST、LIST 30 语句;
2. 用于标题说明和内容注解的 REM 语句;
3. 存放语句行的“盒子”——存储器;
4. 怎样从存储器中抹去一行语句;
5. 怎样在原来语句行之间加上一行语句;
6. 替换一行语句的方法;
7. 用 PRINT 语句绘图。

在使用电脑时,对“命令”和“语句”这两个词似乎显得有些不同。其实,它们之间的差异是人为造成的,在 BASIC 语言中并不严格地区分它们。以后,在学到一行上有几个语句或命令时,我们会对它们的含意作进一步的解释。

从现在起就需要让学生懂得:程序即使在屏幕上没有显示出来,也仍可能存放在存储器内。对于 LIST 语句,除了本课介绍的两种用法外,还有其他用法,例如:LIST 100-300 和 LIST-300,这在以后会学到。

为了形象地说明存储器的概念,我们把它比喻为一只只存放东西的盒子,这会有助于学生理解变量的概念,以及了解语句

中复杂表达式的详细工作情况。

学习 REM(注释)语句对学生来说,开始可能会引起混淆:如何区分在屏幕上显示的究竟是 PRINT 语句打印的内容,还是由键盘输入的注释内容,这需要给他们作些解释。在每堂课结束时用 PRINT(打印)语句绘几张图能顶上说许多话,这样学生们能很快地理解其中的意思。第四课后面的绘图将有助于提高学生编辑语句行的技巧。

复习题:

1. 怎样把你不再需要的语句行抹掉?
2. 在键入 HOME 语句之后,怎样把存储器里的所有语句显示在屏幕上?
3. 怎样用修改后的一行语句来替换有错的一行语句?
4. 假如要把新的一行语句插入到存储器内原来的两行语句之间,应该怎样做?
5. 解释一下电脑怎样把程序行放进存储器的“盒子”里的,电脑在盒子的前面写上了些什么?

课 文

先输入:

HOME

NEW

每课开始时用这两个语句,是为了把屏幕擦干净,把存储器内的程序都抹掉。

现在输入:

10 HOME

```
20 PRINT "LISTEN"  
30 PRINT CHR$ (7)  
40 PRINT "DID YOU HEAR IT?"
```

执行这个程序,然后输入 HOME 语句将屏幕擦掉;于是程序在屏幕上消失了。但是程序并没有丢失,电脑已经把程序放在存储器里了,它可以随时再显示出来。

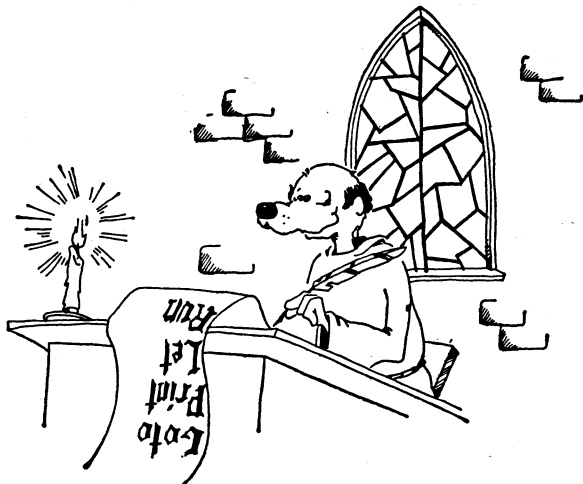
LIST(列表)命令可以列出程序内容

要想看一下刚才输入的程序中第 30 行是什么语句,可以输入:

```
LIST 30
```

电脑还可以把其他任何一行的内容显示出来,只要相应地改变 LIST 后面的数字。如果要看一下存储的整个程序,那么输入:

```
LIST
```



后面不跟任何数字。请试一下。

存储器

电脑的存储器是用来存放程序 and 数据的，它就象一个个放东西的盒子一样。用到一个盒子时，就把东西放在盒子里面，并在这个盒子的前面写上一个名字，以便在需要时，可以根据这个名字找到盒子里的东西。开始时，所有盒子都是空的，盒子上也没有名字。当输入：

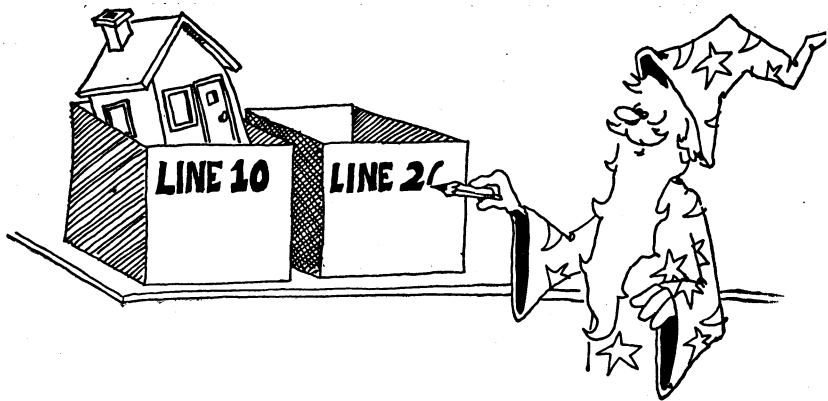
10 HOME

时，电脑取第一只空盒子，并取名为“第10行”，写在盒子前面的标签上，然后把 HOME 语句放进盒中存放起来。

接着你又输入：

20 PRINT "LISTEN"

电脑又取一只盒子，同样把“第20行”写在它的标签上作为它的名字，再把 PRINT "LISTEN"这一语句放入盒内。用这一方法，



可以把你输入的所有语句都放在存储器内。

从存储器中删去一行语句

有时候需要把原来程序中的某一行语句删掉，这只要输入该行的行号，后面不跟什么东西就行了。例如要删掉第 20 行语句，只须输入：

20

这时虽然在屏幕上仍然可以看到这一行，但是你用 LIST(列表)命令显示一下存储器里的内容，就会发现第20行语句已没有了。

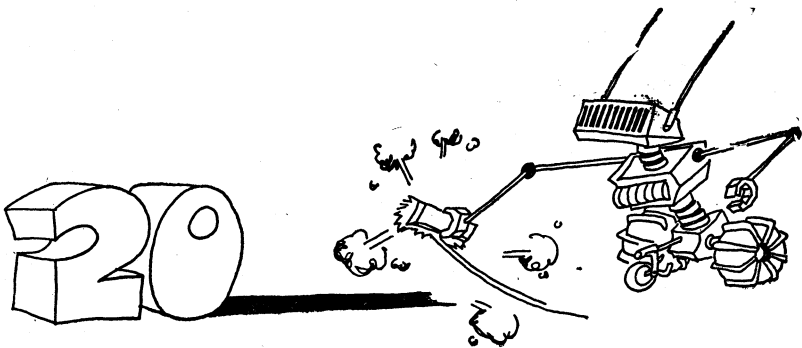
当你只输入某一行号，后面不跟什么东西时，电脑做的事情就是：找到写有这行号的那个盒子，将盒子里的东西倒空，然后把盒子前面的行号也抹去，使它成为一只空盒子。

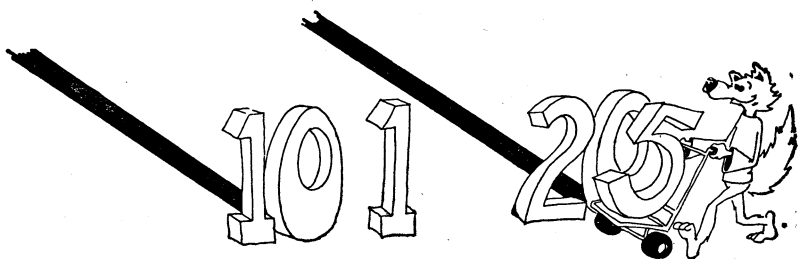
如果要把整个程序都清除掉，那该怎么办吗？（先看一下前面的课文，思考一下再看下面的提示。）

当输入 NEW 命令时，电脑对存储器盒子做些什么？

插入一行语句

如果要在程序的任何地方，甚至是在原来两行语句之间插





入一行新语句,那只要先在原来的行号之间选一合适的行号,然后将要加入的行输入,电脑便会自动地把它放到相应的位置上。

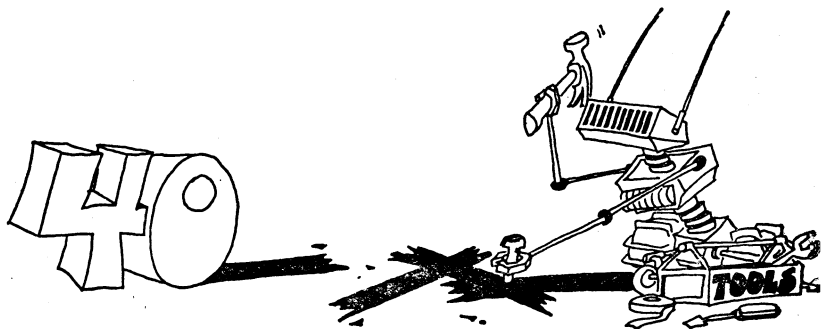
键入 NEW 并输入:

```
10 REM MORE AND MORE
20 PRINT "MORE LINES WANTED"
40 PRINT "HERE THEY ARE"
```

列表显示并执行一下,然后加上下面这一行。

```
15 PRINT "STILL"
```

再列表显示和执行一次,看看和上一次有什么不同。



语句行的修改

如果一行语句错了,要修改,只须重新输入一次。例如:在上面的程序中,如果再输入: 40 PRINT "NEEDS FIXING",就把行号为40的原来语句改掉了。

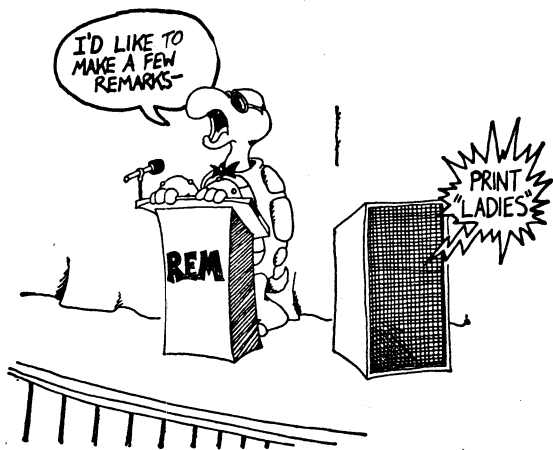
想一想,当输入这一行新内容时,电脑对标有名字为“第40行”的那个盒子做了些什么?

REM(注释)语句

你编写的程序,有时候需要加上几个小标题,这可以方便地用 REM 语句来实现。

输入:

```
10 REM PROGRAM 2  
20 HOME
```



I'D LIKE TO MAKE A FEW REMARKS——我想做些注释

```

30 PRINT "LINE 10 DOES NOTHING"
35 REM THIS LINE DOES NOTHING
40 LIST
RUN

```

REM 意思为“注释”，用 REM 语句可以在程序的任何地方写上一些说明，这样便于阅读的人理解程序的内容，而这些注释语句丝毫也不影响原来程序的执行。

绘图

可以利用 PRINT 语句打印的字符在屏幕上绘图，这里简单地画一辆汽车，在画图前先输入 NEW，然后输入：

```

10 HOME
20 PRINT
30 PRINT "×××××××"
40 PRINT "×××××××××××××"
50 PRINT "□                □"

```

别忘记，在 PRINT 语句中，应根据需要放上一些空格。绘图时，空格也是图案的一个组成部分。

作业3:

1. 什么语句能在屏幕上显示出程序中第10行的内容？
2. 怎样在屏幕上显示出整个程序？
3. 程序执行时遇到 REM 语句如何处理？
4. REM 语句的用途是什么？
5. 使用 HOME、CHR\$(7)、REM 和 PRINT 语句在屏幕上画三只飞鸟，画好后能使每只鸟发出“唧”的叫声。

第四课 专 用 键

教学参考:

本课主要讲箭头键和 REPT(重复)键的使用。箭头键是用来移动光标的,它只对目前正在编辑的行起作用。光标移动对原来行内的字符没有影响,它无论停在什么地方,都可以输入新的字符,但不能在原来字符之间插入新字符,也不能把原来的字符删除掉,而只能够用新的字符或者空格来替换原来的字符。在一行内容处理完之后,只要象通常一样,按一下回车键,该行就又重新输入电脑。

注意:刚才这样处理的那一行内容,只是将从这行左边开始一直到光标目前位置之间的那部分内容送入了存储器,而光标右边余下的那部分内容就丢失了。因此想要存入该行的所有内容,就应将光标移到本行的末尾。

箭头键不能用来修改前面已经输入的语句行,如果要修改前面的语句行,需要用编辑的方法,这个方法以后会介绍。

REPT(重复)键能重复输入字符;在与箭头键联用时能快速地移动光标;在画虚线时也很有用。

复习题:

1. 什么叫“光标”?它有什么用处?
2. 当光标处在语句行的中间位置时,若按一下回车键,那

么该行的哪一部分被丢失？

3. 怎样来编写一行语句,包括用箭头键把光标移至行的中间,修改字符,然后在按回车键之前将光标移到末尾。

4. 学会使用 REPT 键。

课 文

光标是一个闪烁的方块

闪烁的小方块称为“输入光标”。它用来显示要输入的下一个字符在屏幕上的位置。光标好象是一个“引路者”,指引你把输入的字符一个个地填写在屏幕上,它沿着屏幕移动,但总是指向下一个字符要在屏幕上出现的位置。

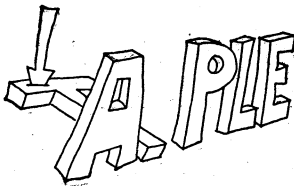
箭头键

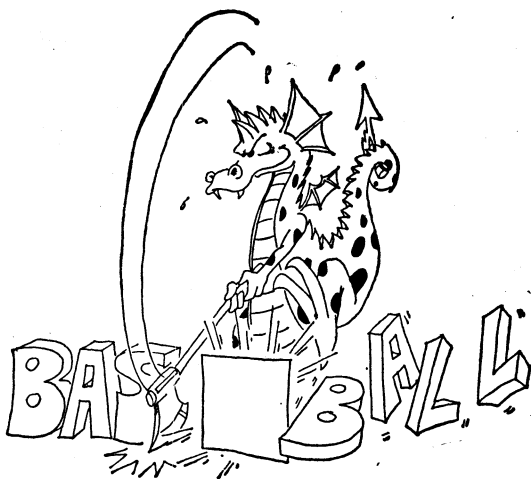
“箭头键”可以帮助你修改输入语句中存在的错误。

输入:

```
10 REM ATPLE. (不要按回车键)
```

上面一个单词写错了,要对它进行修改,方法是:按几下左箭头键,将光标移至字母“T”上,键入字母P,用来取代字母T。再用





右箭头键把光标移到该行的末尾，即移到句号之后。现在这一行语句就正确了，读作：

```
10 REM APPLE.
```

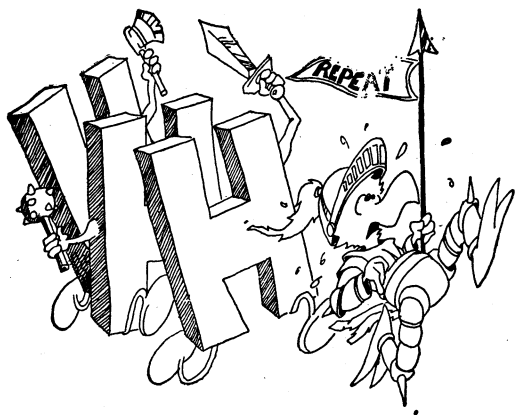
按回车键，将正确的一行内容放入存储器。

注意：在按回车键之前，要将光标移回到该行的末尾，如果不是这样，那么这一行的后半部分内容就会丢失。不妨再试一下。

REPT(重复)键

先在键盘上按 REPT 键和任何一个字母键，看看屏幕上出现的情况。接着再按 REPT 键和左箭头键，再看一看发生什么情况。

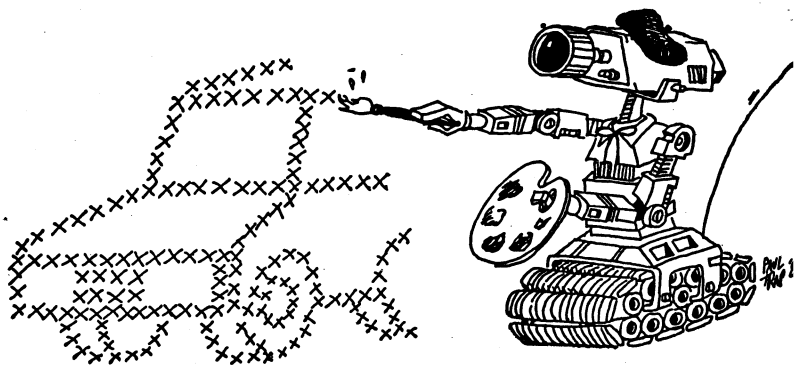
规则：REPT 键可以与键盘上任何其他键一起使用。例如可以同空格键与回车键一起用，请试一下。



REPEAT——重复

画图

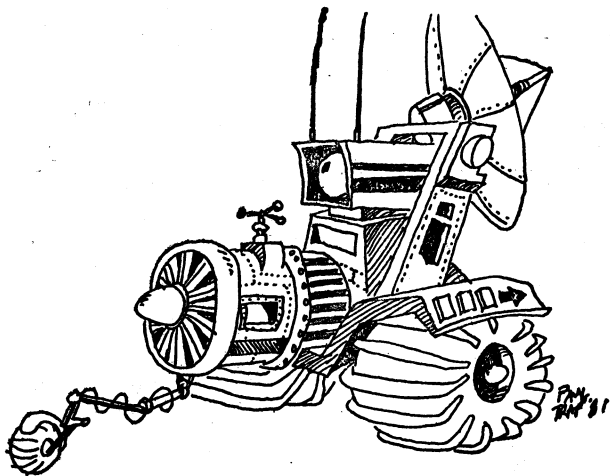
REPT(重复)键在画图中使用是很方便灵活的,一幅图案中,可用它来重复一些字母和其他的符号。



作业4:

1. 输入一行语句并用箭头键使光标在该行中移动,改变这一行中的一些字母,完成后把光标移到行的末尾,再按回车键把该行内容送入存储器。

2. 你能在屏幕上画一幅“微笑的脸”吗?



第五课 INPUT(键盘输入)语句

教学参考:

本课要介绍 INPUT(键盘输入)语句和字符串变量的概念。先介绍形式最简单的键盘输入语句:

```
INPUT A$
```

这样做是为了能使学生把注意力集中在 INPUT 语句的主要特性上。同样道理,在14课之前的其他语句也只讲一些主要特性,这就能使我们很快地概括出编写程序的要点,以便使学生不让一些枝叶遮住眼睛,而能及早“见到森林”,早些写出有意义的程序来。有趣的程序所需要的语句有:

```
PRINT      打印语句
```

```
INPUT      输入语句
```

```
GOTO      转向语句
```

```
IF         条件语句
```

```
RND       游戏中用的随机数函数
```

现在我们再次用“盒子”的概念来介绍字符串变量。为了简单起见,不妨暂时限制变量名称只用一个字母来表示,这在短程序中不会引起混乱,反而能提高输入的速度。在学过 RND 函数之后,再来对它的复杂命名规则进行讨论。

我们在程序中尽量用字符串,而不用数字,因为字符串能使

程序更为有趣,也不至于在编制程序时在逻辑概念上造成混淆。

复习题:

1. 电脑可以把哪两样不同的东西放在盒子里?
2. 程序用什么方法要求你输入一些内容?
3. 怎么知道电脑在等待你回答?
4. 在后面带有美元符号(\$)的字母称作什么?
5. 用 HOME、PRINT 和 INPUT 语句写一段小程序。
6. 如果在输入一项内容之后电脑回答“EXTRA IGNORED”,说明什么问题? 造成这样的原因是什么?

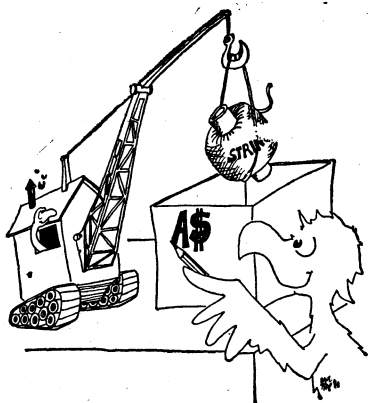
课 文

电脑用 INPUT(键盘输入)语句来要求外界输入一些东西。
输入:

```
10 HOME
15 PRINT "SAY SOMETHING"
20 INPUT A$
25 PRINT
30 PRINT "DID YOU SAY"
40 PRINT A$
```

执行这个程序。当屏幕上出现了问号时,就在键盘上输入“HI”,并按回车键。

刚才出现的问号是由第20行 INPUT(键盘输入)语句产生的,闪烁的光标告诉你要输入一些东西进去。接着当你输入了“HI”之后,电脑就把这个词存放在称为 A\$ 的盒子中。程序执行到第40行时,就要求电脑把 A\$ 那个盒子中的内容打印出来。



STRING——字符串

再执行一次这个程序，这一次可以不要再输入“HI”，而输入一些逗人发笑的内容。

字符串变量

上面程序中用到的A\$称为“字符串变量”，这名称用来写在盒子的前面，而字符串则放在盒子内。

规则：字符串变量的名称要以美元符号“\$”结尾，可以用任何一个你喜欢的字母作它的名字，然后在字母后面放一个美元符号。

A\$称为字符串变量，是因为在程序中可以在不同的时刻把不同的字符串放进这个盒子。因此这个盒子里的字符串不是一成不变的。当然，盒子一次只能容纳一个字符串。

输入错误的内容

规则：不要把任何逗号、问号、分号或冒号键入回答电脑的字符串中去。

如果不留意偶然放进了一个标点符号,电脑会回答:

? EXTRA IGNORED

出现这种情况,电脑就去掉标点符号后面的东西,然后继续执行程序。

作业5:

1. 写一段可以询问某人姓名的程序,然后用该名字说一些引人发笑的话。

2. 写一段程序,先输入你最喜欢的颜色,并将它放入称为C\$的盒内;接着程序会问你最喜欢的动物是什么,也将它放入C\$的盒子内,然后再打印C\$的内容。你认为应该打印出什么来?执行一下这个程序,看猜得对不对。

第六课 打印方法与SPEED(速度控制)语句

教学参考:

这一课中要介绍:

1. 结尾带有分号的PRINT(打印)语句;
2. 两项内容之间带有分号的PRINT语句;
3. PRINT语句中看不见的打印指针;
4. SPEED(速度控制)语句。

在PRINT语句中,逗号的运用可以忽略。因为在40列的屏幕上并没有多大用途。

这一课将要介绍电脑屏幕上看不见的打印指针。它可指出PRINT语句要打印的下一个字符应在屏幕的什么位置上(它与输入光标不同,输入光标是看得见的闪烁的小方块,在编辑方式和输入语句中非常有用)。

当PRINT语句以分号结尾时,打印指针就保持在原来位置上;下一个PRINT语句要打印的第一个字符就紧跟在这句语句的最后一个字符后面。

如果结尾没有分号,PRINT语句打印完本语句的内容后就

把打印指针移至下一行的开头。

一个PRINT语句可以打印几项内容,每一项可以是字符串,常数,变量或表达式。各项之间可由分号分开。这儿常数和变

量还没有介绍过。

被打印的一连串内容中,各项之间会有一些连接的字符。不知你记住没有,空格也是字符。例如在“HAM AND EGGS(火腿炒鸡蛋)”程序中,需要一些空格,而空格必须直接放入字符串内。

复习题:

1. 什么语句会使屏幕上出现闪烁的光标?
2. 什么语句使用看不见的打印指针?
3. 如何把两个PRINT语句打印的内容放在同一行上?
4. 在执行下列语句时,两个单词之间会有空格吗?

```
10 PRINT "HI"; "THERE!"
```

如果没有,如何在它们之间放入一个空格?

5. 什么样的语句使打印速度变得非常慢?
6. 如何使打印回到正常速度?

课 文

显示一行还是多行

输入下面的程序:

```
10 REM FOOD  
20 PRINT  
30 PRINT "HAM"  
40 PRINT "AND"  
50 PRINT "EGGS"
```

然后执行这个程序。程序中的每个PRINT(打印)语句各打印一行。

现在请你再输入：

```
30 PRINT "HAM";
```

```
40 PRINT "AND";
```

要小心，不要改变或抹掉其他行的内容，并在“HAM”的末尾和“AND”的末尾各加上一个空格，然后在两行的结尾都加上一个分号。

执行一下这段修改过的程序。看看与第一次有什么不同的地方。

隐蔽的指针

你还记得闪烁的方块吗？这是输入光标。在用键盘输入时，它用来指示下一个字母在屏幕上出现的位置。

PRINT语句也有类似光标的指针，但看不见。在电脑进行打印时，它能标明下一个字母应该打印在什么地方。

规则：分号使PRINT语句不可见的打印指针在原来的位置上等待；下一个PRINT语句把打印的内容连接到上面已经写有内容的那一行上去。

下面请你做两个例题：

1. 输入：

```
10 HOME
```

```
20 PRINT "ENTER A NAME"
```

```
30 INPUT A$
```

```
35 HOME
```

```
40 PRINT "ENTER ANOTHER"
```

```
50 INPUT B$
```

```
60 HOME
```



```
70 PRINT "PRESENTING THAT FAMOUS  
TWOsome" (在电脑上应写成一行,下同)
```

```
75 PRINT
```

```
80 PRINT A$; " AND "; B$
```

一定要在“AND”前后都加上一个空格。

2. 键入NEW后再输入:

```
10 PRINT "ROCK"; "AND"; "ROLL"
```

执行后再试一试:

```
10 PRINT "ROCK "; "AND "; "ROLL"
```

不要忘记在ROCK和 AND之后各加上一个空格。

SPEED(速度控制)语句

有时候觉得电脑以通常的速度打印显得太快了,速度控制语句能放慢打印速度。

键入并执行这个程序:

```
10 REM ::::: SPEED :::::
```

```
20 HOME
```

```
30 PRINT "MISSISSIPPI"
```

```
40 SPEED=5
```

```
50 PRINT "MISSISSIPPI"
```

```
60 SPEED=255
```

程序中第40行用了SPEED(速度控制)语句。速度的快慢可以从1到255的任何一个数中进行调节。速度值=1时,非常慢;速度值=255时,为正常速度。

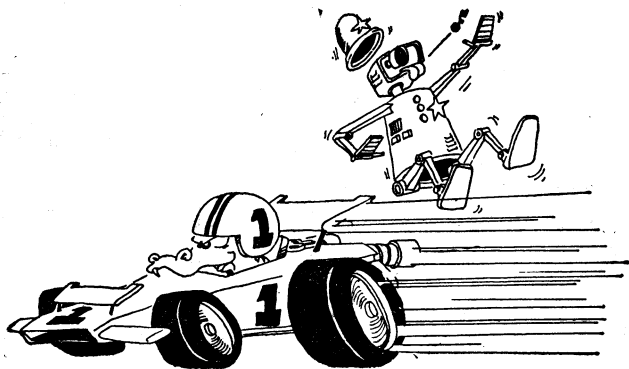
看第60行。这一行非常重要,在程序做完后,它使电脑回到正常速度。

作业6:

1. 编写一个程序,使这个程序能询问一个音乐小组的名称和一首曲调的名称。然后用一个PRINT语句打印小组名称和曲调名称并把词“PLAY”放在二者之间。

2. 按上题要求编写同样的程序,但要使用三个PRINT语句,并把内容打印在同一行上。

3. 用白底黑字方式让电脑慢慢地打印出你的姓名(汉语拼音)。



第七课 LET(赋值)语句

教学参考:

我们还是用存贮器盒子的概念来介绍LET(赋值)语句。

盒子模型是用来强调:LET是一句替换语句,其作用与算术中所使用的“等于”符号不同。

利用盒子能很好地分清“变量名”与“变量值”的概念。变量名写在盒子的标签上,而变量值放在盒子里。盒中的值可以取出来供反复使用。新的值也可放入盒子里。

确切一些说:变量使用时,盒中的值可以被复制和使用,而原来的值仍保持不变。对这一点应解释清楚。

到目前为止,学到的语句和命令有:

HOME(清屏幕)语句、NEW(清内存)命令、PRINT(打印)语句、RUN(执行)命令、PRINT CHR\$(7)(发“嘟”声)语句、FLASH(闪烁)语句、INVERSE(白底黑字)语句、NORMAL(黑底白字)语句、LIST(列表)命令、REM(注释)语句、INPUT(键盘输入)语句、SPEED(速度控制)语句、LET(赋值)语句等。

到目前为止学到的专用键有:

RETURN(回车)键, REPT(重复)键, 左、右箭头键以及SHIFT(换档)键等。

我们可以把学到的语句连贯起来使用。

复习题：

1. LET语句把东西放入盒子里,INPUT语句也具有这样的功能。两者不同在哪里?

2. 如果执行这个小程序:

```
10 LET A$="HI"
```

```
20 LET B$=A$
```

那么在A\$盒子中是什么内容?在B\$盒子中又是什么内容?

3. 在下面的程序中:

```
10 Q$="MOM"
```

"MOM"叫什么?这个程序中的字符串变量名是什么?程序执行后,字符串变量值是什么?

4. 下面的程序执行后,每只盒子里是什么内容?

```
10 LET H$="FAT"
```

```
20 LET K$="SAUSAGE"
```

```
30 LET P$=H$+K$
```

课 文

LET叫赋值语句,它的作用是把右边的东西放入左边的盒子里。

执行下列程序:

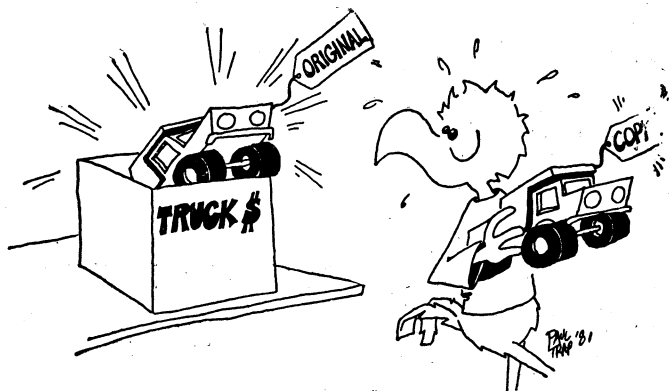
```
10 HOME
```

```
20 LET W$="MOPSEY"
```

```
40 PRINT W$
```

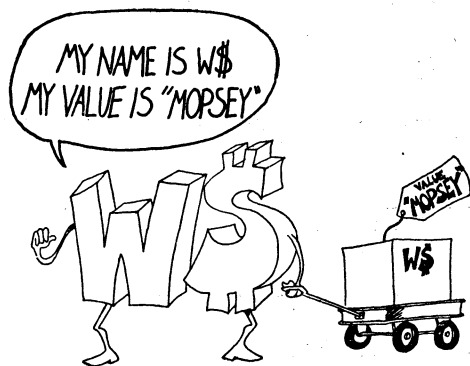
执行过程如下:

第10行:电脑将屏幕清除干净。



ORIGINAL——原物 COPY——复制品

第20行：电脑需要一只称为“W\$”的盒子；电脑先在存贮器中检查一下是否有“W\$”标签，因为“W\$”在前面的程序中没有使用过，所以一只也没有找到；于是就取出一只空盒并在前面写



MY NAME IS W\$——我的名字叫W\$
 MY VALUE IS “MOPSEY”——我的值是“MOPSEY”

上“W\$”，然后把字符串“MOPSEY”放入这个盒子里。

第40行：电脑明白它要打印“W\$”盒内的内容；于是找到了那只盒子，并把里面的字符串“MOPSEY”进行复制，再把复制品放在电脑屏幕上。而原来在“W\$”盒内的字符串“MOPSEY”原封不动，仍在里面。

变量名与变量值

W\$就叫变量名，变量值是指放在盒子中的内容。在上面的程序中，W\$的值为“MOPSEY”。

请你自己再做一个例子。

输入并执行：

```
10 LET D$="PICKLES"  
20 LET A$="AND"  
30 PRINT "WHAT GOES WITH PICKLES?"  
35 INPUT Z$  
40 HOME  
50 PRINT D$; A$; Z$
```

请解释电脑在每一行中做些什么？

字符串的连接

下面讲的是如何把两个字符串连接在一起，组成一个较长的字符串，输入下列程序：

```
10 HOME  
20 LET W$="HAR DE"  
25 LET X$="HAR"  
30 LET L$=W$+X$  
40 PRINT L$
```

```
50 PRINT
60 LET L$=L$ + X$
70 PRINT L$
```

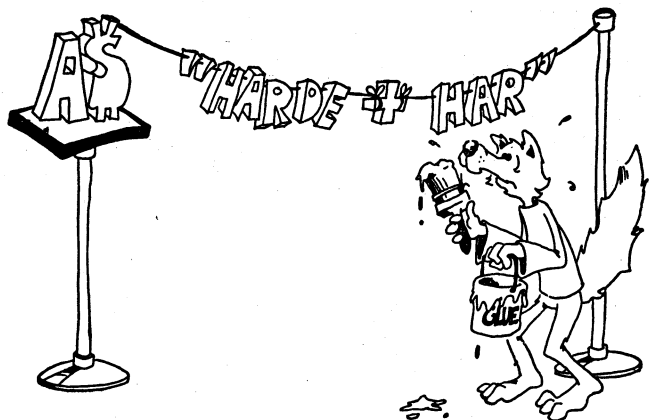
在执行这个程序以前，先猜一下第40行与第70行将会打印出什么。

现在执行程序，看一看你猜得是否正确。

规则：程序中的“+”号能把两个字符串连接在一起。注意，上面程序中的第50行只打印一空行，这对于把打印内容分隔开来是很有用处的。

作业7:

1. 编写一个使用LET语句的程序，并解释它是如何把内容放入盒内的。
2. 编写一个能输入两个字符串的程序，并把它们连接在一起，然后再打印出来。



第八课 GOTO(转向)语句与RESET (复位)键

教学参考:

GOTO(转向)语句能使程序无条件地转向到别处去执行。等下一课学到IF(条件)语句时,两者结合使用将会有助于学生改变程序中语句执行的次序,而不是让电脑仅仅简单地依次按语句行号执行。

就目前来说,转向语句的主要用途是让程序反复地连续执行一段时间。在进行每一次循环时,可以修改其中有关的一些内容。

这里遇到的一个问题是如何使它停止循环。如果你用的是带有自动启动器ROM的苹果II型电脑,可按RESET(复位)键即能较好地完成这一工作。如果是其他型号的苹果电脑有时则要把RESET改成“CTRL-C”(暂停)键。甚至在有的苹果电脑上,CTRL-C键必须要与RESET键一起使用才能进行复位。

现在我们已经学会了编制实用程序的四种语句中的三种。它们是:PRINT(打印)语句、INPUT(键盘输入)语句及GOTO(转向)语句。所缺少的是IF(条件)语句。IF语句可以使电脑具有判别周围情况和作出决定的能力。

复习题:

1. 当执行这个小程序时,屏幕上会出现什么?

```
10 PRINT "HI"  
20 GOTO 40  
30 PRINT "BIG"  
40 PRINT "DADDY"
```

2. 执行这个程序时,屏幕上又会出现什么呢?

```
10 PRINT "APPLE"  
20 PRINT "PIE";  
30 GOTO 20
```

3. 在上面的程序中,如何使程序停止运行?

4. 编写一个能发出“嘟嘟”声的短程序:问你自己最喜欢的电影明星的姓名是什么,然后使它一遍又一遍地反复这样做。

课 文

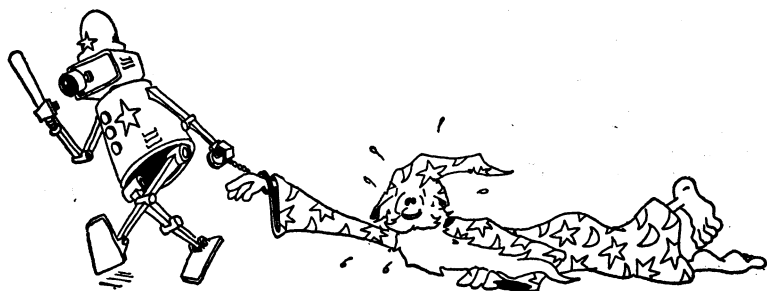
反复循环的程序

输入下列程序:

```
10 HOME  
20 PRINT "YOUR NAME?"  
25 INPUT N$  
30 PRINT N$  
35 PRINT  
40 GOTO 30
```

在执行这个程序时,它不会自己停下来。当你的姓名“呼”的一下在你眼前闪过时,如果你要让它突然停下来,请同时按:

CTRL(控制)键和RESET(复位)键(或CTRL-C——暂停)



键)。

第40行使用了GOTO (转向) 语句。每当电脑执行第40行时,它就自动转到第30行去执行,重新打印出你的姓名。

以后我们在许多程序中都要使用 GOTO 语句。

转向

输入这个程序:

```
20 PRINT "SAY SOMETHING"  
→30 INPUT S$  
40 PRINT "DID YOU SAY'"; S$; "'?"  
45 PRINT  
—50 GOTO 30
```

请执行一下。每当看到“?”及同时出现的闪烁光标时,请输入一个回答的内容进去。按一下RESET (复位) 键可以停止程序的执行。

注意: 程序中第50行到第30行画着的箭头,表明GOTO 语句在做些什么。在你编写的程序中也可以试画一下这样的箭头。

转向的种类

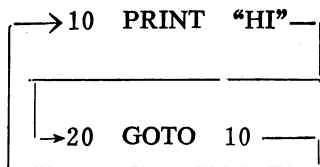
只有两种转向: 向前转或向后转。

向后转实际上是做一个循环。

```
10 PRINT "HI"
```

```
20 GOTO 10
```

程序所走的路如下：



电脑在这个循环中一遍又一遍地回转，直至按RESET（复位）键才停止。

向前转移就是跳过一部分程序再继续执行。目前还不太有用，以后在IF（条件）语句中会使用它。



RESET(复位)键

CTRL-C(暂停)键和RESET(复位)键可称是“救命键”。当遇到麻烦时,按一下CTRL-C键或RESET键,电脑就会发出“嘟”声,并重新进行启动,等待接收你的下一条命令。原来的程序仍保存在存贮器内,安然无恙。

作业8:

1. 编写一个能一遍又一遍打印出你姓名的程序。
2. 怎样使你的程序停止执行?
3. 再编写一个程序:在一行上打印你的姓名,在另一行上打印你朋友的姓名;并使它一遍又一遍重复执行。使用 SPEED 语句来放慢打印速度,用RESET 键来停止执行这个程序,然后输入SPEED=255,使打印速度回到正常速度。
4. 编写一个程序,将下列语句都用进去: HOME, PRINT CHR\$(7), PRINT, INPUT, LET及GOTO,并使它能将两个字符串连接在一起。

第九课 IF(条件)语句

教学参考:

本课将介绍 IF (条件)语句,并对两个字符串是相同还是不同的两种情况作出不同的处理。

IF语句是功能很强的语句,它是电脑中具有逻辑判断能力的关键语句。这是一种比较复杂的语句,学生在学习这种语句时可能需要指导。

学习IF语句需要想象力。课文中的“蛋糕”插图及“三叉路口”插图可以帮助你理解。程序中语句行的执行次序可以改变,这已经和GOTO语句一起介绍过了。这一点IF语句也能达到,只是需要加上一定的条件,经电脑判断以后然后再执行:如果语句中表达式是真的,电脑就会转向执行;如果是假的,电脑就会顺次执行下面的语句。IF语句中的“条件”是一种判断正确与否的表达式;“语句C”(见课文中的条件语句)是用于在“条件”满足的情况下电脑要执行的语句。

条件语句的表达式中出现的关系式对学生来说可能容易产生混淆。这里必须搞清楚,“=”表示判断左、右两个字符串是否相同;而“<>”符号则相反,表示判断它们是否不相同。

学生对“<>”符号可能不够熟悉,在这一点上最容易产生混淆。遇到这种情况,可以让学生多用几次,并与“=”关系式作些比较,举一反三就会掌握的。熟悉IF(条件)语句的使用对学生

是有好处的。这有利于处理好许多种关系：

<(小于符号)、>(大于符号)、=(等于符号)

=<(小于等于符号)、=>(大于等于符号)

复习题：

1. 如何使下面这个程序打印出“THAT'S FINE”?

```
15 PRINT "DOES YOUR TOO HURT?"
17 INPUT T$
20 IF T$="NAH" THEN GOTO 90
40 IF T$="SOME" THEN GOTO 15
90 PRINT "THAT'S FINE"
```

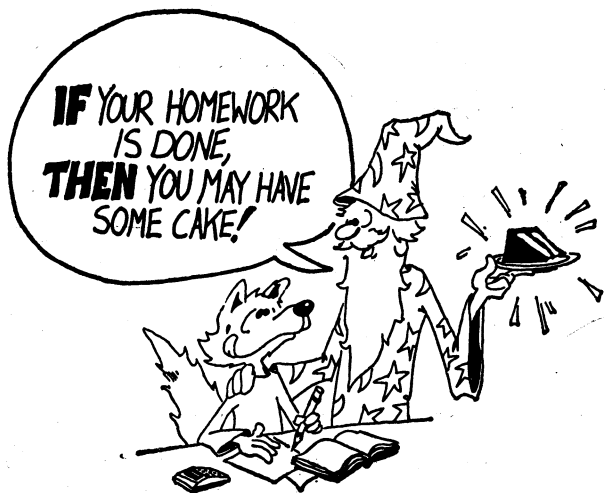
2. 编写一个问你是不是喜欢 CHOCOLATE (巧克力)或者 VANILLA ICE CREAM(香草冰淇淋)的短程序。回答是“C”或“V”。是“C”就打印“YUMMY!”(美味),是“V”就打印“MMMMMM!”。

课 文

先清除存贮器,然后输入:

```
10 HOME
20 PRINT "ARE YOU HAPPY? (YES OR
   NO)"
30 INPUT A$
40 IF A$="YES" THEN PRINT " I'M
   GLAD"
50 IF A$="NO" THEN PRINT "TOO BAD"
```

让这个程序执行几次,在每次出现“?”时,分别回答“YES”、



IF YOUR HOMEWORK IS DONE——IF(如果) 你的家庭作业做好了

THEN YOU MAY HAVE SOME CAKE——THEN (那么) 你可以吃蛋糕

“NO”或“MAYBE”。看一看发生什么情况。

IF(条件)语句

IF语句主要有两个组成部分：

10 IF 条件 THEN 语句 C

一开始电脑看着“条件”：如果条件是真，电脑就执行语句 C；如果“条件”是假，电脑就执行下面一行，而不执行语句 C。

整个过程的顺序如下：

10 IF 条件为真(正确) THEN 执行语句 C，
然后再执行下面一行语句。

或者：

10 IF 条件为假(错误) THEN 执行下一行。
这里再举几个IF语句的例子。

```
10 REM IF TEST PROGRAM
50 IF A$="YES" THEN PRINT "GOOD"
55 IF "YES"=A$ THEN PRINT "GOOD"
60 IF A$=B$ THEN LET C$="NO
    WAY!"
70 IF N$="BIRD" THEN PRINT CHR$(7)
75 IF A$="READY" THEN INVERSE
```

第50行中的IF语句与第55行中的 IF语句做的事完全一样。

作业9A:

1. 把下面几行语句加到上面的程序中去。

```
15 INPUT A$
17 LET C$="WHAT?"
20 LET B$="PAY UP"
25 LET N$=A$
85 PRINT C$
87 NORMAL
99 GOTO 15
```

执行这个程序,然后再输入这些词:

YES, BIRD, READY, NO WAY

请你自己再挑选其他一些词试试,看看程序打印出什么, IF 语句是否象你所希望的那样在工作(记住:如果“条件”为真,就执行THEN之后的语句)。

2. 先清除存贮器,然后写一个问你是“BOY”(男孩)还是

“GIRL”(女孩)的程序。如果答案是“BOY”程序将打印“SNIPS AND SNAILS”(剪子和蜗牛)。如果答案是“GIRL”则打印出“SUGAR AND SPICE”(糖果和调味品)。

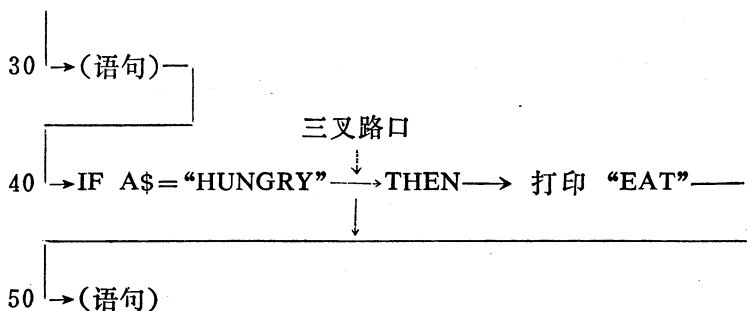
IF(条件)语句与三叉路口

电脑看到IF语句,就象人走到三叉路口一样,必须选择要走的是哪一条路。

如果“条件”为真,电脑必须经过“THEN”,执行那里的语句。然后再执行下一行语句。

如果“条件”为假,它就立刻执行下一行语句。

这里有一张地图(见下页),图上标有道路的分支。



“不等”号

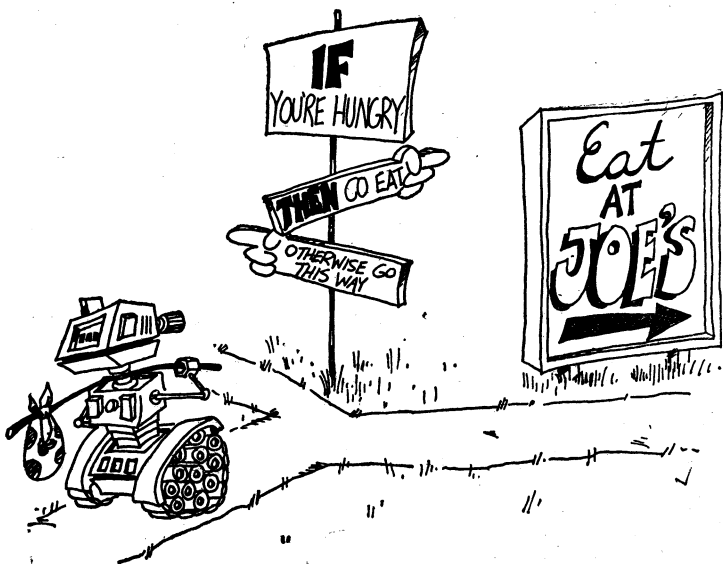
“< >”符号意思为“不等于”。它与平时用的“=”符号正好相反。

要产生“< >”符号,可先按“<”键,然后再按“>”键。

```
40 IF 条件 THEN PRINT "NO SMOKE"
```

“条件”是一个成立(为真)或不成立(为假)的短语,如果成立(为真),那么电脑就打印“NO SMOKE”。看下面的“条件”

```
Q$< > "FIRE"
```



IF YOU'RE HUNGRY——IF(如果)你饿了
 THEN GO EAT——THEN(那么)去吃东西
 EAT AT JOE'S——请到JOE店去吃
 OTHERWISE GO THIS WAY——否则请走这条路

然后把它放入IF语句中：

```
40 IF Q$ < > "FIRE" THEN PRINT "NO SMOKE"
```

如果Q\$盒子中是“COLD”，那么Q\$就不等于“FIRE”，于是条件Q\$ < > “FIRE”为真，电脑就打印出“NO SMOKE”。

如果Q\$盒子中是“FIRE”，那么条件

Q\$ < > FIRE 为假

于是电脑就不再打印出任何内容。

下面是一个例子，请试一试。

```

10 PRINT "IS YOUR HOUSE ON FIRE?"
20 PRINT "(ENTER 'FIRE' OR 'COLD')"
```

30 INPUT Q\$

```

40 IF Q$ < > "FIRE" THEN PRINT "NO
    SMOKE"
```

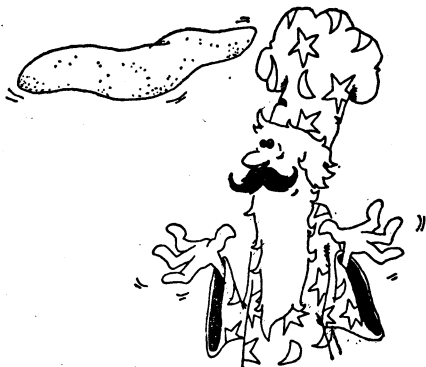
```

50 IF Q$ = "FIRE" THEN PRINT "HELP"
```

作业9B:

1. 编写一个“PIZZA(烘肉饼)”程序,提出一些需要选择的材料,使电脑对每一种不同的选择作出一个逗人发笑的回答。例如你可以选择 MUSHROOMS (蘑菇)、PEPPERONI (红辣椒)、ANCHOVIES(鲱鱼)、GREEN PEPPERS(青椒)等。也可问一下肉饼的大小。

2. 编写一个彩色猜谜游戏。玩游戏时,一个人用字符串C\$输入一种颜色,另一个人不断输入字符串G\$猜颜色。使用两条IF语句行,一条带有“条件”G\$ < > C\$表示猜错了。另一条带有“=”符号代表猜对了。“语句C”打印出“假”或“真”的结果。



第十课 数 字

教学参考:

本课将介绍数字变量及其运算。对LET(赋值)语句、INPUT(键盘输入)语句,及PRINT(打印)语句要重新进一步认识。

本课将把存贮器“盒子”的概念进一步扩展到数。变量名暂时仍限制每次使用一个字母。

电脑的算术运算在使用时都采用十进制,这是大家很熟悉的。只是这儿代表乘法的符号“*”学生可能对它不太熟悉。但只要多使用几次,就会习惯的。

字符串中的数字并不是算术中直接可使用的“数”。学生对这一点也许会感到很奇怪。本书后面将介绍的字符串函数VAL及STR\$将会有助于消除学生们的疑点,它们可以使数和字符串相互转换。

用PRINT语句可以打印出字符串与数值的混合体。BASIC语言中“=”符号属于非标准用法,它表示的是“置换”关系而不是“相等”关系。这在LET $N=N+1$ 语句中就表现得很突出。在课文的图片中将使用盒子概念来说明“=”的含意。

复习题:

1. 给存贮器中三种类型的“盒子”命名(即根据存放在盒子内的东西种类来命名)。

2. 解释为什么电脑使用的“ $N=N+1$ ”和算术中的“ $7=7+1$ ”不一样。

3. 举一个用 LET 语句编写的错误运算的例子，其中要使用 *，或 / 符号。

4. “TYPE MISMATCH ERROR”在电脑中表示什么意思？

5. 举一个有“TYPE MISMATCH ERROR”程序行的例子。

6. 请解释数字变量的“变量名称”及“变量值”各是什么意思。对字符串变量呢？

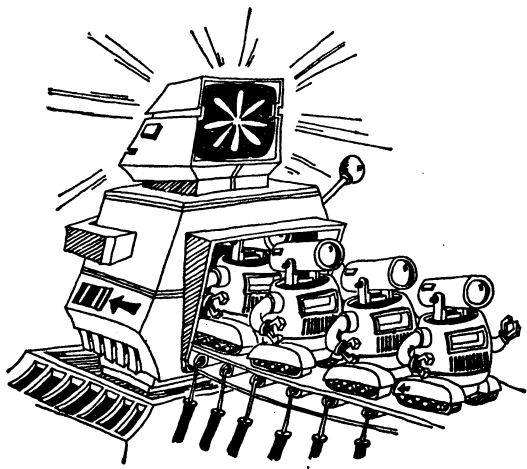
7. 如盒子 A 和 B 中各放有数 5 和 8，那么 PRINT A;B 将打印出 58。如何使电脑打印出 5 8 而不打印 58 呢？

课 文

到现在为止，在 INPUT(键盘输入)语句、LET(赋值)语句及 PRINT(打印)语句中，我们只使用了字符串，其实数字也可使用。请输入并执行下列程序：

```
10 HOME
20 PRINT"GIVE ME A NUMBER"
30 INPUT N
40 LET A=N+1
45 PRINT
50 PRINT "HERE IS A BIGGER ONE"
60 PRINT A
```

第 20 行打印的意思是“给我一个数”，第 30 行要求你输入一



个数。

算术运算：

在电脑中，减号就象“=”号一样在字符键上；电脑用“*”表示乘法符号，以代替平常用的“×”符号。

试做：改变上面程序的第40行，使N乘以5，看一看电脑是怎么执行的。

电脑使用“/”符号来表示除法。答案用十进制数表示。

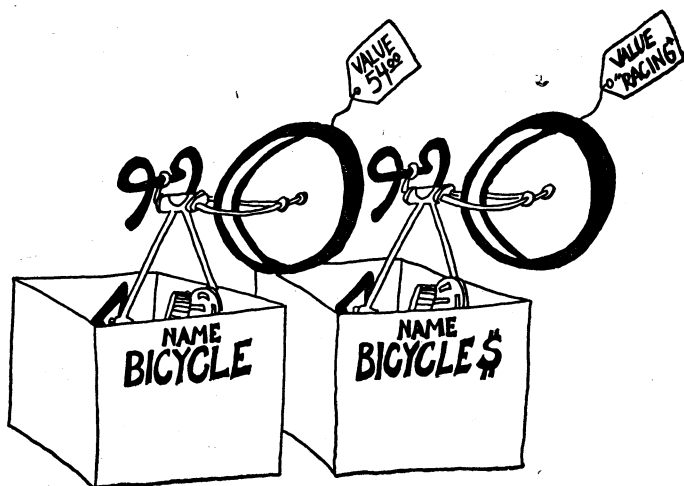
变量

装有字符串的盒子名称必须以美元符号结尾。例如：N\$、A\$、Z\$等。

而装有数字的盒子的名称就没有美元符号。例如：N、A、Z等。

放入盒内的东西称作变量值。

LET(赋值)语句中的运算



VALUE——值 NAME——名字

在电脑上用 LET 语句中可以进行算术运算。下面是一个例子。

```

10 LET A=2
20 LET B=3
30 LET C=B-A
40 PRINT A; " "; B; " "; C

```

再举一例：

```

10 LET B=15
20 LET A=B/5
30 LET X=A*4+2
40 PRINT X; " "; A

```

注意：数字与字符串是不同的。例如：“1984”不是一个数，

而是一个字符串常数,因为它处在引号中。

规则:字符串即使是由数字字码组成的,仍不能算是数字。5,22,3.14,-50等,是一些数字常数;而“HI”,“7”,“TWO”,“3.14”等,则是一些字符串常数。

字符串中的数不能用来做算术运算。

正确:10 LET A=3+7

错误:10 LET A\$=3+7

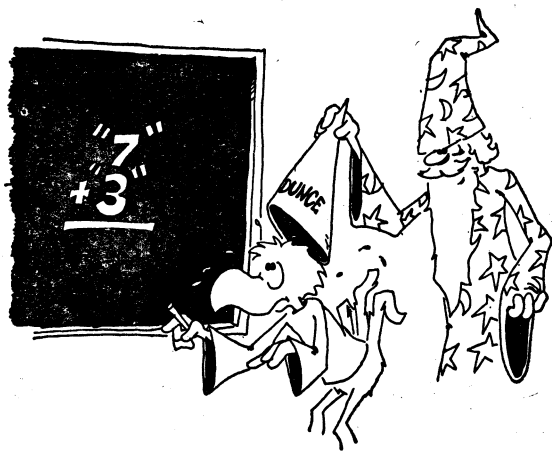
错误:10 LET A="3"+"7"

如果执行上面两个错误行中的任何一个,电脑会打印出:

TYPE MISMATCH ERROR IN LINE 10

现在我们知道了变量类型有两种:“字符串变量”和“数字变量”。决不能将字符串和数混在一起运算。例如:

10 LET A=5



DUNCE——笨学生


```
20 LET B$="10"
```

```
30 LET C=A+B$
```

第 10 行和第 20 行是正确的,而第 30 行却错了。当执行这个小程序时,电脑会做些什么呢?请试一试。

先猜一下下列语句打印出什么,然后把它们分别送入电脑执行,看你猜得对不对。

```
PRINT 5
```

```
PRINT "5"
```

```
PRINT "5+3"
```

```
PRINT "5"+"3"
```

```
PRINT 5+3
```

打印字符串和数值的混合体

可以在同一个 PRINT 语句中打印数和字符串。但需记住,不能把它们混合起来运算。下面这几个例子是正确的:

```
PRINT A;"SEVEN";"7"
```

```
PRINT A;B$
```

请执行

```
10 PRINT 5/2;"IS EQUAL TO 5/2"
```

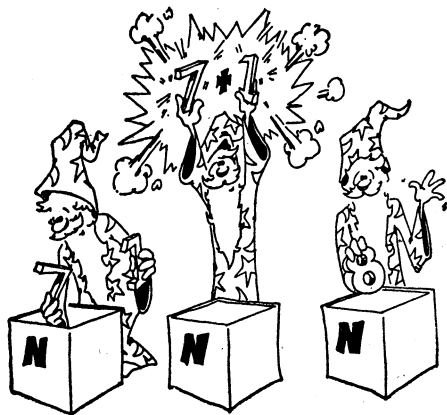
看看在屏幕上出现了什么。

趣味等号

电脑中的“=”符号意思并不是“等于”。看下面的程序:

```
10 LET N=N+1
```

这在算术中是没有意义的。假定 N 为 7 也就是说, $7=7+1$ 显然是荒唐的,不正确的。但是在电脑中,说 $N=N+1$ 就对了。因为“=”符号的真正意思是“替换”,可以理解为 $N \leftarrow N+1$ 。就



是把 $N+1$ 送到 N 盒中去。

再看这一句：

```
10 LET N=N+1
```

其中 $N=7$ 。电脑跑到盒子上写有 N 的盒前，从盒中取出数字 7，再把 1 加到 7 上得 8。然后电脑又把 8 放回 N 盒内。

当然你也可以这样来理解：

```
10 LET N=N+1
```

意思是让 (LET) 新 N 等于旧 N 加 1。

作业10:

1. 编写一个可以询问你年龄及今年是哪一年的程序。然后减去并打印出你出生的年份。注意一定要用 PRINT 语句来说明你要什么以及最后一个数的意思是什么？

2. 编写一个可以询问两个输入数的程序，然后打印出它们的积(将两数相乘)。

第十一课 TAB(打印格式)函数与延时循环

教学参考:

TAB(打印格式)函数给屏幕显示增加了灵活性。延时循环可以降低程序执行的速度,这给观察整个程序的操作过程带来极大的方便;它也可用于必须以一定速度执行的那部分程序,所以也叫做“计时循环”。

TAB 函数能用在 PRINT 语句中,就象打字机上的 TAB 键一样。它比你最初认识到的要有用得多。它可以用字符串来移动图案和显示各种有趣的信息。

如果学生对打字机不熟悉,那就需要老师在他们学习TAB函数时给予特别帮助。

几个 TAB 函数可同时用在一句 PRINT 语句中,但是在括号()中的自变量必须每次增长,因为无法用 TAB 函数把指针移回左边。以后我们还要学习 HTAB(横向定位)语句和 VTAB(纵向定位)语句,这些语句可以把指针放置到屏幕的任何地方。

在 TAB 函数和要打印的内容中间并不是一定要使用分号,但建议还是使用分号为好。

本课将以一种简单的方法来介绍延时循环。

延时循环都在同一行上,用冒号将 NEXT 语句分开,延时的时间量由循环变量的大小来确定。当取值 1000 时,大约延时

一秒钟。

延时循环的主要工作是单纯地进行计数；当计数达到一个特定值时循环就结束；然后执行下一个命令。了解这一点以后，学生就比较容易掌握循环体的执行情况了。

复习题：

1. 如何编写出能持续 2 秒钟的延时循环程序？

2. 下列程序是用于延时循环的吗？

```
120 FOR Q=1000 TO 5000
```

```
122 NEXT Q
```

3. 讲出在下列各种情况下，电脑将做些什么？

```
10 PRINT "HI"; TAB(20); "GOOD LOOKING!"
```

```
10 TAB(5); PRINT "OH-OH!"
```

```
10 PRINT TAB(15); "NOP"; TAB (1); "NOT  
HERE"
```

课 文

TAB(打印格式)函数

PRINT 语句中的 TAB 函数就象打字机上的 TAB 一样，能把打印指针向右移动几个空格。不过打印指针不象光标，它是看不见的。试做一下下面各行语句，看看打印指针是怎样指出下一个要打印的位置的：

```
10 PRINT "123456789ABCDEF"
```

```
20 PRINT 1; TAB(5); 2
```

```
30 PRINT TAB(3); "Y"; TAB(9); "Z"
```

规则：在 TAB(N) 执行后，下一个字符就被打印在第 N 列

上。

注意:当你想执行 10 TAB(5)时,屏幕上会出现SYNTAX ERROR IN 10. TAB(),意思是“第10行 TAB()函数语法错误”。因为 TAB()必须被用于 PRINT语句中,不能单独地使用 TAB()函数。TAB()函数也不能倒退执行。试执行下面的程序:

```
10 PRINT "123456789ABCDEF"  
20 PRINT 1; TAB(9); 9 ; TAB (3); 3
```

TAB()函数只能将打印指针向右移动,而不能移往左边。上面程序错在哪儿,是显而易见的。

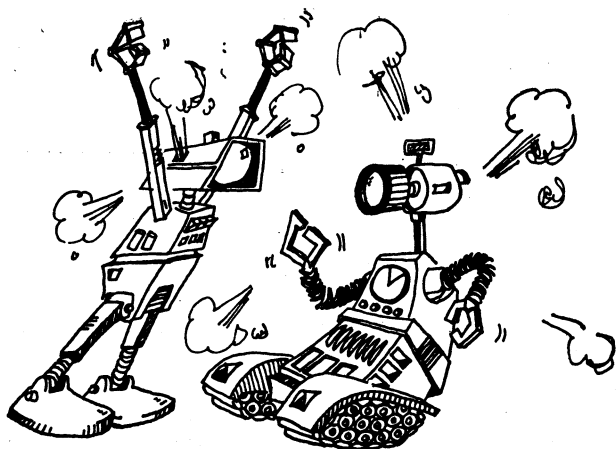
下面的有趣程序能使你的名字往下掉:

```
10 HOME  
15 LET N=1  
20 PRINT "YOUR FIRST NAME"  
30 INPUT W$  
40 PRINT TAB (N); W$  
50 LET N=N+1  
60 GOTO 40
```

按 CTRL=RESET 键可以停止程序的执行。这个程序在屏幕上沿着对角线从左上角到右下角打印出你的名字,就象你的名字往下掉一样。用其他N值试试看,并改变一下行的内容:

```
15 LET N=30  
50 LET N=N-1
```

TAB()函数能打印一个多大的空格



电脑的显示屏幕共有 40 个空格,在TAB()函数的括弧内可以使用 1 到 40 中的任何一个数。大于 40 时,电脑按数字的次序,将打印指针移向下面的行。当数字大于 255 时,电脑在屏幕上会发出错误信息:

ILLEGAL QUANTITY ERROR IN XX

式中 XX 为行号,意思是第 XX 行不合法的数字错误。

TAB()函数也可以和字符串连用,如

```
10 PRINT F$; TAB(15); M$; TAB(25); L$
```

这里 F\$、M\$ 及 L\$ 分别表示你名字的三个字符串。

函数名与自变量

TAB()就是一个“函数”名。以后我们还要研究其他函数,象 RND(), INT(), LEFT\$()等等。()内的数叫做“函数的自变量”。TAB说:“移动指针”,而自变量说:“把它移到某某位置去”。

作业 11A:

1. 编写一个可以询问一个人姓和名的程序。然后从第 5 列开始打印出姓, 在第 25 列开始打印出名; 再使用 GOTO 语句, 使程序可以处理第二个人的姓和名。

2. 请你编写一个“别名”程序。它先问你的真实姓名, 然后发出“嘟嘟”声, 同时显示出你的真实姓名; 接着, 用 TAB 函数将你的别名打印在屏幕规定的位置上。

延时循环

还记得 SPEED(速度控制)语句吗? 它可以用比普通速度慢得多的方式进行打印。

还有一种放慢程序执行速度的方法, 那就是现在要介绍的“延时循环”。

执行下列程序:

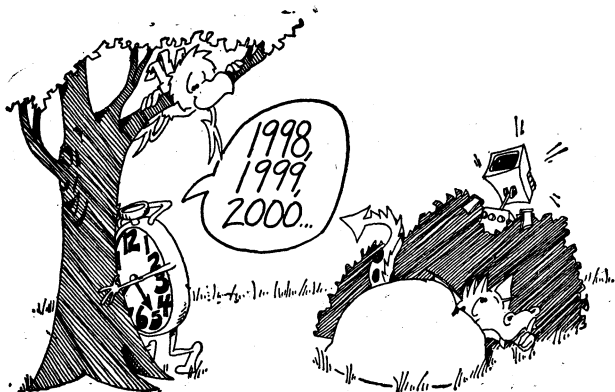
```
10 REM DELAY LOOP
20 HOME
30 PRINT "WAIT"
40 FOR I=1 TO 2000: NEXT I
50 PRINT "DONE"
```

第 40 行就是延时循环。程序从 1 开始计数。当计到 2000 时, 再转入下一行去执行, 它就象捉迷藏游戏中的数数计时一样。

请将第 40 行内的数“2000”改变成其他数, 再试一试。

在延时循环中, 每计数 740 大约就相当于一秒钟时间。请试一试下列程序, 看看是不是大约一秒钟执行完。

```
10 REM.....TICK TOCK.....
```



```
20 HOME
30 INPUT "WAIT HOW LONG?"; S
36 LET T=S*740
40 FOR I=1 TO T: NEXT I
45 PRINT: PRINT CHR$(7)
50 PRINT S; "SECONDS ARE UP"
```

作业 11B:

编写一个“慢性子人”姓名的程序,使它能打印出三个词。每个词之间有几秒钟间隔,并使电脑在每个词之前都发出“嘟”声。

第十二课 带有数字的IF(条件)语句

教学参考:

本课中用的 IF(条件)语句将扩展到算术表达式。本课中用到的逻辑关系有:

=(等于)、>(大于)、<(小于)、< >(不等于)

课文中还要介绍 IF 语句的嵌套用法。

在本课作业第二题的猜谜游戏中,对循环程序已作了说明,但还没有讨论过。循环从第 50 行开始到第 80 行为止。判断是否跳出循环在第 70 行进行。这种循环逻辑就是“直到……才”方式。

复习题:

1. 在 IF 语句中哪部分可能是真或是假?
2. 在 IF 语句中,THEN 后面跟的是什么?
3. 在下列程序执行后,盒子 D 中是什么内容?

```
10 LET D=4
```

```
15 IF 3<7 THEN LET D=9
```

4. 在上题条件中,改成 $3>7$,其他不变,那么盒子 D 中是什么?

课 文

请试一下下面名叫“少年”的程序:

```

10 REM * * * TEENAGER * * *
15 HOME
20 PRINT "YOUR AGE?"
30 INPUT A
40 IF A<13 THEN PRINT "NOT YET A
    TEENAGER!"
50 IF A>19 THEN PRINT "GROWN UP
    ALREADY!"

```

这儿的 IF 语句有点象以前使用过的字符串函数。IF 语句的形式为：

IF 条件为真 THEN 执行语句 C

所谓“条件”可能要用到下列几种算术符号：

= 等于

> 大于

< 小于

<> 不等于

每一个“条件”都是一个短句，可以用“数学语言”来写。例如：

$A < > B$ 读作 “A 不等于 B”；

$5 < 7$ 读作 “5 小于 7”。

练习：

在下面的例子中，使 LET A=7, LET B=5 及 LET C=5。说出每个“条件”是真还是假。

A = B

A > B

$A < B$

$A = C$

$A < C$

$A > C$

$B = C$

$B > C$

$B < C$

$A < > B$

$B < > C$

包含在 IF 语句中的 IF 语句

上面的“少年”程序缺少一点东西，现在补上：

```
60 IF A>12 THEN IF A<20 THEN PRINT  
    "TEENAGER!"
```

为了弄清它，可先把它分为两部分：

语句 C

式中语句 C 就是

```
IF A<20 THEN PRINT "TEENAGER!"
```

这整个一行首先问：“年龄大于 12 岁吗？”

如果答案是“YES”（是的），这一行就问第二个问题：“年龄小于 20 岁吗？”

如果答案仍然是“YES”，这一行就打印出“TEENAGER!”（少年）

如果在这两个答案中有一个回答是否定的，它就无法执行 PRINT 语句，因而也就不会打印。

作业 12:

1. 给上面第 60 行画一张“三叉路口”的示意图，要有两个分叉(参看第 9 课中的示意图)。

2. 请看下面的“猜谜游戏”程序：

```
10 REM ... GUESSING GAME ...
15 HOME
20 PRINT "TWO PLAYER GAME"
25 PRINT
30 PRINT "FIRST PLAYER ENTER A
NUMBER FROM 1 TO 100"
35 PRINT "WHILE SECOND PLAYER ISN'T
LOOKING"
37 PRINT
40 INPUT N
45 HOME
50 PRINT TAB(12); "MAKE A GUESS";
55 INPUT G
60 IF G<N THEN PRINT "TOO SMALL"
65 IF G>N THEN PRINT "TOO BIG"
70 IF G=N THEN GOTO 90
80 GOTO 50
90 REM THE GAME IS OVER
92 PRINT
93 FLASH
95 PRINT "THAT'S IT!"
99 NORMAL
```

如果你要在磁盘上存入这个程序,请学习第 14 课内容。

第 80 行的作用是转向到第 50 行去,使你能多次猜谜直至猜中为止;如果 $G = N$ 时,执行第 70 行就跳到第 90 行;然后空一行打印出“THAT'S IT”。

3. 说出在下面的条件下,上面的猜谜游戏程序里第 50 行到第 80 行中各行发生的情况:

(1) 如果 G 为 31, 而 N 为 88。

(2) 如果 G 为 88, N 为 88。

4. 这里又是一个程序,它打印出什么? 一共打印多少次?

```
10 LET N=1
20 IF N=13 THEN PRINT "UNLUCKY!"
30 LET N=N+2
40 IF N>30 THEN GOTO 99
50 GOTO 20
99 PRINT "DONE"
```

如果第 10 行变成:

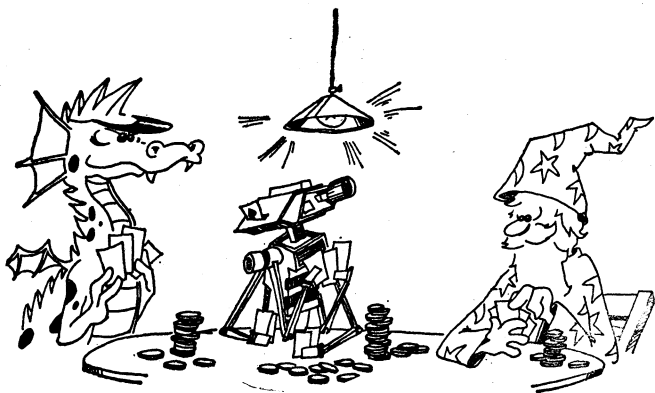
```
10 LET N=2
```

它会打印出什么呢?

5. 编写一个程序,它能说出从 1 到 10 每个数所代表的意义。玩游戏者输入一个数,电脑就打印出这个数的意思。例如:“3 即三次击球,你出界了”或“7 很幸运”等等。

6. 编写一个“数字钟”程序:用计时循环来计算秒,先输入游戏开始时的时、分及秒;时钟于是就开始运行了,一秒一秒地加,并把秒值打印出来。达到 60 秒时,在分上加 1,再把秒值返回零。对小时也同样处理。长时间地使用“时钟”并不断调节计

时循环,这样“时钟”就能走得很准确。



第十三课 RND(随机数)函数和 INT(取整)函数

教学参考:

本课将介绍两种函数: RND(随机数)函数和 INT(取整)函数。这两种函数在游戏程序中非常有用,而且能方便地使电脑屏幕进行万花筒般的趣味显示。

RND 函数产生出一个 0.0 到 1.0 之间的随机数。这种数的取得就象在掷骰子得到 1~6 或发扑克牌得到 1~13 中的任意一个数一样。

一般的学生对小数运算可能不太熟练,但这并没有多大的影响,这里用到的是由一个整数乘上小数随机数,有时也许还要加上一个整数;而且乘法是由电脑来做的,所以只要对运算的结果有一个粗略的了解就行了。

有时我们还需要将随机数的范围扩展到比 0~1 更大些,并把它转换成整数。INT(取整)函数能通过简单地截去尾数,即“把小数部分扔掉”来做到这一点(遇到负数时,情况要复杂一些,这里不讨论这种情况)。

本课中再次使用了“函数”的概念,我们将作进一步说明。

一个函数在另一个函数括弧内的嵌套,将通过在 INT 函数的自变量中的 RND 函数来加以说明。

复习题:

1. 说明电脑在下列各种情况时第10行语句将打印些什么?

```
10 PRINT INT (G)
```

“盒子G”取值: 2, 2.1, 2.95, 3.001, 67, 0, 0.2

2. 说出 INT() 函数与“四舍五入”数有何不同, 对你来说哪一种方法比较容易掌握?

3. 想一想如何用 INT() 函数对一个数进行四舍五入处理。

4. RND(8) 函数会做些什么呢?

5. 如何能得到所有的从 0 到 10 的随机整数?

提示: INT(RND(8) * 10) 并不完全对。

6. 如何获得从 5 到 8 的随机整数呢?

课 文

RND(随机数)函数

在掷骰子时, 无法预料会出现什么数; 在发扑克牌时, 也无法预料每个人会得到什么牌。

电脑可以用某种方式来让你“掷骰子”, 或“发扑克牌”, 并且能做其他许多难以预料的事情。

它是用 RND 函数来实现的。RND 表示“随机”的意思。

执行下列程序:

```
10 REM RANDOM NUMBERS
20 HOME
25 LET N=RND(8)
30 PRINT N
```




JACKPOT——头号号码

```
40 IF N<95 THEN GOTO 25
```

在屏幕上可以看到许多小数,这就是第25行中的 RND 函数起的作用。

括弧内不管放上什么数都没有关系,只要是整数就行了。这里选择“8”,是因为在键盘上靠近“()”符号,比较容易打入。

RND 函数给出大于 0 但小于 1 的小数。要使数大于 1,则要做乘法。例如将上面的程序改变为:

```
25 LET N=RND(8)*52
```

```
40 IF N<45 THEN GOTO 25
```

并重新执行这段程序。

现在数的范围就在 0 到 52 之间了。它们可用来选择一付扑克牌中的 52 张纸牌。但是,我们通常需要的是象 7 和 8 这样的整数,而不要象 7.03454323 及 8.89746582 这样的小数。这样

我们就要请 INT(取整)函数来帮忙了。

INT(取整)函数

INT ()函数是将括弧内的数去掉小数部分, 留下整数部分。

看下列各个小程序中的 INT 函数:

```
10 LET I=INT (6.3)
```

```
20 PRINT I
```

```
10 LET X=0.3
```

```
20 PRINT "X="; X; TAB(10); "INT(X)=";  
INT(X)
```

```
10 LET X=3
```

```
20 LET Y=2.5
```

```
30 LET P=X+Y
```

```
40 LET Q=INT(X+Y)
```

```
50 PRINT P, Q
```

执行后请查一下答案中的小数部分有没有去掉。再试试:

```
10 REM ..... INT .....
```

```
20 HOME
```

```
30 PRINT "GIVE ME A DECIMAL NUMBER"
```

```
32 INPUT D
```

```
35 LET I=INT(D)
```

```
40 PRINT "DECIMAL"; D; TAB (20); "IN-  
TEGER"; I
```



INTEGER——(取)整数

```
50 IF I < > 0 THEN GOTO 30
```

输入零可以结束这一个程序。

下面向你提供一个“掷骰子游戏”的程序，它执行起来真象是在掷一只骰子一样，不妨请试试：

```
10 REM // // // // ONE DIE // // // //  
20 HOME  
30 LET R=RND(8)  
40 PRINT "RANDOM NUMBER"; TAB(20); R  
50 LET S=R*6  
55 PRINT "TIMES 6"; TAB(20); S  
60 LET I=INT(S)  
65 PRINT "INTEGER PART"; TAB(20); I
```

```

70 LET D=I+1
75 PRINT "DIE SHOWS"; TAB(20); D
77 PRINT
80 PRINT "ANOTHER? <Y/N>"
82 INPUT Y$
85 IF Y$="Y" THEN GOTO 20

```

()里放什么

下面的例子可以说明在 INT 函数和 RND 函数的括号内可以放什么内容：

数：10 LET X=INT(34.7)

变量：10 LET X=INT(J)

表达式：10 LET X=INT(3 * Y + 2)

函数：10 LET X=INT (RND(8))

这里介绍一种可以节省大量空间的方法。

原程序为：

```

30 LET R=RND(8)
50 LET S=R*6
60 LET I=INT(S)
70 LET D=1+I

```

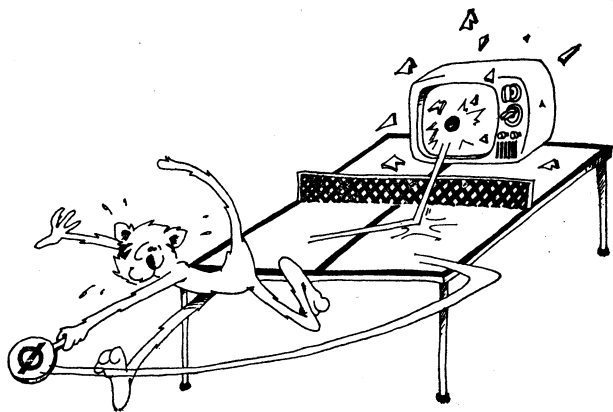
用 70 LET D=1+INT(RND (8)*6) 替代原来的 70 行就能达到这样的目的。

作业13:

1. 写一个程序,能掷出两只分别叫 D_1 和 D_2 的骰子,并打印出每次掷出的在 D_1 和 D_2 上的数字以及它们的和。这里不需要用到上面课文中介绍的程序中的变量 R、S 和 I,但这些变

量可以指示你找到比较理想的答案。

2. 编写一个“PAPER(纸头)、SCISSORS(剪刀) 和 ROCK(石头)”的游戏程序。你与电脑对玩(纸包石头、石头砸剪刀、剪刀剪纸)。电脑用 RND()函数选择一个数: 1 或 2 或 3。1 是纸, 2 是石头, 3 是剪刀。你的选择用 INPUT(键盘输入)语句输入 P、R 或 S 来决定, 让电脑判别输赢并记分。



第十四课 信息存入磁盘

教学参考：

本课将阐述如何把程序存放到磁盘上去，以及如何再把程序取出来。如果你用的苹果机没有配置磁盘驱动器，可参考附录2，用盒式磁带来存贮程序。

这节课要介绍的命令有：

SAVE(存盘)命令 LOAD(装入)命令

CATALOG(显示文件目录)命令 DELETE(删除)命令

如果你的磁盘片还没有为学生的专用而作过“加工”，那么应该先做这一步工作。然后再学习课文。在附录中有怎样“加工”的详细说明，并附有问候 HELLO(您好)程序供使用。

其他用到的语句或命令还有：

NEW(清内存)命令 REM(注释)语句

HOME(清屏幕)语句 PRINT(打印)语句 LIST(列表)命令

本课内容在学完第三课后的任何时候都可以学习。到现在才介绍这一内容，是因为以前课文中的大部分程序都相当短而且乏味，不值得存贮。前几课主要强调的是编制程序的过程而不是程序的实用性。这一课内容应该由读者自己决定放在第三课后的什么地方学习比较好。这样学生可以把他们感到满意的程序存放起来，以便随时调用。

复习题：

1. 什么叫文件？
2. 文件名可以有多长？
3. 什么标点符号不能放在文件名中？文件名中能有空格吗？
4. 如果使用了没有文件名的 SAVE 命令，你怎样使电脑回到正常状态？
5. “LOAD 文件名”这命令做些什么？ CATALOG 命令又做些什么？
6. 如何把不需要的文件从磁盘中抹去？
7. 如果一个程序已放到“文件”中去了，电脑的存贮器里还存在这个文件吗？

课 文

如果你没有磁盘，请看书后面的附录 2，那里将介绍怎样把程序存放到磁带里去。

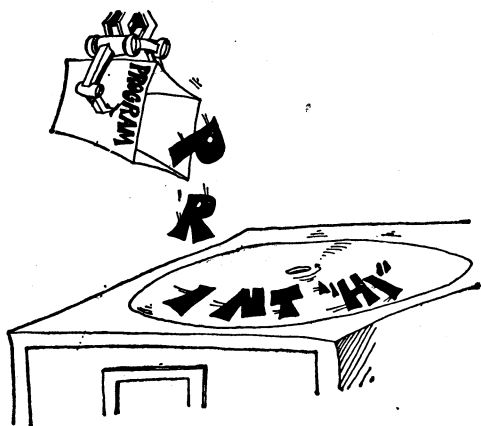
输入一个程序

如果电脑内事先已经有了一个程序，就不必再另写一个程序输入了。否则，请输入：

```
NEW
10 REM ::: HI :::
20 HOME
30 PRINT "HI"
```

存放程序：

请先看一看！在驱动器中已放进磁盘片了吗？如果没有，



PROGRAM——程序

把一片磁盘放进去。一定要把驱动器的门关好。然后输入：SAVEHI 这时候可以看到驱动器上的红灯亮了，还可以听到“呼呼”的声音。当红灯熄灭时，“呼”声就停止了，说明程序已经被存入磁盘中了。

现在，你的程序以“HI”为文件名被保存在磁盘上。可以把磁盘看成是一个文件柜。里面可以放许多文件。刚才送入的程序取名为“HI”的原因是容易记忆。现在它已经存放在文件柜中了。如果你的程序不用这个名字，那可以用另一个正确的名字重新把它存放起来。

注意：

单词“SAVE”后面一定要跟文件名，否则电脑会认为你要把程序存放到盒式磁带里去，而不是存入磁盘里去。

如果你把程序搞乱了，按一下 CONTROL-RESET键就可以

回到正常状态。

CATALOG(显示文件目录)命令

假使要看一下程序是否真正存贮在磁盘上,可以输入:

CATALOG

在发出“呼呼”声,红灯熄灭后,就会看到:

A 002 HI

A的意思是文件目录中包含有APPLESOFT程序。002表示这是一个短程序。在磁盘上只占据二个扇形区,一张磁盘片有496个扇形区可供你存贮信息。可以把每个扇形区看成是文件柜中的一个文件夹子。

程序的装入

既然已确信程序被存放在专用的磁盘里了,那么就可以放心地把它从电脑的存贮器中抹去。

输入:

NEW

HOME

LIST

LIST执行后屏幕上显示不出什么东西来。因为NEW命令把程序从电脑的存贮器中抹掉了。现在让我们把那个程序取回来。

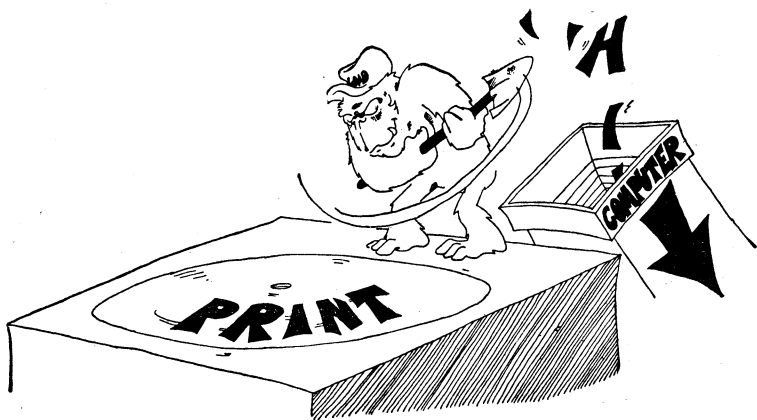
输入:

LOAD HI

听到“呼”的一声,并看到红灯亮过后,程序现在又进入存贮器中了。

再输入:LIST看看是不是这么回事。

删去一个文件



COMPUTER——电脑

如果你突然改变了主意，要把记录在磁盘里的一个文件去掉，用什么办法呢？

只要输入：



DELETE HI

再输入：

CATALOG

检查一下看看 HI 文件是否真的不在磁盘里了。

合法的文件名

使用文件名应注意下列几点：

- (1) 一定要以字母开头；
- (2) 文件名最长不能超过 30 个字符；
- (3) 文件名中可以有数字；
- (4) 文件名可以有标点符号，但是不能有逗号；
- (5) 还可以有空格。

一般文件名最好取短一点，这样可以少按些键。

好的文件名如：

JUMPING CAT(跳猫)

GUESSING GAME(猜谜游戏)

SUB(减法)

错误的文件名如：

7UP(以数开头)

CAT, DOG(里面有逗号)

命令

下面这四个命令与文件名一起使用，

SAVE 文件名

LOAD 文件名

CATALOG

DELETE 文件名

在看到有“文件名”的地方，一定要打上文件名。

在 SAVE 和 LOAD 命令中，总是要用文件名。否则，电脑就认为你是要用磁带录音机而不是用磁盘来存放信息。

如果在 DELETE(删除)命令中忘了使用文件名，电脑就会打印出 ?SYNTAX ERROR(语法错误)，这时应该再补上文件名。请随便举个例子试一下。

作业14:

1. 写一个 4 行短程序，把它存放在磁盘里。
2. 先用 NEW 命令清除存贮器，再另外写一个短程序，也把它存放在磁盘里。
3. 先用 NEW 命令和 CATALOG 命令，然后依次装入上面的程序，并执行这个程序。
4. 用 DELETE 命令删除其中的一个程序。
5. 用 SAVE、LOAD、CATALOG 及 DELETE 等命令进行反复练习，直到对它们已经理解了为止。

第二部分 绘图、游戏及其他

第十五课 语句的简略形式

教学参考：

本课要介绍几种书写电脑语言的简略方法：

1. 用? 来代替PRINT语句；
2. 在赋值语句中LET可以省略；
3. ; 号可将两个或几个语句并列放在同一行上；
4. INPUT语句除用于输入外，还可以打印一些内容。

到目前为止，我们已学会使用许多语句，其中包括 RND 函数以及如何将程序存放到磁盘上去等。这些内容可供学生编写出许多丰富多采的程序来。

用冒号可以把几个语句并列放在同一行上，从而使整个程序缩短。但同一行包含的几个语句应具有一些共同的性质。

这种方法也可能会把一个程序搞乱。例如：用GOTO语句转向到的语句，如果把它移到其他一行的中间，你就会得到错误的结果。

有时连经验丰富的程序设计员偶尔也会搞错。他们把一些语句移到有IF语句行的条件后面，这就改变了程序原有的逻辑

关系。因为只有当IF语句中的条件成立，语句才能执行。

另一方面，利用冒号把由几个语句并列组成的小“程序”放在同一行的IF语句后面，这就不必使用GOTO语句了。这样整个程序就缩短了，显得简洁明了。所以使用冒号是改进程序清晰度的一种有效的办法。在使用INPUT语句时，如果没有提供输入内容，在屏幕上就会显示出“？”符号，同时出现闪烁的光标，这是在提醒你要输入必要的内容；如果已经输入了内容，在屏幕上就不出现“？”符号，而只显示闪烁的光标。所以当输入的内容是一个问题时，你就必须同时输入一个问号，否则屏幕上就没有“？”符号。

复习题：

1. “？”符号可用来表示什么语句？
2. 如何说明在赋值语句中可省略LET？
3. 在一句INPUT语句中，引号内是输入的内容，现在问在引号后应跟什么标点符号？
4. 把两个语句放在同一行上，用冒号分开，这样做有什么好处？
5. 下面各行错在哪里？

```
10 REM BEGINNING; GOTO 1000
```

```
10 GOTO 50; S$="FAST"
```

课 文

PRINT(打印)语句的简略形式

用一个?号可以代替PRINT语句。

请输入：

```
10 ? "HI"
```

```
LIST 10
```

执行这个程序,屏幕上会打印出

```
HI
```

LET(赋值)语句的简略形式

下面这两行作用相同:

```
10 LET A=41
```

```
10 A=41
```

这两行也相同:

```
20 LET B$="HI"
```

```
20 B$="HI"
```

在赋值语句中,可以省略词汇 LET。一行开始,由变量名



后面紧跟着“=”符号,电脑就知道执行LET语句。

INPUT(键盘输入)语句的简略形式

下列程序:

```
10 PRINT "ENTER YOUR NAME"  
20 INPUT N$
```

可以改写成:

```
10 INPUT "ENTER YOUR NAME"; N$
```

很明显,要把两句变一句,只要在输入的一个短句和一个变量名之间放一个分号就行了。再看几个例子:

```
10 INPUT "AGAIN ? <Y OR N>"; Y$  
20 INPUT "LOCATION"; X, Y  
30 INPUT "MONTH, DAY, YEAR"; M$, D, Y
```

LIST(列表)命令的几种功能

LIST命令有5种不同的用法:

1. 单用LIST命令可以显示出整个程序;
2. LIST 48显示出第48行语句;
3. LIST 50—75显示从第50行到第75行之间所有的语句;
4. LIST—27 显示出从开始一行到第27行之间所有的语句;
5. LIST90— 显示从第90行到结束一行之间所有的语句。

缩写语句行

把几行语句合并到同一行内,中间用冒号“:”分开,这样就可以缩短程序,节省空间。

例如下列程序

```
10 Q=17*3
```



```
20 R=Q+2
30 PRINT R
```

可以缩写成：

```
10 Q=17*3; R=Q+2; ? R
```

在存储器中，这一行可以看成是：

```
10 Q=17*3; R=Q+2; PRINT R
```

一般说来，为了使程序更加简洁、明了，我们常常需要合并语句行。在合并时可将相同类型的语句放在同一行上。例如下列程序：

```
10 X=0
12 Y=0
14 Z=0
```

可改写成：

```
10 X=0; Y=0; Z=0
```

我们还常常用 REM 语句放在某行的末尾作注释。例如：

```
40 H=X+Y/66; REM H IS THE HEIGHT
```

其中 H IS THE HEIGHT 意为H是高度。

IF(条件)语句中的冒号

IF语句中使用冒号后，也可以使程序简洁。

例如程序：

```
50 IF A=0 THEN GOTO 80
60 B=Q
62 C=B*D
64 FLASH
66 PRINT "WRONG"
```

```
80 FOR .....
```

可改写成：

```
50 IF A < > 0 THEN B=Q; C=B*D;  
    FLASH; PRINT "WRONG"
```

```
80 FOR .....
```

如果“A<>0为真”，所有要执行的语句都放在THEN后面。

注意：

不要在 IF 语句中随便放入不属于它的东西。例如下面两句：

```
35 IF A=B THEN PRINT "ALIKE"  
40 Q=R
```

把它们并成一句：

```
37 IF A=B THEN PRINT "ALIKE"; Q=R
```

意思就截然不同了。上面两句中，不管条件是真是假，第40行中的Q=R语句总是要做的；但是在第37行中，只有在条件A=B成立时Q=R才能执行。

冒号使用中的一些错误

在用冒号把一行中的几个语句分隔开时，REM语句或GOTO语句必须放在这一行最后一部分的位置上。因为电脑对这两个语句后面的任何命令都不加理会，也就不会去执行了。下面举几个例子：

正确写法：35 P=3; REM P IS THE PRICE

错误写法：35 REM P IS THE PRICE; P=3

正确写法：40 R=P+1; GOTO 88

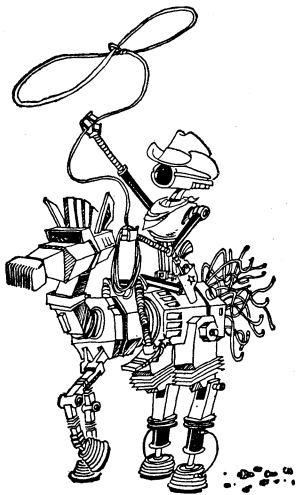
```
42 S=3
```

错误写法：40 R=P+1; GOTO 88; S=3

上面两个错误语句原因在于：电脑在读REM语句或GOTO语句后，不会再去执行后面的P=3或S=3语句了。

作业15:

1. 编写一个程序，要求各种省略方法至少使用一次。
2. 编写一个“假期”程序。先问你要度多少天假期，然后它可以给你当参谋，告诉你去哪儿并干些什么。



第十六课 VTAB(纵向定位) 语句和HTAB(横向定位)语句

教学参考:

用 HTAB(横向定位)语句和 VTAB(纵向定位)语句可以将打印指针移到屏幕上的任何一点。

这些语句对于修改文本或绘制图画很有用处。

在前面课文中曾讲过的 TAB 函数只能用在 PRINT 语句里边。而 HTAB 语句和 VTAB 语句却是用在 PRINT 语句之前的。

为了有效地使用这些语句,要把屏幕看成是由40列、24行组成的。这种规定的优越性在以后处理各种屏幕上的图画时将会体现出来。

复习题:

1. 如果要在第12行上打印一个词,应该用什么语句?
2. 如果在第6行上先空开20个空格,然后再打印一个字符,你要使用哪两个语句?
3. 在回答第2题时,哪个语句先用,这有没有区别?
4. 说明如何在同一行上打印两个单词“FAT”和“CAT”,要求先在第25列处打印“CAT”,然后在第5列处打印“FAT”。

课 文

VTAB(纵向定位)语句

屏幕上一共可以分成 24 行。VTAB(纵向定位)语句可用来选择将一个字符或语句打印在从第 1 行到第 24 行中的任何一行上。

执行这个程序：

```
10 REM VTAB DEMO
15 HOME; SPEED=1
18 VTAB 10; PRINT "LINE 10 FIRST"
20 VTAB 1; PRINT "LINE 1 NEXT"
30 VTAB 24; PRINT "LINE 24 LAST"
40 SPEED=255
```

再执行一个程序：

```
10 REM : : : WHICH LINE : : :
20 HOME
30 INPUT "WHICH LINE"; L
40 VTAB L
50 PRINT L; "HERE,";
60 GOTO 30
```

看看 VTAB 语句是怎样起作用的。

如果 VTAB 语句之后的数为零或大于 24，电脑就会打印：“ILLEGAL QUANTITY ERROR IN LINE 40.”意思是“第 40 行语句不合法的数字错误”。

HTAB(横向定位)语句

屏幕的横向可分成 40 列。HTAB(横向定位)语句可用来选择在从第 1 列到第 40 列中的任何一列上打印字符或语句。以前所学的TAB函数只能用在 PRINT语句中,而 HTAB 命令可以单独使用。不妨请试试下列程序:

```
10 REM...HTAB DEMO...
12 HOME; SPEED=1
20 VTAB 12; HTAB 35; PRINT "HERE"
30 VTAB 12; HTAB 5; PRINT "THEN
   HERE"
40 SPEED=255
```

再试试:

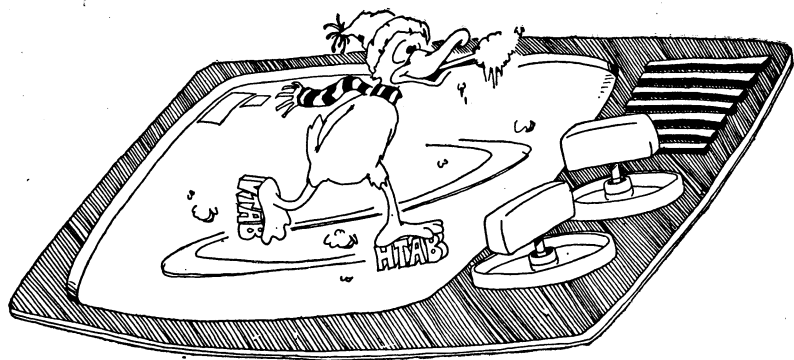
```
10 REM: : : WHICH ROW, COLUMN: : :
20 HOME
30 VTAB 1; HTAB 1; INPUT "WHICH
   ROW, COLUMN?"; R, C
35 VTAB 1; HTAB 1; PRINT " "
40 VTAB R; HTAB C; PRINT "*"
45 FOR T=1 TO 2000; NEXT T
60 GOTO 30
```

按一下RESET(复位)键可以停止程序的运行。

第 30 行意思是要你输入两个数,它们中间用逗号分开。如果你在操作时只打印一个数就按回车键,电脑就会显示出:

? ?

这是在提醒你还要输入另一个数,如果你键入用逗号分开的数太多,屏幕就显示出:



EXTRA IGNORED

意思是：这些多余的数一点也没有用。

屏幕与格子纸

屏幕很象一张格子纸，让我们给“行”自上向下从1到24编号，用变量Y表示；“列”自左向右从1到40编号，用X表示。数字以左上角为开端。执行下列程序，格子纸跃然屏幕上。

```
10 REM...GRAPHS...
20 HOME
25 PRINT "GIVE ME SOME X AND Y'
    S."
27 PRINT "X FROM 1 TO 40, Y FROM
    1 TO 24"
28 PRINT
30 INPUT "ENTER 'C' TO CONTINUE";
    X$
```

```
32 HOME
35 INPUT X, Y
40 HTAB X: VTAB Y
50 PRINT "*"
55 HTAB 1: VTAB 1: PRINT " "
60 HTAB 1: VTAB 1: GOTO 35
```

作业16:

1. 使用 RND() 函数在屏幕上随机地写出你的姓名，然后在整个屏幕范围内随机多次打印它。

2. 编写一个程序，使它能在屏幕的随机点上打印“HERE”。每写好一次，就抹去一次，再写一次，再抹去一次，……在这个程序中还请放进延时循环，这样做是为了让“HERE”不致于跳动太快。

第十七课 FOR-NEXT(循环)语句

教学参考:

本课要介绍形成循环的FOR, NEXT及STEP(步长)语句。

循环由两个语句组成:由一句FOR开始,再用另一句NEXT呼应。这两个语句使用时,中间可以有几行其他语句,但两者之间的关系很密切。这一点对学生来说可能会引起混淆。重温一下前面课文中学到的延时循环可以帮助学生对循环语句的认识,即FOR...NEXT是一对语句。要着重说明循环是重复执行这对语句中间的一组语句而形成的。

“嵌套循环”概念的引入是以延时循环作为一个内循环的情况来介绍的。

本课对循环的一些细节没有进行讨论,有一些在后面的课文中陆续会介绍的。循环至少要兜一圈,因为退出循环是由NEXT语句决定的,而要遇到它只有执行了整个循环体才行。

程序在执行中一遇到FOR语句就进入循环。其中输入循环变量初值的方法和输入其他数字变量相同。STEP(步长)值,终止值以及在FOR语句后面的第一个语句的地址都放在存贮器里。

循环开始以后的执行情况都由NEXT语句来控制。当遇到NEXT时,循环变量按步长值增加,并与终止值比较。如果循环

变量大于(在步长为负的情况下为小于)终止值, NEXT 语句就命令退出循环, 转向执行它后面的语句。否则就继续执行循环体内的语句。

由于循环变量的处理和 BASIC 语言中的其他变量处理一样, 所以在循环体内可以随时改变这些变量。不过在改变时应特别小心, 因为 NEXT 语句是要通过改变这些变量来决定循环是否已结束的。

程序从外部跳入循环体内一般是不允许的。但在 NEXT 语句之前跳出循环是可以的。不过这会给查找程序执行的顺序带来困难, 尤其是涉及到有子程序的地方。

复习题:

1. 编写一个能打印数字 0~20 的循环程序。
2. 编写一个能打印数字 30~20, 步长为 2 的循环程序(注意步长应为负值)。
3. 编写一个嵌套循环程序, 打印数字 100、200、300, 还要求在它们之间不同的行上打印出数字 1、2、3、4、5。

课 文

还记得延时循环吗? 让电脑从 1 计数到 2000, 然后继续执行下面的语句:

```
30 FOR T=1 TO 5: NEXT T
```

其实, 电脑比这还要高明。它可以在计算的同时做其他事情。

执行下列循环程序:

```
10 REM COUNTING
```

```

20 HOME
30 FOR I=5 TO 20
40 PRINT I
50 NEXT I

```

循环可以从任意一个数开始,到任意一个较大的数结束。用下列语句依次来改变第 30 行,再试一试:

```

30 FOR I=100 TO 101
30 FOR I=-7 TO 13
30 FOR I=1, 3 TO 5, 7

```

下列程序用箭头表示出循环部分:

```

10 REM ON PAPER
20 HOME
┌──30 FOR I=0 TO 7
│ 40 PRINT I
└──→50 NEXT I

```

STEP(步长)语句

在上面的程序中每循环一次数值变化的增量为1,现在我们把它改为2。这样第 30 行就写成:

```
30 FOR I=10 TO 30 STEP 2
```

循环变量每次的增量,就称为“步长”。上列程序原来的步长为1,后来步长又改为2。

作业 17A:

请编一个循环程序,让电脑从 0 到 100 计数,步长为5。

负步长循环

也可以使用负步长来执行循环语句。

试做下列程序：

```
10 REM *** APOLLO 11 ***
20 HOME; SPEED=100
30 PRINT "T MINUS 12 SECONDS AND
    COUNTING"
40 FOR I=11 TO 0 STEP -1
50 PRINT I; PRINT CHR$ (7)
60 FOR J=1 TO 740; NEXT J; REM
    TIMING LOOP
70 NEXT I
75 INVERSE
80 PRINT "ALL ENGINES RUNNING, LIFT
    OFF."
82 PRINT "WE HAVE A LIFT OFF."
84 PRINT "32 MINUTES PAST THE HOUR."
86 PRINT "LIFT OFF ON APOLLO 11."
99 SPEED=255; NORMAL
```

第40行中步长为“-1”；第60行为计时循环。

嵌套循环

在上面介绍的程序中，一个循环里面还包含着另一个循环：外部循环从第40行开始，到第70行结束；内循环在第60行。

这就是嵌套循环。它就象小孩子的一套玩具盒，大玩具盒里还套有小玩具盒。

循环变量

为了使每条FOR语句知道哪个NEXT语句和它配对，NEXT



后就紧跟一个“循环变量”名来作记号。还是看上面程序的第60行：

```
60 FOR J=1 TO 740; NEXT J
```

其中 J 就是循环变量。再看第 40 行和第 70 行的循环变量：

```
40 FOR I=12 TO 0 STEP -1
```

...

```
70 NEXT I
```

I 也是循环变量。

错误的嵌套循环

嵌套循环的内循环必须全部在外循环的内部。

```
错误：
    25 FOR X=3 TO 7
      30 FOR Y=3 TO 7
        40 PRINT X * Y
      →50 NEXT X
    →60 NEXT Y
```

正确：

```
—25 FOR X=3 TO 7  
—30 FOR Y=3 TO 7  
40 PRINT X*Y  
→50 NEXT Y  
→60 NEXT X
```

作业17B

1. 编写一个程序，能把你的姓名打印15次。
2. 对上题的程序作改进，要求每打印一次，姓名前面要多加两个空格。在循环中可用TAB(打印格式)函数，使屏幕上呈现斜线形状。
3. 现在要求把你的姓名打印 24 次。从屏幕底部开始向上打印，在循环中要用 VTAB 语句。
4. 现在要求把你的姓名写在同一行上，你朋友的姓名写在下一行，并且不断重复，直到每个姓名写过 5 次为止。

第十八课 编辑方式与执行方式

教学参考:

本课将较为详细地介绍电脑的编辑方式与执行方式。

尽管这些内容属于基础的东西，本书却还是把它排在后面学习。这是因为它们比较抽象，另外也考虑到不要影响BASIC语言中主要语句或命令的学习。

但是，如果需要也可早一些学习这部分内容。

本课中仅用到 PRINT 语句和RUN命令。

其他要用到的一些名称为：

编辑方式：命令方式 立即方式

 计算器方式 直接方式

执行方式：延时执行方式

编辑方式是电脑用户的基础。在编辑方式中，输入一行内容，字符放在输入缓冲区中，一个缓冲区最多可放 256 个字符。

当按回车键时，电脑就观察输入的这一行是否由一个数字开始；如果是这样，它就把这行内容作为程序中的一行存放起来，并按行的编号按次序排列。如果这行不是以一个数字开头，电脑就从输入缓冲区中将它取出并执行它。通常这一行往往是单个命令，象HOME 或 RUN等。但是，立即方式功能很强，即使

一行程序很长,也可以立即执行。这种特性无论在程序设计阶段要在已经写好的程序行之间插入新的语句行,还是在程序调试期间,一边执行语句行的算术运算,一边修改语句等方面都很有用。

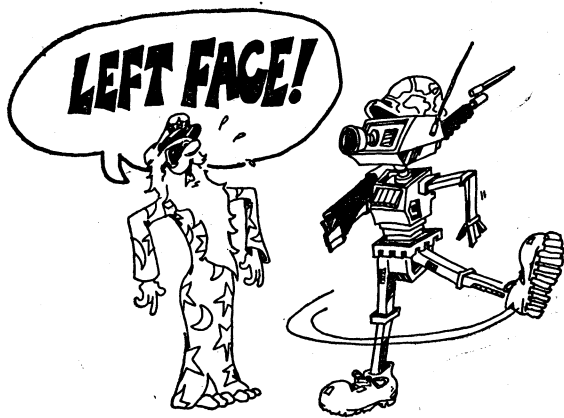
复习题:

1. 电脑在执行方式中是怎样工作的?
2. 怎么能知道电脑是否处在“编辑方式”中?
3. 在编辑方式中,你可以做哪三件事?
4. 如果输入以行号开始的语句,电脑对这一行如何处理?
5. 如何输入没有行号的命令?电脑又是如何处理的呢?

课 文

输入:

NEW



LEFT FACE——向左转

HOME

后准备上课。

执行和运行

“执行”意思就象战士执行“向左转”命令一样。

“执行一段程序”意思就是“运行一段程序”。

延时执行

输入并执行这一个程序：

```
10 PRINT "HI"
```

这是编制和执行程序的一般方法，也叫作“延时执行”。

在延时执行中，电脑一直等到输入RUN命令才执行程序。

延时执行的规则是：如果该行以数字开头，就放入存贮器中，成为存贮器中程序的一部分；程序可由RUN命令来执行。

立即执行

这是一种简便的方法：输入一句不带行号的语句。

```
PRINT "HI"
```

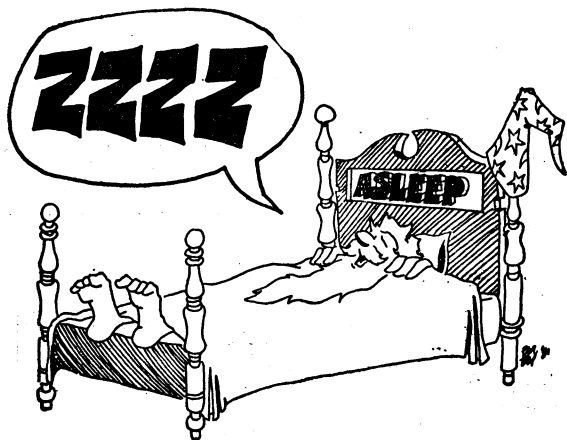
这次电脑会立即打印出“HI”，而不再需要输入RUN命令，这叫作“立即执行”。

立即执行规则是：如果输入的某一行不以数字开头，那么一按回车键电脑就立即执行这一行语句。在立即执行方式中，可以执行由几个语句组成的一行程序。下面是一个例子，请试一试。

```
FOR I=1 TO 20: PRINT I: NEXT I  
PRINT: PRINT "DONE"
```

操作方式

人们在睡眠时处于一种状态，而在清醒时又处于另一状态。



ASLEEP——睡着了

也就是说，人们有两种“操作方式”。

如果某人在打鼾，我们可以说这个人已睡着了。（当然并非所有的人睡着时都会打鼾。但为了解释电脑如何象人一样，不妨就假设所有的人睡着时都打鼾。）

电脑也有两种操作方式，分别称作“编辑方式”与“执行方式”。

编辑方式

按下 CONTROL RESET 键，你在屏幕上可看见“`]`”符号和闪烁的方块。“`]`”符号叫做“提示”符号，意思是电脑处在“编辑方式”。形象些说，“`]`”是电脑处在编辑方式时的“打鼾”信号。

闪烁的方块叫作光标。它告诉我们电脑正在等待输入一些内容，输入的字符就放在光标位置上。

电脑处在编辑方式时，你可以做下面几项工作：

(1) 可以输入一行行以数开头的语句行组成的程序。

(2) 可以象使用袖珍计算器那样来使用电脑，就象使用一架大台式计算器。

(3) 可以纠正程序中的错误，这叫作“编辑”程序。“编辑方式”就是从这儿得名的。后面我们还要详细介绍如何学习编辑程序。

执行方式

输入RUN命令后电脑便离开了编辑方式，进入执行方式。

电脑处在执行方式时，它能自动执行存贮器内的程序；当程序执行完后，又自动回到编辑方式。

作业18:

1. 解释“延时执行”的含意。编写一个有几个语句行的程序，在一行中打印“22 + 67 IS”，然后在另一行做加法，并打印出答案。

2. 解释“立即执行”的含意，把电脑当作计算器使用来做一些算术题。

3. 怎么能知道电脑是处于“编辑方式”呢？

4. 电脑在执行方式中做些什么？

5. 当程序执行完后，电脑又进入什么方式？

6. 你怎样知道下一个输入的字符会出现在屏幕的某个地方？

第十九课 用字符串移动图象

教学参考:

本课要向你提供一个有趣的小程序,它可以在屏幕上“射箭”。

这里将介绍字符串变量的一些作用;加空格能使图象移动,并同时抹去前面的图象;几个字符串可以联结;清晰而形象的活动图案将有助于搞清循环概念和盒子概念。

复习题:

1. 要使图象移动必须抹去旧图象,然后再画上新图象。课文中的“箭”程序是如何完成“抹去”图象这项工作的?
2. 字符串盒子可以存放空格。如何告诉电脑你要一只里面是一个空格的字符串变量D\$呢?
3. 一个循环以NEXT I 语句结束。NEXT对变量 I 做些什么呢? NEXT怎样判定下一步要执行哪一行语句?

课 文

画图象的一种方法是使用字符串。执行下列“箭”程序:

```
10 REM >>>...ARROW.....>
20 HOME
30 S$=" "; REM S$ IS A SPACE
```

```

40 A$=">>>.....>"
42 VTAB 10
44 FOR I=1 TO 29
46 HTAB I
48 PRINT S$+A$; REM GLUING TWO
    STRINGS
50 NEXT I

```

将它存放到磁盘上。

这段程序是射出一支箭，横穿过屏幕。下面对程序中各行语句执行情况作些解释：

第30行：在引号之间有一空格，所以在称作“S\$”的盒子内存贮着一空格。后面的注释语句对此作了说明。

第40行：这个字符串看起来象一支箭。

第42行：VTAB 语句选择第10行，供箭移动。

第44行：这是循环的起点，循环在第50行结束。

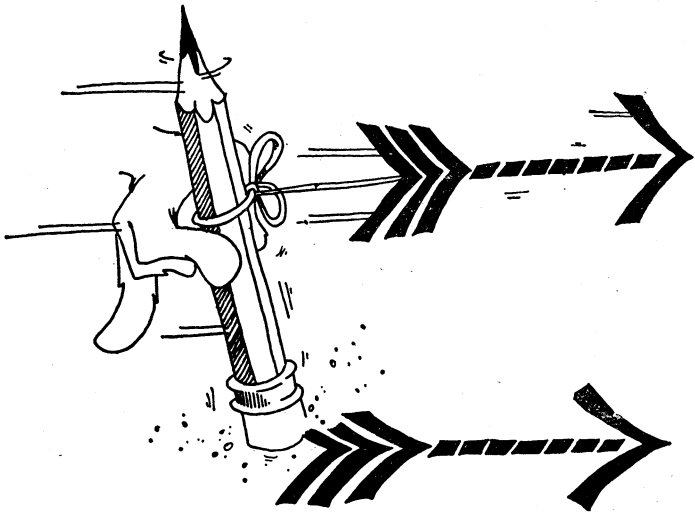
第46行：字符从这行的左边开始打印。

第48行：先把S\$盒内的一个空格连在(A\$盒)箭的末尾的一格上，然后把空格和箭打印在屏幕上。后面的注释对此作了说明。

第50行：NEXT 检查 I 是否大于29。如果不是，程序返回到第46行执行。因为这是 FOR I……语句之后的第一条语句。

第46行：这时从这行的左边开始打印一个空格。

第48行：这时打印的一支箭比刚才向右移过了一格，箭羽尾部被一个空格所代替，旧箭就这样被抹去了。



第50行：NEXT 再重复检查一遍。

现在请你自己把循环过程往下复述一下。

如何抹去旧箭

画出程序最先三次打印的箭。用“b”表示箭尾的空格，以便识别它。于是得到：

```
b>>>.....>
  b>>>.....>
    b>>>.....>
```

每次打印新箭时，旧箭除了羽尾上的最后一个“>”外，其余部分都被新箭覆盖了。最后一个“>”由新箭尾部的空格来覆盖，这就相当于一支箭向右移了一格。

空字符串和带有空格的字符串

执行下列程序：

```
20 A$="A"; B$="B"
```

```
30 S$=" "
```

```
40 PRINT A$; S$; B$
```

电脑在A与B之间打印出一个空格。空格来自第30行，处在引号中。所以名为S\$的盒子内有一个空格。

改变第30行并重新执行：

```
30 S$=""
```

现在引号之间没有什么东西，所以在名叫S\$的盒子内也没有任何东西。

由此可以看出：有一个空格的字符串与空字符串是不同的。

作业19：

1. 设法让箭跑得稍微慢一些，使它后面发出一团团小烟雾。

2. 编写一个程序，能把你的名字在屏幕上从左向右滑行。

3. 编写一个程序，能把你朋友的名字从上向下滑行。

提示：要用VTAB语句，在打印“新”名字之前不要忘记抹去“旧”名字。

4. 再编一个组合程序，使你的名字从左向右移动，而你朋友的名字从上向下移动，并在屏幕中央相遇。

第二十课 变 量 名

教学参考:

本课将介绍变量命名的规则。

取一个好的变量名有助于说明程序的功能。在取变量名时，有两个隐蔽的错误要特别留神。

一是 BASIC 语言只识别一个名字的前面两个字符。这意味着一些长而漂亮的名字不是唯一的。这一点连有经验的程序设计员有时也会疏忽。

另一个是“保留字”不能在名字的任何地方出现，不管是开头，中间还是结尾。因为保留字是 BASIC 语言专用的。

在程序列表过程中，当发现有包含保留字的变量名时，屏幕上会显示一个记号。变量名中的保留字会自动分开。第一次碰到这种情况时，你可能会怀疑是自己打错了键，但是这个错误在输入得到纠正后仍然会出现。唯一的解决办法是重新选用一个新的名字。

在附录 3 上有 APPLESOFT BASIC 的保留字表。

复习题:

1. 为什么给变量取多于两个字符的名字会带来麻烦?
2. 带有数字的名字可能是个好的名字，也可能是一个不好的名字。怎样来辨别?

3. 一个名字里不能用标点符号, 但有一种情况可以例外。你知道允许用什么符号? 放在哪里? 它告诉你什么?

4. 较长的名字可能在它的前面, 中部或后面含有“保留字”。怎样辨别哪些字是保留的?

课 文

关于变量名的规定

1. 字符串变量名要以美元符号结尾。

2. 数字变量名则不需要以美元符号结尾。

3. 变量名是用字母和数字组成的。

4. 变量名的第一个字符必须是字母。

5. 变量名中不能有标点符号或其他符号, 除了字符串名称必须在结尾有一个“\$”符号外。

6. 变量名所用的字母或数字的多少可自由选择。但是, 只有最前面的二个符号起作用, 其余的电脑都不能辨认 (除了字符串名称结尾的“\$”符号以外)。

7. BASIC 语言中的保留字不能作为变量名。因为这些词是用于语句中的。象: REM、LIST、FOR、TO、LET、IF、PRINT 等等。附录 3 中有一张保留字表。

下面是一些正确与错误的变量名的例子:

正确的变量名:

A77

LEFTSIDE

X3AB

APE

NAME\$

D5\$

错误的变量名:

2X

X*

S [\$

\$NAME

3RDNAME\$

LIST

COLOR (其中有OR!)

STATE\$ (其中有AT!)

TOM (其中有TO!)

试一下,将错误变量名STATE\$及TOM放入象10 TOM=
1的行内,然后键入LIST命令,看看电脑有什么反应。

这个名字中有一个保留字,它会自动分开。

意思相同的不同变量名

“只有开头两个字符起作用”这条规定,如稍有疏忽,就会使你的程序运行起来乱了套。因为电脑无法将下面一对对名字分开。它往往会认为:

HOSE与HOP一样

XI与XIQ一样

NAME 1\$与NAME 2\$一样

如果做这样的程序:

10 HOP\$="BUNNY"

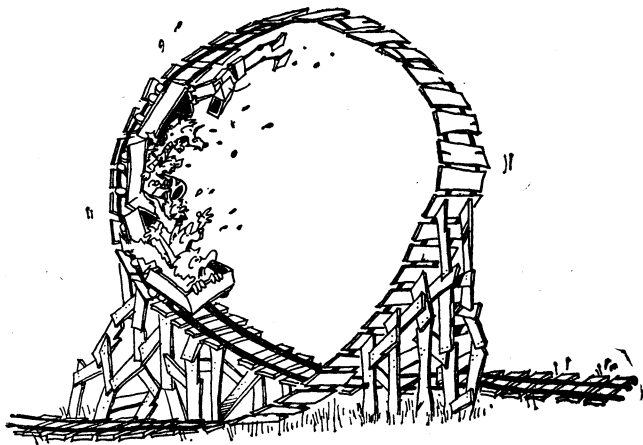
20 HOSE\$="LONG AND NARROW"

30 PRINT HOP\$

你就会亲自体验到 HOSE\$ 与 HOP\$ 是具有相同名字的变量。电脑认为它们都只有“HO\$”写在各自盒子的前面。

作业20:

再读一下本课开始介绍的 1 到 8 的变量名的规定。为每个规定写两个例子：一个是正确的，另一个不符合要求(第6个规定无错误名称)。然后在电脑上试一下,看看有什么反应。



第二十一课 LO-RES (低分辨率) 图形

教学参考:

本课介绍的语句有: GR(低分辨率绘图)语句、TEXT (恢复显示文字)语句、COLOR(选色)语句、PLOT(定点)语句。

下一课介绍的语句有: VLIN(垂直线)语句和HLIN (水平线)语句。

如果你用的是一台黑白显示器,LO-RES (低分辨率) 图形也是很有用的。实际上,要保持图案轮廓的鲜明,不一定非得用彩色显示不可,黑白显示也行,只要不是全黑就行了。

当然,在绘图时如果用了彩色显示总比用黑白显示更能吸引人。在进行大面积绘图时,低分辨率彩色画是显得挺美观的。不过色彩假使选择不当,线条就反而不很清晰了。

如果使用的是一台彩色显示器,学生就可以选择自己喜欢的色彩。在系统盘的盘片上,COLOR DEMOSOFT 程序对学生们的作图很有帮助。可以让学生把这个程序复制到自己的磁盘上去。

LO-RES 意思是“低分辨率”的,表示作图用的点不很小。它们呈长方形。每个长方形与文本显示方式中的字母一样宽,但只有字母一半那样高。使用时,整个屏幕水平方向可以放 40 个方块,垂直方向也可放 40 个。下面还留有的四行可显示一些文

字。这里说的“方块”，实际上指小长方形。

用一个点一个点来画图是非常枯燥乏味的。下一课的VLINE语句和HLIN语句可以大大减轻这方面的工作。这里建议用一个变量来标出图画的一个角或中心参考点，因此可以定出从这点到其他点、线的偏移量。这样在需要的时候就很容易移动整个图象，只要改变变量的值，它就能象动画片那样活动了。

复习题：

1. 向电脑输入GR语句后会出现什么情况？
2. 如果在编写完一个使用GR和PLOT语句的程序后，运行时并未见到什么图案，那么在GR语句后遗忘了什么东西？
3. TEXT语句做些什么？
4. 绘图时有多少种颜色可供选择？
5. 对于语句PLOT X、Y中的X、Y允许取什么范围内的数？它们同语句HTAB与VTAB允许选择的范围有什么不同？

课 文

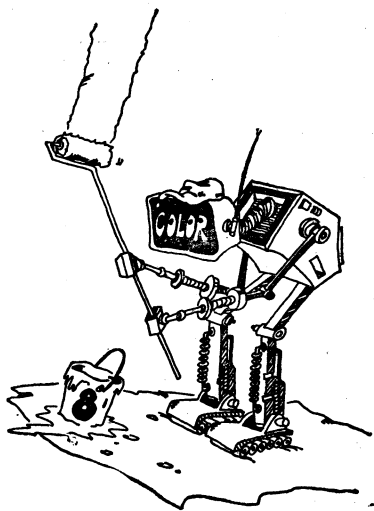
使用彩色小方块来绘图叫作“LO-RES”（低分辨率）绘图。

显示器的色彩调节

如果你用的显示器是黑白的，就跳过这节内容。先学后面的几节内容。

用下面的方法把磁盘中叫作COLOR DEMOSOFT的程序装入电脑，并执行这个程序：

```
LOAD COLOR DEMOSOFT
RUN
```



然后在下面的颜色单上选择表示颜色的号码。要慢慢地调节彩色显示器的颜色和色度旋钮，使显示的色彩与颜色名称相符合。

颜色单

- | | | | |
|---|------|----|------|
| 0 | 黑色 | 8 | 棕色 |
| 1 | 深红色 | 9 | 桔黄色 |
| 2 | 深蓝色 | 10 | 灰色 2 |
| 3 | 紫红色 | 11 | 粉红色 |
| 4 | 深绿色 | 12 | 淡绿色 |
| 5 | 灰色 1 | 13 | 黄色 |
| 6 | 中蓝色 | 14 | 水色 |
| 7 | 淡蓝色 | 15 | 白色 |

绘图语句

执行下面叫做“微笑”的程序：

```
10 REM ( ( ( ( SMILE ) ) ) )
20 GR
30 INPUT "WHAT COLOR? <1-15>"; C
35 COLOR=C
40 PLOT 20, 20
45 PLOT 21, 21
50 PLOT 22, 22
55 PLOT 23, 22
60 PLOT 24, 22
65 PLOT 25, 21
70 PLOT 26, 20
85 PRINT CHR$(7); REM PEEP
88 HOME
90 INPUT "AGAIN? <Y/N>"; A$
95 IF A$="Y" THEN GOTO 30
99 TEXT; HOME
```

SAVE SMILE

请用各种不同的数字(颜色)来回答程序中提出的问题“WHAT COLOR(用什么颜色)?”最后再用 0 试试。显然,由于屏幕本身是黑色的,电脑用“黑色”来画微笑的形象实际上是什么也看不到的。

在这个程序中使用了下列语句:

GR(低分辨率绘图)语句 COLOR(颜色)语句

TEXT(恢复显示文字)语句 PLOT(定点)语句

第20行使用了GR语句,它的意思是“绘图”。

规则: 在开始绘图前,要用GR语句,这样电脑就知道应该做绘图工作了。

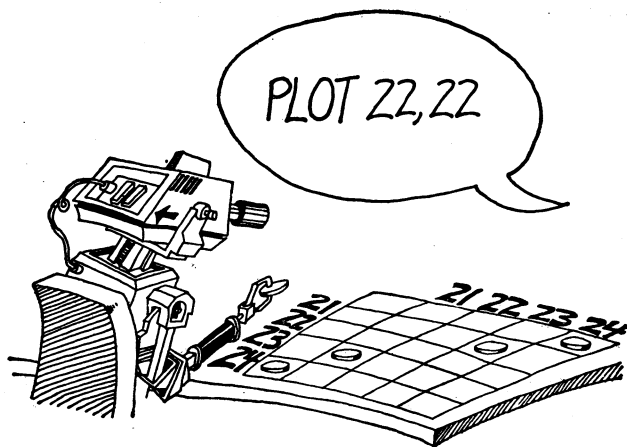
第35行使用了COLOR语句,它是用来决定在绘图时用什麼具体颜色的。

规则: 在使用GR语句后,必须要有COLOR语句跟着,并用数字1到15来决定它的颜色。因为GR语句在执行后,颜色总是黑色的,编号为0。如果忘了改成另一种颜色,在黑色屏幕上就会看不见图画。

第99行使用了TEXT语句。

规则: TEXT语句的意思是结束绘图,这样电脑可以再显示文字。

在程序中多次使用了PLOT语句。



规则：PLOT X, Y 语句意思是在横座标 X 和纵座标 Y 的地方放一个点。X 是一个数字，在 0—39 范围内变化；Y 也一样。

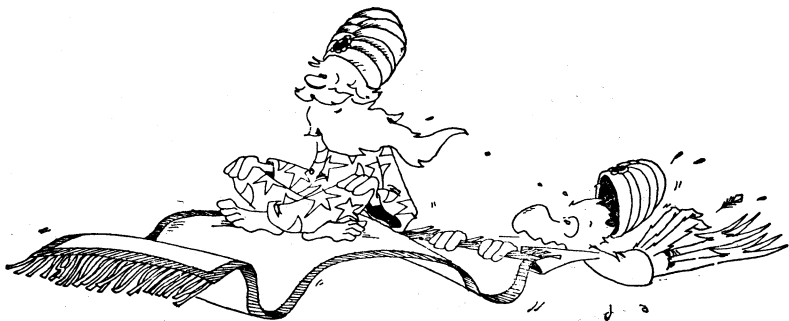
VTAB (纵向定位) 和 HTAB (横向定位) 两个语句在屏幕上的变化范围分别是 1 到 24 和 1 到 40。如果 PLOT 语句中横坐标 X 和纵坐标 Y 变化的范围也是从 1 到 40，而不是从 0 到 39 那就好了。可惜不是这样！一定要记住这一不同点。

作业 21:

1. 在课文中“微笑”程序所显示的图形里，设法在脸上画出眼睛和鼻子。

2. 在十三课的“猜谜游戏”中，加进一个号码，使得彩色星星在答案猜对时能显示出颜色；再使用计时循环，使星星会显示几秒钟，然后重新开始做游戏。

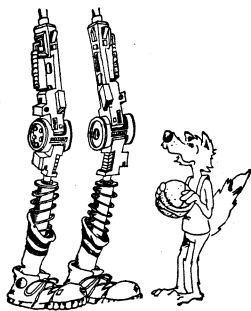
3. 编写一段能画“星巴德魔毯”（《一千零一夜》中的故事）的程序；可以由使用者自己来选择地毯有多少种颜色。如果有困



难,可以输入下面两句语句,看看COLOR DEMOSOFT 程序就行了。

```
LOAD COLOR DEMOSOFT
```

```
LIST 700-799
```



第二十二课 用HLIN(水平线)语句和VLIN (垂直线)语句绘图

教学参考:

这里介绍可以画线的HLIN(水平线)语句和VLIN(垂直线)语句。

低分辨率绘画要用到水平线和垂直线，一块块涂满颜色的画面也可以用填满线条来实现。

复习题:

1. HLIN A, B AT C 语句是要画什么样的线? 语句



中的字母 A、B 及 C 分别告诉电脑要做什么？

2. 对语句 `VLIN A, B AT C`, 回答上面提出的同样问题。

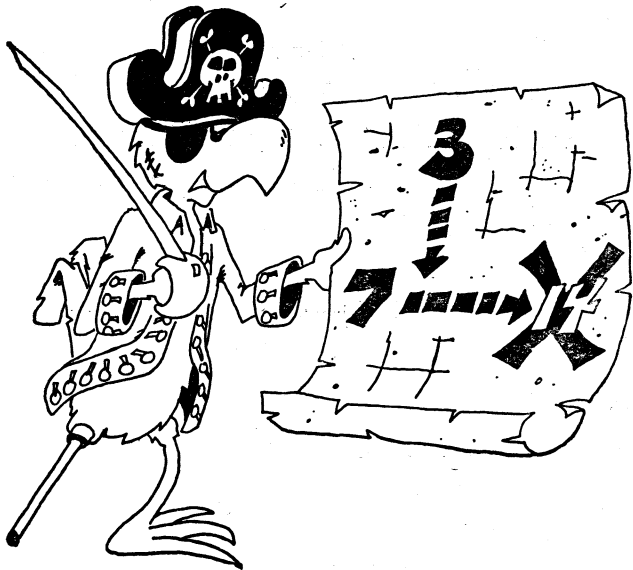
3. 如果将一条线画在另一条线的顶部或穿过另一条线, 那么会发生什么情况？

4. 如何画一个四边是蓝色的、里面是红色的正方形？

课 文

现在我们要用线条, 而不是用点来画图了。

首先要从磁盘中取出 `COLOR DEMOSOFT` 程序, 并执行这一个程序。然后慢慢地调节显示器的色彩旋钮, 使色彩与你选



择的号码颜色相符。

HLIN (水平线)语句

如果要画一条绿色的水平线,就应该给HLIN语句三个数,每个数一定要在0~39范围内。

执行这个程序:

```
10 GR: COLOR=12: REM COLOR IS GREEN
30 HLIN 7, 14 AT 3
```

电脑在屏幕顶部的第三行(这就是AT3的意思)上画出了一条绿线;绿线从屏幕左边起第7格开始,到第14格结束。

换句话说:30 HLIN 7, 14 AT 3意思是:从顶部数下来第3行的第7格开始,到第14格为止,画出一条水平线。

VLIN(垂直线)语句

如果把上面的第30行改写成:

```
30 VLIN 8, 15 AT 4
```

那么水平线就换成了垂直线。这一语句的意思是:

在从左边数起的第4列上,从第8行开始,一直到第15行为止,画出一条垂直线。

试做下列程序:

```
10 GR: REM.....MAKE LINES.....
12 HOME
15 INPUT "HORIZONTAL OR VERTICAL?
    <H/V>"; D$
20 INPUT "COLOR? <1-15>"; C
25 COLOR=C
40 IF D$="V" THEN GOTO 70
```

```

42 PRINT "HORIZONTAL LINE"
44 INPUT "HOW FAR FROM THE TOP?
    <0-39>"; Y
46 INPUT "START WHERE? <0-39>"; X1
48 INPUT "END WHERE? <0-39>"; X2
50 HLIN X1, X2 AT Y
55 GOTO 12
70 PRINT "VERTICAL LINE"
74 INPUT "HOW FAR FROM THE LEFT?
    <0-39>"; X
76 INPUT "START WHERE? <0-39>"; Y1
78 INPUT "END WHERE? <0-39>"; Y2
80 VLIN Y1, Y2 AT X
85 GOTO 12

```

用文件名 MAKE LINES (画线) 把这个程序存入磁盘中去。

请再做另一个名叫“捡起拐杖”的程序。

```

100 REM: : : PICK UP STICKS: : :
105 HOME
110 GR
120 FOR I=1 TO 50
125 FOR T=1 TO 300: NEXT T
130 REM VERTICAL LINES
135 COLOR=RND (8) * 16
140 X=RND (8) * 39

```

```

142 Y1=RND (8) * 39
144 Y2=RND (8) * 39
146 IF Y1>Y2 THEN YT=Y1: Y1 =Y2: Y2
    =YT
148 VLIN Y1, Y2 AT X
155 COLOR=RND (8) * 16
157 FOR T=1 TO 300: NEXT T
160 REM HORIZONTAL LINES
162 Y=RND (8) * 39
164 X1=RND (8) * 39
166 X2=RND (8) * 39
168 IF X1>X2 THEN YT=X1: X1=X2: X2
    =YT
170 HLIN X1, X2 AT Y
190 NEXT I
192 PRINT "DONE"
195 FOR T=1 TO 5000: NEXT T
199 TEXT: HOME

```

在上面的程序中请找出下列东西：

1. 三个计时循环在什么地方？每一个计时循环做了些什么？
2. 一个主循环在哪里？
3. 指出主循环中两种主要的活动是什么？
4. 程序执行完后，将屏幕显示回到正常情况的是哪一行语句？

5. 告诉电脑准备画图的是哪一行语句?

6. 绘图中一共用了多少种颜色?有拐杖是黑色的这一内容吗?哪一行是选择颜色的?

将这个程序运行后,以文件名“PICK UP STICKS”(捡起拐杖)存放到磁盘中去。

移动图形

要使图形移动就必须:

1. 画图;
2. 将图抹去;
3. 移动一些位置后再画图;
4. 再把图抹去;
5. 象这样循环重复多次。

做这件事的最好办法是用一个子程序来画图。第二十四课将会教你如何做。

作业22:

1. 编写一个画正方形的程序(正方形用什么颜色由你自己选择);并将它存放到磁盘中去。
2. 在上面的程序中再加些语句行,使正方形象骰子一样,面上画有1到6六个点。
3. 编写一个能画出你姓名第一个字母图形的程序。开始用红色(颜色1),并且发出“嘟”的声音,然后改成深蓝色(颜色2),一直变到颜色15为止。

第二十三课 密写与GET(取数)语句

教学参考:

本课要介绍GET(取数)语句。

GET语句能使电脑从键盘上得到单一的字符。语句执行时,电脑会一直等到按下键值为止;在等待的过程中,屏幕上一点东西也不显示,也没有提示符和光标显示,即使按下键后,屏幕也无显示。不过,这时实际上电脑已接收到按下的字符。

GET语句的用途就基于这一特点。例如:电脑要求输入一个关键词,你又不想让旁观者看到,这时可以用一系列GET语句让电脑来接收。

与INPUT语句相比,GET语句的另一个优点是不要按回车键。这就使得GET语句在“方便用户”、编制程序方面很有用。

在苹果电脑中,最好将字符放在字符串变量中,而不要放在数字变量中。如果要用GET取一个数字,可以把数字作为字符串取进来,然后用后面课程要讨论的VAL()函数,把它们转换为数字。

复习题:

比较INPUT语句和GET语句:一个一次只能取一个字母,另一个一次能取一个词甚至一句话;一个有光标,另一个则没有;一个能在屏幕上打印出输入的内容,另一个则不打印;一个

需要按回车键，另一个则不需要。试回答哪一种情况属于哪一种语句。

课 文

INPUT(键盘输入)语句

使用INPUT语句有以下两种方法：

1. 不带短句的：

7. 10 INPUT A\$
 10 INPUT N
 10 INPUT NAME\$, AGE, DAY, MONTH\$, YEAR



TOP SECRET——绝密
KEEP OUT——切勿入内
KEY BOARD——键盘

2. 带有短句的:

```
10 INPUT "NAME, AGE"; NAME$, AGE
```

```
10 INPUT "HOW ARE YOU?"; FEEL$
```

单字、句子或数都可以用其中的一种方法输入。

GET(取数)语句与密写

GET语句与INPUT语句不同,它仅从键盘上获得一个字符。在程序中出现GET语句时,电脑一直会等到有键按下为止。在整个过程中屏幕上无任何显示:短句、问号、光标以及你按下的键的任何内容都不显示。

但是,电脑一直在等你按下一个键,以后不需要再按回车键,电脑就会立即继续执行程序。

在猜谜游戏中,GET语句有它特殊的妙用。因为你输入要猜的字母或数字,伙伴们是无法看到的,只有你自己知道。下面请试做两个例子:

```
10 REM.....GET.....
```

```
20 HOME
```

```
30 PRINT "PRESS ANY KEY"
```

```
40 GET K$
```

```
45 PRINT CHR$(7)
```

```
47 FOR T=1 TO 1000: NEXT T
```

```
50 PRINT "THE KEY YOU PRESSED  
WAS"; K$
```

```
10 REM ### BACKWARDS ###
```

```
20 HOME
```

```

30 PRINT "TYPE IN A 5 LETTER WORD"
35 PRINT
40 GET A$; GET B$; GET C$; GET D$; GET
   E$
50 PRINT "NOW HERE IT IS BACKWARDS"
55 PRINT; INVERSE
60 PRINT E$; D$; C$; B$; A$
70 NORMAL

```

用字母组成词

GET语句一次只能取一个字母,要组成词,需要用字符串连接才行。下面是一个例子:

```

10 REM GET A WORD
20 HOME
30 PRINT "TYPE A WORD, END IT WITH
   A PERIOD."
35 W$=" "; REM WORD STRING IS
   EMPTY
40 GET L$; REM GET A LETTER
50 IF L$="." THEN GOTO 80; REM TO
   TEST FOR END
60 W$=W$+L$; REM ADD LETTER TO
   END OF WORD
65 GOTO 40; REM TO GET ANOTHER
   LETTER
80 REM WORD IS FINISHED

```

85 PRINT W\$

那么电脑怎么会知道一个字已经打完了呢？它是通过检查字后面是否有句号来实现的。第50行就是用来检查句号的，一旦发现是句号，就立即结束打印那个词。

用GET语句来取数

GET语句可以用来输入数字，但有时也会出现一些麻烦。这一点将在字符串和数的转换中再进行解释。

作业23:

1. 编写一个程序，使它有一个“颜色单”供使用者选择。使用者通过打入一个字母来选择其中的一项，字母的获得可用GET语句。例如：

```
PRINT "WHICH COLOR? R=RED, B=BLUE,  
G=GREEN"
```



2. 编写一个做造句游戏的程序。句子中包含一个名词作主语，一个动词谓语以及一个宾语。第一位游戏者送入一个名词(例如“驴子”)，第二位送入一个动词(例如“唱歌”)，第三位送入另一个名词(例如：“牙签”)，看看句子是否能成立；使用 GET 语句可以使游戏者无法看到其他人用的单词，而且只要在名词前加一个形容词就可以扩大游戏的范围。

第二十四课 GOSUB(转子程序)语句、 RETURN(返回)语句和END(结束)语句

教学参考:

本课主要讲子程序,还要介绍END(结束)语句。因为子程序一般放在程序里的较高行号处,所以必须在中间的行号处用结束语句。

子程序不但在较长的程序中有用,而且在较短的程序中,为了使结构清晰,子程序也很有用。

在有些学生中很难形成一种习惯:即将结构加到程序上。结构在实施中可以有許多名称,如“结构程序编制”和“由顶向下程序编制”等,这些技术可以用来训练程序设计员。

要提醒学生们注意:可以用结构的方法使所设计的程序具有语句流畅、思路清晰的优点。本课主要介绍如何编写较好的REM语句以及在编写程序中怎样使用模块化结构的技术。

在BASIC语言中,用GOSUB(转子程序)语句来建立程序模块。本课将通过一个程序的例子来说明什么是模块结构。

复习题:

1. 执行END语句时会发生什么情况?
2. GOSUB语句与GOTO语句有什么不同?
3. RETURN语句执行时会发生什么情况?

4. 如果RETURN语句在GOSUB语句之前执行, 又会发生什么情况呢?

5. “调用子程序”是什么意思?

6. 一个程序中允许放多少条END语句?

7. 程序中为什么要用子程序?

课 文

执行下列程序, 然后把它存放到磁盘中。

```
100 REM MAIN PROGRAM
101 :
105 HOME
110 PRINT "READY TO GO TO THE
SUBROUTINE"
120 GOSUB 200
130 PRINT "BACK FROM THE SUBROUTINE"
133 PRINT
135 PRINT "GO TO THE SUBROUTINE
AGAIN"
140 GOSUB 200
150 PRINT "BACK AGAIN"
190 END
199 :
200 REM SUBROUTINE
201 :
210 PRINT "IN THE SUBROUTINE"
```



```
215 FOR T=1 TO 2000: NEXT T
217 PRINT CHR$(7)
290 RETURN
```

这是一个长程序的轮廓,主程序从第100行开始,到第190行结束。

在有 PRINT 语句的地方,需要时还能放一些其他的语句行进去。

在第190行中的 END 语句告诉电脑:程序已结束。于是电脑就自动回到编辑方式去了。

第120行和第140行是“调用子程序”语句。这意味着电脑要去执行子程序中的语句,执行完后再回到主程序来。

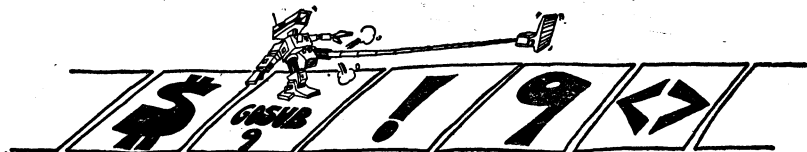
GOSUB 200 语句要电脑记住它在程序中的位置,使得电脑执行完子程序后能够回到这个地方再继续执行下去。除此之外,它的作用就和一条 GOTO 200 语句差不多了。

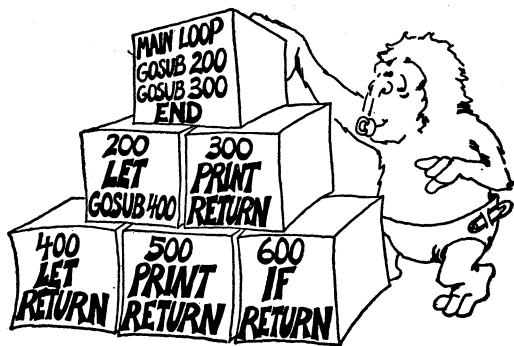
RETURN 语句告诉电脑:要返回到 GOSUB 语句之后的一行语句去执行。

用子程序有什么好处

在短程序中,子程序优点不很明显,而在长程序中两个好处很突出:

1. 可以大大减少你的工作量,节省许多存贮器的空间。你





不需要在程序的不同地方重复地写相同的程序行。

2. 使程序更容易理解,也能较快地编出程序和调试程序。

END(结束)语句

程序中可以有END语句,也可以有一个或许多个END语句。

规则: END语句告诉电脑要停止程序的执行并回到编辑方式去。这就是END语句的功能。因此可以把END语句放到程序的任何地方,例如:可以放在 IF语句中的THEN后面。

活动图象

```

10 REM??? JUMPING J???
20 HOME; GR
22 X=15; Y=15; D=1
25 FOR J=1 TO 10
26 FOR I=1 TO 10
30 COLOR=8; GOSUB 100; REM DRAW
35 COLOR=0; GOSUB 100; REM ERASE

```

```

45 Y=Y-D
50 NEXT I
55 D=-D
60 NEXT J
90 TEXT: HOME: END
100 REM
101 REM DRAW THE J
102 REM
110 HLIN X, X+5 AT Y
120 VLIN Y+1, Y+7 AT X+3
130 HLIN X, X+2 AT Y+7
140 PLOT X, Y+6
190 RETURN

```

这个程序是画字母“J”。从第100行开始的子程序能画出“J”来。在GOSUB语句之前可以用COLOR语句挑选“J”所要的颜色。瞧第30行和第35行，如果挑选的颜色号码是“0”，那么子程序就会逐渐地将“J”抹去。

子程序画出的“J”，是从屏幕左上角的X、Y那点开始的，只要改变X和Y的值，“J”就会被画在不同的地方。第22行表明：第一个“J”将画在靠近屏幕的中部。

字母“D”用来说明将“J”从一处移到另一处要移多少。第22行取D等于1，但是第55行在图画了10幅后将D变成了-1。

第45行表明每张图要画的位置是由比上一次大一个偏移量D的Y值来决定的。

作业24A:

输入并执行课文中会跳跃的“J”程序，然后作出下列改变：

1. 改变子程序的内容，使它能打印出你自己姓名的第一个字母。
2. 把那个字母的颜色变成蓝色。
3. 把“J”变成“横向移动”而不是“垂直移动”。
4. 把起始点改成在屏幕的右下角而不是在屏幕的中部。
5. 把移动图象的距离改成20步而不是10步。
6. 把每一步的距离从1变成2。
7. 改变移动方向，使它斜向移动（用 $X=X+D$ ； $Y=Y-D$ ）。
8. 改变程序，使“J”在移动时颜色由红色(颜色1)依次换成各种颜色，最后变成白色(颜色15)。
9. 把第35行中的 $COLOR=0$ 变成 $COLOR=1$ 后会发生什么情况？

如何编写一个长程序

让我们来编写一个“刽子手”游戏程序。这是一个猜字游戏。在游戏中，每当你猜错一个字母时，程序就会画上“绞死者”的一部分。

首先写整个程序的轮廓，可以在纸上或者直接在屏幕上这样做。如果碰到困难不知道怎么办，那么可以看下面向你提供的参考程序。

先写出整个程序的轮廓：

```
10 REM *** HANGMAN GAME ***
200 REM INSTRUCTIONS
300 REM GET THE WORD TO GUESS
```

```

400 REM MAKE A GUESS
500 REM TEST IF RIGHT
600 REM ADD TO THE DRAWING
700 REM TEST IF GAME IS OVER
800 REM END GAME MESSAGE

```

再填进较详细的内容：

```

10 REM * * * HANGMAN GAME * * *
99 :
100 REM MAIN LOOP
101 :
120 INPUT "NEED INSTRUCTIONS?
    <Y/N>"; Y$
122 IF Y$="Y" THEN GOSUB 200
130 GOSUB 300; REM GET WORD
132 STOP
135 GOSUB 400; REM MAKE GUESS
140 GOSUB 500; REM TEST GUESS
145 GOSUB 700; REM TEST IF GAME IS
    OVER
190 GOTO 135; REM MAKE ANOTHER
    GUESS
199 :
200 REM INSTRUCTIONS
... write the instructions last
290 RETURN

```

```

299 :
300 REM GET THE WORD TO GUESS
... use INPUT to get a word form player 1
... draw dashes for the letters to be guessed
390 RETURN
399 :
400 REM MAKE A GUESS
... player 2 guesses a letter
490 RETURN
499 :
500 REM TEST IF GUESS IS RIGHT
... if wrong, GOSUB 600; REM draw
    hangman part
... if right, GOSUB 700; REM see if game is
    over
590 RETURN
599 :
600 REM ADD TO THE DRAWING
... add to the hangman drawing
... test if drawing is done
... if so, then GOSUB 800
690 RETURN
699 :
700 REM TEST IF GAME IS OVER
... see if all letters have been guessed

```

```

... if yes, GOSUB 900
790 RETURN
799 :
800 REM END GAME MESSAGE
... message for when guesser loses
890 RETURN
899 :
900 REM END OF GAME MESSAGE
... message for when guesser wins
990 RETURN

```

如果对这个程序是如何结束还感到含糊不清的话，那就把它留在以后再说吧。请先编写和测试第一部分的程序，这里把STOP语句放在第132行，电脑就可以先执行第一个子程序了。请从第300行开始写一个子程序，要求用INPUT语句输入从第一个游戏者处所得到的一个字，然后对所猜的字画出相应的图形。

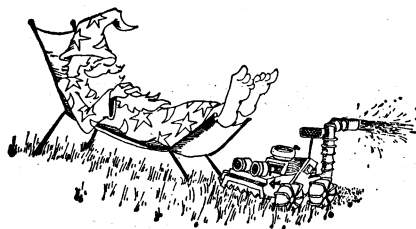
作业24B:

1. 编写一个可以使用子程序的短程序。它不必做任何有用的事，只要打印出一些趣味的内容就行了；但要求是主程序中要有三个子程序，其中主程序要二次调用一个子程序；一个子程序要调用另外一个子程序；用IF语句来调用一个子程序。

2. 编写一个能在屏幕上写上你姓名的三个缩写字母的的程序。每一个字母使用不同的颜色，然后使它们上下移动一次。

3. 继续编写“刽子手”游戏程序。这是一个较长的程序。现

在先做一部分工作(你能做多少就做多少),然后将它存放在磁盘中,以后再完成这个游戏。



第三部分 进一步编制程序的方法

第二十五课 行 编 辑

教学参考：

在程序设计的过程中，我们常常需要增删或修改语句行，这就叫“行编辑”。本课要介绍用 J ^I K 四个键来移动光标的方法：先按一下 ESC 键，然后用 I、M、J、K 键来上、下、左、右移动光标。

一旦要修改程序中的某一行时，用 J 键把光标移到该行号的第一位数字上，千万不要使用箭头键，这一点很重要。当光标放在第一个数字上之后，才可用箭头键在行内来回移动光标，这时不要使用 I，J，K 及 M 键了。

在 APPLESOFT BASIC 编程参考手册的附录中，比较全面地介绍了行编辑，包括有关语句行的输入与删除等情况。这部分内容希望在掌握了本课介绍的行编辑之后，再去作进一步的学习。

复习题：

1. 如果要移动光标，在按 I、J、K 或 M 键时，是否要把 ESC 键同时按下？

2. 如果要编辑一行语句, 并且把它放回到存贮器中去, 能不能用 J 和 K 键来代替箭头键? 为什么?

3. 如果你用箭头键把闪烁的光标放在一行语句的中间位置, 然后按回车键, 这时会发生什么情况?

课 文

在这一课中, 我们要介绍使用键盘上的“ESC”键。用它可以让闪烁的光标从一个语句行中跑出来, 移到屏幕的任何地方去。

移动光标的办法

如果要把光标移到屏幕的其他地方去, 可以按下列规则办。

规则: 先按一次ESC键, 然后按I、J、K、M四个字母键, 就可以移动光标。每个键按的次数不受限制。四个键的作用如下:

I ↑ 向上
向左← J K → 向右
M ↓ 向下

当然, 这些键也可以同时与REPT键一起配合使用。

注意: 这时不要使用箭头键来向左或向右移动光标。除I、J、K、M这四个字母键外, 也不要使用任何其他键。如果需要使用其他键, 必须再按一次ESC键。

下面请练习一下。

用上面介绍的方法试一试把光标移到屏幕的顶部, 然后把它向左移动几步再向右移动; 当光标向右跑到了屏幕的边缘时, 如果再向右跑一步, 光标就会出现在下一行的最左端; 接着请再把光标移回屏幕的底部。掌握了从上、下、左、右四个方向上移



ASTRO·ZAP——星球大战

动光标后，你能进一步按圆周移动光标吗？请试试。

修改语句行

如果要修改刚输入的一行语句，可以按以前学过的方法做：

1. 先用箭头键把光标移到有错误的地方；
2. 重新输入一个正确的字母；
3. 再把箭头键移到这一行的末尾；
4. 按下回车键，把这一行内容放入存贮器内。

行编辑——改变存贮器内的语句行

有时候语句行已经放入存贮器之后才发觉有错误，或许有时候你突然改变主意了，想改变前面输入的某一行，那怎么办呢？是否还要重新打一次呢？不必了！可用下列方法修改：

1. 用LIST语句把需要的行先在屏幕上显示出来。
2. 再用ESC键和I、J、K、M键，把光标移到要修改的那一行的开头。

3. 同前面一样,用箭头键, REPT键和回车键来进行修改。

注意: 在将光标移到要修改的那一行左边的过程中,只能用ESC键和I、J、K及M键。一旦光标移到该行第一位数字上之后,就不要再使用这四个键了。这时只要用箭头键移动光标到要改错的地方,再输入正确的字符就行了。在改错后,要用右箭头键把光标移到这行末尾,并按下回车键把这一行放到存贮器中。

局限性: 在原来的语句行中不能插入额外的字母,但可以在原来字母位置上用一空格来代替。

下面请你做些行编辑的练习:

输入:

```
623 PRINT "I LIKE IPPLES"  
HOME
```



DON'T——不

I LIKE SPINACH——我喜欢未加修剪的胡子

现在输入：

LIST 623

这时从屏幕显示上发现，该语句有一个字母打错了，句中“IPPLES”应为“APPLES”，需要对它进行修改。先按一下ESC键，再按I键，使光标移到这一行语句的“2”上；然后按J键，把光标移到数字“6”上。

接着再用右箭头键和REPT键，使光标移到“IPPLES”单词的第一个字母上，按字母键A，把这单词改为“APPLES”。

连续按右箭头键直到光标移到这行末尾为止。别忘了，再按一下RETURN(回车)键。

最后用LIST命令把这行内容显示一下，看看是否修改正确了。

作业25:

1. 从磁盘中取出一个旧程序，显示其中的一行语句，然后做修改练习。用ESC键和I、J、K、M键把光标移到这行的左端，接着改为用箭头键不断移动光标，修改里面的内容。最后用回车键把修改好的内容放入存贮器中。

2. 编写一个APPLE TREE(苹果树)程序。先画蓝色为背景，再画灰色的地面，棕色的树干和绿色的树叶，树上结着果实。然后让苹果从树上落下来，掉在地上。用子程序来画苹果树的一部分以及苹果下落部分。用行编辑来修改编得不够简洁的语句。

第二十六课 字符串函数

教学参考:

本课中要介绍的函数有:

LEFT\$函数、MID\$函数、RIGHT\$函数、LEN函数

这些函数都有截取字符串的功能。课文对 MID\$ 函数带有三个自变量作了说明,但对第三个自变量的省略情况没有作介绍。运用这些函数以及前面介绍过的连接运算符“+”,就可以随意截取字符串,并能将这些字符串按任何次序再联接起来。

和前面已经解释过的情况一样,函数的功能只介绍它的主要特性,一些特殊情况这里不作介绍,尤其是那些会产生错误信息的情况,否则会把课文学习的内容搞乱。如果学生在学习这部分内容时有困难,那么老师或家长可以参考一下苹果电脑使用手册来帮助学生解决这些问题。

复习题:

1. 如果要从“STARS AND STRIPES”字符串中截取“STAR”,应该用什么函数?取什么自变量?
2. 如果要截取“AND”,应使用什么函数和自变量?
3. 如果要计算字符串 PQ\$中字符的个数,用什么函数,什么自变量?
4. 下列这些语句行错在哪里?

```
10 A$=LEFT$ (A, D$)
```

```
10 RIGHT$ (R$, I)
```

```
10 F$=MID$(A, 3)
```

```
10 J$=LEFT (R$, YT)
```

5. RIGHT\$()函数需要哪两个自变量?

6. 用什么语句能从某一单词中截取第3和第4个字母?

7. 编写一个小程序,先输入单词“computer”,然后使它变成“putercom”。

课 文

字符串的联接

你已经知道如何把几个字符串联接在一起的方法了。如:

```
55 A$="CON"+"CAT"+"EN"+"ATION"
```

```
60 PRINT A$
```

“联接”的意思就相当于手拉手地连在一起。这样的字符串或许可以称为“链”。

截取字符串

让我们从字符串上截下一段来。看看下面的程序:

```
10 REM >>>> SCISSORS >>>>
```

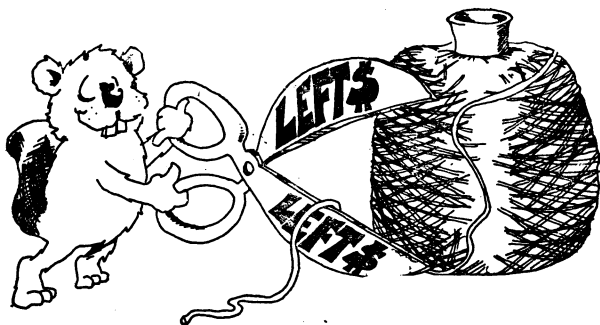
```
20 HOME
```

```
30 N$="123456789"
```

```
35 Q$=LEFT$(N$, 4)
```

```
40 PRINT Q$, N$
```

第35行中的LEFT\$函数是用来截取一个字符串左边部分的,截下的部分字符可放在存贮器内或者被打印出来等等。



规则 LEFT\$()函数在括号内要有两样东西：

1. 你要截取的字符串盒子名；
2. 需要截取下来的字符的数目。

如果把第40行改成：

```
40 PRINT RIGHT$ (N$, 3)
```

程序执行时，电脑就会打印出：

```
789
```

除了从字符串右边一头开始截取字符外，其他情况RIGHT\$()函数和 LEFT\$()函数都一样。

截取与联接

执行这一个程序：

```
10 REM : : : SCISSORS AND GLUE : : :
20 HOME
30 N$="123456789ABCDEF"
35 FOR I=1 TO 15
40 L$=LEFT$(N$, I): R$=RIGHT$(N$, I)
50 PRINT I; TAB(5); L$; TAB(25); R$
```



```
60 NEXT I
```

程序中截下的字符串可按不同次序重新联接在一起。请在程序中加进下面这一行并执行：

```
55 IF I=4 THEN PRINT: PRINT R$  
    +L$: PRINT
```

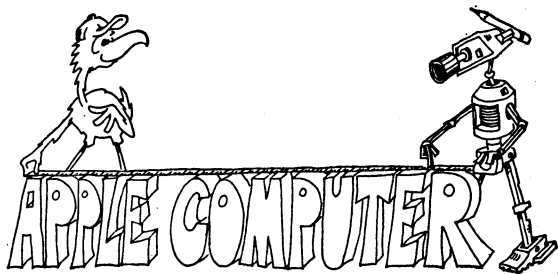
LEN函数

请执行下列程序，看看LEN函数有什么作用。

```
10 REM : : : LONG ROPE : : :  
20 HOME  
30 INPUT "GIVE ME A STRING:"; N$  
40 L=LEN(N$)  
50 PRINT "THE STRING:"; N$; " , "  
55 PRINT: PRINT "IS"; L; "CHARACTERS  
    LONG"
```

LEN()函数可以说出字符串内字符的个数；它把字符串内的一切字符，包括空格都计算在内。这就好象它用一根绳子在量字符串的长度一样。

从中间截下一段



MID函数用来截取字符串中间的某一段字符。执行这个程序：

```
10 REM ### MIDDLE ###
20 HOME
30 N$="123456789"
40 P$=MID$(N$, 3, 4)
50 PRINT P$
```

其中第40行 40P\$=MID\$(N\$, 3, 4)表示的意思是：

1. 从N\$盒子中取出字符串；
2. 在这个字符串中，自左向右从第三个字符数起，共数出四个字符——“3456”；
3. 把这四个字符保存在P\$盒子中。

下面的程序叫“无空格”程序，请试一下，看看它的显示过程。其中第60行语句每次从字符串中间取出一个字符。

```
10 REM >>> NO SPACES >>>
11 :
20 PRINT: PRINT
25 PRINT "GIVE ME A LONG SENTENCE":
   PRINT
30 INPUT S$
40 L=LEN (S$)
45 T$=" "
50 FOR I=1 TO L: REM LOOK AT EACH
   LETTER
60 L$=MID$ (S$, I, 1)
```

```

70 IF L$ < > " " THEN T$=T$+L$;
   REM SAVE ONLY LETTERS
90 NEXT I
92 PRINT; PRINT T$
95 PRINT; PRINT; PRINT

```

作业26:

1. 写一个密码程序: 你输入一个句子, 它会先计算这个句子的长度, 然后将每个单字的第一个字母和第二个字母对换, 第三个字母和第四个字母对换……等等。例如:

THIS IS AN APPLE. 变成 HTSI SI NA PALPE.

2. 写一个回答问题的程序: 你问它一个以动词开头句子的问题, 它会把动词和名词的位置颠倒过来回答你的问题。例如:
ARE YOU A TURKEY? 变成 YOU ARE A TURKEY.



3. 编写一个名叫“PIG LATIN”(颠倒字母顺序)的程序。它要求你输入一个单字,然后它会取出这个单字的第一个字母并把它移到这个字的尾部,再加上AY。假如取出的字母是元音字母(A、E、I、O、U)那么它就会在该单词的结尾按不同的元音字母分别加上LAY、LEE、LI、LO或LU。



第二十七课 字符串和数的变换

教学参考:

本课要学习另外两个字符串函数: STR\$ 和 VAL, 还要总的复习一下函数的概念。

STR\$ 函数的作用是把一个数转换成字符串。

VAL 函数的作用恰恰相反, 它把字符串转换成为一个数值。

两种变量可以相互转换, 这给涉及到数的程序增加了很大的灵活性。

复习题:

1. 如果一个数在本课作业第 2 题的程序中“跑”得太快, 你如何使它放慢速度?

2. 请编写一个含有字符串: “GEORGE WASHINGTON WAS BORN IN 1732.”(乔治·华盛顿生于1732年)的程序, 使它能回答这个问题: “HOW LONG AGO WAS WASHINGTON BORN?”(华盛顿生于哪一年?) 并从字符串中获得诞生日期, 还能将这个日期转换成一个数。

3. 什么是“数值”? “函数值”是什么意思? 用数值能做什么?

4. 函数的“自变量”是指什么? RIGHT\$ 函数有多少自变

量? CHR\$()函数又有多少个自变量?

5. 命令(象 RUN, LIST 等)通常是放在一行的什么位置上? 可以把函数放在一行的开头吗?

课 文

本课还要介绍两种字符串函数: VAL()函数和 STR\$()函数。

把字符串变成数

变量有两种: 字符串变量和数字变量。我们可以使这两种变量相互转换。

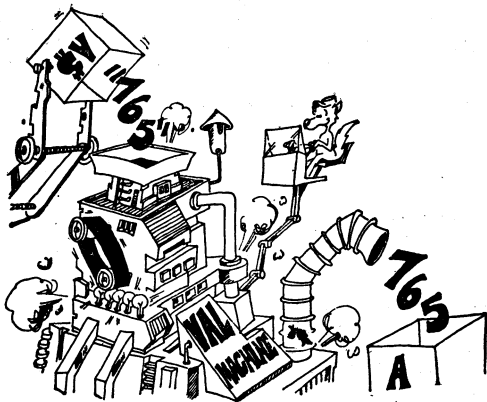
执行下面的程序, 你就会明白了。

```
10 REM MAKING STRINGS INTO NUMBERS
20 HOME
30 L$="123"
40 M$="789"
50 L=VAL(L$)
60 M=VAL(M$)
70 PRINT L
72 PRINT M
74 PRINT "... "
76 PRINT L+M
```

其中 VAL 表示的意思是“数值”, 它能把字符串转换成数值。

把数变换成字符串

请执行下面的程序:



```
10 REM MAKING NUMBERS INTO STRINGS
11 :
20 PRINT
25 INPUT "GIVE ME A NUMBER"; NB
30 N$=STR$ (NB)
35 L=LEN(N$)
37 PRINT
40 FOR I=L TO 1 STEP -1
45 B$=B$+MID$(N$, I, 1)
50 NEXT I
60 PRINT "HERE IT IS BACKWARDS"
65 PRINT, PRINT B$
```

STR\$ 表示的意思是“字符串”，它能将数转换成字符串。

函数

在这本书中使用的函数有：

RND(), INT(), LEFT\$(), RIGHT\$(),
MID\$(), LEN(), VAL(), STR\$(),
ASC(), CHR\$(), SCRN(), PDL()

有几个函数我们还未学过，后面几课再介绍。

使用函数有下面几条规则：

1. 函数总是用括号()来包含一个或多个自变量。例如：
MID\$(D\$, S, J)函数的括号内有三个自变量 D\$, S 及 J。
2. 自变量可能是数或字符串，也可能两者都是。
3. 函数不是“语句”，不能把它当作语句来使用。

例如：

正确 10 LET D=LEN (CS\$)

错误 10 LEN (CS\$)=5

4. 一个函数运用起来很象一个数或一个字符串。可以这样来理解，函数能产生一个数，这个数象其他任何数或字符串一样可以被放入盒内或被打印出来。一个函数甚至可以作另一个函数中的自变量。

5. 函数自变量能说明给出的是哪些数值。

6. 必须记住：字符串应放在表示字符串变量的盒子内；数值应放在表示数字变量的盒子内。

下面做些练习。

一、在下面列出的六个函数中：

1. 说出每个函数各有多少个自变量，名称是什么？
2. 说明各个函数值是字符串还是数。

RND (8)


```

INT (Q)
LEFT$(U$, Y)
MID$(RIGHT$, E, 2)
VAL(ER$)
STR$ (INT (RND(8)))

```

二、下面每一行都有错误，解释一下错在哪里。

```

10 INT(Q) = 65
10 D$ = LEFT(R$, 1)
10 PW$ = VAL(F$)
10 PRINT CHR$

```

作业27:

1. 编制一个程序：先取一个数，然后再取另一个数，它是第一个数的反向排列数（如先取12，再取的数是21），把此两数相加，并按加法列式打印三个数字，要求放上+号及一根横线，形如

$$\begin{array}{r}
 \times \times \times \times \\
 + \times \times \times \times \\
 \hline
 \times \times \times \times \quad .
 \end{array}$$

2. 编写一个“走数”程序：先把一个数写在屏幕上，然后把这个数的最左边的一个数码移至最右边，并不断重复这样做。移动时别忘了让程序把每次移动过的最左边的那个数抹去。

第二十八课 游 戏 棒

教学参考:

这篇课文较长, 主要介绍 PDL() 函数、SCRN() 函数和 PEEK 函数。使用 PEEK(-16287) 和 PEEK(-16286) 可以弄清是否已按过游戏棒的按钮。

游戏棒通常用在趣味横生的绘图游戏中。本课要说明, 画低分辨率图形可以用移动点来模仿射击; 要让学生理解 X、Y 是表示 40×40 的屏幕上的一点; 在画移动物体时, 要在下一个物体的新图案画出之前把旧图案抹去。

要查明射击是否“命中”目标, 只要用 SCRN 函数来检查一下子弹前面的正方形就行了。如果子弹前面有几种类型的目标, 最好检查一下正方形是否是“背景”色; 如果不是, 那么就转向到询问色彩的子程序去, 这样目标就会被打印出来。

绘图游戏可能比较长, 用 BASIC 语言来编排的这些游戏执行时的速度还可能比较慢。如果将程序的“工作”部分放在开头, 初始化部分放在结尾 (这种安排只要在程序的头几行用一个语句调用一下就行了), 那么就可以获得执行游戏程序的最大速度。

游戏在以最大速度执行时, 应避免重复将数字转换成移动的点。在课文中“PADDIES” (游戏棒) 程序中的第 60 行最好改

写成GO PEEK(QO),句中的变量QO=-16287是由程序的初始化部分来确定的。这是获得快速执行游戏程序的最重要的一步。

复习题:

1. PDL()函数能给出什么范围的数?
2. 如何把这个范围的数改变为0~20。
3. PEEK(-16287)函数可以说明游戏棒上的按钮是否已被按动。如果按钮已被按过,PEEK的值将是多少?如果没有被按动呢?
4. 如果要问游戏棒1上的按钮情况,PEEK()中的自变量应取什么值?
5. SCRNX(x, y)函数有什么作用?对x来说允许取什么范围的数? y呢?
6. 如果一个程序能使一个移动的点来穿过屏幕,那么怎样来判断点是否已击中屏幕上的某些目标呢?

课 文

游戏棒和按钮

如果你使用的电脑带有游戏棒,那么请执行下列名字也叫“游戏棒”的程序:

```
10 REM: : : PADDLES: : :  
20 HOME: PRINT  
30 PRINT "TURN THE PADDLE KNOBS"  
35 PRINT: PRINT "AND PUSH THE BUTTONS"  
38 REM CHECK THE KNOBS
```

```

40 P0=PDL(0)
45 VTAB 10
47 PRINT "PDL(0)=";P0;TAB(15);
48 P1=PDL(1)
50 PRINT "PDL(1)=";P1
55 REM CHECK THE BUTTONS
60 B0=PEEK(-16287)
62 B1=PEEK(-16286)
65 VTAB 12
70 C$=" "; IF B0>127 THEN C$="BANG"
72 PRINT C$;
74 C$=" "; IF B1>127 THEN C$="BANG"
76 PRINT TAB(15); C$
99 GOTO 40

```

用 RESET 键可以结束程序的执行。

如果电脑有四根游戏棒，可分别编号为0、1、2和3（有的电脑可能只有两根游戏棒，可以把它们编号为0、1）。

PDL(0) 函数可以知道 0 号游戏棒上的按钮旋转到什么位置；如果数靠近0，旋钮已旋转到左边；如果数靠近255，旋钮已旋转到右边。这儿要注意：在同一行中，不能两次使用PDL()，甚至在相互靠近的行中也不要连续使用 PDL() 函数。理由很简单：电脑需要花费一些时间才能计算出游戏棒上的旋钮的位置。如果频繁地从游戏棒上取数，反而会将会将电脑运行的程序搞乱。在上面的程序中，第40行和第48行中就包含PDL() 函数，它们被第45行和第47行分开了。

如果要弄清游戏棒 0 上的按钮是否已被按动，只要查看一下存贮器中的位置 - 16287 的情况就行了。这正是 PEEK (- 16287) 函数所做的事。如果它发现的数大于 127，这说明按钮已被按动。游戏棒 1 上的按钮是否被按动，可以由看一下存贮器中的位置 - 16286 的情况来决定。

在屏幕上移动点

执行这个程序：

```
10 REM MOVE A SPOT
15 :
20 HOME; GR
30 PRINT "USE PADDLE 0"
50 Y=39
55 COLOR=0
60 PLOT X, Y
65 X=PDL(0) * 39/255
67 COLOR=3
70 PLOT X, Y
99 GOTO 55
```

用 RESET 键停止程序的执行。用文件名 "MOVE A SPOT." (移动点) 把程序存放到磁盘中去。

记住：屏幕只有 40 个小方块长和 40 个小方块宽，每次只要从 0 算到 39 就行了。

程序中第 50 行的意思是：“把最后一行的点放在屏幕底部。”

抹去和画上

“抹去，画上；抹去，画上……”程序每次在屏幕上画上一个

点时必须先把它抹去后才能把它画到屏幕的其他地方去。否则屏幕上的点就会越画越多。如果要观察这一现象，可以把55行改成 $COLOR=3$ ，或去掉第60行。

上面的程序中第55行和第60行的任务是消除点。第55行使颜色变成黑色，就象屏幕的底色一样；第60行就用上面规定的黑颜色画出一个黑点来取代屏幕上原先的彩色点。

第67行把点变成蓝色；第70行就在屏幕上画上一个蓝点。

第65行在问游戏棒：点应该跑到底线的什么地方去？游戏棒可以用0到255之间的一个数进行回答，但在屏幕上只能从0算到39，所以我们必须先把这个数除以255，得到0和1之间的一个数，然后再乘39，再得到0到39之间的一个数。在这个数没有确定之前，我们称它为数 x 。

发射一束激光

取出刚才存入磁盘中的程序“MOVE A SPOT”，再添加一些新的语句行：

```
10 REM SHOOT A DOT
20 HOME: GR
30 PRINT "USE PADDLE 0"
50 Y=38
55 COLOR=0
60 PLOT X, Y
65 X=PDL(0) * 39/255
67 COLOR=7
70 PLOT X, Y
75 IF PEEK (-16287)>127 THEN GOSUB
```

200

```
99 GOTO 55
200 REM SHOOT
205 COLOR=12
210 FOR I=38 TO 0 STEP -1
220 PLOT X; I
230 NEXT I
240 FOR I=38 TO 0 STEP -1
245 COLOR=0; PLOT X, I
250 NEXT I
299 RETURN
```

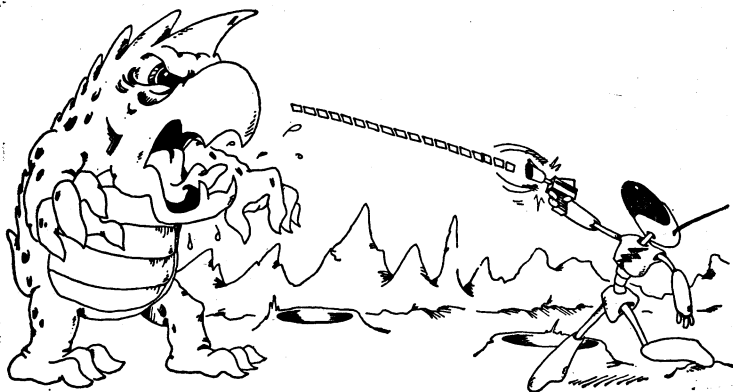
执行这个程序,用 RESET 键来停止程序的执行。把程序的文件名命名为“SHOOTING.”(射击),并把它存入磁盘中。

第75行问游戏棒上的按钮是否已被按动。如果是的话,第200行开始的子程序就会射出一条彩色线,然后再把它抹去。

射星星和 SCRN()函数

从磁盘中取出“SHOOTING”(射击)程序,并把下列这些语句行加进去,这样原程序就变为“射星星”程序了:

```
10 REM *** SHOOTING STARS ***
40 GOSUB 300
215 IF SCRN(X, I) < > 0 THEN PRINT
    CHR$(7);
300 REM INITIALIZE STARS
305 COLOR=1
310 FOR I=1 TO 10
```



```
320 X=RND(8) * 40  
330 Y=RND(8) * 20  
350 PLOT X, Y  
360 NEXT I  
399 RETURN
```

重新执行这个程序，仍用 RESET 键来停止程序的执行。用文件名“SHOOTING STARS”（射星星）把程序存入磁盘中去。

这个程序在执行中调用了一次在第300行上的子程序，它能在靠近屏幕顶部的地方随机地画10颗星星。

程序在画激光线时，它要看看下一个要画到的小方块是否是背景色（黑色）。如果不是，它肯定就是一颗星星，这时电脑会发出“嘟”的声音来说明你已打中了一颗星星。

要看目标，可以用第215行中的 SCRN(x, y) 函数，它的作用是问在屏幕 x、y 位置上的点是什么颜色。这里举例的情况是：背景为颜色0，代表星星的小方块颜色为9。

作业28:

1. 课文上“SHOOTING”(射击)程序中第200行开始的子程序能画出引人入胜的激光线。请你清除掉这个程序后再编写一个子程序,它也可以不断地在画出下一个小方块之前把前面的小方块抹去。

2. 在课文上的“SHOOTING STARS”(射星星)程序中:

在第300行开始的子程序中,请用红色和蓝色两种颜色来画星星;在第200行开始的子程序中,让激光发出一束红光。

第二十九课 ASCII 代码、键盘 和 ON-GOTO(多路转向)语句

教学参考:

本课要介绍表示字符的 ASCII 代码以及 ASC() 函数和 CHR\$() 函数。这两个函数可以分别用来把字符转换成数字或将数字转换成字符。

ASCII 代码是一种标准代码的表示方法,它专门用于电脑与打印机,电脑主机与电脑终端以及电脑与电脑之间的信号交换。在电脑的程序中 ASCII 代码也非常有用。由于它编排代码的方法是按英语字母的次序来进行的,同时代码本身的数字也是按次序的,因此 ASCII 代码在按字母次序处理的程序中就很有用。标点符号也有相应的 ASCII 代码。

有一些与电脑硬件有关的信号象“报警”、“换行”、“回车”等也有 ASCII 代码。这种代码能放在 PRINT 语句中,用来控制屏幕的显示以及扬声器声音的大小。

复习题:

1. ASC(S\$)函数是把它值作为一个字符串还是作为一个数输入电脑的?
2. ASC(S\$)函数的自变量是字符串还是数?
3. 对 CHR\$(N)函数回答上面提出两个同样的问题。

4. 字母 B 和字母 W, 哪一个字母的 ASCII 代码较大?
5. 你知道字符“1”的 ASCII 代码吗? 它是数字 1 吗?
6. 如果执行下面的一行, 电脑会做些什么?

```
10 PRINT CHR$(13); CHR$(7)
```

(如果你不知道, 请在电脑上试一下。)

课 文

按英语字母次序给字母编号

看了标题你也许会说:“这很容易, A 为 1, B 为 2, C 为 3 等等, 这不就行了吗?”事实上, 由于某些原因, 在电脑中是这样来进行编码的: 例如, A 为 65, B 为 66, C 为 67 等等。

这些数码就叫作字符的 ASCII 代码。标点符号和数字也有相应的 ASCII 代码。

ASC()函数

使用 ASC()函数可以把字符串变成 ASCII 代码。



请执行下列程序：

```
10 REM * * * WHAT NUMBER IS THIS  
    KEY? * * *  
20 PRINT  
25 PRINT "PRESS KEYS TO SEE ASCII  
    NUMBER"  
30 GET C$  
40 PRINT C$; TAB(5); ASC(C$)  
50 GOTO 30
```

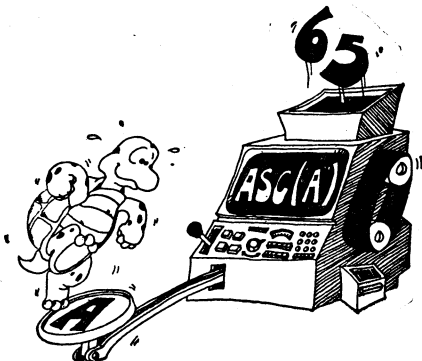
用键盘输入一些字母,数字或标点符号,看看它们的 ASCII 代码;再按一下 RETURN(回车)键和其他键试试。

按 RESET 键结束程序的执行,然后把它存放到磁盘中。

字母表

使用 ASCII 代码到底有什么好处呢?它能帮助我们打印字母表或按字母次序排列的表格。

执行下面的程序：



```

10 REM ALPHABETIZE
20 PRINT
30 INPUT"GIVE ME A LETTER:"; A$
35 PRINT
40 INPUT"GIVE ME ANOTHER:"; B$
45 A=ASC(A$); B=ASC(B$)
47 REM PUT IN ALPHABETICAL ORDER
   BY
48 REM SEEING WHICH HAS THE LOWER
   ASCII NUMBER.
50 IF A>B THEN X=A; A=B; B=X; REM
   SWAP THEM
55 PRINT
60 PRINT "HERE THEY ARE IN ALPHA-
   BETICAL ORDER"
65 PRINT;PRINT CHR$(A); TAB(5); CHR$(B)

```

请注意观察 ASC()函数和 CHR\$()函数在程序中的作用。

ASC()函数给出字符串内第一个字符的 ASCII 代码。

CHR\$()函数正好相反。它给出与 ASCII 代码相对应的字符。

把这个程序存入磁盘中去。

字符的 ASCII 代码

英语字母的 ASCII 代码从65(代表字母 A)开始, 到90(代表字母 Z)为止;

数码字的 ASCII 代码从48(代表数字0)开始,到57(代表数字9)为止;

标点符号则从32(代表空格)开始到47,又从58开始到64,再从91开始,一直到95为止;

RETURN(回车)键的 ASCII 代码为13。

数字转换成字符

用 CHR\$() 函数可把一个 ASCII 代码转换成一个字符。

执行下列程序,看看 CHR\$() 函数的作用。

```
10 REM / / / DISPLAY ASCII / / /
11 :
20 HOME
25 PRINT "NOTE: FIRST 32 CHARACTERS
   ARE INVISIBLE"
30 FOR I=0 TO 255
40 PRINT I, CHR$(I)
50 FOR T=1 TO 500: NEXT T
60 NEXT I
```

把这个程序存放到磁盘中。

由于代码到95为止,因此第30行中大于95的数就不起作用了。

数值从0到31都不能使电脑打印出任何内容。只是当取数7时,能使电脑发出“嘟”声;当取10和13时,电脑能使打印格式移到下一行。具体地说,ASCII 代码7表示“铃”,在打印时,使电脑发出“嘟”声;ASCII 代码10表示“换行”,当往电脑内输入或让电脑打印某些内容时,它把屏幕上的光标移到下面一行;ASCII

代码13表示“回车”，即打字机上的回车键，它把屏幕上的光标移到一行的开头。与ASCII代码不同的是，在打字机上是由“回车”键来做换行工作的。

游戏和键盘

GET 语句能让电脑等待你按键输入某些内容。另外有一种方法也可以按键输入，但不需要让电脑等待，它常用于游戏程序中。请看下面的程序（注意：这里的SNAKE——蛇程序还要作些调试，请参看33课最后一节“程序的纠错”）。

```
2   GOTO 1000; REM SNAKE
100  REM MAIN LOOP
101  ;
102  CL=CL+11
103  IF CL > 12 THEN CL=1
104  COLOR=CL
105  DR=PEEK (AR)
107  S=PEEK (ST)
110  IF DR=LL THEN D=D-1; IF D=0
      THEN D=4
111  IF DR=R THEN D=D+1; IF D=5
      THEN D=1
113  FOR T=1 TO 50; NEXT T
115  ON D GOTO 120, 122, 124, 126
120  Y=Y-1; GOTO 130
122  X=X-1; GOTO 130
124  Y=Y+1; GOTO 130
```

```
126 X=X+1
130 PLOT X, Y
140 A=B; B=C; C=E; E=F; F=G; G=X
142 L=M; M=N; N=O; O=P; P=Q; Q=Y
145 COLOR=0; PLOT A, L
199 GOTO 100
999 END
1000 :
1001 REM *****SNAKE *****
1002 :
1010 REM BY EDWARDH, CARLSON
1011 :
2000 REM BORDER
2010 GR; COLOR=7
2020 HLIN 0, 39 AT 0
2022 VLIN 0, 39 AT 39
2024 HLIN 0, 39 AT 39
2046 VLIN 0, 39 AT 0
2100 LL=149
2102 R=136
2105 AR=49152
2107 STROBE=49168
2110 X=20; Y=20
2115 A=X; B=X; C=X; E=X; F=X; G
=X
```



```

2116 L=Y; M=Y; N=Y; O=Y; P=Y; Q
      =Y
3000 REM INSTRUCTIONS
3010 INPUT "DO YOU WANT INSTRUCTIONS?";
      Y$
3015 IF Y$ < > "Y" THEN 3999
3020 PRINT "TURN LEFT, LEFT ARROW
      KEY"
3022 PRINT "TURN RIGHT, RIGHT ARROW
      KEY"
3999 REM
9990 COLOR=1
9999 GOTO 100

```

键盘盒

在上面的程序中,你每次按键输入,电脑总是把字符放进名字为“49152”的盒子内,这就是 INPUT 语句去找字符的地方;也是 GET 语句去找字符的地方。盒子内的东西我们也可以去查看,这只要使用 PEEK 语句就行了。SNAKE (蛇)程序中使用了下面的两行语句:

```

105 DR=PEEK(AR)
107 S=PEEK(ST)

```

句中,AR=49152,ST=49168。在 PEEK 语句中,用变量名而不用常数可使程序执行得更快。

第105行查看的目的是要知道是否有键盘输入。如果有,那么这只键的 ASCII 代码就从键盘的盒子中取出,放入名为 DR

的盒子内。

第110行询问这个键是否是左箭头键 (ASCII 代码 149, 见 2100行)。

请你自己想一想第111行做些什么。

第107行告诉电脑把键盘盒里的内容倒空, 这样它就准备接受从键盘输入的另一个字符。这儿使用了 `107 S=PEEK(ST)` 语句行。

你知道盒子内的数字 49168 是从哪里得到的并放进 ST 盒子中去的吗?

ON—GOTO(多路转向)语句

```
115 ON D GOTO 120, 122, 124, 126
```

这一句的意思是:

如果D为 1	则转向120
2	122
3	124
4	126

如果D为其他值, 就自动转向下一行语句执行。

在GOTO之后, 可以放一两个或多个行号, 但是每一个行号应该与程序中某个地方的行号相同。

作业29:

1. 编写一个程序, 先由你输入一个单字, 然后能以字母次序重新安排这个单字中的所有字母。

2. 编写一个“无法理解的言语”的程序, 当输入一句句子后, 它会去掉句子中所有的元音字母, 再把这个句子打印出来。

3. 编写一个程序, 它能使用GET语句得到一个A到C的

字母，把字母变成 1 到 3 的数，然后用 ON—GOTO 语句来选择菜单(即字母单)上的某一项。



第三十课 数组与DIM(数组说明)语句

教学参考:

本课将介绍数组的概念,同时还介绍 DIM(数组说明)语句。

首先讲一维数组。如果我们把数组比作家庭,那么数组的元素就是家庭成员,下标值是成员的“名字”。

二维数组可比作为电视屏幕上的矩形阵列。

更高维的数组在本课中只不过是提一提而已,没有给出例子来说明。

复习题:

1. DIM AD\$(5)语句做些什么?
2. 在程序中,应把 DIM 语句放在什么地方?
3. 数组有哪两类?
4. 数组的“下标”或“下标变量”是什么?
5. 语句 DIMSR(5, 9)又做些什么?

课 文

数组家庭

22 F\$(0) = "DAD"

24 F\$(1) = "MOM"

26 F\$(2) = "BRIAN"

上面三行语句组成一个家庭，家庭中的每个成员都含有一个变量，也许不说你也看得出这是一个字符串变量的家庭。



下面是一个数字变量的家庭：

35 N(0) = 43

37 N(1) = 13

39 N(2) = 0

41 N(3) = 0

如果一个家庭中的每个成员都有一个姓，比如象A()或B\$()等一样，各人的姓又都是一样的；而且每个成员还都有一个名，它是用一个数填在括号()内来命名的，这个名字又总是以“0”开始，依次增加；那么这个家庭就叫做“数组”，而名字叫做“下标变量”。

用 DIM 语句保留盒子

当一个家庭的全体成员都去看电影时，事先总要为每个人保留座位。用 DIM 语句就能做到这一点。

DIM 语句告诉电脑要给数组留一排空盒子。例如：

```
18 DIM A(3)
```

意思是要留四只存贮盒子。每一只盒子分别给变量 A(0)、A(1)、A(2)和 A(3)使用。要记住这些盒子的编号是从“0”开始的。下面再举一例：

```
30 DIM A(3), B$(4)
```

这一次 DIM 为 A()数组保留四只盒子，为字符串数组 B\$ 保留五只盒子。从 B\$(0)到 B\$(4)的盒子是供字符串 B\$ 用的，开始时每个盒子都是空的。

规则：在调用数组以前，要先把 DIM 语句写入程序中。

制表格

下面的程序是用 DIM 语句来制表格的：

```
10 REM + + + IN A ROW + + +
20 HOME: PRINT
30 DIM A$(5)
35 PRINT"ENTER A WORD"
40 FOR N=0 TO 5
45 IF N>0 THEN PRINT "ANOTHER"
50 INPUT A$ (N)
55 PRINT
60 NEXT N
70 PRINT
100 REM PUT IN A ROW
```

```

105 PRINT"HERE THEY ARE IN A
      ROW"
106 PRINT
110 FOR I=0 TO 5
120 PRINT A$(I); " ";
130 NEXT I

```

请输入并执行这个程序,然后把它存入磁盘中。

数组中的成员可以单独地使用。看这一行:

```
40 B$(2)="YELLOW SUBMARINE"
```

数组也可以由下标值的不断变化来循环使用。上面程序中的第50行与第120行就是这样做的。

在下面的程序中你只要填进朋友的名字和电话号码就能列出一份清单,象电话号码簿一样。这是由程序中的 DIM 语句同时制作了二个表格所形成的。

```

10 REM PHONE LIST
20 HOME: PRINT
30 DIM NAME$(20), NUMBER$(20)
35 I=0
40 PRINT "ENTER NAMES AND
      NUMBERS"
50 PRINT: INPUT "NAME?"; NA$(I)
60 INPUT "NUMBER?"; NU$(I)
70 I=I+1: GOTO 50

```

请输入并执行这个程序,然后按下 RESET 键停止程序的执行。再把程序存放在磁盘中。

一维、二维和多维数组

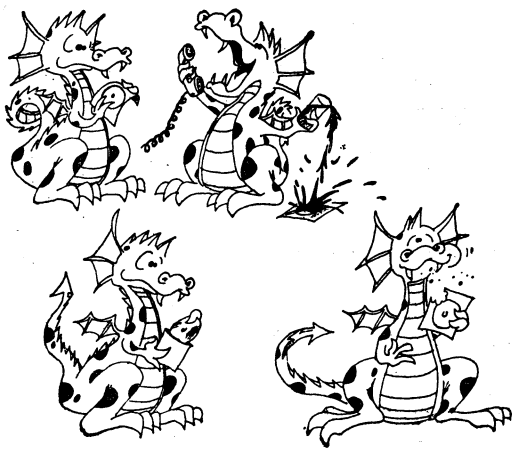
只有一个下标的数组叫作一维数组。数组也可以有二个或更多个下标。二维数组中的家庭成员要确定它们自己在屏幕上的位置就象去查看年历表中的某月某日一样，每个成员都是由两个下标值来决定的。下面请试一个例子。

```
10 REM + + + TWO-DIM ARRAY + + +
15 :
18 HOME
20 DIM T(5, 6)
30 FOR X=0 TO 5
40 FOR Y=0 TO 6
50 T(X, Y)=X+Y
60 NEXT Y, X
65 :
70 REM * * * * * PRINT OUT THE
    ARRAY
72 :
80 FOR J=0 TO 6
82 VTAB (5+3*J)
85 FOR I=0 TO 5
87 HTAB(3+4*I); PRINT T(I, J);
90 NEXT I,J
```

作业30:

1. 编写一个电话号码簿的程序，列出名字表并在每一个人名字旁边写上相应的电话号码。

2. 用二维数组来编制一个“周历”程序。要用 DIM AR\$(7, 24) 语句, 使程序对于一周中的每一天中的每一小时都打印出一项。



第三十一课 AND(与)、OR(或)、NOT(非)逻辑关系

教学参考:

本课要介绍 AND (与)、OR (或)、NOT (非) 逻辑关系和用数值表示的真与假。这些内容对某些形式的 IF 语句非常有用。在第12课中,有一个“TEENAGER(少年)”程序,里面用了一个隐蔽的IF 语句来打印“你是一个少年”。如果改用“OR”来表示程序中的逻辑关系,则会更明了一些。本课中有些概念比较抽象,难以掌握。用数字 1 和 0 来判断结果的真和假似乎欠佳,但是电脑常常会把非 0 的值都判断为真,这一点倒可以弥补一些不足。

复习题:

1. 说说下列各行IF语句,哪些能打印出东西来:

```
10 IF 3=3 THEN PRINT "HI"
```

```
10 IF NOT(3=3) THEN PRINT "HI"
```

```
10 IF 3=3 OR 0=2 THEN PRINT "HI"
```

```
10 IF 3=3 AND 0=2 THEN PRINT "HI"
```

```
10 IF "A"="B" THEN PRINT "HI"
```

```
10 IF NOT("A"="B") THEN PRINT "HI"
```

2. 下面每一行语句打印出什么值?

```
10 A=1; PRINT A, NOT A
```

```

10 A=0; PRINT A, NOT A
10 A=1; B=1; PRINT A AND B
10 A=0; B=1; PRINT A AND B
10 A=0; B=0; PRINT A AND B
10 A=0; B=1; PRINT A OR B
10 A=0; B=0; PRINT A OR B
10 PRINT NOT 23
10 PRINT NOT 0
10 PRINT 3 AND 7
10 PRINT 3 AND 0

```

课 文

输入下面的程序,看看它是否比第12课里的“TEENAGER”(少年)程序在逻辑上更明了些。

```

10 REM <<<AND, OR, NOT>>>
20 HOME; PRINT
30 INPUT "YOUR FIRST NAME"; N$
35 PRINT
40 INPUT "YOUR AGE"; A
45 PRINT
50 IF (A>12) AND (A<20) THEN PRINT
   N$ "IS A TEENAGER."
55 NFLAG=(A<13) OR (A>19)
60 IF NFLAG THEN PRINT N$ "IS NOT
   A TEENAGER."

```

```
65 PRINT
```

```
70 IF (NOT NFLAG) AND (A=16) THEN  
    PRINT "AND" N$ "IS SWEET SIXTEEN."
```

请执行这个程序,并把它存入磁盘中。

AND(与)逻辑关系

看上面程序中的第50行,意思是:在两种情况同时成立时对“IS A TEENAGER (是一个少年)”来说是真的,即他们是超过12岁而又不满20岁的人。

```
IF(超过12岁)AND(不满20岁)THEN(你是一个少年)
```

OR(或)逻辑关系

在第55行中使用了“OR(或)”的逻辑关系。说了两种情况:“年龄不足13岁”或者“年龄超过19岁”。其中只要有一种情况符合,就不是少年。

```
IF(不满13岁)OR(超过20岁)THEN(你不是一个少年)
```

用数字表示真和假

电脑是怎样表示一个条件是真还是假呢?它是用数字来表示的。

规则: 真 用数 1 表示

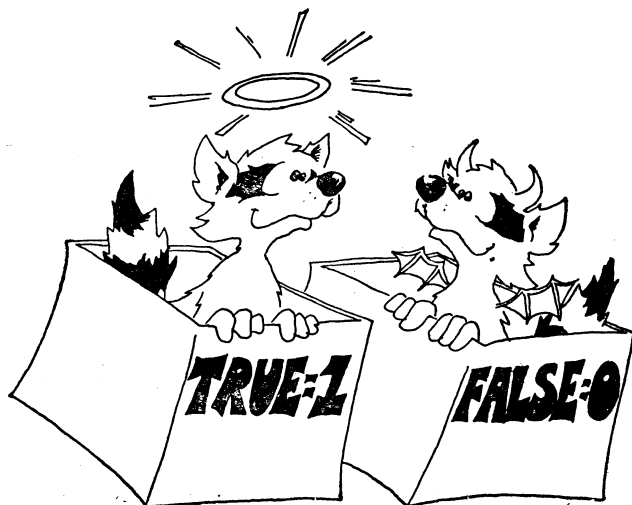
假 用数 0 表示

很容易记住 0 是表示假的。如果作业都错了得分就是 0。

例如,在编辑方式中输入:PRINT 3=7电脑就进行检查,看看 3 是否确实等于 7。结果不是,就打印“0”,意思是假的。

再输入:PRINT 3=3

同样,电脑也要进行检查,看看3=3是否成立。结果是对的,就打印“1”,意思是真的。



TRUE——真 FALSE——假

把真与假放入盒子里

真与假的值能象其他数一样可以进行处理，还可以存贮在前面写有数字变量名称的盒子内。例如执行下列程序：

```
10 N=(3=22)
```

```
20 PRINT N
```

数字 0 会被存贮在盒子 N 中，因为 $3=22$ 是假的。再执行：

```
10 N="B"="B"
```

```
20 PRINT N
```

数字 1 会被存贮在盒子 N 中，因为引号中的两个字母是一样的，所以 $"B"="B"$ 是真的。

用IF语句来判断真和假

以前学过的IF语句如下：

10 IF条件 A成立 THEN语句 C

现在用编辑方式,把下面两句做一下:

```
IF 0 THEN PRINT "TRUE"
```

```
IF 1 THEN PRINT "TRUE"
```

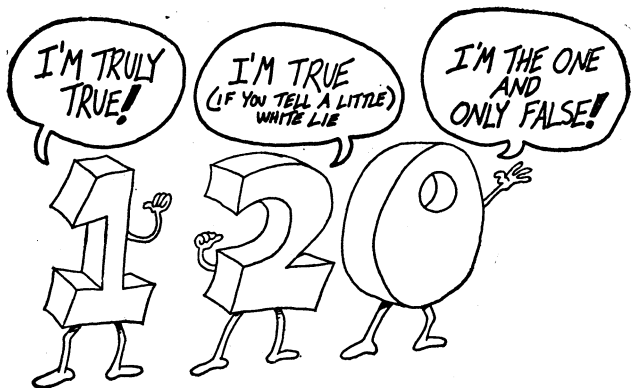
再试一下:

```
IF 22 THEN PRINT "TRUE"
```

这里用的是另一种 IF 语句的形式,它的规则是:在 IF 语句中,先判断条件。如果数为 0,电脑就认为“条件是假的”,并跳过 THEN 后面的语句;如果不是 0,电脑就认为“条件是真的”,接着就执行 THEN 后面的语句。

IF 语句判断条件时,对于条件为真可以是数字“1”,但扩大化以后,只要不是 0 的任何数都被认为是真的。

NOT(非)逻辑关系



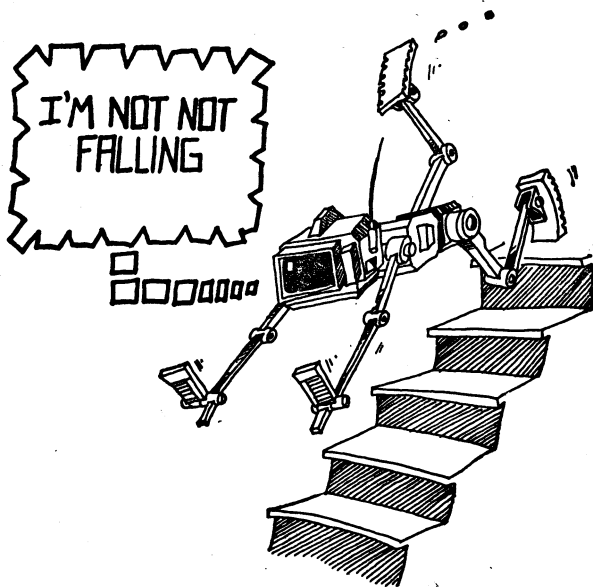
I'M TRULY TRUE——我是真正的“真”

I'M TRUE——我也算是“真”

I'M THE ONE AND ONLY FALSE——只有我一个是假

逻辑NOT(非)是把假改成真或把真改成假。试做：

```
10 REM ??? DOUBLE NEGATIVE ???  
20 N=3  
30 PRINT "N"; TAB(20); N  
40 PRINT "NOT N"; TAB(20); NOT N  
50 PRINT "NOT NOT N"; TAB(20); NOT  
    (NOT N)  
60 REM The computer knows that "I don't  
    have no..."  
61 REM means "I do have....."
```



I'M NOT NOT FALLING——我不是没有摔下去

把程序存入磁盘内。

处理 NOT 逻辑关系时,其含意同样可以扩展一些。上面程序中的 N 为了电脑就判断为真,而 NOT3 就是假,数值为 0。同样,NOT(NOT3)的值与 NOT(0)、NOT(假)、真、或者数字 1 它们的结果都是一样的。

逻辑符号

在条件语句的逻辑表达式中可以使用这 6 个关系符号:

= 等于

< > 不等于

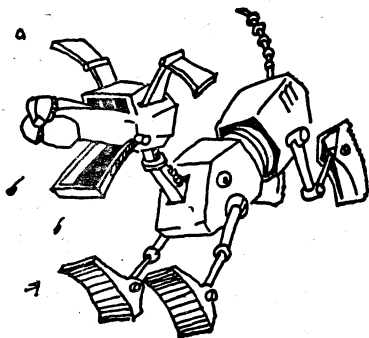
< 小于

> 大于

<= 小于等于

>= 大于等于

要组成<>、<=及>=符号必须按两次键。六个符号中的最后两个符号是新的,看一看下面这些例子就可以弄清<与<=之间的差别了。



$2 \leq 3$ 是真 $2 < 3$ 是真

$3 \leq 3$ 是真 $3 < 3$ 是假

$4 \leq 3$ 是假 $4 < 3$ 是假

下面两个逻辑表达式是相同的:

$2 \leq Q$ $(2 < Q)$ 或 $(2 = Q)$

作业31:

1. 请说说看 N 盒子里的内容应是什么?

$N = 4 = 4$

$N = "G" < > "S"$

$N = 5 > 7$

$N = 3 > 2$ AND $3 < 2$

$N = 4 = 3$ OR $4 = 4$

$N = \text{NOT } 0$

$N = 5 > = 4$

2. 下列语句中,哪些会打印出单词“JELLYBEAN”?

IF 0 THEN PRINT “JELLYBEAN”

IF 1 THEN PRINT “JELLYBEAN”

IF 9 THEN PRINT “JELLYBEAN”

IF $3 < > 3$ THEN PRINT “JELLYBEAN”

IF 2 AND 4 THEN PRINT “JELLYBEAN”

IF 0 OR 1 THEN PRINT “JELLYBEAN”

IF NOT 3 THEN PRINT “JELLYBEAN”

IF “A” = “2” THEN PRINT “JELLYBEAN”

IF NOT (3) AND 2 THEN PRINT “JELLY-
BEAN”

```
IF NOT (0) OR 0 THEN PRINT "JELLY-  
BEAN"
```

```
IF 4<=5 THEN PRINT "JELLYBEAN"
```

3. 编写一个可以用来测试具有双重否定的句子的程序：使它找出象NOT, NO, DON'T, CAN'T NOTHING等这样的否定词，并数一下总共有几个；如果有两个这样的单词，那么就判断出这是一句双重否定句。请用“COMPUTERS AIN'T GOT NO BRAINS.”(电脑不是没有脑袋)这句话来测试一下你编的程序。

第三十二课 方便用户的程序

教学参考:

本课将介绍一些有用的程序,为电脑用户提供方便。

编写程序不仅要达到一定的功能,而且还要考虑到能让使用者用起来感到方便。这意味着:屏幕显示的内容要容易阅读;键盘输入尽可能少用回车键;程序中若有错误,能及时得到提示并给于使用者有进行修改的机会。

方便用户的程序必须浅显易懂,随时可得;每个程序还都要加以说明。初学者需要有详细的说明,而有经验的使用者只要一些简单的提示就行了。

要教会编写这方面的程序是有一定困难的。能否学会一些行之有效的编写方法与技巧得依靠不断的实践。一般地说,大部分年轻学生在这一方面不会有很大进展的,但只要能了解编制程序的要求,学会使用一些能编制这方面程序的技能,那就算已有一定的收获了。

复习题:

1. 不管用户需要与否,一个程序是不是都应该提供说明?
2. 什么是程序中的提示? 试举两例说明。
3. 什么是电脑显示屏幕上的“向上翻滚”现象? 怎样把语句行写入屏幕而不产生向上翻滚?

4. 如果使用者要从键盘上打入一个字母,那么用什么语句最好(要避免使用回车键)?

5. 什么是“错误陷阱”?如果要求输入一个1到5的数,但是输入的却是另外一个数,那么怎样才能查出错误?

6. GOSUB语句一般放在程序的哪一部分?

7. 为什么要把程序的“STARTING STUFF”(起始部分)放在程序的结尾(高行号数上)?

课 文

电脑用户一般有两种类型:

一、大部分用户在执行一个程序时,往往需要有:

1. 指导性的说明;
2. 提示部分;
3. 屏幕上显示的内容清楚而不杂乱;
4. 自动从屏幕上抹去一些旧的内容;
5. 按键次数不要太多;
6. 对一些错误操作有保护措施。

二、有一部分用户常需要改变程序中的内容,他们希望程序能:

1. 一个程序由几个部分组成;
2. 每一个独立部分都用REM语句注上小标题;
3. 在程序中要有解释。

不要忘记编制程序的人自己也是使用者,因此在编制时也要为方便自己使用而多考虑一下。

程序的构成

程序主要由三部分组成:



STARTING STUFF——起始部分

一、“起始部分”有时写在程序的开头，它的作用是：

给用户提供有关的说明；

提供屏幕显示；

给所有的变量设置初值；

询问用户有关开始时的信息。

二、“主程序”部分，它的作用是：

控制任务执行时的次序；

调用子程序来完成某些任务。

三、“子程序”部分，它的作用是：

完成程序的一部分工作。

如何来安排程序的各个部分

下面是一个程序完整的轮廓：

```
1 GOTO 1000; REM * * * program name
```

```
* * *
```

```

...
100 REM MAIN LOOP
...
... calls subroutines
...
199 END
1000 :
1001 REM * * * program name * * *
1002 :
... REM's that give a description of the
... program, variable names, etc,
...
1999 :
2000 REM STARTING STUFF
...
... ask for starting information
... set variable values
... give instructions

```

一般地说,主程序部分要放在整个程序靠近前面的地方,因为那里电脑执行起来比较快。

起始部分往往放在程序的末尾,因为这段程序可能最长,要做的事情很多,随时都可能要加入一些内容,它也不需要很快地执行,所以这样放用户会感到很方便。

子程序部分就不同了,它要看具体情况而定:

1. 要快速执行的一些子程序,可以放在第2行至第99行之

间;

2. 一些做初始准备工作的子程序可以放在第1999行之后;
3. 其他类型的子程序就放置在第200行至第1999行之间。

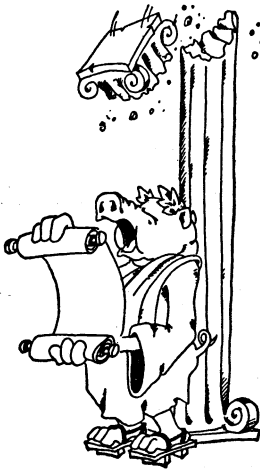
“选择法”提示

```
280 PRINT "DO YOU WANT INSTRUCTION  
<Y/N>"
```

这一行语句的意思是问“你要看说明吗 <Y/N>”，它让初学者通过按“Y”（表示 YES）键来看看程序的有关说明，而其他人用不着去看说明时，就按“N”（表示 NO），这样显示屏幕上就不出现文字的说明。

用“选择”法来提供必要的提示，既简洁明了，又满足了各种不同的要求，会使用户感到很方便。

制止屏幕上的翻滚现象



电脑工作时屏幕上经常会出现语句行向上翻滚的情况：电脑不停地在屏幕底部写着程序中的语句行，每写好一个新行，原来的语句行就依次向上推进一行，这种向上翻卷的现象会使人目眩头晕。这种情况就象古代罗马人用的画轴一样，底部不断展开，而顶部不断绕紧。为了避免这种翻卷，可以用 VLIN 语句和 HLIN 语句来定位，使程序在适当的位置上打印出你需要看的语句行；在不需要的时候，可以在这个位置上打印一串空格把它抹去。利用延时循环可以把显示的内容长时间地留在屏幕上，便于你仔细阅读。

少按键的方法

用 GET 语句输入一个字母，就不必按回车键，这样可以减少按键的次数。下面是个例子。

```
380 PRINT "DO YOU NEED INSTRUCTIONS?  
      <Y/N>"  
382 GET R$; IF R$="Y" THEN GOSUB  
      3400
```

预防出错

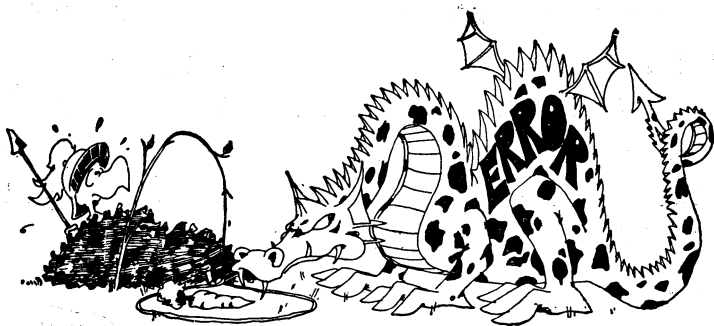
把下面一行语句加到上面的程序中去：

```
384 IF R$ < > "N" THEN GOTO 380
```

第380行只提供两种选择：Y或N。如果按了其他键，第384行语句就对键入的字符不作处理，并仍转回到第380行，等待你重新再输入一次。程序有了这种防错处理功能就不会把事情弄糟。

作业32:

1. 看一看下面的 COLOR EATER (吃颜色者) 程序，用



ERROR——错误

REM 语句来解释程序中的一些语句行；并对打印出来的句子内容作些修改，例如“**I HAVE NO FOOD**”一句把打印在第 2200 行和 2210 行上的内容破坏了，你会修改好吗？

```
1 GOTO 1000; REM *** COLOR EATER
  ***
2 ;
100 NC=0
101 FOR I=X-1 TO X+1
102 FOR J=Y-1 TO Y+1
104 IF I<0 THEN I=0
105 IF I>39 THEN GOTO 120
106 IF J<0 THEN J=0
107 IF J>39 THEN GOTO 120
109 CC=SCRN (I, J)
110 IF CC=C THEN X=I, Y=J, COLOR=0,
    PLOT X, Y; GOTO 100
```

```

112 IF CC < > 0 THEN NC=1
115 NEXT J, I
120 C=C+1; IF C>15 THEN C=1
125 IF NC=0 THEN GOSUB 300
130 VTAB (22); HTAB (1); PRINT X;" "
    TAB (5)Y " ";
135 VTAB (24); HTAB(1); PRINT C" ";
199 GOTO 100
300 REM COLOR EATER HAS NO FOOD
310 X=X+1; IF X>39 THEN X=1
320 PRINT "I HAVE NO FOOD!";
330 PRINT CHR$ (7)
335 FOR T=1 TO 500; NEXT T
340 HOME
399 RETURN
1000 REM
2000 REM STARTING STUFF
2010 HOME: GR
2020 FOR I=0 TO 39
2030 FOR J=0 TO 39
2040 C=INT (RND (8) * 15)+1
2045 COLOR=C
2050 PLOT I, J
2060 NEXT J, I
2100 X=20: Y=20

```

```
2200 VTAB 22; HTAB 8; PRINT "X, Y  
LOCATION";  
2210 VTAB 24; HTAB 8; PRINT "EATING  
COLOR";  
2999 GOTO 100
```

2. 编写一个密码程序,使用者只要输入一个“口令”(词),电脑便会打印出一个字母表,例如口令用的是词 APPLE,电脑就把重复的字母 P 去掉,得到 APLE,然后把它放在这个字母表的其他字母的前面,剩余的字母按正常次序排列,

APLEBCDFGHIJKMNOQRSTUVWXYZ

使用者可以从一份密码单上来选择是做编码工作还是做译码工作。

第三十三课 程序的调试和STOP(暂停)语句、 CTRL-C(控制)键、CONT(继续执行)命令

教学参考:

要完全掌握修改程序的技术不是一件容易的事情,这里仅仅介绍一些小的技巧,能有利于较快地排除程序中的故障。只有通过不断地实践,才能使學生逐步掌握这方面的技巧。

复习题:

1. 程序在执行时可以用哪两种方法使电脑打印:
BREAK IN 55
2. STOP 语句与END 语句有什么不同?
3. STOP 语句与CTRL-C 键有什么差别?
4. CONT 命令做些什么?
5. 为什么要把STOP 语句放入程序内?
6. 延时循环怎样帮助你调试程序?
7. 怎样附加一些PRINT语句来调试程序?
8. 程序在修改后为什么要把 STOP 语句和附加的 PRINT 语句划掉?
9. 在哪一些语句行中可以选用 CTRL-C 键来停止程序的执行? 这时能否改用STOP语句?

课 文

STOP(暂停)语句

输入并执行下列程序:

```
10 REM SECRET STOP
20 HOME
25 R=INT(RND(8) * 200)
30 FOR I=0 TO 200
40 IF I=INT (RND(R) * 200) THEN STOP
50 NEXT I
```

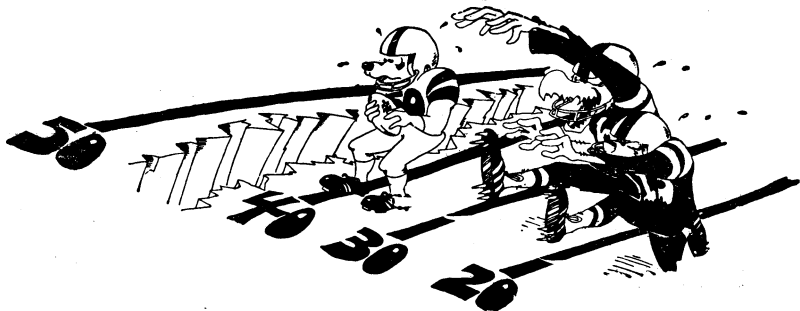
电脑运行了一会儿就发出“嘟”声,并停止程序的执行,这是第40行中 STOP语句起的作用。这时在屏幕上显示出:

```
BREAK IN 40
```

意思是“程序在第40行停止执行”。如果你想要知道这时 I 的秘密值是多少,就直接用打印语句:

```
PRINT I(不要带行号)
```

能立即打印出 I 的值。



如何使程序继续执行

程序停止执行后，可以用CONT命令使程序在原来的基础上继续执行下去。

“STOP”语句象“END”语句一样

STOP语句的作用是停止执行原来的程序，使电脑进入编辑方式。它除了发出“嘟”声并指出目前程序停在哪一行之外，其余情况与END语句完全一样。

在程序中要使用多少条STOP语句可以由你自己决定，这样程序将按照你的需要随时停下来，便于你调试或修改程序。

停止程序执行的另一种方法

用“CTRL-C”（控制）键也可以使程序的执行暂时停下来，操作时先按上面标有CTRL的键，然后再按“C”键。

执行下列程序：

```
10 REM GO FOREVER
15 PRINT: SPEED=100
20 PRINT "MUD TURTLES OF THE WORLD"
30 PRINT "UNITE!": PRINT
35 SPEED=255
40 FOR T=1 TO 1000: NEXT T
99 GOTO 10
```

程序在执行的过程中，不管什么地方都能用CTRL-C键使它暂停执行。电脑暂停时会发出“嘟”声，然后进入编辑方式并且屏幕上显示出：

```
BREAK IN LINE XX
```

其中XX表示程序暂停的行号位置。

可以用 CONT (继续执行)命令使程序从原来的地方继续往下执行。

用上面的程序试一下,使它在第20行停下来,然后又使它停
在第30行上,接着再停在第40行上。

程序停止执行后你可以做些什么

你可以将STOP语句放在程序运行不正常的那一部分上,然后执行这段程序,等停下来后,看看发生了一些什么异常情况,这样你就能找出原因,对症下药,修改好程序。具体做法是:

1. 你可以列表显示这部分程序,仔细琢磨一下,看看有什么错误;

2. 用 PRINT 语句检查有关的变量值,看看它们是否和我们所要求的相一致;

3. 一旦发现问题,可以根据需要加入、修改或删除一部分语句行(这时电脑处在编辑方式中,你完全可以这样做)。

当然用 CTRL-C 键也能停止程序的执行,但是它不一定能使程序正好就停在出问题的地方,这样就不便于修改程序。

使程序继续执行的方法

一共有四种方法可以使停下来的程序继续往下执行:

1. 如果程序不需要修改,可以用 CONT 命令让程序继续执行下去;

2. 用 GOTO XX 语句(其中XX为行号)可以让程序从任意指定的语句行处开始继续往下执行;

3. 如果对一个程序已作过修改,而你又想看看修改得对不对,那么可以用RUN XX命令(其中XX为行号),使程序从指定的行号处重新开始执行;

4. 用 RUN命令让程序从头开始执行,对这你已很熟悉了。现在比较一下上述四种方法之间的差别:

如果在原来的程序中没有加入语句、删除语句或用编辑方式改变某一行语句的内容,那么就可以用 CONT 命令使程序继续执行下去,也能输入GOTO XX语句让程序直接从指定的语句行开始往下执行。

假如程序已经作了修改,那么只能用 RUN命令让程序从头开始重新执行或者用 RUN XX 从指定的行号处执行该程序。

这里还要着重说明一下: CONT命令和GOTO XX语句所使用的变量值就是上面程序执行时保留在变量盒子里的内容; RUN命令和RUN XX命令都是把上次程序中用到的变量值全部清除后再执行程序的。

CONT 命令只限于用在 CTRL-C 键或者STOP语句的后面,使原来没有执行完的程序继续执行下去。而其他三种方法: RUN命令、RUN XX命令和 GOTO XX语句则不受此限制,它们还可以随意地启动一个新的程序。

捉程序中的“害虫”

程序中出现的一些小错误常常被称为“害虫”,排除错误就叫“捉害虫”。初次编写程序出现一些错误是在所难免的,如果你的程序执行得不顺利,可以做下面四件事:

1. 如果电脑打印出出错信息,就会指出现在程序停止在哪一行上,这便于你寻找错误;但要注意,真正产生错误的原因可能出在另外某一行上。

2. 如果电脑一直在执行程序,但就是得不到正确的结果,这时就应停止程序的执行,把一些 PRINT语句放入程序的适当

地方,以便观察程序执行时发生的情况。

3. 或者把 STOP 语句放入程序的适当地方。

4. 如果程序执行得太快而无法知道正在发生的情况,那么应加入一些延时循环来降低程序执行的速度。

程序修改完以后,要把调试用的 PRINT 语句、STOP 语句以及延时循环等附加的东西统统从程序中去掉。

作业33:

1. 在第29课提到的“SNAKE”(蛇)程序中,你能捉出一些“害虫”来吗?例如:当蛇撞到一堵墙时程序就被“撞破”了,你能修改那个程序给蛇喂一些“食料”,使得蛇一碰到了墙就让程序结束吗?

2. 把你以前编写的一些程序仔细修改一下。

附录一

磁盘使用法

磁盘保养

学生应该知道如何使用和保养磁盘，特别要注意下面几点：

1. 不要随使用手去触摸盘片上的呈褐色或灰色的长方形孔格。
2. 不要弯折盘片。
3. 在盘片插入磁盘驱动器或者从中取出时应小心行事。
4. 盘片使用后一定要放回保护袋里。
5. 不要把食物或饮料掉在盘片上，也不要不要在电脑旁吃东西。

为学生准备一张专用的盘片

输入下列程序，这是一个问候HELLO(您好)程序：

```
10 REM === HELLO ===  
20 :  
30 REM GREETING PROGRAM FOR  
40 :  
50 REM "KIDS AND THE APPLE"  
55 :  
60 HOME; PRINT
```

```
70 PRINT "STUDENT' S NAME' S DISK"  
75 PRINT CHR$(4); "CATALOG"  
77 PRINT  
80 PRINT "ENTERING EDIT MODE"  
85 NEW
```

请在程序中写有“STUDENT' S NAME”的地方放进某一学生的姓名,这样程序在执行时会打印出这学生的姓名;然后便显示磁盘上的目录,最后 HELLO 程序又将自己从存储器中自动抹去。这样做的目的是使学生不会把他们自己编写的程序和 HELLO 程序相混。

在输入上述程序后,把一张新的盘片插入 1 号驱动器,接着从键盘上输入:

```
INIT HELLO
```

这时磁盘驱动器就会旋转一段时间,这是在为向磁盘上存放程序做些必要的准备工作。

随后换一张 DOS 3.3 系统盘片放入驱动器,并且输入:

```
LOAD COLOR DEMOSOFT
```

接着再把学生用的磁盘放回驱动器,输入:

```
SAVE COLOR DEMOSOFT
```

再输入:

```
LOCK HELLO
```

```
LOCK COLOR DEMOSOFT
```

LOCK 是给上面的两个程序上锁,使它们不会被偶然的原因而从磁盘上擦掉。在目录中,凡是文件名前面带有“*”符号的,就表示这文件已经上了锁,能保护文件,使它不能轻易地被

擦掉。

如果上了锁的文件要修改或者要把它删掉，那就要先把它的锁打开。例如要对磁盘上已上了锁的HELLO 程序进行修改，可以输入：

```
UNLOCK HELLO
```

```
LOAD HELLO
```

然后就对这程序进行修改，完毕后再把它送到磁盘上保存起来，使用下列语句：

```
SAVE HELLO
```

```
LOCK HELLO
```

附录二

将程序存入磁带

如果电脑没有配置磁盘驱动器，可以用磁带录音机来保存一些有用的程序。要把程序录制到磁带上，有两个步骤：

首先调整好磁带录音机上的音量和音调；

其次必须输入正确的程序录制命令，使电脑完成信息存入磁带的工作。

录音机的调节

录音机第一次与苹果Ⅱ型电脑联接使用时，要进行音量调节，使它处于合适的音量控制状态。这项工作做完后，以后每次使用都调节到这一位置。APPLEⅡ BASIC编程手册中对录音机的调节也有较详细的说明。

程序存入磁带

将程序存放到磁带上，最简单的办法是每盘磁带上只存放一个程序，这只要选用记录数据的10分钟磁带就行了。这种磁带也比较便宜。当然用一般的磁带同时存放几个程序也行。

在磁带上录制程序的过程如下：

1. 先列表显示程序的内容，看一下是不是你要录制的程序；
2. 把磁带倒转到起始位置；
3. 输入SAVE命令，但不要急于按回车键；

4. 在录音机上同时按下RECORD和PLAY按钮,启动录音机使它处于录音状态;

5. 接着再按回车键。

随后电脑就自动完成录制程序的工作,你看到的现象是:

1. 光标从屏幕上消失了;

2. 在你慢慢地从1数到15的过程中,电脑一直在往磁带送一个平稳的音调;

3. 当电脑发出一声“嘟”时就真正开始录制程序了;

4. 当录制结束时,电脑又会发出“嘟”声,同时闪烁的光标也重新出现。

这时电脑认为已经把程序存放到磁带上去了。为了证实程序是否确实已经存放在磁带上,可以用LOAD命令把程序再送回电脑。

把磁带上的程序装入电脑

1. 检查录音机,看它是否已连接、调节好了;

2. 使磁带倒转到头;

3. 输入LOAD命令;

4. 按下录音机的PLAY键,使它处于放音状态;

5. 按回车键。

这时会发生下面的情况:

1. 光标消失了,电脑在耐心地收听它先前输给磁带的平稳音调,当然你这时是听不到这声音的。

2. 大约过了15秒钟,电脑就会发出一声“嘟”,表示它已听完了这平稳的音调,并且开始把程序从磁带上取出来,送到存储器中去。

3. 当电脑把这个程序读完以后又会发出一声“嘟”，闪烁光标会重新出现。在两次“嘟”声之间，大约花10秒钟就可以取出一个短小的程序，而花几分钟则可以取出一个很长的程序。

从磁带上取程序时出现的差错

如果电脑在读取程序的过程中出了问题，会显示一些错误信息，可能还会发出“嘟”声。如果看到下面几种错误信息：

```
* * * SYNTAX ERR
```

```
ERR
```

```
* * * MEM FULL ERR
```

```
ERR * * * MEM FULL ERR
```

就应请老师帮助或者参阅APPLE II BASIC编程手册。

工作前的准备

用磁带录制程序，有时会出现一些较大的故障，从而使你辛辛苦苦编写的一个程序丢失了，这是一件很可惜的事。为了防止这类问题，事先应做些准备工作，以确保程序能正确录制在磁带上。

当你坐到电脑旁准备使用机器时，第一件要做的事，就是检查一下磁带系统工作是否正常。试验方法如下：

1. 从磁带上读入一个短程序放入电脑的内存中；
2. 接着把它从电脑里录制到另一盒磁带上；
3. 然后用NEW命令清除电脑存储器中原来的程序；
4. 再把刚刚录制的程序读入电脑，以验证程序是否确实存在磁带上了。

磁带上不用文件名

用磁带存储程序，不管是录制还是从磁带上读出都不用文

件名，这一点和磁盘的使用不一样，操作时须注意。

错误用法(用磁带) **LOAD SNAKE**

错误用法(用磁带) **SAVE SNAKE**

正确用法 **LOAD**

正确用法 **SAVE**

附录三

保留字表

&
AND AHC AT ATN
CALL CHR\$ CLEAR COLOR= CONT
 COS
DATA DEF DEL DIM DRAW
END EXP
FLASH FN FOR FRE
GET GOSUB GOTO GR
HCOLOR= HGR HGR2 HIMEM, HLIN
 HOME HPLOT HTAB
IF IN# INPUT INT INVERSE
LEFT\$ LEN LET LIST LOAD
 LOG LOMEM,
MID\$
NEW NEXT NORMAL NOT NOTRACE
ON ONERR OR
PDL PEEK PLOT POKE POP
 POS PRINT PR #
READ RECALL REM RESTORE RESUME

	RETURN	RIGHT\$	RND	ROT=
	RUN			
SAVE	SCALE=	SCRN(SGN	SHLOAD
	SIN	SPC(SPEED=	SQR
	STEP	STOP	STORE	STR\$
TAB(TAN	TEXT	THEN	TO
	TRACE			
USR				
VAL	VLIN	VTAB		
WAIT				
XPLOT	XDRAW			

附录四

作业题答案

1-3

```
10 HOME
20 PRINT "HI THERE,"
30 PRINT "APPLE COMPUTER!"
```

2-1

```
10 HOME
20 FLASH
25 PRINT "MINDA"
30 INVERSE
35 PRINT "ANNE"
40 NORMAL
45 PRINT "CARLSON"
```

2-2

```
10 HOME
20 FLASH
22 PRINT CHR$ (7)
25 PRINT "MINDA"
30 INVERSE
32 PRINT CHR$ (7)
```

35 PRINT "ANNE"
40 NORMAL
42 PRINT CHR\$ (7)
45 PRINT "CARLSON"
3-5
15 HOME
16 PRINT
17 PRINT
20 PRINT "....."
25 PRINT "O"
27 PRINT CHR\$ (7)
30 PRINT
32 PRINT
34 PRINT
36 PRINT ""
38 PRINT " O"
40 PRINT CHR\$ (7)
50 PRINT
52 PRINT
54 PRINT
56 PRINT "...O..."
58 PRINT " O"
60 PRINT CHR\$ (7)
4-2
10 HOME

```

20 PRINT
22 PRINT
24 PRINT
26 PRINT
30 PRINT " OO"
32 PRINT " OO"
34 PRINT
36 PRINT
38 PRINT
40 PRINT
42 PRINT " * "
44 PRINT " * "
46 PRINT " * * * * "
48 PRINT " ***** "
5-1
10 HOME
20 PRINT
22 PRINT
24 PRINT
30 PRINT "HELLO, WHAT IS YOUR NAME?"
35 INPUT N$
37 HOME
38 PRINT
40 PRINT "WELL,"
41 PRINT

```

```
42 PRINT N$
45 PRINT
50 PRINT "IT IS SILLY TO TALK TO COMPUTERS."
    5-2
10 HOME
20 PRINT "WHAT IS YOUR FAVORITE COLOR?"
25 INPUT C$
27 PRINT
30 PRINT "I PUT THAT IN BOX C$"
32 PRINT
35 PRINT "NOW YOUR FAVORITE ANIMAL?"
40 INPUT C$
42 PRINT
45 PRINT "I PUT THAT IN BOX C$ TOO"
47 PRINT
50 PRINT "NOW LET'S PRINT WHAT IS IN BOX
    C$"
52 PRINT
55 PRINT "IT IS:"
57 PRINT
60 PRINT C$
    6-1
2 REM BY DAN CLARK, AGE 10
10 HOME
20 PRINT "NAME A MUSICAL GROUP"
```

30 INPUT A\$
40 PRINT
50 PRINT "NAME ONE OF THEIR SONGS"
60 PRINT
70 INPUT B\$
75 PRINT
80 PRINT A\$; "PLAYS"; B\$

6-2

10 HOME
20 PRINT "GIVE ME THE NAME OF A MUSI-
CAL GROUP"

22 INPUT G\$

25 PRINT

30 PRINT "WHAT IS ONE OF THEIR TUNES?"

35 INPUT T\$

40 PRINT

42 PRINT

50 PRINT G\$;

52 PRINT "PLAYS";

54 PRINT T\$

6-3

10 HOME

20 PRINT "YOUR NAME?"

25 INPUT N\$

30 INVERSE

40 SPEED=1
42 PRINT
44 PRINT
46 PRINT
50 PRINT N\$
60 NORMAL
70 SPEED=255

7-2

10 HOME
15 PRINT
17 PRINT
20 PRINT "HOW IS THE WEATHER?"
25 INPUT W\$
28 PRINT
30 PRINT "AND HOW DO YOU FEEL?"
35 INPUT F\$
37 PRINT
40 PRINT "YOU MEAN:"
42 PRINT
45 S\$=W\$+"AND"+F\$
50 PRINT S\$; "?"

8-3

10 HOME
15 LET M\$="MINDA"
20 LET Y\$="KAREN"

30 PRINT M\$

35 SPEED=1

40 PRINT Y\$

50 GOTO 30

9A-2

10 REM BOYS AND GIRLS

20 HOME

22 PRINT

23 PRINT

25 PRINT "ARE YOU A BOY OR A GIRL?"

30 INPUT A\$

32 PRINT

33 PRINT

34 PRINT

35 IF A\$="BOY" THEN PRINT "SNIPS AND SNAILS"

40 IF A\$="GIRL" THEN PRINT "SUGAR AND SPICE"

9B-1

2 REM PIZZA

3 REM BY CHRIS CLARK, JR. AGE 14 GOING ON
(YOU FIGURE IT OUT)

4 HOME

5 SPEED=100

6 PRINT "HALLO. AY AM MARIO, YOUR PIZZA
MAN."

7 PRINT "JUST TELL ME ZE GORY DETAILS

```
      AND I'LL"
8  PRINT "DO ZE REST."
9  PRINT
10 PRINT "WHAT SIZE SHOULD ZIS PIZZA BE
      (S/M/L)?"
20 INPUT S$
30 IF S$="S" THEN PRINT "ON A DIET? HO
      HO!"
31 PRINT
33 IF S$="M" THEN PRINT "GOOD CHOICE-NOT
      TOO BIG, BUT FILLING!"
35 PRINT
38 IF S$="L" THEN PRINT "YOU MUST HAVE
      A BIG BUNCH AT HOME!"
39 PRINT
40 PRINT "NOW, YOU WANT DOUBLE CHEES ON
      ZIS (Y/N)?"
42 INPUT CH$
45 REM ETC.
50 REM MUSHROOMS, ETC.
60 REM ANCHOVIES, ETC.
80 REM PEPPERS, ETC.
90 REM MEAT, ETC.
150 INVERSE
151 PRINT "HOKAY, HERE IS YOUR PIZZA!"
```

```
152 NORMAL
153 PRINT
154 IF S$="S" THEN PRINT "WAN SMALL PIZZA
    WITH";
155 REM ETC.
160 IF BASE$="P" THEN PRINT "PEPPERONI"
165 REM ETC., ETC.
238 PRINT
240 FOR J=1 TO 2000
242 NEXT J
244 SPEED=255
```

9B-2

```
10 REM===COLOR GUESSING GAME===
20 HOME
22 PRINT
23 PRINT
24 PRINT
25 PRINT "PLAYER 2 TURN YOUR BACK"
27 PRINT
30 PRINT "PLAYER 1 ENTER A COLOR"
35 INPUT C$
40 HOME
42 PRINT
43 PRINT
44 PRINT
```

```
50 PRINT "PLAYER 2 TURN AROUND AND GUESS"  
52 PRINT  
53 PRINT  
54 PRINT  
55 INPUT G$  
60 IF G$ < > C$ THEN PRINT "WRONG!"  
65 IF G$=C$ THEN PRINT "RIGHT"  
67 PRINT  
70 GOTO 55  
    10-1  
10 REM BIRTH YEAR  
20 HOME  
30 PRINT "HOW OLD ARE YOU?"  
32 PRINT  
34 INPUT A  
36 PRINT  
40 PRINT "AND WHAT YEAR IS IT NOW?"  
50 INPUT Y  
52 B=Y - A  
55 PRINT "HAS YOUR BIRTHDAY COME YET THIS  
    YEAR?"  
56 PRINT "<Y/N>"  
60 INPUT Y$  
65 IF Y$="N" THEN B=B - 1  
67 PRINT
```

70 PRINT "YOU WERE BORN IN "; B; ". "

10-2

10 REM MUL TIPLICATION

20 HOME

22 PRINT

23 PRINT

24 PRINT

30 PRINT "GIVE ME A NUMBER"

35 INPUT A

37 PRINT

38 PRINT

40 PRINT "GIVE ME ANOTHER"

45 INPUT B

48 $C = A * B$

50 PRINT

52 PRINT

60 PRINT "THEIR PRODUCT IS"; C

11A-1

10 HOME

12 PRINT

13 PRINT

14 PRINT

15 PRINT "THIS IS A PARTY GAME"

16 PRINT

17 PRINT

```

20 PRINT "WHAT IS YOUR NAME?"
22 PRINT
24 PRINT
25 INPUT F$
27 PRINT
30 PRINT "PLEASE TURN YOUR BACK,"; F$
31 FOR I=1 TO 5000; NEXT I
32 HOME
34 PRINT "SOMEONE PRINT IN A NICKNAME";
36 INPUT N$
40 HOME
42 PRINT
44 PRINT
50 PRINT CHR$ (7); REM BELL
55 PRINT F$; TAB (10);"IS CALLED"; N$;" "
60 GOTO 12
      11A-2
10 REM! ☐ * # $ INSULTS $ # ☐!
12 HOME
14 PRINT
16 PRINT
20 PRINT "HEY YOU! WHAT IS YOUR NAME?"
21 PRINT
22 PRINT
23 INPUT N$

```

```

25 REM DELAY LOOP
30 FOR T=1 TO 2000
35 HOME
36 PRINT CHR$ (7)
38 PRINT
40 PRINT CHR$ (7)
42 PRINT
44 PRINT CHR$ (7)
46 PRINT
48 PRINT CHR$ (7)
50 PRINT "BAH!!!"
52 PRINT
55 PRINT TAB(10); N$
57 PRINT
60 PRINT TAB(15);"YOUR FATHER EATS LEEKS!!!"
    11B
2 REM "PATIENCE" BY DAN CLARK, AGE 10
10 HONE
15 SPEED=2
18 PRINT CHR$ (7)
20 PRINT "YOUR . . ."
22 PRINT
23 FOR I=1 TO 2000; NEXT I
25 PRINT CHR$ (7)
30 PRINT ". . . WAIT'S . . ."

```

```
31 PRINT
32 FOR I=1 TO 2000: NEXT I
35 PRINT CHR$ (7)
40 PRINT "    ...OVER!!"
50 SPEED=255
    12-5
10 REM I GOT YOUR NUMBER!
20 HOME
25 PRINT
26 PRINT
27 PRINT
30 PRINT "GIVE ME A NUMBER BETWEEN ZERO
    AND TEN."
35 PRINT
40 INPUT N
45 PRINT
46 PRINT
50 IF N=0 THEN PRINT "I GOT PLENTY OF
    NOTHING!"
51 IF N=1 THEN PRINT "I'M NUMBER ONE!"
52 IF N=2 THEN PRINT "TWO IS COMPANY."
53 REM ETC.
70 IF N>10 THEN GOTO 99
90 GOTO 25
99 PRINT "THAT'S ALL FOLKS!"
```


13-1

```
10 REM ** A PAIR OF DICE **
15 HOME
20 LET D1=1+INT (RND (8) * 6)
22 LET D2=1+INT (RND (8) * 6)
25 D=D1+D2
30 PRINT "THE ROLL GAVE:"
32 PRINT
35 PRINT TAB(10); "THE FIRST DIE"; D1
40 PRINT TAB(10); "THE SECOND"; D2
45 PRINT TAB(10); "THE DICE";D
47 PRINT
48 PRINT
50 PRINT "AGAIN? Y/N";
55 INPUT Y$
57 PRINT
58 PRINT
60 IF Y$="Y" THEN GOTO 20
```

13-2

```
10 REM PAPER, SCISSORS AND ROCK
12 HOME
13 PRINT
14 PRINT
15 PRINT
16 PRINT "PLAY THE"
```

```

17 PRINT
18 PRINT
19 PRINT TAB(5) "PAPER"
20 PRINT TAB(15) "SCISSORS"
21 PRINT TAB(31) "ROCK"
22 PRINT
23 PRINT "GAME AGAINST THE COMPUTER"
24 PRINT
25 PRINT "PRESS 'RESET' KEY TO END GAME"
26 PRINT
27 PRINT "ENTER YOUR CHOICE, <P, S, R>"
29 REM COMPUTER CHOOSES
30 C=INT (RND (8) * 3) + 1
32 IF C=1 THEN C$="P"
33 IF C=2 THEN C$="S"
34 IF C=3 THEN C$="R"
35 REM C$ IS THE COMPUTERS CHOICE
37 INPUT Y$
38 REM Y$ IS YOUR CHOICE
39 :
45 REM IS THERE A TIE?
46 :
50 IF C$ < > Y$ THEN GOTO 60
52 REM IF C$=Y$ THERE IS A TIE
55 PRINT "TIE"

```

```

57 GOTO 30
59 :
60 REM NO TIE, WHO WINS?
61 :
62 IF C$="P" THEN IF Y$="S" THEN GOTO 70
63 IF C$="S" THEN IF Y$="R" THEN GOTO 70
64 IF C$="R" THEN IF Y$="P" THEN GOTO 70
65 REM COMPUTER WINS
66 PRINT "COMPUTER WINS"
69 GOTO 30
70 REM YOU WIN
72 PRINT "YOU WIN"
79 GOTO 30
90 REM END THE GAME BY PRESSING
92 REM THE "RESET" KEY
  OUTLINE 15-2
10 REM !!! VACATION !!!
12 HOME
20 REM HEADING
30 REM INSTRUCTIONS
50 REM CHOICE OF VACATIONS
90 REM ENDING OF PROGRAM
  PROGRAM 15-2
10 REM !!! VACATION !!!
12 HOME

```

13 PRINT
14 PRINT
15 PRINT
20 REM HEADING
21 PRINT "VACATION CHOOSING PROGRAM"
22 PRINT
23 PRINT "PICKS YOUR VACATION BY THE"
24 PRINT AMOUNT YOU WANT TO SPEND"
25 PRINT
30 REM INSTRUCTIONS
31 PRINT "ENTER THE AMOUNT IN DOLLARS
THAT"
32 PRINT "YOU CAN SPEND."
33 PRINT
35 REM GET DOLLAR AMOUNT
37 INPUT D
40 M\$="FLIP PENNIES WITH YOUR KID BROTHER."
43 P\$="SPEND THE AFTERNOON IN BEAUTIFUL
HOG WALLOW, MICH."
45 Q\$="ENTER A PICKLE EATING CONTEST IN
SCRATCHY BACK, TENN."
47 REM ETC.
49 REM ETC.
58 Z\$="BUY A COSY YACHT AND CRUISE THE
CARIBBEAN SEA."

```
70 IF D<0.49 THEN PRINT M$, GOTO 90
72 IF D<0.99 THEN PRINT P$, GOTO 90
74 REM ETC.
76 REM ETC.
86 IF D<1000000 THEN PRINT Z$, GOTO 90
88 PRINT "TREAT YOUR WHOLE SCHOOL TO A
ROUND THE WORLD TRIP!"
90 REM ENDING OF PROGRAM
```

16-1

```
10 REM RANDOM NAME
20 HOME
30 INPUT "YOUR NAME?"; N$
40 X=RND (8) * 35
50 Y=1+RND (8) * 19
60 HTAB (X)
70 VTAB (Y)
80 PRINT N$
90 GOTO 40
99 END
```

16-2

```
10 REM JUMPING "HERE"
20 HOME
30 X=RND (8) * 35 + 1
35 Y=RND (8) * 22 + 1
50 VTAB (Y)
```

```

60 HTAB (X)
70 PRINT "HERE";
75 FOR I=1 TO 500: NEXT
80 HTAB (X)
90 PRINT "      "
999 GOTO 30

      17A
10 REM COUNTING BY FIVES
15 SPEED=1
20 FOR I=5 TO 100 STEP 5
30 PRINT I
40 NEXT I
99 SPEED=255

      17B-2
10 HOME
20 REM SLIPPING NAME
25 N$="BRIAN"
27 SPEED=200
30 FOR I=1 TO 15
35 PRINT TAB(I*2); N$
40 NEXT I
99 SPEED=255

      17B-3
10 REM CLIMBING NAME
12 HOME

```

```
15 N$ = "STANLAUS MAZURSKI"  
17 SPEED = 225  
20 FOR I = 23 TO 1 STEP - 1  
25 VTAB I  
27 HTAB I  
30 PRINT N$  
40 NEXT  
90 HOME  
99 SPEED = 255
```

17B-4

```
10 REM + + + FRIENDS NAMES + + +  
12 HOME  
15 SPEED = 100  
20 INPUT "NAME?"; N$  
22 PRINT  
25 INPUT "OTHER NAME?"; M$  
30 FOR I = 1 TO 5  
35 FLASH  
36 PRINT  
40 PRINT N$  
45 INVERSE  
46 PRINT  
50 PRINT M$  
60 NEXT I  
98 SPEED = 255
```

99 NORMAL

19-2

```
10 REM SLIDING NAME
12 HOME
20 INPUT "YOUR NAME?"; N$
25 VTAB 15; HTAB 1
30 SPEED=225
40 FOR I=1 TO 30
45 N$=" " + N$
50 PRINT N$;
55 HTAB 1
60 NEXT I
99 SPEED=255
```

19-3

```
10 REM FALLING FRIEND
12 HOME
20 INPUT "YOUR FRIEND'S NAME."; F$
25 B$=" "
29 HOME
30 FOR I=1 TO 23
31 PRINT B$
32 HTAB 15
34 VTAB I
40 PRINT F$;
42 HTAB 15
```



```

45 FOR T=1 TO 100, NEXT T
49 NEXT I
      19-4
10 REM CROSSING FRIENDS
12 HOME
20 INPUT "YOUR NAME"; Y$
22 PRINT
24 INPUT "YOUR FRIEND'S NAME"; F$
26 B$="      "
28 HOME
35 :
36 REM MAIN LOOP
37 :
40 FOR I=1 TO 23
45 IF I>16 THEN GOTO 60
50 HTAB 2 * I
51 VTAB 12
53 PRINT B$;
55 HTAB 2 * I+2
57 PRINT Y$;
60 HTAB 12; VTAB I
65 PRINT B$;
70 HTAB 12; VTAB I+1
75 PRINT F$;
90 NEXT I

```

```
91 :  
92 REM DELAY AT END  
93 :  
95 FOR T=1 TO 2000; NEXT T  
99 HOME
```

21-2

```
100 REM MAKE A STAR  
105 GR  
110 COLOR=13  
120 FOR I=15 TO 25  
125 PLOT 20, I  
130 PLOT I, 20  
140 NEXT I  
148 COLOR=11  
150 FOR I=16 TO 24  
155 PLOT I, I  
158 J=40-I  
160 PLOT J, I  
170 NEXT I  
180 FOR T=1 TO 3000; NEXT T  
195 TEXT  
197 HOME
```

22-1

```
10 REM DRAW A BOX  
12 HOME
```

```

13 PRINT, PRINT, PRINT
20 PRINT "WHAT COLOR SQUARE? <1-15>"
22 INPUT C
30 GR: COLOR=C
40 HLIN 10, 30 AT 10
42 HLIN 10, 30 AT 39
44 VLIN 10, 39 AT 10
46 TLIN 10, 39 AT 30
60 FOR T=1 TO 5000, NEXT T
95 TEXT
97 HOME

          21-3
100 REM SINBAD'S MAGIC CARPET
101 :
102 GOTO 1000
200 :
201 REM MAIN LOOP
202 :
205 X=1
206 A=RND (8), B=RND (8)
207 C=RND (8)
210 FOR I=0 TO 19, FOR J=0 TO 19
212 K=I+J
214 COLOR=C(X)
215 PRINT C(X)

```

```

216 X=X+A*(I+3*B)/(J+3)+C*(20-I-J)/53
217 REM CHANGE LINE 216 TO GET A VARIETY OF PATTERNS
218 IF X>NC * 1 THEN X=X-NC; GOTO 218
219 IF X<1 THEN X=1
230 PLOT I, K; PLOT K, I
232 PLOT 39-I, K; PLOT K, 39-I
234 PLOT 39-K, I; PLOT I, 39-K
236 PLOT 39-I, 39-K; PLOT 39-K, 39-I
290 NEXT J, I
999 END
1000 :
1001 REM <<< SINBAD'S MAGIC CARPET >>>
1002 :
1080 DIM C(15)
1090 GR
2000 :
2001 REM INPUT COLORS
2002 :
2010 INPUT "NUMBER OF COLORS IN THE RUG";
NC
2020 FOR I=1 TO NC
2025 INPUT "COLOR? <0-15>"; C
2030 C(I)=C
2032 FOR J=1 TO C; X=RND (8); NEXT J

```

```
2033 REM THIS CHANGES THE RANDOM NUMBER
      FROM ONE RUG TO THE NEXT
2035 NEXT I
2040 REM SOME OF THE PRETTIEST RUGS HAVE
      ONLY 2 COLORS
9999 GOTO 200
      22-2
10 REM DRAW A DIE
12 HOME
13 PRINT, PRINT, PRINT
20 PRINT "WHAT COLOR SQUARE? <1=15>"
22 INPUT C
30 GR, COLOR=C
40 HLIN 10, 30 AT 10
42 HLIN 10, 30 AT 39
44 VLIN 10, 39 AT 10
46 VLIN 10, 39 AT 30
100 :
101 REM ADD SPOTS
102 :
110 PRINT "HOW MANY SPOTS?"
115 INPUT NS
118 COLOR=14
120 IF NS=1 THEN PLOT 20, 25
125 IF NS=2 THEN PLOT 13, 14, PLOT 27, 35
```

```
130 IF NS=3 THEN PLOT 20, 25; PLOT 13, 14;  
    PLOT 27, 35  
135 IF NS=4 THEN PLOT 13, 35; PLOT 27, 14;  
    NS=2; GOTO 120  
140 IF NS=5 THEN PLOT 20, 25; NS=4; GOTO  
    120  
145 IF NS=6 THEN PLOT 20, 14; PLOT 20, 35;  
    NS=4; GOTO 120  
160 FOR T=1 TO 3000, NEXT T  
195 TEXT  
197 HOME
```

23-1

```
9   :  
10  REM MENU MAKER  
11  :  
12  HOME  
15  PRINT; PRINT; PRINT  
19  :  
20  PRINT "WHICH COLOR DO YOU LIKE?"  
21  PRINT  
22  PRINT " <M> MAGENTA"  
24  PRINT " <Y> YELLOW"  
26  PRINT " <G> GREEN"  
28  PRINT " <B> BLUE"  
29  PRINT
```

```

30 :
31 GET C$
32 :
35 IF C$="M" THEN C=1
36 IF C$="Y" THEN C=13
37 IF C$="G" THEN C=12
38 IF C$="B" THEN C=6
39 :
40 COLOR=C
50 :
51 REM MAKE A STAR AS IN THE PROBLEM
    2 OF ASSIGNMENT 21
52 :
        23=2
10 REM SILLY SENTENCES
12 HOME
13 PRINT, PRINT, PRINT
14 PRINT "SILLY SENTENCES"; PRINT, PRINT
15 PRINT "WANT INSTRUCTIONS? <Y/N>"; PRINT
16 GET Y$; IF Y$="Y" THEN GOTO 100
18 PRINT , PRINT
20 PRINT "THE SUBJECT; (END WITH A PERIOD)";
    PRINT
22 GET L$
24 IF L$="." THEN GOTO 30

```

```

28 S$=S$ + L$
29 GOTO 22
30 S$=S$ + " "; REM ADD A SPACE AFTER THE
    LAST WORD
32 PRINT "THE VERB; (END WITH A PERIOD)";
    PRINT
34 GET L$
36 IF L$="." THEN GOTO 42
38 S$=S$ + L$
40 GOTO 34
42 S$=S$ + " "
50 PRINT "THE OBJECT; (END WITH A PERIOD)";
    PRINT
52 GET L$
54 IF L$="." THEN GOTO 70
56 S$=S$ + L$
58 GOTO 52
70 S$=S$ + "."
80 PRINT
85 PRINT S$
99 END
100 :
101 REM INSTRUCTINS
102 :
104 HOME

```



```
110 PRINT "THREE PLAYERS ENTER PARTS OF  
A SENTENCE"; PRINT  
115 PRINT "NO PLAYER CAN SEE WHAT THE  
OTHERS ENTER"; PRINT  
120 PRINT "THE FIRST ENTERS THE SUBJECT";  
PRINT  
122 PRINT "(THE PERSON DOING SOMETHING)";  
PRINT  
130 PRINT "THE SECOND ENTERS THE VERB";  
PRINT  
132 PRINT "(THE ACTION WORD)"; PRINT  
140 PRINT "THE THIRD ENTERS THE OBJECT";  
PRINT  
142 PRINT "(THE PERSON OR THING TO WHOM";  
PRINT  
144 PRINT "THE ACTION IS DONE)"; PRINT  
148 FOR T=1 TO 5000; NEXT T  
150 GOTO 18
```

25-2

```
10 REM APPLE TREE  
12 HOME; GR  
20 REM MAIN LOOP  
25 GOSUB 200; REM SKY  
30 GOSUB 300; RNM MAKE TREE  
40 GOSUB 400; REM MAKE APPLES
```

```
50 GOSUB 500; REM APPLES FALL
99 END
200 REM
201 REM BACKGROUND
202 REM
205 COLOR=6; REM MAKE APPLES
210 FOR I=0 TO 39
215 HLIN 0, 39 AT I
220 NEXT I
300 REM
301 REM MAKE TREE
302 REM
310 REM GROUND
312 COLOR=10; REM GREY
314 HLIN 0, 39 AT 39
315 HLIN 0, 39 AT 38
320 REM TRUNK
322 COLOR=8; REM DARK GREEN
324 VLIN 25, 38 AT 19
326 VLIN 25, 38 AT 20
330 REM GREEN PART
332 COLOR=4
333 HLIN 17, 21 AT 25
334 HLIN 15, 25 AT 24
335 HLIN 13, 27 AT 23
```

```
336 HLIN 10, 30 AT 22
337 HLIN 9, 31 AT 21
338 HLIN 8, 32 AT 20
339 HLIN 9, 32 AT 19
340 HLIN 8, 31 AT 18
341 HLIN 10, 29 AT 17
342 HLIN 11, 29 AT 16
343 HLIN 13, 26 AT 15
344 HLIN 12, 26 AT 14
345 HLIN 14, 24 AT 13
346 HLIN 16, 23 AT 12
347 HLIN 17, 21 AT 11
390 RETURN
400 REM
401 REM MAKE APPLES
402 REM
405 COLOR=9, REM ORANGE
408 FOR K=1 TO 40
409 REM PUT APPLES IN A RECTANGLE
410 X=INT (RND (8) * 28) + 8
415 Y=INT (RND (8) * 25) + 10
420 L=SCRN (X, Y)
424 REM ONLY PUT APPLES ON GREEN TREE
425 IF L=4 THEN PLOT X, Y
430 NEXT K
```

```
490 RETURN
500 REM
501 REM FALLING APPLES
502 REM
509 REM LOOK IN RECTANGLE FOR APPLES
510 FOR K=1 TO 1000
515 X=INT (RND (8) * 28)+8
520 Y=INT (RND (8) * 25)+10
530 L=SCRN(X, Y)
534 REM IF FIND ORANGE APPLE, DROP IT
535 IF L=9 THEN GOSUB 600
540 NEXT K
590 RETURN
600 REM
601 REM DROP THE APPLE
602 REM
610 LL=4; REM GREEN
620 COLOR=LL; PLOT X, Y; REM COVER EMPTY
    SPOT
625 Y=Y+1; REM LOOK BELOW APPLE
630 L=SCRN(X, Y)
634 REM IF SEE GROUND, PLOT APPLE AND
    RETURN
635 IF L=10 THEN COLOR=9; PLOT X, Y - 1;
    RETURN
```

```

640 LL=L, REM COLOR TO COVER EMPTY SPOT
645 COLOR=9; PLOT X, Y; REM APPLE ONE
    SQUARE LOWER
690 GOTO 620
    26-1
10  REM CIPHER MAKER
12  HOME
13  PRINT
14  PRINT
20  PRINT "CODE MAKING PROGRAM"
21  PRINT
22  PRINT
30  PRINT "ENTER A SENTENCE FOR CODING;"
31  PRINT
32  INPUT S$
33  PRINT
35  L=LEN (S$)
40  FOR I=1 TO L STEP 2
45  P$=MID$ (S$, I, 2)
50  Q$=RIGHT$ (P$, 1)+LEFT$ (P$, 1)
55  L$=L$+Q$
60  NEXT I
65  PRINT
66  PRINT "HERE IS THE CODED SENTENCE:"
67  PRINT

```

```

70 PRINT L$
      26-2
10 REM QUESTION ANSWERER
12 HOME
13 PRINT, PRINT, PRINT
20 PRINT "ENTER A QUESTION"
21 PRINT
25 INPUT Q$
28 L=LEN(Q$)
30 :
31 REM TAKE OFF THE QUESTION MARK
32 :
33 Q$=LEFT$(Q$, L - 1) + "."
34 PRINT
35 :
36 REM LOOK FOR THE END OF THE FIRST WORD
38 :
39 FOR I=1 TO L
40 C$=MID$(Q$, I, 1)
45 IF C$=" " THEN S1=I:I=L
46 NEXT I
47 :
48 REM LOOK FOR THE END OF THE SECOND
   WORD
49 :

```

```

50 FOR I=S1+1 TO L
52 C$=MID$ (Q$, I, 1)
54 IF C$=" " THEN S2=I:I=L
56 NEXT I
57 :
58 REM TURN THE WORDS AROUND
59 :
60 S$=MID$ (Q$, S+1+1, S2 - S1)
62 V$=LEFT$ (Q$, S1)
65 PRINT S$+V$+RIGHT$ (Q$, L - S2)

```

26-3

```

10 REM * * * PIG LATIN * * *
12 HOME
13 PRINT; PRINT; PRINT
20 PRINT "THIS IS A PIG LATIN PROGRAM"
21 PRINT
25 PRINT
30 INPUT "GIVE ME A WORD."; W$
32 PRINT; PRINT
40 F$=LEFT$ (W$, 1)
42 E$=F$+ "AY"
44 IF F$="A" THEN E$="LAY"
46 IF F$="E" THEN E$="LEE"
48 IF F$="I" THEN E$="LIE"
50 IF F$="O" THEN E$="LO"

```

```

52 IF F$="U" THEN E$="LU"
70 L$=RIGHT$ (W$, LEN (W$) - 1)+E$
80 PRINT L$
85 PRINT; PRINT
90 GOTO 30

          27-1

10 REM BACKWARD ADDED TO FORWARD
12 HOME
13 PRINT; PRINT; PRINT
20 INPUT "GIVE ME A NUMBER;"; N
30 N$=STR$ (N)
35 L=LEN (N$)
40 FOR I=1 TO L
45 B$=B$+MID$ (N$, L+1 - I, 1)
50 B=VAL (B$)
55 NEXT I
57 PRINT ; PRINT ; PRINT ; PRINT ; PRINT
60 PRINT " "; N
61 PRINT "+"; B
65 PRINT " "; LEFT$ (".....", L+1)
70 A=N+B
72 A$=STR$ (A)
75 IF LEN (A$)=L THEN PRINT " "; A
76 IF LEN (A$)=L+1 THEN PRINT " "; A
99 END

```


27-2

```
10 REM MARCHING NUMBERS
12 HOME
13 PRINT ; PRINT ; PRINT
20 INPUT "GIVE ME A NUMBER"; N
22 B$=" "
25 N$=STR$ (N)
27 L=LEN (N$)
28 VTAB 15
31 :
33 REM LOOP
34 :
40 FOR I=1 TO 39 = L
45 HTAB I
50 PRINT " ";
55 HTAB I+1
60 PRINT N$;
62 FOR T=1 TO 200; NEXT T
65 N$=RIGHT$ (N$, L = 1)+LEFT $ (N$, 1)
70 NEXT I
80 :
82 REM END
84 :
99 SPEED=255
```

28-1

```
200 REM SHOOT
202 REM ADD TO SHOOT PROGRAM
210 FOR I=38 TO 0 STEP - 1
230 COLOR=12
240 PLOT X, I
250 COLOR=0
260 PLOT X, I
290 NEXT I
299 RETURN
```

28-2

```
10 REM * * * SHOOTING STARS * * *
20 HOME : GR
30 GOSUB 300
40 PRINT "USE PADDLE 0"
50 Y=39
55 COLOR=0
60 PLOT X, Y
65 X=PDL (0) * 39/255
67 COLOR=7
70 PLOT X, Y
75 IF PEEK ( - 16287) > 127 THEN GOSUB 200
99 GOTO 55
200 REM SHOOT
210 FOR I=38 TO 0 STEP - 1
212 CS=SCRN(X, I)
```

```
215 IF CS=1 THEN PRINT CHR$(7); CS=0
217 COLOR=12
220 PLOT X, I
230 COLOR=CS
240 PLOT X, I
280 NEXT I
299 RETURN
300 REM INITIALIZE STARS
310 FOR I=1 TO 10
320 X=RND (8) * 40
330 Y=RND (8) * 20
340 COLOR=10
342 IF RND (8) < .5 THEN COLOR=1
345 PLOT X, Y
350 NEXT I
399 RETURN
```

29-1

```
10 REM ALPHABETICAL
12 HOME
13 PRINT : PRINT : PRINT
19 :
20 PRINT "THIS PROGRAM ARRANGES THE LET-
TERS"
21 PRINT "OF A WORD IN ALPHABETICAL OR-
DER."
```

```
22 :
23 PRINT
29 :
30 INPUT "GIVE ME A WORD:"; W$
31 :
32 PRINT
35 L=LEN (W$)
37 :
38 REM TEST LETTERS IN ALPHABET
39 :
40 FOR I=65 TO 65+26
41 :
42 REM TO SEE IF IN WORD
43 :
45 FOR J=1 TO L
50 G=ASC (MID$ (W$, J, 1))
55 IF G=I THEN H$=H$+CHR$ (G)
60 NEXT J; NEXT I
61 :
70 PRINT
75 PRINT "HERE IT IS IN ALPHABETICAL OR-
    DER;"
76 PRINT
80 PRINT H$
```

```

10 REM   □□□ DOUBLE DUTCH   □□□
11 :
12 HOME ; PRINT ; PRINT ; PRINT ; PRINT ;
    PRINT
20 :
21 REM INPUT THE SENTENCE
22 :
25 PRINT "GIVE ME A SENTENCE."; PRINT
30 INPUT S$; PRINT
40 L=LEN (S$)
42 HOME ; PRINT ; PRINT ; PRINT ; PRINT;
    PRINT
46 :
47 REM CHECK EACH LETTER
48 :
50 FOR I=1 TO L
52 L$=MID$ (S$, I, 1); REM GET A LETTER
60 IF L$="A" THEN GOTO 72
62 IF L$="E" THEN GOTO 72
64 IF L$="I" THEN GOTO 72
66 IF L$="O" THEN GOTO 72
68 IF L$="U" THEN GOTO 72
69 SS$=SS$+L$
72 NEXT I
73 :

```

```

74 REM PRINT THE SENTENCE IN DOUBLE DU-
    TCH
75 :
76 PRINT "HERE IT IS IN DOUBLE DUTCH"
77 PRINT
80 PRINT SS$
    29-3
10 REM ON. . : GOTO SAMPLE
12 HOME
13 PRINT : PRINT : PRINT
20 :
21 REM MAKE A MENU
22 :
30 PRINT "MAKE YOUR CHOICE;"
31 PRINT : PRINT "<A> TAKE A NAP"
32 PRINT : PRINT "<B> EAT AN APPLE"
33 PRINT : PRINT "<C> CALL A FRIEND"
40 PRINT : INPUT X$; PRINT
41 X=ASC (X$) = 64
49 :
50 ON X GOTO 60, 70, 80
51 :
52 GOTO 30
59 :
60 PRINT "YOUR BED IS NOT MADE!"

```

```
61 END
70 PRINT "YOUR SISTER ATE THE LAST ONE!"
71 END
80 PRINT "YOUR FATHER IS ON THE PHONE!"
81 END
```

30-1

```
9 :
10 REM PHONE LIST KEEPER
11 REM RUN 1000 TO INITIALIZE PHONE LIST
13 D$=CHR$ (4)
14 DIM NA$(20), TE$(20)
15 :
16 REM MENU
17 :
18 GOSUB 800
20 HOME
21 PRINT : PRINT : PRINT "PHONE LIST": PRINT
22 PRINT "<A> ADD NAMES AND NUMBERS":
PRINT
24 PRINT "<L> LIST THE NAMES AND NUMBERS":
PRINT
25 PRINT " <E> END THE PROGRAM": PRINT
26 PRINT " <I> INSTRUCTIONS": PRINT
27 GET Y$
28 IF Y$="A" THEN GOTO 100
```

```

30 IF Y$="L" THEN GOTO 300
31 IF Y$="E" THEN END
32 IF Y$="I" THEN GOSUB 900
35 GOTO 27
95 END
100 :
101 REM ADD NEW NAMES AND NUMBERS
102 :
105 HOME
110 INPUT "ENTER NAME:"; A$
115 PRINT
120 INPUT "ENTER NUMBER:"; T$
130 FOR I=1 TO 20
132 IF NA$(I)=" " THEN Z=I:I=20
134 NEXT I
140 NA$(Z)=A$; TE$(Z)=T$
190 GOSUB 500
193 HOME
195 GOTO 20
300 :
301 REM LOOK UP A NUMBER
302 :
305 HOME
310 FOR I=1 TO 20
320 PRINT NA$(I); TAB(25); TE$(I)

```



```

325 IF NA$(I)=" " THEN I=20
330 NEXT I
390 PRINT "PRESS <M> FOR RETURN TO THE
MENU"; GET A$
395 GOTO 20
500 :
501 REM PUT THE PHONE LIST ON DISK
502 :
515 PRINT D$; "OPEN PHONE LIST"
520 PRINT D$; "WRITE PHONE LIST"
530 FOR I=1 TO Z+1
535 PRINT NA$(I)
536 PRINT TE$(I)
540 NEXT I
545 PRINT D$; "CLOSE PHONE LIST"
595 RETURN
800 :
801 REM LOAD PHONE LIST FROM DISK
802 :
810 PRINT D$;"OPEN PHONE LIST"
811 PRINT D$;"READ PHONE LIST"
815 FOR I=1 TO 20
820 INPUT NA$(I)
821 INPUT TE$(I)
825 IF NA$(I)=" " THEN Z=I;I=20

```

```
830 NEXT I
840 PRINT D$;"CLOSE PHONE LIST"
895 RETURN
900 REM
995 RETURN
1000 :
1001 REM INITIALIZE DISK FILE "PHONE LIST"
1002 :
1003 D$=CHR$ (4)
1004 PRINT D$;"OPEN PHONE LIST"
1005 PRINT D$;"WRITE PHONE LIST"
1007 DIM NA$(20), TE$(20)
1010 FOR I=1 TO 20
1015 PRINT NA$(I)
1016 PRINT TE$(I)
1020 NEXT I
1040 PRINT D$; "CLOSE PHONE LIST"
1095 END
```

31-3

```
2 :
10 REM "AIN'T GOT NO"
11 :
12 HOME
13 PRINT ; PRINT ; PRINT
20 :
```

```

21 PRINT "SEARCH FOR DOUBLE NEGATIVES";
   PRINT
22 :
30 PRINT "ENTER A SENTENCE,"; PRINT
31 :
32 INPUT S$; PRINT
33 S$=S$+" "; REM END THE SENTENCE WITH
   A BLANK SPACE.
35 L=LEN (S$)
40 NM=0; REM NUMBER OF NEGATIVE WORDS
42 S1=1;S2=1; REM START LETTER OF TWO
   WORDS
45 FOR I=1 TO L
50 L$=MID$ (S$, I, 1); REM GET A LETTER
54 REM IS IT A SPACE?
55 IF L$=" " THEN S1=S2;S2=I+1; GOSUB 200
60 NEXT I
65 PRINT
70 IF NN=2 THEN PRINT "THIS SENTENCE HAS
   A DOUBLE NEGATIVE."
72 IF NN=0 THEN PRINT "THIS SENTENCE AIN'T
   GOT NO DOUBLE NEGATIVE!"
74 IF NN=4 THEN PRINT "THIS SENTENCE HAS
   TWO DOUBLE NEGATIVES."
100 :

```

```

101 REM TEST THE PROGRAM ON THE SENTEN-
    CE;
102 :
103 REM I NEVER EAT NO JUNK FOOD!
104 :
198 END
199 :
200 REM IS THE WORD A NEGATIVE?
201 :
205 LW=S2 - S1 - 1; REM LENGTH OF THE
    WORD
210 W$=MD$ (S$, S1, LW)
220 PRINT W$
232 IF W$="NO" OR W$="NOT" OR W$="DON'T"
    THEN NN=NN+1
234 IF W$="NOTHING" OR W$="NEVER" OR
    W$="AIN'T" THEN NN=NN+1
240 REM ETC.
295 RETURN

```

32-2

```

1 GOTO 1000; REM * * * CODE_DECODE * * *
2 :
100 REM MAIN LOOP
110 REM
113 GOSUB 400; REM GET PASSWORD

```

```

115 PRINT, PRINT "CODE OR DECODE? <C/D>";
    GET Y$
120 IF Y$ = "C" THEN GOTO 500; REM CODE
    MESSAGE
130 IF Y$ = "D" THEN GOTO 600; REM DECODE
    MESSAGE
140 GOTO 115
199 END
399 :
400 REM GET PASSWORD AND FORM CODE AL-
    PHABET
401 :
402 HOME : PRINT : PRINT
405 INPUT "INPUT PASSWORD"; PW$
406 REM REMOVE REPEATED LETTERS FROM
    THE PASSWORD
408 F$=LEFT$ (PW$, 1)
410 FOR I=2 TO LEN (PW$)
411 L1$=MID$ (PW$, I, 1)
412 FOR J=1 TO LEN (F$)
415 L2$=MID$ (F$, J, 1)
420 IF L1$=L2$ THEN GOTO 430
421 NEXT J
422 F$=F$+L1$
430 NEXT I

```

```

432 PW$=F$
433 PRINT ; PRINT ; "THE SHORTENED PASS-
    WORD IS"; PW$
434 :
435 REM REMOVE PASSWORD LETTERS FROM
    THE ALPHABET
436 :
440 FOR J=1 TO LEN (PW$); L2$=MID$ (PW$,
    J, 1)
441 IF L2$=LEFT$ (A$, 1) THEN A$=MID$ (A$,
    2); GOTO 460
442 FOR I=2 TO LEN (A$); L1$=MID$ (A$, I,
    1)
445 IF L1$=L2$ THEN A$=LEFT$ (A$, I-1)+
    MID$ (A$, I+1)
455 NEXT I
460 NEXT J
461 :
462 REM FORM CODE ALPHABET
463 :
465 A$=PW$+A$
470 PRINT ; PRINT "THE CIPHER ALPHABET
    IS"; PRINT ; PRINT A$; PRINT B$
498 RETURN
499 :

```

```

500 REM FORM A CODED MESSAGE
501 :
505 PRINT ; PRINT "INPUT MESSAGE, END WITH
    ' * ' SIGN"; PRINT
510 GET L$,L=ASC (L$)
515 IF L$="*" THEN GOTO 590
520 IF L <65 OR L > 91 THEN P$=P$+L$: GOTO
    540
530 P$=P$+MID$ (A$, L ÷ 64, 1)
540 PRINT L$;
589 GOTO 510
590 PRINT ; PRINT P$
598 END
599 :
600 REM DECODE A MESSAGE
601 :
610 PRINT ; PRINT "TYPE IN THE CODED ME-
    SSAGE"
612 PRINT "END THE MESSAGE WITH A ' * '
    SIGN"
613 PRINT ; PRINT
615 GET L$
617 IF L$="*" THEN GOTO 690
620 FOR I=1 TO 26
625 IF L$=MID$ (A$, I, 1) THEN PRINT MID$

```

```
(B$, I, 1); GOTO 615
630 NEXT I
635 PRINT L$;
640 GOTO 615
690 END ●
1000 :
1005 REM * * * CODE = DECODE * * *
1010 :
2015 A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
2020 B$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
2999 GOTO 110
```


附录五

错误信息表

CAN'T CONTINUE 不能继续执行

在电脑不允许继续运行的情况下使用了 CONT 命令。

DIVISION BY ZERO 除以零

在程序中出现了除以零或者除以值为零的变量。

ILLEGAL DIRECT 非法指示

在编辑方式中使用了 INPUT、GET、DEFFN 或者 DATA 之类的语句。

ILLEGAL QUANTITY 非法数值

出现了用负数作为数组的下标，例如 LET A(-1)=34；或者使用了错误的自变量，例如

错误

L=VAL(R)

TAB(-3)

SPEED=460

HTAB 55

正确

L=VAL(R\$)

TAB(3)

SPEED=255

HTAB 40

所谓错误的自变量是指：在需要数字的时候用了字符串，在需要字符串的地方用了数字；需要用正数时出现了负数或者是一个大于所允许范围的数。

NEXT WITHOUT FOR 有NEXT语句而缺少FOR语句

在执行FOR语句之前就出现了NEXT语句，或者控制变量

不一致,例如FOR I=1 TO 5: NEXT M。

OUT OF MEMORY 超出存储器范围

通常是由于程序中FOR.....NEXT循环用得太多,或者子程序太多,或者表示式中的括弧太多所造成的。

OVERFLOW 溢出

在进行一项计算时得到的结果很大,超出了电脑处理的范围。

REDIM'D ARRAY 数组说明出错

在执行DIM语句之前就使用了数组,例如
LET A(3)=7: DIM A(20); 或者对同一个数组进行了二次定义。

RETURN WITHOUT GOSUB 有RETURN (返回)语句而无GOSUB语句

在执行GOSUB语句之前先执行了RETURN语句;它常常发生在这种情况下:当程序中意外地出现“进入”到子程序内部去执行时,就会产生上述错误信息。

STRING TOO LONG 字符串太长

使用的字符串多于255个字符。

BAD SUBSCRIPT 数组下标出错

使用数组时,下标选用不当,例如
DIM A(5, 5): A(1, 1, 1)=77 下标编号错误;
DIM A(5): A(14)=7 下标大于5。

SYNTAX ERROR 语法错误

通常是用不正确的标点符号、括号不成对、错误的字符或者拼写错误等。

TYPE MISMATCH 类型不匹配

混淆了数字和字符串的使用，例如

LET A="9" 或者 LET A\$=33 或者 A\$=LEFT(A\$,1) 等等。

UNDEF'D STATEMENT 没有定义行

使用GOTO或者GOSUB语句时，转入的语句行在程序中不存在。

娃娃用电脑

胡金初 王钧权 周宠安 编译

徐培南 审校

简毅 装帧

少年儿童出版社出版

(上海延安西路1538号)

新华书店上海发行所发行

上海市印刷十二厂排版 上海市印刷三厂印刷

开本 787×1156 1/32 印张 9.125 字数 192,000

1985年12月第1版 1985年12月第1次印刷

印数 1—7,000

统一书号: R13024·207 定价: 1.20元