

算 法 语 言 扩 展 BASIC

简 明 使 用 手 册

(NOVA 计算机用)

清 华 大 学

电子工程系计算数学教研组翻印

1975. 2.

目 录

§ 1. 引言	1
§ 2. 说明部份	1
1. 数的表示	1
2. 简单变量	2
3. 数组	2
4. 算术表达式	2
5. 函数	3
6. 字符串	4
§ 3. 基本语句	5
1. 赋值语句	5
2. 循环语句	5
3. 转移语句	5
4. 转子和返回语句	5
5. 条件语句	6
6. 开关语句	6
7. 暂停和结束语句	7
8. 注释语句	7
§ 4. 输入输出语句和格式	7
1. 程序输入	7
2. 基本输出格式	8
3. 键盘输入	9
4. 自选输出格式	9
§ 5. 矩阵语句	12
§ 6. 文件的输入输出	13
1. 文件名	13

2.	文件的开启和关闭	13
3.	二进制读写	14
4.	符号读写	15
5.	矩阵文件读写	15
6.	链接语句	16
§7.	鲍盘命令	16
1.	与程序运行有关的命令	16
2.	目录管理命令	18
3.	交换信息命令	18
4.	由BASIC语句构成的命令	18
§8.	调用汇编语言子程序	18
§9.	系统生成	21
1.	RDOS 下的操作手续	21
2.	SOS 下的操作手续	24
附录1	BASIC语句简表	28
附录2	文件输入输出错误表	30
附录3	BASIC 错误表	32

§1. 引言

BASIC 是 BEGINNER'S ALL-PURPOSE

SYMBOLIC INSTRUCTION CODE (初学者通用符号指令代码) 一词的缩写, 是一种解释型对话语言。

NOVA系列计算机配备的BASIC语言分单用户的、多用户分时的和扩展的三种, 后者又有单用户或多用户、独立操作系统(SOS)或实时磁盘操作系统(RDOS)以及多用户交替使用内存(Swapping)的磁盘操作系统等五种方案。单用户BASIC 见我室1972年9月编的“算法语言BASAC使用说明”。这里介绍的扩展BASIC比单用户的功能有所增加, 主要是:

1. 处理文件的输入输出, 建立公用的和用户专用的程序库;
2. 有字符串操作;
3. 进行矩阵运算;
4. 提供自定义输出格式;
5. 扩充了键盘命令;
6. 可以调用汇编语言子程序;
7. 增加了开关语句, 条件语句中允许直接用语句, 赋值语句中LET可省略等。

分时BASIC的功能介于扩展的和单用户的二者之间, 例如没有1、4等项。

本资料参照美国DGC公司提供的文件EXTENDED BASIC USER'S MANUAL(093-00065-01)和日文版(093-500065-02)编成, 使用时请注意版本是否一致。

为阅读8、9两节需了解NOVA系列的汇编语言及操作系统; 二般运算人员可以略过。

§2. 说明部分

1. 数的表示

一律采用十进制。输出有效数字少于六位时用定数表示, 否则用浮点表示, 输入可以有七位有效数字。有浮点中间用双精度时13位有效数字。

例:	数 字	输出形式
	2000,000	2.00000 E+6
	-108,999	-108,999
	1/16	.0625

机内用NOVA软件通用的浮点表示方法，其范围 $5.4 \times 10^{-7} - 7.2 \times 10^7$ 。超过上界为溢出。

2. 简单变量

用一个字母或一个字母加一个数字表示，如A、A3、Z6等。

3. 数组

可定义一维和二维数组，名字用一个字母（不能与简单变量重复）表示，数组下界为0，上界需说明，如：

```
DIM A(15) B(2,3)
```

方括号圆括号通用。

一维组上界不能超过255。二维数组在内存中按行排列。括号中第一个数表示行标上界，第二个数表示列标上界，变量总长不能超过1024。不加说明时，BASIC解释程序给一维数组留11个位置。

运算中引用数组中变量时要在括号中注明下标（称下标变量）。下标可以是整数或表达式，后者解释程序自动取整数部分。

例： $E(0,5)$ ， $E(X*Y+Z,I)$

利用语句DIM可以改变数组的编号方式或缩小边界（但不能扩大）。这样节省的内存单元不能再作他用。

例： $DIM B[3,2]$ 将前面例子中定义的 3×4 的数组B重新编号成 4×3 的数组。

4. 算术表达式

常数、变量和函数用算术运算符连结成表达式。运算符有：

+	正号, 加	$A + B$
-	负号, 减	$A - B$
*	乘	$A * B$
/	除	A / B
↑	乘方	$A \uparrow B$

按通常顺序进行运算, 括号内的最优先。注意: *号不能省略。

5. 函数

用户自己可定义单变量函数(不超过26个), 其格式为

$$DEF \quad FN \quad Z(X) = \text{表达式}$$

Z 是A~Z中任一字母, X 是表达式中出现的形参变量。定义必须在一行内写完。它可以再引用其他函数, 但不超过四重。运算中引用已定义的函数, 需将括号中的形参变量换成已赋值的实际变量或常数。

扩展 BASIC 提供的标准函数有:

$SIN(X)$	求正弦, X 用弧度表示, $ X < 10^{15}$
$COS(X)$	求余弦, 同上
$TAN(X)$	求正切, 同上
$ATN(X)$	求反正切, 数值在 $-\pi/2$ 与 $+\pi/2$ 之间
$LOG(X)$	求自然对数, 要求 $X > 0$
$EXP(X)$	求指数, 要求 $-178 < X < 175$
$SQR(X)$	开平方根, 要求 $X \geq 0$
$ABS(X)$	求绝对值
$INT(X)$	求整数部份, 要求 $X < 2^{31}$
$RND(X)$	求出一个在0到1间的准随机数, 要求写一个宗量 但结果与之无关。若在程序中加语句

RANDOM

就使随机数发生器复位，再使用时产生的数与开始时相同，以便查错或输出检查。

$SGN(X)$ 符号函数，取值 $-1, 0, 1$
 $LEN(S\$)$ 求字符串 S 的长度
 $DET(X)$ 计算刚求过迹的那子矩阵的行列式结果与点无关。

函数 TAB 及 EOF 的功能分别见第 4、2 及第 6、3。

6、字符串（简称串）（ SOS 及第二版以前的 $RDOS$ $BASIC$ 无字符串操作）字符串常用引号标出，其中空格有数。例：“ $\Delta NOV, --- 1200 \Delta SOFTWARE \Delta$ ”（ Δ 表示空格）

字符串变量用一字母或一个字母加一个数字后加符号表示。字符串变量与字符串，或连接构成字符串表达式，用 DIM 语句说明串变量的最大长度（即容量）。

例： $DIM \quad A\$ (10), \quad R\$ (30)$

在赋值、输入、输出、条件等语句中可以引用串变量（加或不加下标）。条件语句中按字符逐个比较，ASCII 代码高者为大。

例： $A\$$ 全字符串

$A\$ (2)$ 从第二个字符开始

$A\$ (I, J)$ 从第 I 个至第 J 个字符， $J \geq I$ 。

例： 100 $DIM \quad A\$ (30), \quad B\$ (30)$

120 $LET \ A\$ = \Delta THE \Delta SQUARE \Delta OF \Delta 12 \Delta IS \Delta 4 \Delta$

140 $LET \ B\$ = A\$ (1, 15), "12", A\$ (17, 20), "14 \frac{1}{4}"$

执行后， $B\$$ 成为“ $\Delta THE \Delta SQUARE \Delta OF \Delta 12 \Delta IS \Delta 14 \frac{1}{4} \Delta$ ”。

例： 10 $LET \ A\$ = ""$ 串 A 变成空串

10 $LET \ A\$ = A$, B\$$ 串 B 附在串 A 后面

10 LET A\$ = A\$.B\$ 运算结果为字符串B, 串A原来的内容已被破坏。

§3. 基本语句

BASIC 程序由一系列语句构成, 每个语句不超过一行。语句前面加标号, 是 1~9999 间的任意整数, 不一定连续。每句结束时按回车键 (CR, 用 $\downarrow$ 表示)。一般情况下按顺序执行 (人为改变除外)。语句行内的空格无意义 (用引号括出的字符串除外)。

1. 赋值语句

变量 = 表达式

或 LET 变量 = 表达式

等式左端的各变量必须先赋值。

左端可以是简单变量、下标变量或字符串变量。

2. 循环语句

FOR 循环变量 = 表达式1 TO 表达式2 STEP 表达式3
:
NEXT 循环变量

FOR 和 NEXT 语句中用的循环变量要一致, 且必须是简单变量。表达式1为初值, 2为终值, 3为步长, 步长可正可负。若省去 STEP 及表达式3, 表示步长为+1。循环可以嵌套, 不超过四重, 但不能交叉。

3. 转移语句

GO TO 语句标号

表示人为地改变执行顺序。

4. 转子和返回语句

主程序中: GOSUB 子程序入口标号

子程序返回处: RETURN

同一个子程序可以有不同的入口和返回处, 可以自己调用自己。子程序可以嵌套, 但不得超过四重。

5. 条件语句

IF 表达式1 比较算符 表达式2 THEN 语句标号
GOTO
GOSUB

或 IF 表达式1 比较算符 表达式2 THEN 语句

如果条件成立, 执行指定标号的语句或 THEN 后面的语句, 否则往下执行。若省去比较算符与表达式2 则表示比较条件是“表达式计算结果不为零”。

比较算符有:

=	等于
<	小于
<=	小于或等于
>	大于
>=	大于或等于
<>	不等于

例: 200 IF A\$ <> "YES" GOSUB 650

230 IF X THEN IF .01 > SQR(X) GOTO 410

(若 $X \neq 0$ 且 $\sqrt{X} < .01$ 执行 410)

6. 开关语句

ON 表达式 THEN 语句标号清单
GOTO
GOSUB

根据表达式计算结果, 取整数下标, 决定转向第几个语句,

若算出的结果超过X的数目或小于等于0，开关语句无效，往下执行。

例： 30 ON X↑I GOSUB 100, 200, 300, 400

7. 暂停和结束语句

STOP

表示暂停，执行到这一条时，终端上印出：STOP AT 语句标号。

END

表示程序结束。如果不写，解释程序按逻辑顺序执行全部语句后也自动停止。

8. 注释语句

REM 注释内容

程序不执行，但保存。

§4. 输入输出语句和格式

1. 程序输入

READ 变量清单

DATA 常数清单

DATA 是非执行语句，可放在程序任意位置，通过 READ 依次赋值给各变量。后者可以是算术变量或字符串变量，但必须与相应的常数类型一致。清单中各项用“，”号隔开，每一行开头都要有 DATA 或 READ。可以设想有一个“箭头”，指明现在应读哪个数，利用语句：

RESTORE

可使箭头返回起始位置。

例: 50 READ A, A\$, B, C\$, D
 550 DATA 1, "TIME:", 10.5, "TEMPERATURE:" 43
 2 基本输出格式

PRINT
;

表达式清单

可以打印数值、字符串及其组合 数的表示方式见 2.1。
 解释程序提供一种标准的输出格式，每行打印5个数。数据用
 “,”号隔开就表示使用这种格式。

例: 10 LET X = 5
 20 ; X, (X * 2) 6, X - 100

输出形式:

0	14	23	← 字符位置，每段14个字符
↓	↓	↓	
5	1.00100E+6	-95	← 数据

若用“;”号，则空一格打印下一个数据。语句末尾的“,”
 或“;”号表示不换行并指明下一个数据的打印位置。不写打印
 内容表示空一行。

利用函数

TAB (表达式); 打印内容

可以自选间隔。解释程序取表达式整数部分作为开始打印下一个
 数据的位置(以最左端为零算起)。若打字头已处于该位置右端，
 则 TAB 无效，按“;”号执行。若算出的数值超过一行允许的
 字符数 n ，则取按 n 的模。TAB 表达式后的“;”号不能省略。

例: 10 DATA 5, -7 9
 20 READ A, B, C
 30 TAB(5); A; TAB(10); B; TAB(15); C

输出结果

0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95

← 字符位置

← 数据

3. 键盘输入

INPUT 变量清单

执行这一语句时，终端上印出“?”号，等待键盘输入数据。输入完每个数据后加“,”号或回车。对应一个INPUT语句的数据敲完后必须按回车。可以输入数字或字符串。但必须与程序中要求的变量类型一致。输入字符串时不加引号，这时第一个空格无效，但“,”号必须放在引号内。常在INPUT语句前加一行打印语句，说明需输入的内容。

例: 40 ; "VALUES OF A, A\$, B"
45 INPUT A, A\$, B
50 ;
55 ; A, B, A\$

打印式样:

VALUES OF A, A\$, B: 2.4, ITEM-1.6
2.4 ~ 1.6 ITEM

4. 自由输出格式

PRINT USING "格式字符串"。来打印清单;

1. 格式字符串由一个或多个格式字与字符串组成。格式字的专用字符是 + - # . / 等。出现一个非专用字符就表示格式字结束。

表示 0-9 中的一数字;

二 为小数类，前面加一个零，后面由规定位做，超过时捨入；

十，一 表示固定位置（最前或最后）符号，“+”表示正负号都印，“-”表示只印负号，正号留空格；

十，十，二—— 或更多个连串，表示做浮动，置于有效数字前，“+”“-”号含义同上，从第二个符号开始起代替井号作用；

三 为三位整数或任何数的符号号，若前面为负数则加一空格；

↑ 是读名字表示小数点位数，印成五样为五位数。

若级数超过给定的位数表示的范围，印出*号，其数目与格式字符数相同，这里删去了对取用元用为美元符号使用说明。

例：	格式字	数据	表示	说明
	井井井井井	30	△△30	向正时齐
	井井	1.95	△2	捨入，符号无效
	井井	387	*未	溢出
	井井井井井井	29.347	△△△29.35	小数第三位捨入
	井井井井	20	20.00	
	+ 井井井井	1.01	+△1.01	符号在固定位置
	井井井井井井	0.01	△△0.01	小数类前加一个零
	井井井井井井	234	234.00	
	井井井井井井	-234	234.00-	
	井井井井井井	-2	△-2.00	符号浮动，二、三位代替井

格式字	数据	表示	说明
+#.###.##	30.1	+△△△?△60	逗号空格代替
++##,###	33	△+000003	非位置的保留,逗号也保留。

+ + # # . # # ↑ ↑ ↑ 6002.33 + 600.24 E + 01

2) 规定格式字后, 在表的格式符 TAB, “,”、“;”等号失效, 但语句末尾的“,”或“;”号仍表示不换行。

3) 有的专用字符串出现时作普通字符用, 两个以上连续出现时作专用字符, 解释程序可以区别。

例: “ANSWER IS -85” “-”号作普通字符, 后面是普通字符 85

“ANSWER IS -###” “-”号是格式字符, 后面跟是专用符 #

4) 若输出数据的位数超过格式字个数时, 就按格式字输出。

例: “###②##.## PER ###” 共三个格式字。

第1、4、7等数据按第一种格式输出, 余类推。

5) 表达式清单中出现字符串时, 解释程序也按格式字中的字符个数输出, 不足的补以空格, 超过的不予删去。

例: 10 PRINT USING “##.##” “TEST”, “SEVENTY”
输出结果: TEST △△ SEVEVT.

这里两次使用由6个字符形成的格式字, 第一个字符串含7个字符, 补两个空格, 第二个串7个字符, 删去一个。

例: 10 USING “A[##]△=△###.##”, I, A[I]
可能的输出结果: 4 [1] = 17.9

3.5. 矩阵语句 语句格式

说 明

1) MAT READ A, B
或 MAT READ A(3), B(2,5)

从数据区给矩阵赋值

2) MAT INPUT A, B

从键盘给矩阵赋值

3) MAT PRINT A, B

打印矩阵

4) MAT A = B

矩阵: $B \rightarrow A$

5) MAT A = B + C

矩阵加法, 允许 $A \leftarrow A$

MAT A = B - C

+ C, 但不允许 $A = B + C - D$

只法有一个运算符

6) MAT A = B * C

矩阵乘法, 不允许 $A = A * B$

7) MAT A = (表达式) * B

标量乘, 括号不能省略

8) MAT A = INV(B)

求逆, $B^{-1} \rightarrow A$

9) MAT A = TRN(B)

转置, $\widetilde{B} \rightarrow A$, 不允许

MAT A = TRN(A)

10) MAT A = ZER

置全“0”矩阵

11) MAT A = CON

置全“1”矩阵

12) MAT A = IDN

置单位矩阵

需要注意:

1) 矩阵可作为一维数组用 DIM 语句定义, 但矩阵下标从“1”开始“0”行与“0”列无效。一维矩阵也要作为二维数组说明。

2) 对已定义的矩阵, 在两项中可只写矩阵名字, 否则应在括号中说明矩阵大小。第一个数表示行数, 第二个表示列数。只写一个数表示是一列矩阵。对于非正方矩阵第12条保证右下角元素为1。

例: 5 DIM A (3, 2)

10 MAT A = IDN

结果是

0	0
1	0
0	1

3) 矩阵运算应满足通常的要求。如相同维数的矩阵才能加、减，正方形矩阵才能求逆等。

4) 用 INPUT 语句输入时，每个矩阵前应加逗号，每行元素敲完后按回车，按标准格式输出时每行打印5个矩阵元，若矩阵每行超过5个元素，则每行元素印完后空一行，再从头开始印。

3.6 文件的输入输出

所谓文件是指用文件名命名的某些信息集合。本系统 BASIC 可与实时磁盘操作系统 (RDOS) 或独立操作系统 (DOS) 结合，具备处理文件输入输出的功能。DOS 只能把可设备作为文件，RDOS 还可以处理磁盘文件。磁盘文件的名字都列在文件目录中，分公用的程序库目录和用户专用目录。

1. 文件名

BASIC 程序语句中使用的文件名可以是字符串常量或字符串变量。磁盘命令中只能用常量。

例：常量 "A\$", "A\$.1", "A\$.SR"

变量 F\$, F\$(2), FS(1, J+1)

文件名由字母、数字和符号组成，不超过6个字符，可以在“.”号后加后缀。外扩设备是特殊文件，其名字由符号串开始（除磁带外），非特殊文件都是磁盘文件。文件名必须用引号括出。

2. 文件的开启和关闭

文件在读写前要打开，使它与一个固定的软件通道连接，并指定使用方式，每个用户最多能打开8个文件，编号0—7。

`OPEN FILE` [数表达式 1, 数表达式 2], 文件名

数表达式 1 计算结果为通道编号 (0-7 中的一个数),
数表达式 2 为 0-3 中的一个数, 规定使用方式, 其中:

0 — 随机存取, 只适用于磁盘文件, 用户目录中无此文件
时补入这个名字;

1 — 向磁盘或输出设备写一个新文件: 代替原有同名文件
的内容;

2 — 在原有文件后续一个新文件, 目录中无此文件时补入;

3 — 由磁盘或输入设备读入一个文件, 目录中必须有这个
文件。

例: 110 `OPEN FILE` [0, 1] "TEST.1"

120 `OPEN FILE` [1, 3], "\$PTR"

130 `OPEN FILE` [1, 1], \$

输入输出结束后要立即关闭文件, 切断通道

`CLOSE FILE` [数表达式 1]。

数表达式 1 是通道编号, 改变文件使用方式也要先关后开。

3. 二进制读写

`READ FILE` [数表达式 1], 变量清单

`READ FILE` [数表达式 1, 数表达式 3], 变量清单

将数据以二进制形式赋予数字或字符串变量。数表达式 1
是通道编号, 数表达式 3 是随机记录个数 (下同)。第一种语句适用
于串联文件, 事先按方式 3 打开, 第二种适用于随机存取的文件。
数据类型必须与变量类型一致。

利用函数

`EOF` (通道编号)

可以判断是否有文件结束标志 (有为 1, 无为 0), 再利用条