

MULTI-20

BASIC 语言

参考手册

(1973.5. 出版)

一九七四年四月译

目 录

- §1. BASIC 的介绍
- §II. 语言的描述
- §II. 1. 语言的特性
- §II. 2. 语言的成分
- §II. 2. 1. 字—标 (Mots-cla)
- §II. 2. 2. 数 码
- §II. 2. 4. 标 符
- §II. 2. 5. 简单变量
- §II. 2. 6. 下标变量
- §II. 2. 7. 数码和变量的内部表示
- §II. 2. 8. 函数指示符
- §II. 2. 9. 标号
- §II. 3. 表达式
- §II. 3. 1. 算术表达式
- §II. 3. 2. 关系式
- §II. 4. 说明
- §II. 4. 1. 图表说明
- §II. 4. 2. 函数说明
- §II. 4. 3. 注释
- §II. 5. 指令
- §II. 5. 1. LET—赋值
- §II. 5. 2. GOTO—无条件转移
- §II. 5. 3. GOSUB 和 RETURN—转子程序和返回.
- §II. 5. 4. PRINT—打印
- §II. 5. 5. IF…… THEN—条件转移
- §II. 5. 6. FOR-NEXT—循环
- 1- §II. 5. 7. INPUT—键盘输入

- §II 5.8. READ-DATA-RESTORE — 常數赋值
- §II 5.9. STOP — 停止
- §II 5.10. END — 程序終了
- §II 5.11. CALL — 調出汇编程序的子程序
- §II 6. BASIC 的 8 K 型式
 - §II 6.1. 指令組
 - §II 6.2. 程序庫的函數目錄
- §II 7. BASIC 的 16 K 型式
 - §II 7.1. 赋值 — MATREAD
 - §II 7.2. 打印 — MATPRINT
 - §II 7.3. 逃標的標符
 - §II 7.4. 恢復的標符
- §III. 使用
 - §III 1. 概述
 - §III 2. 紙帶索引
 - §III 3. 程序的裝入
 - §III 4. 起動
 - §III 5. 編輯
 - §III 6. 各種命令
 - §III 7. 命令方式時的返回
- §IV. 附錄
 - 1. 標準函數表
 - 2. 錯誤信息
 - 3. 12 K BASIC 性能
 - 4. 16 K 型式下的矩陣標符
 - 5. BASIC 的原理 — 程序方框圖

31 言

现在的 BASIC 参考手册是以 12 K 型式 BASIC 研究为中心。

它符合于 INTERTECHNIQUE 的小计算机和 MULTI-8/M301 和 MULTI-20/01, 20/05 和 20/06 的使用。

8 K 型式对于这种基本型来说允许删去某项目。这些改变在本文中²⁾指出。

16 K 型式完全允许 12 K 的功能，并增多了矩阵计算机的功能。

这些增加了的功能要在附录中讲到。

第一章

BASIC 的介绍

BASIC 对于没有太多程序编制经验的计算机使用者来说是一种简单的语言。学习比较容易，并能够解答从最简单到很复杂的题目。

对于计算机来说，程序是一系列的指令，这些指令表示求解题目的整个操作。

处理时分为几个阶段：

- 1° 分析题目，寻找求解的方法
- 2° 以 BASIC 语言写程序
- 3° 输入程序和数据到计算机中
- 4° 编辑和执行
- 5° 输出结果

为了更好地了解程序的编辑，举一个简单的例子：假定计算机求二次方程的根，也就是说

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

从数据 a ， b 和 c 着手

程序写成如下形式：

```
15      LET      A=1
20      LET      B=1
25      LET      C=-6
30      LET      D=B↑2-4*A*C
35      IF      D<0, THEN 65
40      LET      E=SQR(D)
45      LET      X1=(-B+E)/A/2
50      LET      X2=(-B-E)/A/2
55      PRINT    X1, X2
60      GOTO     70
```

```

65 PRINT "PASSE RACINES"; "REELLES"
70 STOP
999 END

```

这个程序是由一系列指令行组成的。每行代表一条实施指令或提供计算机必须的数据。

每行的开头有一个顺序号码或称为标号。这里标号是每次加5，在这些标号之间可以插入原来编辑时没有的指令，除非遇到一条转移指令(不久我们就要讲到)。否则指令按照行的编号确定的顺序执行。

除了第35行外，从15到50行都是用字LET开始。LET是赋值指令。其功能就是把等号右边的算术表达式的值赋予等号左边的变量。

——例如：LET A=1

算术表达式中使用的各种标符有加(+), 减(-), 乘(*), 除(/)和乘方($\uparrow$)

机中首先进行括号内的计算

没有括号时，将按照如下优先次序进行计算：

首先是 $\uparrow$ 接着*或/再接着+或-；在没有标符优先时，计算从左到右进行。

例如，第45行，它首先计算 $-B+E$ ，用A除其结果，再用2除新的结果。

第35和60行指令是转移指令：

1) IF THEN

如果在IF的后面表示的条件被满足时，则执行的下一行指令的标号就是表示在该行THEN后面的标号。

在我们的例子中，如果 D 是负的，计算机将执行第 65 行指令。

如果条件没有实现 ($D \geq 0$)，计算机如同 IF THEN 指令不存在一样处理。按顺序继续进行。

2) 指令标号 GOTO

被执行的下一行指令是这样的，其标号指示在 GOTO 以后。这个转移不需要履行任何先决条件。

第 55 行指令是 PRINT 指令，它要求计算机打印后面的变量值。

当我们希望打印文字的时候，我们就将这些文字放在引号之间，如同第 65 行指令一样。

第 40 条指令将调出一个计算机存储器内具有的标准函数：这里是求平方根，并把 $\sqrt{D}$ 的值赋与 E 。

标号 70 的 STOP 指令行止程序。

通常应该具有标号最高的 END 指令指出程序終了。

当我们要求计算机打印一个值的时候，这个值通常以规格化的格式表示：

- 尾数的符号 (+ 或 -)
- 小数点
- 尾数的七个十进制数字
- 字母 E (为了指示阶码)
- 阶码符号 (+ 或 -)
- 阶码 (两位数字)

例如：

• $1036492E+10$ 表示 0.1036492×10^{10}

• $2140000E-12$ 表示 -0.214×10^{-12}

我们可以看出程序的每行完成一个加工过程。

在完成对简单或复杂题目详细分析以后，接着找出求解的方法，此方法我们很容易用 BASIC 语言表示出来。程序是由一条指令一条指令组成的，这些指令是我们要求计算机来执行的。

再举第二个例子：

假定计算机下式的值。

$$Y = A\sqrt{I} + B \ln I + C$$

对于 $I = 1, 2, 3, \dots, 19, 20$ 。

程序如下：

```
10      READ  A, B, C
20      FOR  I=1  TO 20
30      GOSUB 90
40      LET  Y=A*X+B*Z+C
50      PRINT Y
60      NEXT
70      DATA 1, 2, 3.
80      STOP
90      LET X=SQR(I)
100     LET  Z=LG N(I)
110     RETURN
999     END
```

这里我们首先注意到的是 READ 指令，它要求计算机将 A, B, C 读入内存。

如何读？

回答：在 DATA 指令以后，给 A，B，和 C 值各自为 1，2，3。

READ 指令是涉及到 DATA 指令后面的数据程序。

代替 READ 和 DATA 指令，我们可以安排输入指令 INPUT A，B，C。

在执行这些指令时，计算机打印一个问号，使用者只有在电传机上把要赋予 A 的值打出，并按一下回车键才行。为了要求给出赋予 B 的值，BASIC 打印一个新的问号，并依次类推。程序此时就用这个值展开。

第 20 行表示 FOR 指令；它必须配有 NEXT 指令，这里它在第 60 行。位于这两条指令之间的语句被相继循环执行，每循环一次 I 的值加 1，并指示 I 从 1 到 20。这叫做程序的循环(boucle)

随后，我们迁到带有标号的 GOSUB 指令。它用来调出子程序。在求解题目的时候常常重复出现同样的计算的情况，它能够避免重写若干次。

计算机此时执行所指示标号的指令，(例中的 90)，接着，按顺序连续执行一直到迁到 RETURN 指令；在这个时刻，执行的指令是跟在 GOSUB 指令后面的那条指令，也就是：

LET Y = A * X + B * Z + C

其他指令我们现在已经都熟悉了。并且我们能够练习来编写一个小程序，以确定你是否掌握了。

第二章 语言的描述

§ II.1 语言的字符

程序是借助于电传打字机键盘上可供使用的字符组成的。

这些字符有：

1. 字母：A, B, C ... Z

2. 数字：1, 2, 3 ... 9

3. 符号：+, -, *, /, =, <, >, ≠, ., (,),
空白

用这些字符我们能够组成字，这些字按照规定的语法组织起来就形成语句。

§ II.2 语言的成份

一个 BASIC 的程序行由如下几项组成：

—— 包括在 1 和 9999 之间的 1 到 4 个数字组成的行的标号。

—— BASIC 语句。

—— 表示行的结束的回车字符。

一个程序行最多包括 72 个字符，其中包括行的标号。

处在一个 BASIC 程序行中的空白不表示意义，并且可以为行的清晰而使用。

在这一章中，我们将看到构成 BASIC 语句和成份的描述和它们的意义的说明。

§ II.1 字一标 (MOES-CL2)

组成语句的基本部份的这此字指示如下：

- 说明类型；
- 指令类型；
- 限定界限；
- 标准函数的名称。

说明：* DEF 函数的定义

 * DIM 表格的说明

 * REM 注释

指令：* CALL 调用汇编程序的子程序

 DATA 常数的定义

 * FOR...NEXT 循环

 * GOSUB 转移到子程序

 GOTO 转移

 IF...THEN 条件转移

 * INPUT 通过键盘输入数据

 LET 赋值

 * PRINT 打印

 READ 常数的赋值

 * RESTORE 常数记录田置“0”

 * RETURN 子程序的返回

 STOP 程序展开的停止

界限：END 程序的结束指令

标准函数：ABC 绝对值

 * ATG 反正切

 COS 余弦

EXP	指数
INT	取整部分
LGN	对数
SGN	符号
SIN	正弦
SQR	平方根
* TAN	正切
* RND	随机数的产生

* 不适用于 8 K 的形式

§ II. 2.2 数

数的最一般形式包括：

- 一个符号：+ 或 -；符号“+”可有可无
- 一个整数部分
- 一个小数部分
- 一个指数部分

整数部分由 1 到 7 个十进制数码组成。

小数部分是 1 到 7 个十进制数码组成，前头有一个小数点。

阶码部分是 1 或 2 个带或不带符号的数码组成，前面有一个字母 E。

三个部分的每个是任意的。但是，在数具有阶码部分的时候，它必须具有整数部分或小数部分。

由整数部分和小数部分组成的十进制数的

总的数字最多不超过 7 个。

例子：

$$348.52 E + 4 = 348,52 \times 10^4$$

$$.12 = 0,12$$

$$1 E + 20 = 10^{20}$$

$$-434 E - 2 = -4,34$$

在机田上处理的所有数以浮点的形式表示
(见 § II. 2. 7)

限定：数值应该大约包括在 10^{-38} 和 10^{+38} 之间

精度：7 位有效数字。

§ II. 2.3 运算符

我们把运算符分为：

——算术的运算符

——关系的运算符

算术运算符：	加	+
	减	-
	乘	*
	除	/
	乘方	↑
	赋於	=
关系运算符：	小于	<
	大于	>
	等于	=
	小于或等于	< =

大于或等于 $\geq$
不等于 $\neq$

§ II.2.4 标识符

当写程序时，人们必须给出如下成份的名称：

- 简单变量
- 表格
- 函数

这些名称叫做标识符，并且在它是一个简单变量和表格时，用下述构成：

- 或者一个字母
- 或者一个字母跟着一个数字

例如：

A
A0
B1
Z9

我们不久就要讨论函数标识符的情况。

§ II.5 简单变量

一个变量代表一个单独的值或表中的一个成份。

人们经常用的有简单变量和下标变量。
所有的简单变量用标识符表示。

§ II. 2.6 下标度量 (不适用于 8 K 型式)

下表 标度量 的元素 ^{的索取} 只有在表格说明以后才被利用。

表名是一个单一字母组成的标识符。

表的标识符和度量的标识符可以有同样的名称。

下标度量是用一个标识符来表不在后面跟着一个下标表，其“下标表”允许选择与表格相应的成份。这个下标表是由 1 或 2 个下标构成，一个和另一个用逗号相互分开并放在括号之间。

下标：

下标的数目最大等于 2，它应该与出现在表格说明中的下标的数目是相同的。

一个下标可以是一个没有符号的整数或者是一个简单度量。

下标的下限经常是 1；它的上限则是相应于出现在表格中的这个下标的数值。

例：

$A(1)$	表示一维的 A 表的第一个分量。
$B(J1)$	如果 $J1=3$ 则代表一维的 B 表的第三个分量。
$Z(3, J)$	代表 2 维的 Z 表的分量 $(3, J)$

§ II.2.7 数和变量的内部表示

数，变量，下标变量和下标以 MULTI-20 的小数形式按 4 个 octet (字节) 表示在存储器中。

尾数将占用 3 个 octet (字节) (23 个 bit 和一个符号 bit (位))，阶码将占用一个 octet (字节) (7 个 bit 和一个符号 bit)。

§ II.2.8 函数指示符

人们以函数指示符形式的字母指定跟在函数指示符后面的实在参数是自变量，它被括在括号内。

函数指示符确定一个位，该位借助于以下计标：

——用函数名称确定的算术表达式

——用计标的当时时间的实在参数值确定的自变量。

一个实在参数可以是一个数，一个简单变量，一个下标变量，一个函数指示符或一个算术表达式，其算术表达式我们不久就要定义。

人们分为两类函数指示符：

——代表“标准函数”的有：正弦，余弦，正切……并且对于标准函数，有以子程序形式存在机中的标准标法。

这些指示符不应该作他用；它们的名字