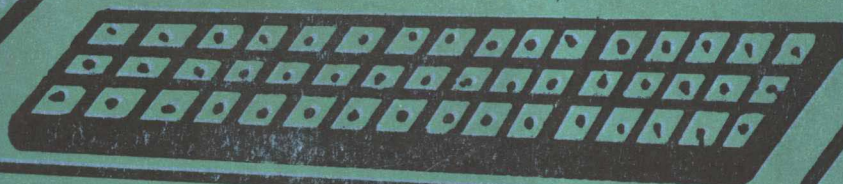


陶龙芳 徐孝凯 编

BASIC

语言实验



中央广播电视大学出版社

BASIC语言实验

陶龙芳 徐孝凯 编

中央广播电视大学出版社

BASIC 语言实验

陶龙芳 徐孝凯 编

中央广播电视大学出版社出版

新华书店北京发行所发行

北京印刷二厂印装

开本 787×1092 1/16 印张10.5 千字262

1988年1月第1版 1988年5月第1次印刷

印数 1—150,000

定价 2.00元

ISBN 7—304—00186—0/TP·13

前 言

BASIC语言是当前广泛流行和使用的一种计算机高级语言，世界上几乎所有 的计算机，特别是微型计算机上都配有此种语言，这为BASIC语言实验提供了极为有利的条件。学习BASIC语言离不开做实验。通过实验，不但能够熟悉机器的操作，包括熟悉键盘、显示器、打印机和磁盘等操作，而且能够巩固所学的知识，提高程序设计的水平以及应用计算机解决实际问题的能力。

本书是一本BASIC语言实验教材，它能够与谭浩强主编的《BASIC语言》教材以及其它的《BASIC语言》教材配套使用。书中以APPLE系列机及兼容机和IBM-PC系列机及兼容机作为实验机型，详细讨论了十二个BASIC语言实验。实验一、二、三主要介绍了微机系统中的键盘、显示器和打印机的基本操作以及输入、修改和运行一个程序的全过程；实验四、五、六、七主要讲述了分支程序、循环程序、子程序和数组程序的设计方法以及跟踪一个程序的执行过程；实验八和九分别介绍了APPLE系列机和IBM-PC系列机上的图形显示操作，通过做这两个实验，能够在计算机的显示屏幕上绘制出各种彩色图形；实验十、十一和十二主要介绍了磁盘的使用以及程序文件、顺序文件、随机文件的存取等操作。

本书除作为中央广播电视大学BASIC语言课的实验教材外，亦可作为其它高等院校、中等专业学校以及各种计算机培训班的BASIC语言课的实验教材和教学参考书。

本书的前九个实验由徐孝凯同志编写，后三个实验由陶龙芳同志编写，谭浩强同志负责全书的审定工作。

由于作者学识水平和实验教学经验的限制，编写过程又较仓促，错误和不足之处在所难免，希望读者和同行专家们批评指正。

编 者

1987年11月

说 明

实验周次安排与机时分配表

序 号	实 验 名 称	周 次	机 时 数
实 验 一	键 盘 操 作 练 习	7	4
实 验 二	屏 幕 显 示 操 作	8	3
实 验 三	打 印 机 操 作	9	2
实 验 四	分 支 程 序 实 验	10	4
实 验 五	循 环 程 序 实 验	12	4
实 验 六	子 程 序 实 验	14	3
实 验 七	数 组 程 序 实 验	16	4
实 验 八	APPLE 机上的图形显示	17	3
实 验 九	IBM-PC 机上的图形显示	17	3
实 验 十	文 件 操 作	18	2
实 验 十 一	顺 序 文 件	18	4
实 验 十 二	随 机 文 件	19	4

几点说明:

1. 做每个实验只需选择使用APPLE系列机及兼容机或IBM-PC系列机及兼容机中的一种机型。
2. 总实验机时数不得低于20小时, 具体做哪几个实验, 可由教学班自选。
3. 每个实验包括实验目的、实验预备知识、实验内容和实验要求四个部分, 上机做每个实验之前, 都必须预习好那个实验, 并根据实验内容编写好上机运行程序。
4. 凡是按照实验要求去做, 并得到满意结果者, 则表明完成(或通过)了该实验, 否则为不通过, 应重新补做。

目 录

实验一	键盘操作练习	(1)
实验二	屏幕显示操作	(23)
实验三	打印机操作	(35)
实验四	分支程序实验	(41)
实验五	循环程序实验	(54)
实验六	子程序实验	(68)
实验七	数组程序实验	(79)
实验八	APPLE机上的图形显示	(93)
实验九	IBM-PC机上的图形显示	(104)
实验十	文件操作	(118)
实验十一	顺序文件	(129)
实验十二	随机文件	(145)
附录一	APPLE机BASIC错误信息表	(155)
附录二	IBM-PC机BASIC错误信息表	(158)
附录三	参考书目	(164)

实验一 键盘操作练习

一、实验目的

1. 掌握微机基本系统的开、关机操作。
2. 了解和掌握BASIC操作状态和操作方式。
3. 了解和掌握机器键盘上字符键和控制键的功能。
4. 熟练掌握几种常用的键盘命令。
5. 会输入、修改和运行BASIC程序。

二、实验预备知识

(一) 微机基本系统的结构

微机基本系统包括主机(又叫系统部件)、显示器(或者监视器)和键盘三个部分。主机是微型计算机的核心部件,其内部有存贮器装置、运算器装置、控制器装置,以及与外部设备交换信息用的接口电路装置等。存贮器用来存贮大量的信息,运算器用来进行算术和逻辑运算,控制器用来解释和执行程序或者命令,在解释执行的过程中,控制整个系统的各个部件有条不紊的工作。键盘是微型计算机的输入设备。通过键盘,不但能够打入命令,让计算机按照命令所规定的功能去动作;而且能够打入程序和数据,使之保存在存贮器中。显示器是微型计算机的输出设备。显示器的屏幕不但能够随时显示你从键盘上输入的程序,数据和命令;而且能够通过显示(又叫打印)命令和语句显示出内存(即主机内的存贮器)中的程序和数据以及一些计算机系统的提示信息。

一般来说,主机和键盘被放在同一平面上,如放在一张工作台上或一张办公桌上,并且键盘放在主机的前面;显示器一般放在主机的上面。这样放置的系统,便于用户使用键盘和屏幕。

(二) 开关机操作

1. 开机顺序

- (1) 分别把显示器和主机的电源插头插入配电盘上相应的电源插座上。
- (2) 打开显示器上电源开关。
- (3) 打开主机的电源开关。

2. 关机顺序

关机顺序正好与开机顺序相反。

- (1) 关主机
- (2) 关显示器
- (3) 拔掉显示器和主机的电源插头或者关掉供给机器的总电源。

(三) BASIC操作状态

按照上面的开机顺序启动机器后,计算机将自动进入BASIC操作状态。但在有的APPLE机上,开机后需同时按下`CTRL`键和`RESET`键(确切地说,`CTRL`键应先按下,后

松开), 才能进入BASIC操作状态; 在有的IBM-PC机上, 开机后需按下 $\overline{\text{F1}}$ 功能键, 才能进入BASIC操作状态。

APPLE机进入BASIC操作状态后, 屏幕上显示出:

┐

其中字符“┐”为APPLE机BASIC操作状态提示符, 提示符后闪烁的方块是光标。

IBM-PC机进入BASIC操作状态后, 屏幕上显示出:

OK

—

其中字符“OK”为PC机BASIC操作状态提示符, 提示符下面是闪烁的光标。

当然, 计算机被启动后, 屏幕上除了显示出BASIC操作状态提示符和光标之外, 还显示出其它一些系统提示信息。如所用的BASIC解释程序的版本号、内存中可用的存储字节数等。

在BASIC操作状态下, 又有两种子状态: 一种是命令状态, 另一种是执行状态。

在命令状态下, 计算机等待用户(用户就是指上机操作的人)从键盘上进行操作。此时, 用户可以从键盘向上计算机输入一个新的程序; 还可以从键盘上打入命令, 用以指挥计算机执行命令所具有的操作功能; 也可以通过键盘上的控制键对内存中的程序进行修改; 等等。

在命令状态下, 打入RUN命令(即运行内存用户程序命令)并按下回车键后, 计算机立即转入执行状态, 执行内存中的用户程序, 程序执行结束后, 又自动地返回到命令状态。在程序执行过程中, 计算机不接受从键盘上输入的任何信息(个别控制键除外)。

(四) BASIC操作方式

BASIC语言允许用户以命令和程序两种操作方式使用它的语句。但是INPUT语句和DATA语句只能用于程序方式, 不能用于命令方式。

所谓程序方式就是将BASIC语句书写在程序中。每一个程序可包含若干个程序行, 每一个程序行里可书写多条语句(当然也可以只书写一条语句), 各语句之间用冒号分开, 每一程序行的开始必须书写有行号(即标号)。行号为一个十进制正整数, 其取值范围在0~65529之间。程序中的各个程序行一般按照行号从小到大的顺序排列书写, 行号一般以10开始, 行号的增量一般也取值为10。也就是说, 在一个程序中, 各程序行的行号依次取10、20、30、40、……。

从键盘上输入程序, 也是从第一个程序行开始依次输入。每输入一个程序行, 就是把该行中的所有字符按照从左向右的顺序依次从键盘上输入。从键盘上每输入一个字符时, 就是按下标有该字符的按键, 此时计算机接受这个字符, 并把它显示在屏幕光标所在的位置上, 同时光标向右移一格(即一个字符的位置), 准备用来显示下一个字符。每当输入完一个程序行并按下回车键后, 计算机就把它送往内存中, 接着上一程序行存储的地方保存起来。待整个程序都输入完毕后, 再通过打入运行命令(即RUN命令)执行(又叫运行)内存中的程序。计算机在执行一个程序时, 一般也是按照行号从小到大的顺序, 一行一行地解释执行, 直到END语句(即结束执行语句)为止。当然, 也可以使用其它一些命令对内存中的程序进行操作。如从键盘上打入LIST命令后, 将立即在屏幕上显示出内存程序的清单(即所有内存中的用户程序)。使用BASIC操作命令(除掉删除和修改)对内存中的程序进行操作后, 该程序仍

然保留在内存中。因此可以对内存程序进行多次操作。

所谓命令方式，就是把 BASIC 语句直接当作命令来使用。每一命令行可以包含若干条命令，每两条命令之间用冒号分开，但一般一个命令行只使用一条命令，每一命令行的开始不能有行号。当从键盘上打入一条命令(或命令行)并按下回车键后，计算机立即执行该命令所具有的操作功能。如从键盘上打入语句命令：

```
PRINT 15 * 20 + 32↵ (“↵”表示回车键)
```

计算机立即执行这条打印语句命令，首先计算出 $15 * 20 + 32$ 的值(即332)，然后把这个值显示在屏幕光标所在的位置上。再如从键盘上打入语句命令行：

```
LET X=30:LET Y=40↵  
PRINT SIN(X), COS(Y), TAN(X+Y)↵
```

当第一行命令打入后，计算机立即执行两条赋值语句的功能，即把30送给x，把40送给y。当第二行命令打入后，计算机立即执行该打印语句的功能，即在屏幕上打印(即显示)出 $\sin x$ 、 $\cos y$ 和 $\tan(x+y)$ 的值。

总之，从形式上看，程序行带有行号，命令行不带行号。从计算机的处理过程来看，计算机接受一程序行后，暂不执行，只是把它保存起来，待整个程序输入完并打入 RUN 命令后，才解释执行每个程序行中的 BASIC 语句；计算机接受一命令行后，立即执行命令行中各个语句命令或其它操作命令的功能，命令执行结束后，该命令就消失了，并不保存在内存中。

(五) BASIC 键盘命令

1. NEW 命令

此命令为清内存命令。在键盘上打入：

```
NEW↵
```

即在键盘上顺序按下 N、E、W 三个字符键和回车键后，计算机清除内存中的所有程序和变量，释放出内存中所有的用户工作空间，等待用户向内存输入一个新的程序。

在向计算机输入一个新的程序之前，一般要使用这条命令，清除内存中原有的用户程序。

但要注意：NEW 命令不能随便使用，若你在输入程序后，使用了此命令，计算机就把你刚刚输入到内存的程序全部清掉。

2. LIST 命令

此命令为显示内存程序清单命令。命令格式为：

```
LIST line1—line2
```

其中 LIST 为命令定义符，line1 为用户要开始显示行的行号，line2 为最后要显示行的行号。若要显示出整个内存中的用户程序，只需打入：

```
LIST↵
```

此时计算机立即把内存中的程序送到显示器，在屏幕上按照行号从小到大的顺序显示出来。若从键盘上打入：

```
LIST 30↵
```

此时，内存程序中行号为30的程序行立即被显示在屏幕上。因此，若用户只需要显示某一个程序行，只要在 LIST 命令符后打入该程序行的行号并按下回车键即可。若从键盘上打

入:

```
LIST 50—↵
```

此时, 屏幕上立即显示出内存程序中行号为50的程序行和行号大于50的所有程序行。因此, 若要显示某程序行之后的所有程序行(包括该程序行), 只要在LIST命令符后接着打入该程序行的行号并打入减号和回车键即可。若从键盘上打入:

```
LIST 20—80↵
```

此时, 屏幕上立即显示出内存程序中从行号为20、到行号为80(包括20行和80行在内)的所有程序行。因此, 若要显示内存程序中某一个程序段(即若干个连续的程序行), 只要在上面的命令中改变一下开始行号(即line1)和结束行号(即line2)即可。从键盘上打入:

```
LIST -100↵
```

此时, 屏幕上立即显示出内存程序中行号小于100和行号等于100的所有程序行。因此, 若要在屏幕上显示出内存程序中某一程序行上面的的所有程序行, 只要将上面命令中的行号换为该程序行的行号即可。

当一个程序输入后, 一般要使用LIST命令, 在屏幕上显示出你刚输入的程序, 以便检查和修改程序的错误。

3. RUN命令

此命令为运行程序命令简称运行命令或执行命令。命令格式为:

```
RUN line
```

其中RUN为运行命令的定义符, line为用户给出的开始运行的程序行行号。若在命令中省略行号, 则从内存中行号最小的程序行开始执行用户程序。

假定在内存中已有下列两个程序:

程序1:

```
10 X=12:Y=-8
20 Z=(X+Y)*(X-Y)
30 PRINT "X=", X, "Y=", Y, "Z=", Z
40 END
```

程序2:

```
100 A=10:B=6
110 C=SQR(A+B)
120 PRINT "A=", A, "B=", B, "C=", C
130 END
```

当从键盘上打入:

```
RUN 10↵
```

计算机立即由命令状态转入执行状态, 从10程序行开始执行内存中的程序1。执行10程序行时, 将12赋值给X、-8赋值给Y; 执行20程序行时(因20程序行只有一条语句, 所以又可叫做20语句), 首先计算出赋值号右边算术表达式的值, 然后再把这个值(即80)赋值给Z; 执行30程序行即30语句时, 在屏幕上立即打印(即显示)出如下内容:

```
X=12    Y=-8    Z=80
```

执行完40语句时, 计算机结束了整个程序的执行, 由执行状态返回到命令状态。

由于计算机执行程序的速度相当快，所以运行程序 1 时，瞬间即可完成。

当从键盘上打入：

```
RUN 100↵
```

计算机立即由命令状态转入执行状态，从100程序行开始执行内存中的程序 2。执行100程序行时，将10赋给A、6 赋给B；执行110语句时，计算A+B的算术平方根(即4)并赋值给C；执行120语句时，在屏幕上立即打印出如下内容：

A=10 B= 6 C= 4

执行完130语句时，计算机结束了整个程序 2 的执行过程，由执行状态返回到命令状态。

当从键盘上打入：

```
RUN↵
```

计算机将从内存中行号最小的程序行开始执行程序。因程序 1 的10程序行行号最小，所以从10程序行起执行程序。该命令同RUN10↵命令具有相同的作用。因此，当需要从内存程序中最小的程序行开始运行程序时，在打入运行命令中，可以省去这个最小的行号；直接打入RUN和按下回车键即可。

4. CONT命令

此命令为继续运行程序命令。当计算机正在执行程序的过程中 碰到STOP语句，或者碰到用户从键盘上打入暂停控制键时，就暂停执行程序，返回到命令状态。此时可以利用打印语句命令查看内存中各变量的值，也可以利用赋值语句命令修改各变量的值。当需要继续运行程序时，则打入：

```
CONT↵
```

此时，计算机立即由命令状态转入执行状态，从STOP语句或暂停控制键中断的地方继续往下执行程序。

若在暂停期间，修改了程序，则不能使用CONT命令来继续运行程序。

5. DEL命令

此命令为删除程序行命令。命令格式为：

```
DEL line1, line2
```

其中DEL为命令定义符，line1和line2表示用户给出的两个程序行的行号。使用该命令将删去内存程序中从line1程序行到line2程序行之间的所有程序行。

例如：从键盘上打入：

```
DEL 30, 60↵
```

计算机立即删除内存程序中从30程序行到60程序行之间(包括30程序行和60程序行)的所有程序行。

当只需要删除内存程序中某一程序行时，只要在命令状态下，打入该行的行号并按下回车键即可。如，从键盘上打入：

```
40↵
```

计算机立即从内存程序中删去行号为40的那个程序行。

此命令仅适应于APPLE机及兼容机。

6. DELEE命令

此命令为PC机，TRS—80等微型机上的删除命令。命令格式为：

DELETE line1—line2

其中DELETE为删除命令的定义符，line1和line2分别表示程序行的行号，两行号之间用减号字符连接。使用该命令能删去内存程序中若干个程序行。

例如，从键盘上打入：

DELETE 20—50↵

计算机立即删除内存程序中20程序行至50程序行之间(包括20和50程序行)的所有程序行。若要删去内存程序中某一程序行，也只需打入该程序行的行号并按下回车键即可。

7. HOME命令

此命令为清屏命令。在键盘上打入：

HOME↵

计算机立即清除显示屏上的所有内容，并把光标移至屏幕的左上角。

此命令仅适应APPLE机及兼容机。

(六) APPLE机键盘上的控制键操作

APPLE机键盘示意图如下所示：

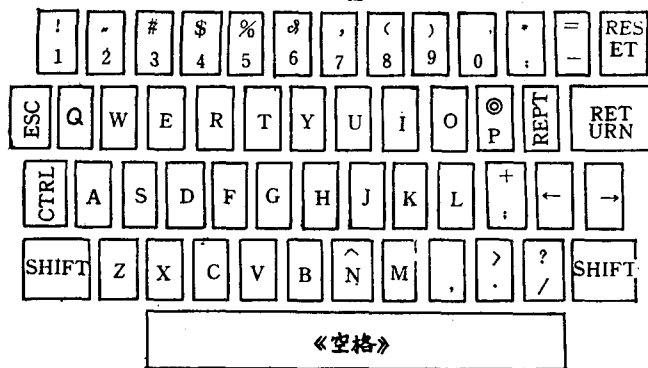


图1.1 APPLE机键盘示意图

键盘上共有52个按键，最下方的长方格为空格键，其它的有英文字母键(A~Z)、数字键(0~9)、专用符号键(如\$、#、?、;等)和控制键(如RETURN键、SHIFT键等)。其中字母键、数字键和专用符号键统称为字符键。在有的按键上，同时标有上、下两个字符，此类键又叫双字符键。

当从键盘上按下字符键时，该键上的字符将立即显示在屏幕上，当从键盘上按下控制键时，该键上的字符串或符号并不显示在屏幕上，只是立即执行该控制键所具有的控制功能。APPLE机键盘上的控制键共有八个，下面分别介绍它们的控制功能。

1. SHIFT键

此键为换挡键。在键盘上的左下角和右下角各有一个，使用时可任选其一。当需要输入双字符键的上面一个字符时，必须同时按下此键。如同时按下SHIFT键和 $\frac{\%}{5}$ 键(确切地说，SHIFT键要先按下、后松开)，屏幕上显示出“%”字符，表示计算机接收到的字符是%，

而不是5。若只按下

%
5

 键，屏幕上显示出5，而不是%，表示计算机接收到的字符是5。

当按下单字符键时，不管是否同时按下了SHIFT键，屏幕上显示的和计算机接收的都将是该键上的字符。

2. RETURN键

此键是回车键。

当输入完一个程序行，只有在按下回车键之后，该程序行才被存贮到内存中指定的地方。

当修改完一个程序行，只有在按下回车键之后，该修改好的程序行才能取代内存中原有的该程序行。

当输入完一条命令或一个命令行，只有在按下回车键之后，该命令才被立即执行。

当计算机执行程序中的INPUT语句时，需要停下来从键盘上读取数据，用户在键盘上打入相应的数据后，只有当按下回车键，计算机才继续执行程序。

按下回车键，还能使显示屏上的光标移到下一行的开始位置上（即下一显示行的最左边）。

3. ←键

此键为退格键。每按一次，光标退回一格（即向左移动一个字符的位置），并且使光标所经过的字符作废。

如屏幕上显示为：

```
10 PRINT
```

按一次←键后变为：

```
10 PRIM
```

此时被光标覆盖着的T字符作废。再按一次退格键后变为：

```
10 PRI
```

此时，被光标覆盖着的M字符也作废。

若再连续按下三次退格键，光标将左移到P字符的位置上，并且依次使I、R、P字符作废。

4. →键

此键为重打键。每按一次，光标右移一格，并且使原光标位置上的字符有效，就如同从键盘上重新打入该字符一样。

如屏幕上显示为：

```
10 PRMT
```

假定光标覆盖着字符I，当按一次→键后变为：

```
10 PRIT
```

此时I同它左边的字符一样为有效。当从键盘上按一次N键后变为：

```
10 PRIN
```

若不是按下N键，而是按下重打键，则M字符有效。现在键盘上打入N字符后，用正确的字符N代替了错误的字符M。

若再按一次→键后变为：

10 PRINT

此时字符T又变为有效,如同从键盘上重新打入T字符一样。

5. RESET键

这是一个复位键。按下这个键(有的APPLE机必须在按下CTRL键的同时按下这个键才起复位作用),能使计算机立即停止正在执行的工作,返回到BASIC命令状态。

当程序或命令在执行中出现死循环等不正常现象时,或者需要人为地中止执行时,可以按下此键,从而使计算机退出执行状态,回到命令状态。

6. CTRL键

CTRL为control一词的缩写,表示具有控制的功能。它一般同其它键配合使用。当配合使用时,CTRL键与其它键必须同时按下(确切地说,CTRL键应先按下、后松开)。

(1) CTRL-C组合键

在程序的执行过程中,同时按下CTRL键和字符C键,能暂停程序的执行,使计算机脱离执行状态而进入命令状态。此时可以对内存中各变量的值进行检查和修改。当打入CONT命令后,又可继续执行程序。

此控制组合键同程序中的STOP语句具有同样的功能。

(2) CTRL-S组合键

在命令或程序的执行过程中,同时按下CTRL键和字符S键能暂停执行,但并不脱离执行状态。当按下任一字符键或空格键,执行被继续。

在使用LIST命令显示较长的程序期间,可使用此组合键冻结显示,以便检查程序。

(3) CTRL-RESET组合键

此组合键为复位键。无论在什么时候,同时按下CTRL键和RESET键(有的APPLE机上只需按下RESET键)都将中断正在执行的操作,返回到命令状态。此时,若打入CONT命令则不能恢复执行。

7. ESC键

此键一般同其它键配合使用,在配合使用时,应先按下ESC键并松开后再按下其它的键。

常用的ESC组合键有:

(1) ESC-@组合键

先按下ESC键后,同时按下SHIFT键和@键,将立即清除显示屏上的所有内容,并将光标移到屏幕的左上角。

(2) ESC-E组合键

顺序按下ESC键和字符E键,可清除从光标开始至这一行末尾的所有字符。

(3) ESC-F组合键

顺序按下ESC键和F键,可清除光标右边及下面所有行上的字符。

(4) ESC-A、ESC-B、ESC-C、ESC-D分别使光标右移一格、左移一格、下移一格、上移一格。

(5) ESC-K、ESC-J、ESC-M、ESC-I也分别使光标右移一格、左移一格、下移一格、上移一格。不过,ESC键后,K、J、M、I可以连续使用,使光标作连续地移动。如:

ESC—K—K—I

使光标先右移两格，后上移一格。又如：

ESC—J—M—M—M

使光标先左移一格，后下移三格。

当使用ESC、K、J、M、I键移动光标的过程中，按下与ESC键不能组合的任一键（如按下空格键、→键、←键，RETURN键等）将退出移动光标状态（即ESC状态）。

注意：(4)、(5)两点中的组合键只起移动屏幕上光标的作用，对光标所经过的字符不起任何作用。

(8) REPT键

REPT为repeat的缩写，具有重复的含意，因此叫它重复键。当需要重复打印某一个字符时，可在按下该字符的同时，按下此键。当松开任一按键时，就停止重复。

(七) 在APPLE机上修改程序

假定想把下面的程序通过键盘输入给计算机：

```
10 READ X,Y
20 LET Z=SQR (X+Y)
30 PRINT X,Y,Z
40 DATA 25.46, 18.73
50 END
```

现在来看如何修改在程序输入过程中的错误。

1. 对当前行的修改

当前行就是正在输入的程序行（即光标所在的那一程序行）。当发现当前行的输入有错误时（如发现多打、漏打或错打了字符），则可根据不同的错误类型，采取不同的修改办法。

(1) 修正打错的字符

当正在输入一程序行的过程中，若发现光标左边有一个或多个字符被打错，则可连续按下←键，将光标移至出错的字符上，然后从键盘上重新打入正确的字符，错误修正后，再连续按下→键使光标移到此行的结尾。

如屏幕上显示为：

```
10 REAB X,■
```

当要打入10语句最后一个字符Y时，却发现READ中的字符D错打成了字符B，此时要连续按下←键，将光标移到错打的字符B上，屏幕上显示为：

```
10 REA■ X,
```

再重新打入正确的字符D，然后连续按下→键，使光标移到此行的结尾即逗号后面，屏幕上显示变为：

```
10 READ X,■
```

此时便可继续输入该程序行后面的字符。

再如，屏幕上显示为：

```
20 LET T=SON(X■
```

当输入过字符x时发现赋值号左边的字符Z被错打成了字符T，赋值号右边的函数名SQ-

R被错打成了SON，此时连续按下←键，使光标移到T字符上，接着从键盘上打入Z键，再按下→键，此时显示变为：

```
20 LET Z=S■N(X
```

再从键盘上依次打入Q和R，接着按两次重打键，屏幕上显示为：

```
20 LET Z=SQR(X■
```

此时可以继续输入该语句后面的字符。

(2) 删除多打的字符

当在输入一程序行的过程中，若发现光标左边有多打的字符，则可连续按下←键，将光标移至多余的字符上，然后按下ESC键和字符K键（若连续删除几个多余的字符，需连续按下几次K键），多余的字符被删除后，再连续按下→键使光标移到此行的结尾。

如屏幕上显示为：

```
30 PRRINT X,■
```

此时发现PRINT定义符中多打了一个字符R，为了删去这个字符，首先使用←键，将光标移到多余的字符R上，屏幕上显示变为：

```
30 PR■INT X,
```

接着按一下ESC键、一下K键，屏幕上显示为：

```
30 PRR■NT X,
```

此时多余的R字符已经从内存中被删除了，但是并没有从屏幕上删除。再接着按下几次→键后，使光标移到此行的结尾（注意：第一次按下→键时，并不向右移动光标，只是退出ESC状态）。接着便可继续输入该程序行后面的字符。

再如屏幕上显示为：

```
40 DATA 2525.46; 18■
```

此时发现DATA语句中第一个数据的整数部分打了两遍，需删去两个字符即2和5；第一个数据的后面应是逗号被错打成了分号，需改正过来。为此，应首先使用←键把光标移到多余的字符2上，接着按下ESC键和两次K键，便删去了多余的两个字符2和5，再接着按下四次→键，光标便移到了打错的分号字符上，从键盘上打入逗号并按下两次→键后，使光标移到了此行的结尾，接着便可继续输入该行的字符。

(3) 插入漏打的字符

当在输入一程序行的过程中，若发现光标左边有漏打的字符，则首先使用退格键，把光标退到漏打字符的位置上；接着按下ESC键和I键使光标上移一格，并按下不能与ESC组合的任一键（如空格键）使之退出ESC状态；再接着从键盘上打入要插入的字符；然后按一次ESC键和一次M键使光标下移一格（即到原来的行上），并按下若干次J键，使光标左移到漏打的字符位置上；最后连续按下重打键，使光标前进到此行的结尾。

如屏幕上显示为：

```
30 PNT X,Y■
```

此时发现PRINT定义符中漏打了R和I两个字符，为此首先使用←键把光标退到N字符的位置上，接着顺序按下ESC键、I键和空格键使光标上移一行退出ESC状态，此时屏幕上显示为：

```
30 PNT X,Y
```

接着从键盘上打入R和I，屏幕上显示为：

```
RI■
```

```
30 PNT X,Y
```

再接着按下ESC键和M键，屏幕上显示为：

```
RI
```

```
30 PNT■X,Y
```

然后再按两次J键使光标左移到N字符上，最后连续按下→键使光标前进到此行的结尾，屏幕上显示为：

```
RI
```

```
30 PNT X,Y■
```

这样R和I就被插到了P和N之间，待此行输入完并按下回车键后，表明该程序行已经存贮到内存中了，若使用LIST命令显示一下30语句，从屏幕上可以看出：

```
]LIST 30↵
```

```
30 PRINT X,Y,Z
```

```
]■
```

第一行的第一个字符“]”是BASIC操作状态提示符，提示符后是用户打入的显示命令，第二行是执行该命令时显示出的30语句，从显示的30语句可以看出R和I已经插入到P和N之间了。

(4) 从出错的字符开始重新输入

在输入一程序行的过程中，若发现刚打入的字符有错时，可将光标退回到出错的位置上，重新输入正确的字符。

如屏幕上显示为：

```
10 REAM■
```

刚打入的字符M应改为D，所以可按一下←键，使光标退回一格，再继续打入正确的字符。

再如屏幕上显示为：

```
20 LEET Z■
```

在LET定义符中多打入一个字符E，此时可将光标退回到多余的字符E上，再接着从键盘上打入正确的字符，这样多余的字符E就消去了。

又如屏幕上显示为：

```
30 PRNT■
```

此时发现R和N之间漏打了字符I，可将光标退回（光标退回应使用退格键）到N字符上，接着顺序打入PR之后的所有字符。若在PR之后打入I、N、T，则屏幕上显示为：

```
30 PRINT■
```

从出错的字符开始重新输入，往往比上面三种修改当前行错误的方法更方便，更直观，这对初次上机的用户来说更是如此。

2. 对已输入行的修改

已输入行就是已存贮到内存中的程序行。当一程序行输入完并按下回车键后，我们就确