

BASIC 入门

清华大学电子工程系
计 算 中 心 印

1978. 4.

BASIC

INTERNATIONAL

PROGRAM

1978-1

说 明

这份资料根据“入门 BASIC”一书（[日]味村重臣等著）的第二、三、四、六、七、八章译出。

“入门 BASIC”一书的特点是通过较多的例题来讲解 BASIC 语言，比较通俗易懂，对初学用 BASIC 语言解题者有一定的参考价值。文中所举例题是大家熟悉的又简单的数学问题，没有更深的内容，因此确实是一本入门书。

第一章，第五章的内容没有多大参考价值，故没有译出。

第八章中所有题目在 DJS-130 机上解过，个别的地方作了一些修改。

由于译者是初学 BASIC 语言，没有解过更多的实际问题，因此体会不深，又加上外语水平所限，肯定有不少错误，包括翻译的错误。把这份资料译出来目的是积累一点教学资料，以供大家共同学习参考。因此务请大家提出宝贵的批评和建议。

译者：史美林

校译：谢锡迎

1978 年 5 月

目 录

说 明

第一章	BASIC 及其背景 (略)	
第二章	基本会话.....	1
第三章	简单的程序设计 (编制)	7
第四章	简单的应用例题.....	33
第五章	复杂的会话 (略)	
第六章	复杂的程序.....	49
第七章	复杂的应用例题.....	63
第八章	小型机上使用 BASIC 解题.....	86

第二章 基本会话

§ 2.1 命令

BASIC 的特色，如前所述，在于通过与计算机进行会话编书并执行程序。关于会话，我们可以将它比喻为用电话向对方传达事情，这时，仅向对方传达事情是不够的。叫到对方后，还要听取事情传达到对方的情况，然后进行必要的处理。对方一面回答这些情况，一面听取必要事情。这种送与受就是会话。它要有明确的顺序，因此有相应的“语言”是必要的。

在电话里，借助电话机，用自然语言一边说话一边收听。在 BASIC 语言里，则借助电传打字机，以接近自然语言的人工语言，一方面用键盘输入，另一方面用打印输出，进行对话。这个人工语言称为命令。

一个是事情的处理次序，一个是程序，可用以语句 (statement) 来书写。这和 FORTRAN 等汇编类似。

关于基本的命令，在本章中通过简单的会话加以叙述。用于复杂会话的命令，在第五章叙述，关于语句在下章加以说明。

§ 2.2 呼叫 (通话)

电话机有自动式，共电式，磁石式等，它们的呼叫 (通话) 方式各不相同。例如，自动式电话是拿起受话器，听到拨号音后拨对方的电话号码叫对方来。共电式电话，是拿起受话器，告诉接线员对方的电话号码，叫出对方来。

因为电传打字机也有几种型式，它的呼叫方式不一样。例如大学有长距离式和直通线路式、分别接下面那样传呼。

(1) 长距离式：

这是用数据喇叭利用电话线和计算机进行联接的。首先按下“电话键”，拿起受话器 (听筒) 听到拨号音，然后，通过拨号盘呼叫计算中心。到此是和挂普通的自动电话相同的，听到高音处，按下 ORIG 键。这样，电传打字机就呼叫计算机并联接上。

(2) 直通线路式：

和计算机用直通线路接上，它是不带受话器的。因此，接线只按下 ORIG 键就可以了。

无论哪种情况，从电传打字机传接过来后，要输入命令 HELLO 或 HLL，并按回车键 (RETURN)。

若接线成功，电传打字机就，打出例如下面那样的头注：

GE 600—LINE T/S FROM DARTMOUTH

TERMINAL 138 ON AT 11:33 10/20/71

因为电传打字机的操作和头注方式的不同，在实际运用时，必须正确了解其使用手册和操作指南。

§ 2.3 程序输入前的会话

接前节的头注之后，电传打字机打出：

USER NUMBER.....

也就是问用户的代码号，这时，电传输入确定的代码，并按回车键 (RETURN)，这里电传输入一行，必须按回车键。如果不是这样做，这一行就不能输入到计算机。以下相同。

接着，电传打印机打出

NEW OR OLD.....

这是问下面要输入的程序是全新的还是以前已用过的，在计算机内已存放好的，若是前者电传输入 NEW “回车” (按回车键)

若是后者电传输入

OLD 回车

当电传输入 NEW 时，电传打出：

NEW FILE NAME.....

这是问将要输入的程序名称，按照规定，将名称由电传输入，按回车键。マートスマ大学，除了 *，；/ 以外的允许用 8 位以内的任意文字、数字的组合作为名称。例如：

A, SAMPLE, JACK, EXAM—1, NO—1 等等

接着电传输入之后，电传打出

READY,

程序就可以输入了

当电传输入是 OLD 时，电传打出：

OLD FILE NAME

下面把以前用过的程序名输入之后，电传就输出：

READY

这时因为没有必要输入程序，立刻就可转到执行。

§ 2.4 程序的输入

程序按下章所述的语句用电传输输入但必须首先电传输输入 5 位以内的标号，在语句之后必须按回车键；即必须是

标号（5 位以内）语句回车键这样的形式。

当电传输输入错误被立刻发现时，可以按下←键加以改正。←键是仅将它前面的一个文字、数字或记号消去，接着把正确的输入进去，达到改正的效果。

例如，将 READ 错输入为 REED，发现这个错误时，按两次←键，然后打入AD就行了，即

REED←←AD

或者在 REE 时发现了错误，按一次←键，电传输输入 AD 就可以了，即

REE←AD

无论那一情况，对计算机都是正确地输入了 READ。

若想将一行全部消去时，将 CTRL 和 X 二个键同时按下就可以了。

若将已经输入过的标号再输入一次，并输入和前面不同的语句，则能将该行全部修改。

正确地将程序输入之后，下面用电传打入表示执行的命令

RUN

如果是电传打入上述的 OLD 时，按 READY 之后，直接电传打入 RUN 就行。无论什么情况，在 RUN 之后不可忘记按 RETURN（回车键）。

§ 2.5 程序的执行和结束

RUN 输入后，电传打字机打印出下列程序名称，时刻，日期，并进入程序的执行。

SAMPLE	11:14	10/20/71
(程序名)	(时刻)	日期

若程序有可能执行的话就执行之。并根据程序中的打印语句（PRINT），打出求得的结果。然而，这并不是说程序是正确的，只不过是程序在文法上没有错误。

如果，文法上一有错，程序就停止执行，并打印出错误信息。详细的在第五章叙述。例如在标号为 70 的语句中用了不恰当的变量，就打印出：

ILLEGAL VARIABLE IN 70

当关于全部程序文法上的检查结束，必要的错误信息都打印完，就打出机器使用时间，例如：

TIME: .07 SECS

这时，就必须将这些错误修正。为此，令应该修正的语句的标号和正确语句的一并输入，其方式和程序输入时相同，仍由电传输入就可以了。要消去一行时，用电传只输入标号，再按回车键即可。当要插入某一语句的时候，将处于前后语句的标号间的值作为标号，电传输入即可。例如，想在标号为 60 和 70 间插入语句，用标号 61~69 即可。因此，标号一开始用 10, 20, 30, ……间隔为 10。或 100, 200, 300, ……间隔为 100，修正时就很方便。

修正结束，再次电传输入 RUN，程序就执行。当结果有明显的错误，必须使计算机停止，这时即使程序正在执行中，电传打入 STOP 就可以，在电传打字机正在动作中时，需要紧急停止时，只要按一下 S 就可以了。这时 S 不被打印，总之，程序执行被停止。同时打印出到此为止的计算机使用时间。然后，就可以再次修正程序。

假如正确的结果得到了，我们可以用电传打字机进行告别，这时如果电传输入：

GOODBYE 或

BYE

就立刻打印出：

* * * OFF AT 11:14

HANOVER 10/20/71

字样。电传打字机就从回路上断开彻离。

然后取下打印结果，就可离座。

图 2.1 是上述的从呼叫到结束的概

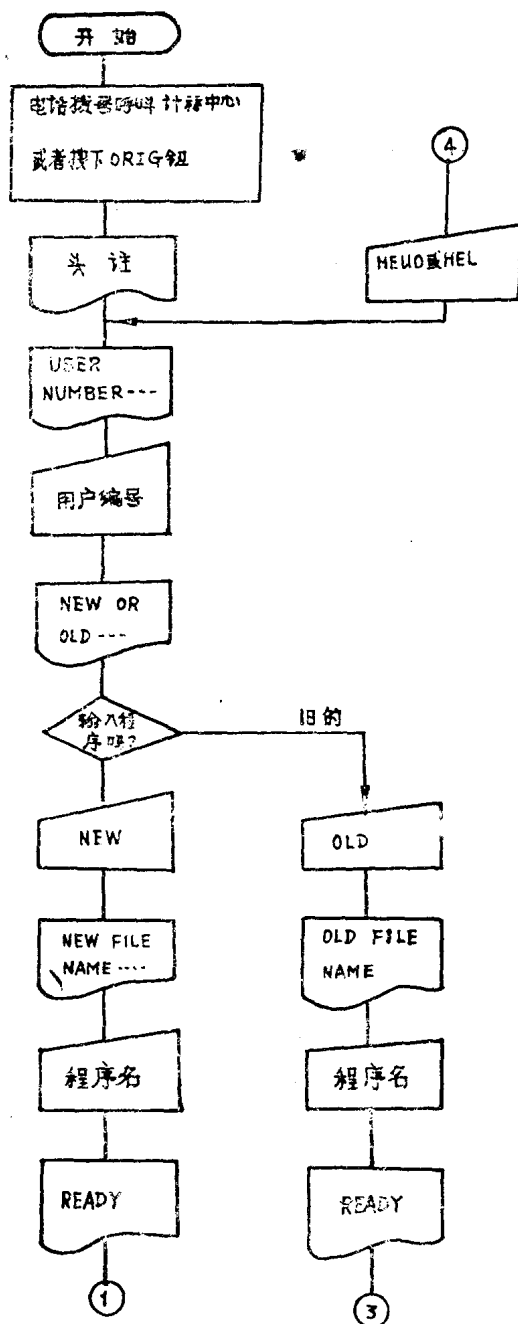


图 2.1 基本电话流程图 ①

略的流程图。

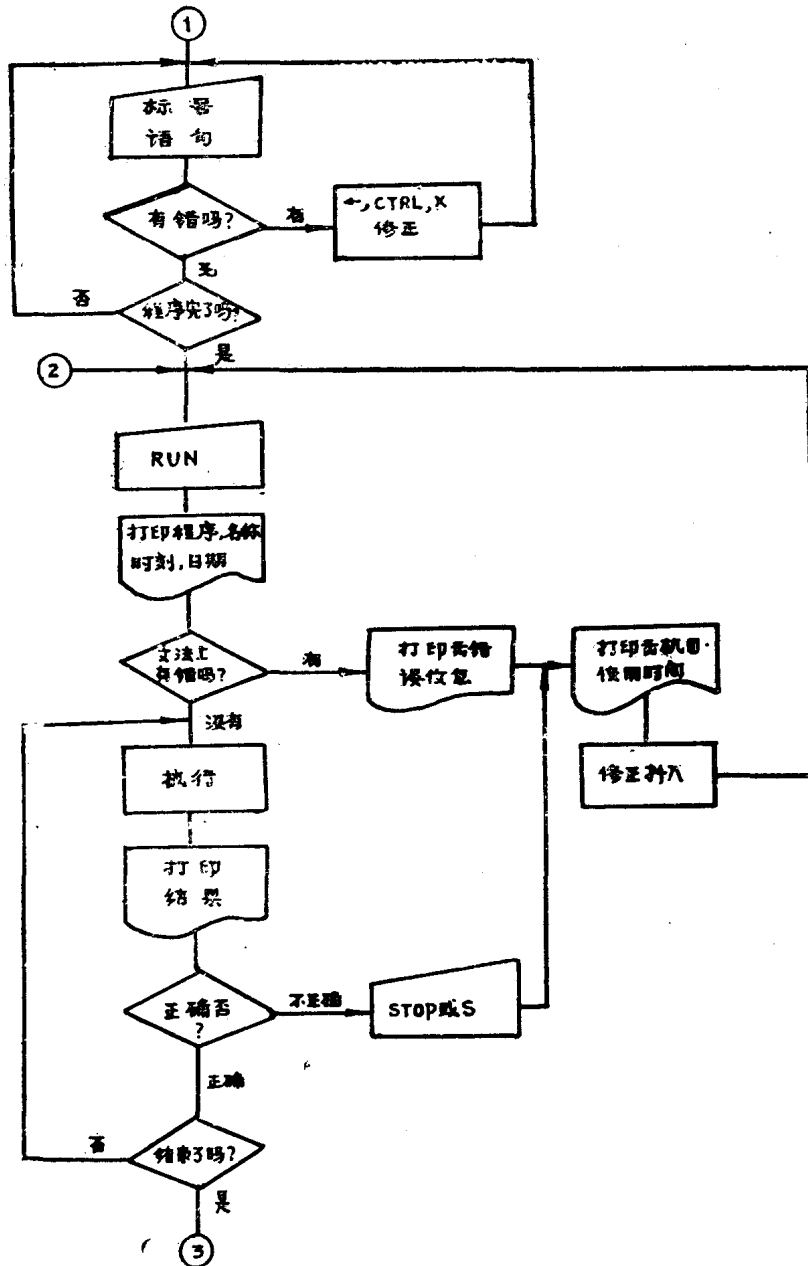
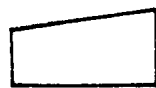
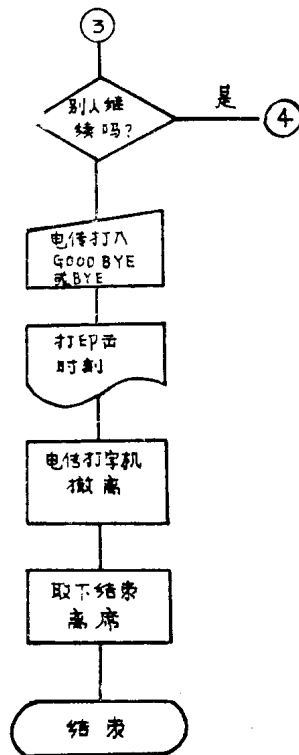


图 2.1 基本会话流程图②



电传输入



电传输出

图 2.1 基本会话流程图③

第三章 简单的程序设计（程序编制）

§ 3.1 例 题

关于 BASIC 的程序设计，我们通过下面的例题来说明

例题：联立方程

$$\begin{cases} ax+by=c \\ dx+ey=f \end{cases}$$

给定 a, b, d, e 的值，当 c 和 f 的值变化时，求解。

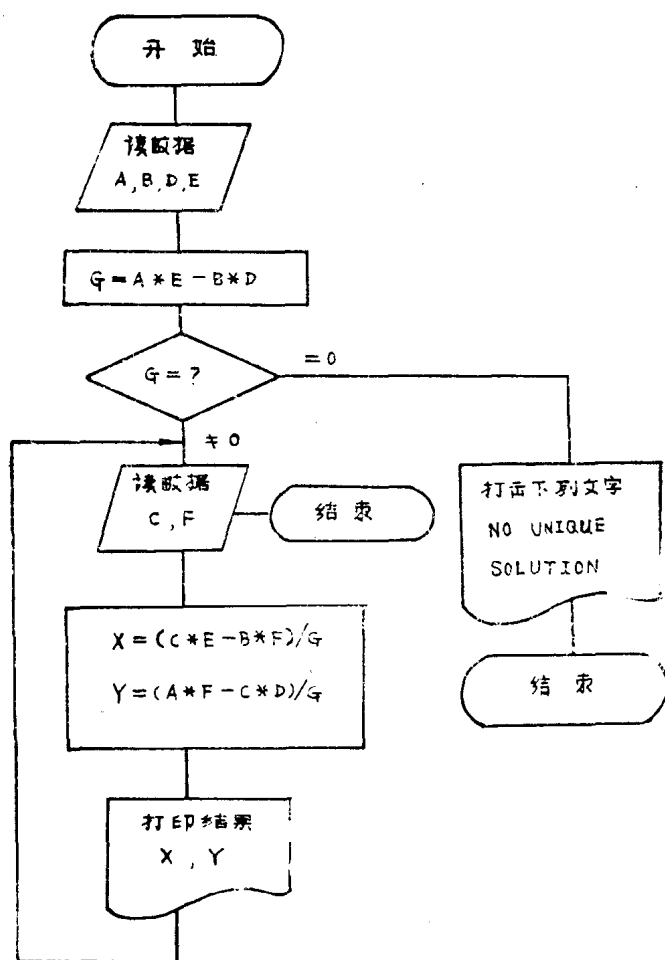


图 3.1 例题的流程图

解此方程时, 首先来看 $ae-bd$ 是否为 0。当 $ae-bd \neq 0$ 时, 方程的解如下:

$$x = \frac{ce-bf}{ae-bd}$$

$$y = \frac{af-cd}{ae-bd}$$

但当 $ae-bd = 0$ 时, 即分母为 0, 解不定, 不能求解。

考虑到以上情况, 求解处理的顺序可用图 3.1 所示的流程图表示之。根据此流程图写出 BASIC 的程序, 如图 3.2 所示。

图中 “NO UNIQUE SOLUTION” 意即 “没有唯一解”

```
10 READ A, B, D, E
15 LET G = A * E - B * D
20 IF G = 0 THEN 65
30 READ C, F
37 LET X = (C * E - B * F) / G
42 LET Y = (A * F - C * D) / G
55 PRINT X, Y
60 GO TO 30
65 PRINT "NO UNIQUE SOLUTION"
70 DATA 1, 2, 3, 4
80 DATA -7, 5
85 DATA 1, 3, 4, -7
90 END
```

图 3.2

通过这个例题, 我们可以得出如下几点:

首先, 程序中只使用大写英文字母。因为打字机只有大写英文字母。有时, 为了区别英文字 O 与数字 0, 将英文字 O 写成 ϕ , $\ominus$ 或 $\bar{O}$ 那样。此外, 在 BASIC 中能使用的数字和记号如下:

大写英文字母 (26 个) A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S,
T, U, V, W, X, Y, Z

数字 (10 个) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

记号 (16 个) +, -, *, /, ↑, =, (,) <, > ., ; ", \$, 空格。

其次, 程序中的各行分别都给一个号码, 这个号码叫做标号 (Line number), 接着就是表示处理内容的语句。也就是说, 程序是由带标号的语句的集合所构成。

标号, 不仅是区别各行, 在 BASIC 中还决定处理顺序。即处理按标号由小到大的顺序进行。因此, 如果语句的标号正确, 那么各语句是否按顺序很好排列都不妨碍。例如, 按 10, 15, 20, 30, 42……90 顺序排列可以, 按 30, 15, 90, 10, 20, 42……的顺序排列也可以。计算机执行时, 按标号从小到大的先后次序执行。这样, 程序的修改, 追加, 删除等很容易。这是会话型语言的特色之一。

第三、语句以一个单字开始。在此例中有 READ, LET, IF, PRINT, GOTO, DATA, END 等七种, 它们表示语句的类型, 以后详细叙述。

第四, 在语句中使用空格。在 BASIC 中, 除 PRINT 语句的引号 (") 间格外, 没有意义。也就是说, 有没有空格没有影响。用空格只不过为了使程序好看。因此写成下面的样子也没有关系。

```
10READ A,B,D,E,
15LET G=A * E-B * D
.....
60GOTO 30
```

现在我们一面参照例题的流程图, 一面一步一步地研究程序。

首先, 标号 10 的 READ 语句, 对应 READ 语句一定伴随着有一个或一个以上的 DATA 语句。在程序执行中, 一碰到 READ 语句, 在 READ 后面写的变量 (这里是 A、B、D、E) 和 DATA 语句中的数值一一相当。READ 语句将这些数据依次赋给各变量。此例中, A 被赋予标号为 >0 的 DATA 语句中的数值 1, B 为 2, D 为 4, E 为 2。

其次, 标号为 15 的 LET 语句, 其意义是计算 $A \times E - B \times D$ 。结果放在 G。式中的 * 表示乘号。又等号 (=) 和数学的意义不同, 这里是将右边的计算结果置换到左边的变量, 也就是说必须注意有“置换”的含意。

在此例中, A, B, C, D 的值由标号为 10 的 READ 语句所确定, G 就为 -6。

$$A * E - B * D \rightarrow G$$

$$1 \times 2 - 2 \times 4 \rightarrow -6$$

现在 $G = -6$ 而不是 0, 但若 $G = 0$ 则求不到唯一解。因此, 由标号为 20 的 IF 语句来判断。 $G = 0$ 时, 即 IF 语句的条件满足, 那么就跳越到 THEN 后面指出的标号的语句。此例题中是标号 65 的 PRINT 语句。例如, 若 $D = 1$, $A * E - B * D$ 为 $1 \times 2 - 2 \times 1 = 0$ 即 $G = 0$, 则下面就执行标号 65 的 PRINT 语句。

标号 65 的 PRINT 语句是将引号中的字符原样打出，即

NO UNIQUE SOLUTION (没有唯一解)

跳过标号 70, 80, 85 的 DATA 语句 (DATA 语句不是执行语句) 在标号 90 的 END 语句停止。

IF 语句的条件不满足时，执行下一标号的语句。此例题中， $G = -6$ ，下面就执行标号为 30 的语句 READ 语句。

同前，参照 DATA 语句，但是标号 70 语词已用完，由标号 80 的 DATA 语句赋值 C 和 F ，即 $C = -7$ ， $F = 5$ ，因此方程式为

$$\begin{cases} x + 2y = -7 \\ 4x + 2y = 5 \end{cases}$$

执行标号 37 和 42 的 LET 语句。这里，必须注意，如果把括号忘了，就变成计算

$$X = CE - \frac{BF}{G}$$

这就错了。 X 、 Y 计算后，下面就执行标号 55 的 PRINT 语句，打印出 X 、 Y 的值。打字的形式如后所述有确定的规则。

此后，由标号 60 的 GOTO 语句迫回到标号 30 语句，再由 READ 语句，标于 85 的 DATA 语句的数值 1, 3 赋于 C 与 F 。方程式变为：

$$\begin{cases} x + 2y = 1 \\ 4x + 2y = 3 \end{cases}$$

由标号 37, 42 的 LET 语句计算 X 、 Y ，由标号 55 PRINT 的语句打印 X 、 Y ，再次由标号 60 的 GOTO 语句迫回到标号 30 的 READ 语句。

现在，标号 85 的 DATA 语句还剩下数值， C 、 F 再次被赋值， $C = 4$ ， $F = -7$ ，方程式变为

$$\begin{cases} x + 2y = H \\ 4x + 2y = -7 \end{cases}$$

下面，同前执行，再迫回到标号 30 的 READ 语句。但现在 DATA 语句中赋值的数值没有了就打出信息：

ONT OF DATA IN 30

程序结束。

由前章所述的命令和上述的程序，此例实际执行结果如图 3.3 所示。

```

NEW OR OLD.....NEW
NEW FILE NAME.....EXAM-1
READY
10 READ A, B, D, E
15 LET G=A * E-B * D
20 IF G=0 THEN 65
30 READ C, F
37 LET X=(C * E-B * F)/G
42 LET Y=(A * F-C * D)/G
55 PRINT X, Y
60 GOTO 30
65 PRINT "NO UNIQUE SOLNTION"
70 DATA 1, 2, 4, 2
80 DATA -7,5
85 DATA 1, 3, 4, -7
90 END

RUN

EXAM-1          11:03          10/19/71
4               -5.5
0.666667        0.166667
-3.66667        3.83333
OUT OF DADA IN 30
TIME:.10 SECS

```

图 3.3

§ 3.2 常数与变量

在 LET 语句的式子中的数值和 DATA 语句中的数值在程序执行过程中它的值是不变的，称为常数 (Constant) 在 BASIC 中，常数可以用整数、带小数的实数、指

数形式的实数等的任何一种给定都可以。其中以指数表示的实数、数值用某一数和 10 的几次方的乘积的形式表示。例如 3.1415×10^4 在 BASIC 中表示为 $3.1415E4$ 。

另外，可以附加符号，当正号时可以省略。

有效位数 9 位，这随制度的不同而不同。

例如下面的这些数。作为常数都是有效的：

整数型：0, 1, 2, 365, -8, +15

小数型：0, 3.14, -6.33, .88, -.43265

指数型： $3E3$, $3.1415E4$, $1.2E-3$, $-1.4E-3$ 。

在语句的式子中或在程序执行过程中其值是变化的数称为变量。(变数)在 BASIC 中，变量必须用一个英文字或一个英文字带一个数字来表示。因此，能够使用的变量， $A \sim Z$ 26 字母和 $0 \sim 9$ 10 个数字组合共 286 种。下面的样子，作为变量都是有效的。

A, B, I, O, X, Y, Z

$A1, B8, I5, O0, X1, Y0, Z2$

但是， O 或 $O0, O1 \dots$ 等等不容易区别，希望不用。

§ 33. 数据的读入

数据的读入，也就是为了给变量赋值，用 READ 语句

标号 READ 变量、变量、……变量。和一个或一个以上的 DATA 语句。

标号 DATA 常数、常数、……常数。在前节的例题中已见过。

可是，在执行 READ 语句中，在利用 DATA 语句中的数值时，它是按顺序地赋值于变重，因此，DATA 语句放在 END 语句前，放在那儿都行但数值必须按程序顺序读入。

例如，前例题中，标号 10 的 READ 语句和标号 70 的 DATA 语句对应，标号 30 的 READ 语句第一次执行时与标号 80 的 DATA 语句对应，第二次和第三次执行时与标号 85 的 DATA 语句一一对应。不过用下面那样的书写也可以：

75 DATA 1, 2, 4, 2, -7, 5, 1, 3, 4, -7

同时下面那样也可以

70 DATA 1, 2, 4

75 DATA 2

80 DATA -7, 5, 1

85 DATA 3, 4, -7

DATA 语句中的数值全部用过之后，还有 DEAD 语句要参照它，那么就打出下

信息，表示程序终了。

OUT OF DATA In 该 READ 语句的标号

DATA 语句的数值已用过之后，不能再使用。但是加标号 RESTORE 这一语句后，就能从头开始使用。例如，图 3.4 的程序就是这种例子。在此程序中，A、B、C、分别被赋值 1, 2, 3, P, Q, R 为 4, 5, 6。标号 60 是 RESTORE 语句，X, Y, Z 也被赋值 1, 2, 3。如果没有 RESTORE 语句，X、Y、Z 就被赋值 7、8、9。

```
10 DATA 1, 2, 3, 4, 5, 6, 7, 8, 9
20 READ A, B, C
30 PRINT A, B, C
40 READ P, Q, R
50 PRINT P, Q, R
60 RESTORE
70 READ X, Y, Z
80 PRINT X, Y, Z
  ⋮
999 END
```

图 3.4

§ 3.4 数据的输出

为了输出计算结果，字符串、跳过一行用 PRINT 语句。有各种多样的使用方法，举例说明。

首先，最简单的使用方法是：

标号 PRINT 变量或表达式

执行此语句，将变量的值或表达式计算结果打印出来，数值打印形式，服从下面规则：

① 数值是整数时，打印出不带小数点的：

符号（正号时为空格），8 位以内的数字。

当 8 位以上时，打印出有效数字 6 位，四舍五入，如下：

符号（正号时为空格），第 1 位，小数点，下面 5 位，空格、E 适当的指数（带正或负的符号的整数）。例如 32437580259 为

3.24376 E + 10