

微型计算机

BASIC

计算机委员会办公室编
上海市计算技术研究所翻印

1981·1

目 录

第一章 BASIC 的立即执行方程

§ 1	算术运算符	3
§ 2	BASIC 的常数类型	5
§ 3	变量及算术表达式	6
§ 4	内部函数	9
§ 5	用户自定义函数	

第二章 basic BASIC

§ 0	预备知识	16
§ 1	BASIC 程序执行方式	18
§ 2	赋值语句	20
§ 3	逻辑运算和关系表达式	21
§ 4	控制转移语句	26
§ 5	循环语句	35
§ 6	启动执行和停止语句	40

第三章 BASIC 输入输出语句及格式

§ 1	输出输入语句	44
§ 2	输出格式编辑语句	53

第四章 BASIC 程序的编写及调试

§ 1	初始化语句	67
§ 2	BASIC 程序编写举例	73
§ 3	BASIC 程序的调试及有关指令	91
§ 4	条件转移指令(续)	104

第五章 字符串处理

§ 1	字符串及字符串变量.....	110
§ 2	正字符串说明格式.....	115
§ 3	字符串的运算和比较.....	127
§ 4	用于处理字符串的几个函数.....	132

第六章 数据文件输入输出

§ 0	预备知识.....	143
§ 1	数据文件的建立和打开.....	146
§ 2	数据文件的读写.....	150
§ 2	数据文件的关闭及文件指针.....	157
§ 4	若干例子.....	160
附录 1		180

第一章 BASIC 的立即执行方式

§ 1 算术运算符

BASIC 具有两种执行方式，一种为程序执行方式，另一种为立即执行方式，本章我们将简单介绍立即执行方式，程序执行方式将在以后各章里讨论。

当采用立即执行方式时，BASIC 如同一个高级计算器。指令一般从键盘输入，机器将立即执行而给出答案。

BASIC 的绝大部分指令均可用立即执行方式。

我们先看几个例子：

例一：计算 $1 + 3 + 5 * 6 < CR >$

执行方式： $>> PRINT 1 + 3 + 5 * 6 < CR >$

结果： 54

例二：计算
$$\frac{5 + 7 * (3 + 8)}{16 + 5}$$

执行方式： $>> PRINT (5 + 7 * (3 + 8)) / 16 + 5 < CR >$

结果： 3

例三：计算
$$\frac{6 + 4}{8} + \frac{7 * 3}{6}$$

执行方式： $>> PRINT (6 + 4) / 8 + 7 * 3 / 6 < CR >$

结果： 4

例四：计算 $\cos 28^\circ + \sin 72^\circ$

执行方式： $>> DEG < CR >$

$>> PRINT \cos (28) + \sin (72) < CR >$

结果： $1.834 \quad 41091563$

说明几点:

- 1: >> 符号为 BASIC 处于工作状态, 等待指令的标志
<见第三章说明>
- 2: <CR> 为按键盘上的 RETURN 键。每一条指令输入后, 只有按一下此键机器才响应。
- 3: DEG 指令见第三章 § 3
- 4: 例二及例三的结果是否正确, 请读者在机器上试验
- 5: PRINT 为向控制台 (或打印机) 输出结果的指令
<见第二章 § 1>

BASIC 的算术运算符如下:

+	加
-	减
*	乘
/	除
()	括号
↑ (或 **)	乘方

与其他语言一样, 各种算术运算是具有先后优先次序的。优先等级如下:

1.	括号	先括号 (从里到外)
2.	↑ (或 **)	后乘幂
3.	* 和 /	再算乘和除
4.	+ 和 -	最后算加减

例: $4 + 8 / 4 = 6$

$$(4 + 8) / 4 = 3$$

在同一优先级中, 运算从左至右进行

*

例: $5 + 8 / 4 * 2 = 9$

$5 + 8 / (4 * 2) = 6$

类型 - 运算符优先级

§ 2 BASIC 的常数类型

前面例二及例三似乎得出了错误的结果。要解释这个问题

BASIC 有四种常数类型: 整型, 浮点型 <长浮点型和短浮点型>, 十六进制型和字符串型。

例	浮点型	整型	十六进制型	(相应的十进制数)
	20000	55	%8000%	- 32768
	3	9995	%FFFF%	- 1
	0.0003765	- 5	% 9 %	9
	12.7	+ 60	% F %	15
	1.25E+5	- 458	% 100 %	256

1. 整型: 一个小于 10000 且不代小数点的数称为整型常数。它在机器内部只占两个字节。

2. 浮点型: 代有小数点或 E 的数认为是浮点数。大于等于 10000 而不代小数点的数也认为是浮点数。

浮点型分两种:

I) 长浮点型 <双精度浮点数>。在机器内占八个字节。

例: $1.67537667890123 E + 07$

II) 短浮点型, 在机器内占四个字节。

例: $1.6753 E + 17$

3. 十六进制型: 两边用 % 号圈夹起来的数。

例: % FABC %

4. 字符串型将在第四章详细讨论

5: 说明几点:

I) 关于在机器内部如何存放这些类型的数请参阅附录, 数在机器内部的存储格式。

II) 一个数的类型由数的表示形式决定。

例: 6 为整数; 6. 为浮点数

6. 00000000 为浮点数 %6% 为十六进制数

III) 几个数运算后, 其结果的类型将决定于参加运算诸数的类型。
请看下列例子

$$6/7 = 0$$

$$6 * 7 = 42$$

若参加运算的全为整型则结果为整型。

$$1) \quad 2. * 3 / 4 = 1.5 \quad 2) \quad 3 / 4 * 2. = 0$$

$$3) \quad 6. / 7 = 0.85714285714285$$

例 1: $2. * 3$ 为浮点数, 故 $2. * 3 / 4$ 为浮点数。

例 2: $3 / 4$ 为整型结果为 0, 故 $3 / 4 * 2. = 0$

例 3: $6.$ 为浮点数

从上面讨论就不难看出最初给出的例二及例三的答案是完全正确的。

§ 3 变量及算术表达式

前面我们举了一些简单的例子。如果我们要计算的式子比较复杂。则可以引进一些变量来简化要计算的内容。

例如要计算 $R * ((2 * \cos^3 A - 3 * \cos A * \sin^2 A) e^{-R/4 * R^3})^2$

其中 $A = \frac{\pi}{3}$ $R = 5$

执行方式 >> $A = 3.1416 / 3$

>> $R = 5$

>> $X = 2 * \cos(A)^3 - 3 * \cos(A) * \sin(A)^2$

```
>> Y = E * P ( - R / 4 ) * R ↑ 3
```

```
>> C = ( Z * Y ) * * 2
```

```
>> PRINT C
```

结果: 4909.8871520946

这里 A, R, X, Y, C 均称为变量。变量顾名思义,在整个运算执行过程中它的值可以改变。在任何时候均可赋予它一个新值。

每个人都有名字,以示与他人相区别。变量也一样,在一个程序中,每个变量都将有一个名字,以便在计算中区别对待。

BASIC 的变量名只能由两个(或一个)字符组成。而且第一个必须是字母(A, B, ...),第二个必须是数字(也可以没有)。所以在—个程序中总共可有 286 个变量名(A, A0, A', ...A9; B, B0, ...B9; ...; Z, Z0, ...Z9)。

BASIC 具有四种类型的变量

1) 整型变量

2) 短浮点型变量

3) 长浮点型变量

4) 字符串变量(将在第四章讨论)

前面三类亦称数值变量。BASIC 规定:一个变量若无特殊说明,则将认为是长浮点型变量。但是如果在赋值号(=)右边各量均为整型数时,结果将为整型。

例: >> A = 7 / 4

```
>> PRINT A
```

结果: 1

```
>> B = A / 3
```

```
>> PRINT B
```

结果: 0.3333333333333333

```
>> A = 7./4
```

```
>> PRINT A
```

结果: 1.75

```
>> B = A/3
```

```
>> PRINT B
```

结果 0.5833333333333333

通过初始化指令 INTEGER, LONG, SHORT 我们可以使某些具体的变量指定为整型, 长浮点型或短浮点型(见第三章 § 3)。

我们亦可利用指令 IMODE, SFMODE, LFMODE 使得在程序中那些没有特殊说明的变量, 全部指定为整型, 长浮点型或短浮点变量(见第三章 § 3)。

算术表达式; 用算术运算符 +, -, *, /, ↑ (或 **) 及括号 () 把一些常数, 变量及正数连接起来的有意义的式子称为算术表达式。它与一般代数式有些相似, 只是书写方法上略有差别。

代数式

算术表达式

$$\frac{1}{2} g h$$

$$G * H / 2$$

$$\frac{b - \sqrt{b^2 - 4ac}}{2a} \quad (B - \text{SORT}(B * B - 4 * A * C) / (2 * A))$$

$$\cos^2 x \sin y - \sin^2 x \cos y \quad \cos(x) \uparrow 2 * \sin(y) - \sin(x) \uparrow 2 * \cos(y)$$

$$a + \frac{b}{c + \frac{d}{e + \frac{f}{g}}} \quad A + B / (C + D / (E + F / G))$$

写算术表达式时，必须注意不能编写聚号。

例 错误的 正确的

4 A C

4 * A * C

A (B + C)

A * (B + C)

2 A

2 * A

在书写时一定要能清楚的区别字母O和数学0之间的区别。

BASIC一般以0代表字母O，而以0代表数字0。

§ 4 内部函数

为方便运算起见，BASIC附有较多的内部函数，这些内部函数将和变量一样出现在算术表达式内进行运算。

格 式	数 学 意 义	例
1. ABS(X) *	X 的绝对值	ABS(-5) = ABS(5) = 5
2. EXP(X)	e ^x 的值	EXP(4.1) = 60.340287597344
3. FRA(X)	X 的小数部分值	FRA(-3.7) = -0.7
4. INT(X)	X 的整数部分	INT(5.7) = 5 INT(-5.7) = -6
5. LOG(X)	log _e X	LOG(3.2) = 1.1631508098056
6. SIGN(X)	符号函数	$\text{SIGN}(X) = \begin{cases} 1 & X > 0 \\ 0 & X = 0 \\ -1 & X < 0 \end{cases}$
7. SQR(X)	$\sqrt{X}$	SQR(2) = 1.4142135623731
8. SIN(X) **	Sin x	
9. COS(X)	COS x	

10. TAN(X) tgx

11. ***ATN(X) arc tgx (tg⁻¹x)

* 这里 X 可为算术表达式, 数值变量或数值常数

* * 若不作特殊说明, 则三角函数的自变量均以弧度计算。

我们可以利用 DEG 指令 (见第三章 § 2) 来预先指定三角函数的自变量以角度来进行计算

例: >> A = SIN (5)

 >> PRINT A

结果: -0.95892427466381

 >> DEG

 >> PRINT SIN (5)

结果: 0.087155742747658

***反三角函数只有 ATN(X) 一种, Sin⁻¹x 和 COS⁻¹x 可以由下面公式计算而得:

$$\text{COS}^{-1}x = \text{arc COS}x = -\text{arc tg}(x/\sqrt{1-x^2}) + 2 \cdot \text{arctg} 1$$

$$\text{Sin}^{-1}x = \text{arc Sin}x = \text{arctg}(x/\sqrt{1-x^2})$$

若不作特殊说明, 以 ATN(X) 的结果为弧度制。同样, 亦可利用 DEG 指令使结果为角度制

例: >> PRINT ATN (0.80)

结果: 0.674740 94222353

 >> DEG

 >> PRINT ATN (0.80)

结果: 38.659808254087

除上述 11 个外还有如下几个:

12. 整型随机函数 IRN (X)

其中 Z 为哑元，即不影响结果的自变量，在使用时可以随意填入一个数。

此函数将产生一个 0 到 + 32767 之间的整型随机数。

例：>> IRN (0) 结果 A = 29284

>> 10 FOR N=1 TO 10 <CR>

>> 20 PRINT IRN (N) <CR>

>> 30 NEXT N

>> 40 STOP

>> RUN

结果： 29284

25801

18835

4647

9295

18846

4924

10105

20210

7652

40 STOP

再一次运算将得到同样的结果，怎么办才能得到不同的结果？利用下面函数

13. 随机化函数 RANDOMIZE

此指令用来重新设置 RND 及 IRN 函数中的哑元变量，使之产生

不同的随机数序列。

例: >> 5 RANDOMIZE
>> 10 FOR N=1 70 10
>> 20 PRINT IRN (6)
>> 30 NEXT N
>> 40 STOP

第一次>> RUN

结果: 28658
24549
16331
32663
32558
32349
31931
31095
29679
26591

第二次>> RUN

结果: 8534
17068
1368
2993
5986
12228
24200
15632
31521
30530

14. 随机函数 RND(Z)

此函数将产生一个(0, 1)范围内的随机数

即 $0 \leq \text{RND}(X) < 1$

例: >> 10 FOR N=1 TO 10
>> 20 PRINT RND(2)
>> 30 NEXT N
>> 40 STOP
>> RUN

结果: 0.3265748425
0.6176820578
0.1240303591
0.1440050664

0.2043298997

0.362674979

0.0632248243

0.7004063556

0.6699768929

0.1906821562

15. 求 N 个算术表述式的极小值 $\text{MIN}(X_1, X_2, \dots, X_N)$

其中 $X_1, \dots, X_N$ 为算术表达式, 数值变量或常数。此函数先计算各表达式的值, 然后给出最小值。

例: >> X = 6
 >> X = 6
 >> Y = 5
 >> M = MIN (X, Y)
 >> PRINT M

结果: 5

16. 求 N 个算术表达式为极大值 $\text{MAX}(X_1, X_2, \dots, X_N)$

其中 $X_1, \dots, X_N$ 为算术表达式, 数值变量或常数。与 MIN 函数一样, 先计算 N 个元素的值, 然后给出极大值。

例: >> X = 6
 >> Y = 5
 >> M = MAX (X, Y)
 >> PRIN M

结果: 6

§ 5 用户自定义函数

BASIC 允许用户自行定义一些函数, 以便参加表达式中的运算。

格式如下:

DEF FN A(X1, X2,, XN) = 算术表达式

其中 DEF FN 为自定义函数固有的格式。A 为自定义函数名, 它规定只能用一个英文字母。所以每个程序中最多只能有 26 个自定义函数。X1, X2, ..., XN 称为自定义函数的形式参数, 它可以与程序中的变量同名而不会混淆。左边出现的参数必定在右边表达式里要出现 (否则, 此参数就没有必要存在)。

先看一个例子:

```
10 X = 7
```

```
20 DEF FNA (X, Y, Z) = X * X + 3 * Y + Z / 5
```

```
30 B = 4
```

```
40 C = 18
```

```
50 D = FNA (B, X, C)
```

```
60 PRINT D
```

```
70 E = FNA (5, 4, 18)
```

```
80 PRINT E
```

```
>> RUN
```

结果: 40

40

1. 第二行为函数定义行, 函数名为 A, 形式参数为 X, Y, Z。这里的 X 与第一行中的 X 具有完全不同的含义。第二行中的 X 无实际意义, 只是当具体调用, 变成实在参数时才有实际意义。例如第五句中的 X 为实在参数。

2. 第五句中 $D = FNA(B, X, C)$ 。为调用自定义函数语句; 其中 B, X, C 为实在参数 (或实在变量)。计算时将以 B 代替第二

行中右边表达式里的 X ，将以 X 代替 Y ， C 代替 Z 。由于此时 $B = 4$
 $C = 18$ $X = 7$ 所以 $FNG(B, X, C) = B * B + Y * X + C / 6 = 40$

3. 如果在等号右边表达式中出现的变量，而在等号左边形式参数表里没有出现，则此变量有实在意义。在运算时取此变量当时的值。

例： 10 DEF FNZ(P) = P * D

20 P = 77

30 D = 5

40 E = 10

50 PRINT FNZ(E)

60 PRINT P

结果 50

77

此例中自定义函数为 $FNZ(P) = P * D$ P 为形式参数，它与下面 $P = 77$ 中的变量 P 毫无关系。在调用此自定义函数时将由实在变量 E 来代替。变量 D 在左边形式参加表中没出现，所以是实在变量，调用时取当前值 ($E = 10$)。

§ 0 预备知识

一、程序的输入:

当机器的磁盘操作系统运行时(如何使磁盘操作系统运行见本书第 册),在控制台显示器上将显示 A。此时输入 BASIC 及回车换行(即按 B, A, S, I, C 及 <RETURN> 键)。过一会控制台显示器将显示出 BASIC 运行标志》。此时读者就可以按本章各例子一行一行的输入要求的程序。每一行必须首先输入行号,就然后是此行的语句,最后,一行输完了后必须按一下 RETURN 表示回车换行。按 RETURN 键之后,将在下一行出现 >> 标志。读者可以输入下一行。

全部输入完毕后,输入立即执行指令 >> LIST, 此时, BASIC 将把刚才你输入的程序显示(打印)出来,最后一行之后,又有 >> 标志出现,这时输入立即执行指令 >> RUN 程序将执行。

注意,每输入一个新程序前,必须输入指令 >> SCR < CR (< CR > 表示 RETURN 键)

二、修改字符:

在一行语句输入时,由于疏忽按错了某个字符则可用

RUB
OUT

 键来修改。

例: >> 30 PRI M < 表示下一个输入字符的位置 > 则按一次

RUB
OUT

 键

>> 30 PRI

--

然后按 N 键和 T 键有

>> 30 PRINT

--