

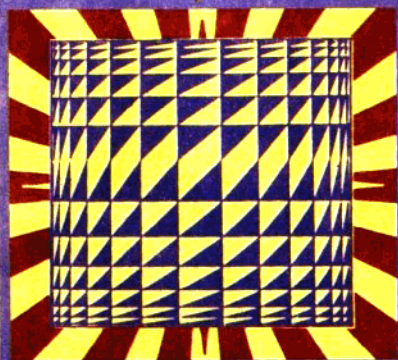


中央广播电视大学教材

BASIC语言实验

BASIC
YUYAN SHIYAN

徐孝凯 编



中央广播电视大学出版社

BASIC 语言实验

徐孝凯 编

中央广播电视大学出版社

(京)新登字 163 号

图书在版编目(CIP)数据

BASIC 语言实验/徐孝凯编. —北京:中央广播电视大学出版社,1995.10

ISBN 7 304 01173 4

I. B… I. 徐… III. BASIC 语言 电视大学 教材 N. TP312BA

中国版本图书馆 CIP 数据核字(95)第 20200 号

BASIC 语言实验

徐孝凯 编

中央广播电视大学出版社出版

社址:北京市复兴门内大街 160 号 邮编:100031

北京印刷二厂印刷 新华书店北京发行所发行

开本 787×1092 1/16 印张:13.25 千字:326

1995 年 8 月第 1 版 1995 年 10 月第 1 次印刷

印数 1—20000

定价 10.30 元

ISBN 7 304 01173 4/TP·59

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、打印机和磁盘等的操作，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，它既可以与谭浩强主编的 BASIC 语言系列教材配套使用，也可以单独使用。本书以 BASIC 和 GWBASIC 为实验背景，给出了九个实验。实验一为键盘操作，介绍了键盘的使用以及如何输入、修改和运行一个 BASIC 程序；实验二、三、四分别为简单程序、分支程序和循环程序实验。在每一个实验中，介绍了相应类型程序的设计方法以及跟踪执行的过程；实验五为数组程序实验，介绍了几种基本类型的数组程序的设计方法；实验六为子程序实验，介绍了如何利用子程序进行模块化程序设计；实验七为绘图实验，介绍了绘图的基本语句和简单绘图举例；实验八和九为顺序文件和随机文件实验，介绍了与文件有关的基本命令和语句以及如何进行简单的文件操作；实验十为 DOS 操作实验，介绍了 DOS 常用命令的功能和行编辑程序 EDLIN 的功能以及上机操作的过程。

本书给出了四个附录。附录 A 为 ASCII 代码表；附录 B 为 BASIC 命令、语句和函数一览表，以便读者查阅；附录 C 为 IBM-PC 机 BASIC 错误信息表，根据此表能够帮助你处理运行中出现的错误；附录 D 为 WPS 文字处理系统简介，WPS 为当前最流行的中文文字处理系统之一，它集编辑、排版和打印为一体，而且简单易学、操作方便，所以把它作为本课程的补充知识，以附录的形式给出，以便同学们了解和掌握，相信一定会受到欢迎。

本书除作为中央广播电视大学 BASIC 语言课的实验教材之外，亦可作为其它各级、各类学校的 BASIC 语言实验教材或教学参考书，还可作为学习家用电脑的 BASIC 语言上机指导书。

由于编者水平所限，编写时间又仓促，不当之处在所难免，希望广大读者和同行专家批评指正。

编 者

1995 年 6 月

实验周次与机时分配表

序号	实验名称	周次	机时数
实验一	键盘操作	4	3
实验二	简单程序	6	3
实验三	分支程序	8	2
实验四	循环程序	10	3
实验五	数组程序	11	2
实验六	子程序	12	3
实验七	绘图	14	3
实验八	程序文件和顺序文件	16	3
实验九	随机文件	17	2
实验十	DOS 操作	18	3

目 录

BASIC 语言概述	(1)
实验一 键盘操作	(17)
实验二 简单程序	(31)
实验三 分支程序	(42)
实验四 循环程序	(58)
实验五 数组程序	(76)
实验六 子程序	(90)
实验七 绘图	(101)
实验八 程序文件和顺序文件	(117)
实验九 随机文件	(133)
实验十 DOS 操作	(150)
附录 A ASCII 代码表	(168)
附录 B BASIC 命令、语句和函数一览表	(170)
附录 C IBM-PC BASIC 错误信息表	(175)
附录 D WPS 文字处理系统简介	(181)

BASIC 语言概述

BASIC 语言是社会上最为流行的一种计算机高级语言，几乎在所有的计算机上都配有这种语言。它具有简单易学、应用范围广、便于上机操作等优点，因此深受广大计算机用户，特别是初学者的欢迎和喜爱。

现在社会上流行着各种不同版本的 BASIC 语言（即 BASIC 系统程序），它们主要有 quick-BASIC, turbo-BASIC, true-BASIC, GW-BASIC, IBMPC-BASIC 等。每一种版本都各有特色和优势，但多数语句和函数是相同的，程序设计思想是相通的。本实验教材为了同主教材相配合，采用 IBMPC-BASIC 语言为蓝本进行讨论。

一、语言构成

BASIC 语言同其它计算机语言和人类自然语言一样，也是由字、词、句所组成。

1. BASIC 语言的基本字符

任何一种语言都有规定的基本字符，如汉语的基本字符是汉字，英语的基本字符是英文字母等。BASIC 语言的基本字符如下：

(1) 英文字母（大小写各 26 个） A~Z, a~z。除在字符串中大小写不同外，其余地方的大写字母与小写字母其效果相同。

(2) 标点符号（8 个）：

0	小数字	□	空格
,	逗号	;	分号
:	冒号	"	双引号
(左括号)	右括号

(3) 数字符号（10 个）：0~9

(4) 算术运算符（5 个）：

+	加号或正号	-	减号或负号
*	乘号	/	除号
^	或 ↑		乘方符号

(5) 关系运算符（6 个）：

<	小于号	>	大于号
<=	小于等于号	>=	大于等于号
<> 或 ><	不等号	=	等号（又作赋值语句中的赋值号）

(6) 逻辑运算符（3 个）：

AND 与
OR 或
NOT 非

(7) 变量类型说明符号 (4 个):

% 整型	! 单精度型
# 双精度型	\$ 字符串型

(8) 其它字符和汉字

在西文 DOS 操作系统下,可在字符串中使用键盘上存在的任何其它字符;在汉字 DOS 操作系统下,可在字符串中使用国际区位码表中任何字符和汉字。

2. BASIC 语言的基本单词

由 BASIC 语言的基本字符按照约定的组词规则排列起来就构成了 BASIC 语言的基本成分——单词。BASIC 语言所定义的单词包括语句定义符、常量、变量、函数和表达式。

语句定义符又叫做语句标识符,它通常用英语单词或英语单词的缩写来表示。如 PRINT 表示打印语句的定义符,LET 表示赋值语句的定义符,END 表示结束语句的定义符等。

BASIC 常量包括常数和字符串两种。如 15, -327, 0.128, 9.42, -368.25, +69400 等都属于常数,“APPLE”, “IBM-PC”, “A1+B2=”, “工资清单” 等都属于字符串。

BASIC 变量包括数值变量和字符串变量两种,每种变量又分为简单变量和下标变量两种表示形式。数值型简单变量同数学上的变量具有相同的含义和表示方法,如 R, F, T1, NAM 等都可作为数值型简单变量来使用。

BASIC 函数包括数值函数和字符串函数两种。数值函数同数学上经常使用的函数具有相同的含义,但表示方法有所不同。如 BASIC 语言中的 $\text{EXP}(x)$ 函数表示数学上的 e^x , $\text{SIN}(x)$ 函数表示数学上的 $\sin x$, $\text{ABS}(x)$ 函数表示数学上的 $|x|$ 等。

BASIC 表达式包括算术表达式、字符串表达式和逻辑表达式三种。其中算术表达式同数学上的算式大致相同,如 $2 * A - B$, $(x + y) \uparrow 2$, $5 * \text{SIN}(x/4)$ 等都是合法的算术表达式。

3. BASIC 语言的基本语句

由 BASIC 语言的基本单词(有时要适当地插入标点符号)按照一定的句法规则排列起来就构成了 BASIC 语句。BASIC 语言就是由具有不同功能的语句所组成。IBM-PC 机上的 BASIC 语言大致具有上百条语句,但最基本的语句如下:

- (1) 数组说明语句 (DIM)
- (2) 赋值语句 (LET)
- (3) 打印语句 (PRINT)
- (4) 键盘输入语句 (INPUT)
- (5) 读数/置数语句 (READ/DATA)
- (6) 结束语句 (END)
- (7) 无条件转向语句 (GOTO)
- (8) 选择转向语句 (ON-GOTO)
- (9) 条件语句 (IF-THEN-ELSE)
- (10) 循环类语句 (FOR/NEXT、WHILE/WEND)
- (11) 子程序类语句 (GOSUB、RETURN、ON-GOSUB)
- (12) 绘图类语句 (COLOR、PSET、LINE 等)
- (13) 文件类语句 (OPEN、CLOSE、INPUT#、WRITE# 等)

4. BASIC 语句的基本结构

一条 BASIC 语句一般由语句定义符和语句体两部分组成。语句定义符通常处于一条语句的开头，该语句的功能可以从语句定义符的英文含义中反映出来，语句体是语句的执行部分，指出具体要进行的操作。例如有四条语句为：

- ① LET X=20
- ② INPUT A, B
- ③ PRINT X, X↑2, EXP(X)
- ④ END

第一条语句的定义符为 LET，英文的含义为赋值，因此该语句叫做赋值语句，或直接叫做 LET 语句，它的功能是把常数 20 赋给变量 X。第二条语句的定义符为 INPUT，英文含义为输入，因此该语句叫做输入语句，或直接叫做 INPUT 语句，它的功能是把用户从键盘上输入的两个常数分别送给变量 A 和 B。第三条语句的定义符为 PRINT，英文的含义为打印，因此该语句叫做打印语句，或直接叫做 PRINT 语句，它的功能是把语句体中三个数据项的值打印在屏幕上。第四条语句只有语句定义符，没有语句体，该语句叫做结束语句，或直接叫做 END 语句，它的功能是使计算机停止程序的执行过程，回到 BASIC 命令状态。

二、程序书写规则

根据解决问题的实际需要，选用若干条合适的 BASIC 语句，就可以编写出一个 BASIC 语言程序，简称 BASIC 程序。下面就是一个根据圆的半径 R，求圆的周长 P 和面积 S 的程序。

```
10 INPUT R
20 LET P=2*3.14159*R
30 LET S=3.14159*R*R
40 PRINT P, S
50 END
```

结合这个程序的例子，可总结出 BASIC 程序的书写规则如下：

1. 一个 BASIC 程序被书写成若干个程序行。如上面的程序共有五个程序行。BASIC 程序中的每个程序行最多允许 255 个字符，若使用了汉字，则每个汉字占用两个字符的位置。如上面程序的第一个程序行共有 11 个字符，其中包括该行末尾隐含的一个回车字符。

2. 每个程序行的开始都有一个正整数，我们把它叫做程序行的行号或标号。如在上面的程序行里，各程序行的行号依次为 10, 20, 30, 40, 50。BASIC 程序中每个程序行的行号可取 1~65529 之间的任意正整数。

相邻程序行的行号不一定使用连续的正整数。在上面的程序里，各行号都是以正整数 10 为间隔的，每两个相邻的程序行之间都可以插入最多九个程序行，这就为修改程序时插入新的程序行提供了方便。

计算机执行程序时，一般是按照行号从小到大的顺序执行的。如执行上面的程序时，用行号表示其执行过程为：10→20→30→40→50。即首先执行 10 程序行（即以 10 为行号的程序行），接着执行 20 程序行（即以 20 为行号的程序行），依此类推，最后执行 50 程序行。

3. 每个程序行里可以书写若干条 BASIC 语句（一般为了简明起见，每行只书写一条语句），语句和语句之间必须用冒号分开。在上面的程序中，每个程序行只书写一条语句，在这

种情况下，我们可用该语句所在的行号来直接称呼这条语句，如可把 10 程序行中的 INPUT 语句称作为 10 语句。在下面的一个程序行里共书写有三条语句：

```
100 □ LET □ A=2; LET □ B=3; PRINT □ A+B
```

当计算机执行含有多条语句的程序行时，将从左向右依次执行每条语句。

4. 书写程序时要适当加入空格。在每个程序行的行号和语句之间、语句定义符和语句体之间都需要加入至少一个空格，这样能够把程序行中的各个部分明显地区别开来，便于阅读和修改程序。在上面的程序中，为了清楚起见，在需要留有空格的地方都用“□”表示出来，实际上只要在这些地方留出空位置即可。

5. 在每个程序的最后一行，一般书写一条 END 语句。END 语句是结束语句，计算机执行 END 语句后，整个程序的执行过程就结束了。当然 END 语句同其它语句一样，可以书写在程序中的任何地方，可以在一个程序中多处使用。

三、BASIC 常量

BASIC 常量、变量、函数和表达式统称为 BASIC 运算对象，它们几乎出现在所有的 BASIC 语句中，掌握它们是学习 BASIC 语言的基础。

BASIC 常量包括常数和字符串两种。

1. 常数

BASIC 常数有两种表示法，其一是采用十进制数的日常表示法，又叫定点表示法；其二是采用带有指数的科学计数法，又叫做浮点表示法。

(1) 定点表示法

定点表示法是指只允许使用十个数字符号 0~9、+（正号）、-（负号）和 .（小数点）表示 BASIC 常数的方法。采用定点表示法表示的 BASIC 常数又叫做定点数。如 326，-0.54，+900，38.25，0.001，-20 等都是采用定点表示法表示的 BASIC 常数（即定点数）。但数学上的常数 90,000,000 不能直接作为 BASIC 常数使用，因为其中包含有逗号，若书写成 BASIC 常数，则定点表示为 90000000。

(2) 浮点表示法

浮点表示法是指在定点表示法的基础上增加以 10 为底的指数部分来表示 BASIC 常数的方法。它同数学上采用科学记数法（又叫指数记数法）表示常数的方法基本相同，所不同的是：在浮点表示法中用指数符号 E 来表示乘方的底数 10。

例如：

数学上的表示	BASIC 浮点表示
-0.17045 × 10 ⁴	-0.17045E4
34.32 × 10 ⁻¹⁵	34.32E-15
1.25 × 10 ⁹	1.25E9
-6.4 × 10 ⁻⁸	-6.4E-8
100000	1E5
0.000147	1.47E-4
150	1.5E2

以浮点表示法表示的 BASIC 常数叫做浮点数。一个浮点数由三个部分组成：左边是一个定点数，叫做该浮点数的尾数；中间为指数符号 E，用来表示乘方的底数 10；右边是一个整数，它可以为正数或负数，可以具有一位或两位数字，被称为该浮点数的指数。如浮点数 $-1.7045E3$ 的尾数是 -1.7045 ，指数是 3。在一个浮点数中，若尾数或指数为正时，其正号可以省略。

将数学上的常数表示成 BASIC 浮点数时，其小数点可以放在尾数的任何位置上。如常数 325×10^6 可以表示成的 BASIC 浮点数为： $325E6$ ， $32.5E7$ ， $3.25E8$ 等。但是，若表示成的浮点数，其小数点前面（即左边）只有一位且该位不为 0 数字，则称此浮点数叫做规格化的浮点数。如 $3.25E8$ ， $1E5$ ， $2.4E-6$ ， $-4.806E12$ 等都是规格化的浮点数。

因此，BASIC 常数包括定点数和浮点数两种。在一般的程序中，大都使用定点数，较少使用浮点数。但对于数学上绝对值较大或较小的数，用浮点表示较为方便。如要表示 11 亿这个数，用定点表示为 1100000000，共有 10 位数字，若用浮点表示则为 $11E8$ ，只使用了四个字符，显得简单明了。

2. 字符串

在 BASIC 语言中，英文字母、空格、十进制数字符号、标点符号、运算符号等统称为字符，或者更广义地说，把能够从键盘上输入的一切符号或汉字统称为字符，用双引号把一个或若干个字符引起来就表示为一个字符串。如“STRING”、“inter-80386”、“A:XXK.DAT”、“姓名：”等都是字符串。

对于一个字符串，其两边的双引号只作为该串的起止定界符，并不属于字符串中的内容。因此，在打印一个字符串时，只顺序打印出双引号内的所有字符，并不打印出双引号，把一个字符串赋值给字符串变量时，变量中不保留该字符串两边的双引号。

字符串可以是一个空串，即串中不包含任何字符；也可以是由若干个字符组成的串，但字符个数最多不允许超过 255 个。

我们知道，计算机是采用二进制形式来存储一切信息的。在 BASIC 语言中，计算机保存一个常数时，首先把它转换成对应的二进制数，然后再把这个二进制数存储起来；在保存一个字符串时，首先对其中的每个字符顺序查找出对应的二进制编码，然后再将这些编码依次存储起来。所谓二进制编码就是按照一定的规则用一串二进制数字来表示一个字符，用不同的一串二进制数字来表示不同的字符。如用 01000010 表示字符 B，用 01000011 表示字符 C 等。

世界上对字符的二进制编码的种类很多，在 BASIC 语言中采用的是美国信息交换标准代码，简称 ASCII 码。在主教材的附录中给出了字符的 ASCII 码表。如字符 \$ 的 ASCII 码为 00100100，表示成十进制数为 36，表示成十六进制数为 24。一般在使用一个字符的 ASCII 码时，都直接采用它的十进制表示，如直接称字符 4 的 ASCII 码为 52，字符 B 的 ASCII 为 66 等。

四、BASIC 变量

在 BASIC 语言中，几乎所有的语句都要使用变量。如在 $LET S=3.14159 * R * R$ 这个赋值语句中，就使用了两个变量 R 和 S，它们分别表示圆的半径和面积。在同一个程序中，为了区别不同的变量，就必须给每个变量取互不相同的名字。如要编写一个根据圆柱体的底面

半径和高度计算底面积和体积的程序，就需要使用四个变量，需要对它们取四个不同的名字，假定分别用 R, H, S 和 V 来表示圆柱体的底面半径、高度、底面积和体积。

在 BASIC 语言中，给变量取名的规定如下：

(1) 以英文字母开头的字母串或字母数字串（字母大、小写等效，本书为统一起见均采用大写）可以作为变量名。如 A、X、B1、NAM、WT3 等均可作为变量名，用以表示不同的变量。

(2) 禁止使用 BASIC 语句定义符、函数定义符（即函数名）、命令定义符等（这些统称为 BASIC 保留字）作为变量的名字。如使用 LET、PRINT、IF、EXP、RUN、LIST 等作为变量名是错误的，因为这些单词（即 BASIC 保留字）在 BASIC 解释程序或编译程序中已经具有了专门的含义。

BASIC 变量包括数值变量和字符串变量（简称串变量）两种类型，每种类型的变量又具有简单变量和下标变量两种表示形式，因此，BASIC 变量共具有四种表示形式：数值型简单变量、数值型下标变量、字符串型简单变量和字符串型下标变量。

数值型简单变量用来表示一个数值。凡符合上面命名规定的所有变量名均可直接作为数值型简单变量来使用，如 I, N, M1, C2 等。

字符串型简单变量用来表示一个字符串，若在符合上面命名规定的变量名后面缀上字符 \$，则表示该变量是一个字符串型简单变量，如 A\$, C1\$, NUM\$ 等。

为了搞清楚什么叫下标变量，首先看一下具有 n 个未知量 m 个方程的线性方程组问题：

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \end{cases}$$

若令：

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \quad X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

则将线性方程组改写成矩阵形式为：

$$AX=B$$

其中 A 称为系数矩阵，它具有 m 行 n 列，共 $m \times n$ 个元素；X 称为未知量矩阵，它具有 n 个元素；B 称为常数项矩阵，它具有 m 个元素。

在 BASIC 语言中，为了进行矩阵运算，必须通过数组说明语句来定义数组，每个数组就是数学上的一个矩阵。下面就是一条数组说明语句：

`DIM A(M,N),X(N),B(M)`

DIM 为数组说明语句的定义符。A, X, B 分别为数组名（或者叫做数组变量），凡符号变量命名规定的任何变量名均可作为数组名。数组名后括号内数据项的个数表示该数组的维数，当只有一个数据项时，称为一维数组，当有两个用逗号分开的数据项时，称为二维数组，依此类推。在这里 X, B 为一维数组，A 为二维数组。数组名后括号内每个数据项可以是常数，也可以是变量，但其值必须是正整数，此值就是该维下标的上界值，各维下标的下界值隐含

为 0。如上面的 DIM 语句定义了数组 A 的第一维下标的上界值为 M，第二维下标的上界值为 N，第一维下标应从 0 变化到 M，第二维下标应从 0 变化到 N，则 A 中共包含 $(M+1) \times (N+1)$ 个元素，它们分别表示为：

$$\begin{array}{cccc} A(0,0) & A(0,1) & \cdots & A(0,N) \\ A(1,0) & A(1,1) & \cdots & A(1,N) \\ \vdots & \vdots & \vdots & \vdots \\ A(M,0) & A(M,1) & \cdots & A(M,N) \end{array}$$

同理，上面的 DIM 语句定义了数组 X 的下标上界值为 N，数组 B 的下标上界值为 M，它们分别包含 $(N+1)$ 和 $(M+1)$ 个元素：

$$\begin{array}{cccc} X(0) & X(1) & X(2) & \cdots & X(N) \\ B(0) & B(1) & B(2) & \cdots & B(M) \end{array}$$

若除了数组 A 中的第 0 行和第 0 列元素外，则正好可用来存储上面的系数矩阵 A，若除了数组 X 和 B 中的下标为 0 的元素外，则正好可用来存储上面的矩阵 X 和 B。

上面 DIM 语句定义的两个数组都是数值型数组，若需要定义字符串型数组，则只要在 DIM 语句里的数组名后面缀上字符串类型说明符 \$ 即可。如：

$$\text{DIM C\$}(5), \text{D\$}(2,3), \text{K}(10)$$

这个语句定义了 C\$ 和 D\$ 为字符串型数组，K 为数值型数组。C\$ 数组中共包含六个元素，它们是：

$$\text{C\$}(0) \quad \text{C\$}(1) \quad \text{C\$}(2) \quad \text{C\$}(3) \quad \text{C\$}(4) \quad \text{C\$}(5)$$

D\$ 数组中共包含 12 个元素，它们是：

$$\begin{array}{cccc} \text{D\$}(0,0) & \text{D\$}(0,1) & \text{D\$}(0,2) & \text{D\$}(0,3) \\ \text{D\$}(1,0) & \text{D\$}(1,1) & \text{D\$}(1,2) & \text{D\$}(1,3) \\ \text{D\$}(2,0) & \text{D\$}(2,1) & \text{D\$}(2,2) & \text{D\$}(2,3) \end{array}$$

K 数组共包含 11 个元素，它们是：

$$\text{K}(0) \quad \text{K}(1) \quad \text{K}(2) \quad \cdots \quad \text{K}(10)$$

在 BASIC 语言中，把数组的每个元素叫做下标变量，其中二维数组中的元素叫做双下标变量，一维数组中的元素叫做单下标变量。数值型数组中的下标变量叫做数值型下标变量，每个数值型下标变量同每个数值型简单变量一样，都能够用来表示一个数值；字符串型数组中的下标变量叫做字符串型下标变量，每个字符串型下标变量同每个字符串型简单变量一样，都能够用来表示一个串值（即一个字符串）。同时，在程序中，下标变量可以出现在简单变量所能够出现的任何地方。

在 BASIC 程序里，数组只有经过 DIM 语句说明后，其下标变量才能使用在以后的语句中，但是，若数组中的每维下标的上界值不超过 10 时，也可以不经过数组说明，而直接使用该数组中的下标变量。

五、BASIC 函数

BASIC 函数分为数值函数和字符串函数两种类型。

1. 数值函数

在 BASIC 语言中,定义了 11 种常用的数值函数。所谓数值函数是指其自变量的值和函数的值均为数值的一类函数。表 1 列出了每种数值函数的名称、表示法及其在数学上的含义。

表 1 数值函数表

函数名称	表示法	数学上的含义
绝对值函数	ABS(X)	$ x $
平方根函数	SQR(X)	$\sqrt{x} (x \geq 0)$
指数函数	EXP(X)	$e^x (e = 2.71828)$
自然对数函数	LOG(X)	$\ln x (x > 0)$
正弦函数	SIN(X)	$\sin x$
余弦函数	COS(X)	$\cos x$
正切函数	TAN(X)	$\operatorname{tg} x (x \neq \frac{n\pi}{2}, n \text{ 为奇数})$
反正切函数	ATN(X)	$\operatorname{arctg} x$
取整函数	INT(X)	求不大于 x 的最大整数
随机函数	RND(X)	由计算机产生一个随机数
符号函数	SGN(X)	$\operatorname{SGN}(x) = \begin{cases} 1 & (x > 0) \\ 0 & (x = 0) \\ -1 & (x < 0) \end{cases}$

结合数值函数表作以下几点说明:

(1) 每个函数都有自己的定义符,如绝对值函数的定义符为 ABS,平方根函数的定义符为 SQR 等。函数的定义符就是函数的名字,如可把正弦函数叫做 SIN 函数,把取整函数叫做 INT 函数。

(2) 函数的自变量可以是一个常数、一个数值变量、一个数值函数或者一个算术表达式,在上面的函数表中统一用形式参数 X 表示出来。自变量必须用圆括号括起来。例如:

SIN (0.5)	自变量是一个常数
TAN (A)	自变量是一个数值变量
EXP (SQR (5))	EXP 函数的自变量是一个数值函数
INT (3 * N - 2)	自变量是一个算术表达式

(3) 三角函数 SIN、COS、TAN 的自变量值的单位是弧度,反正切函数 ATN 值的单位也是弧度。度数化为弧度的公式为:

$$1^\circ = 3.14159/180 \text{ (弧度)}$$

如 $\sin 30^\circ$ 在 BASIC 语言中表示为 $\operatorname{SIN}(30 * 3.14159/180)$ 。

(4) EXP 函数是以 e 为底的对数函数,如 e^{x+y} 可表示为 $\operatorname{EXP}(X+Y)$ 。当指数运算的底数不是 e 时,可采用乘方运算求值,如 10^n 可表示为 $10 \uparrow N$, x^5 可表示为 $X \uparrow 5$ 。

(5) LOG 函数是以 e 为底的对数函数,如数学上的 $\ln(x+1)$ 在 BASIC 语言中表示为 $\operatorname{LOG}(X+1)$ 。如果要计算以 10 为底的对数,则应先利用对数换底公式 $\log_{10} x = \frac{\ln x}{\ln 10}$ 把以 10 为

底的对数转换为 e 为底的对数,然后再利用 LOG 函数求值。如 $\log_{10}(3x^2+5)$ 应表示为 $\text{LOG}(3 * x \uparrow 2 + 5) / \text{LOG}(10)$ 。

(6) SGN 函数叫做符号函数,它的求值公式为:

$$\text{SGN}(x) = \begin{cases} 1 & (x > 0) \\ 0 & (x = 0) \\ -1 & (x < 0) \end{cases}$$

也就是说,自变量的值为正时,函数值为 1;自变量的值为 0 时,函数值为 0;自变量的值为负时,函数值为 -1。如:

$$\text{SGN}(3.5) = 1 \quad \text{SGN}(0) = 0 \quad \text{SGN}(-10) = -1$$

(7) INT 函数叫做取整函数,它的值是不大于自变量值的最大整数。由此可知,当自变量值为正数时,它的值就是这个正数的整数部分;当自变量为负数时,它的值就是这个负数的整数部分加上 -1 (但若这个负数为负整数时,不需要加上 -1)。如:

$$\begin{aligned} \text{INT}(15) &= 15 & \text{INT}(4.63) &= 4 \\ \text{INT}(-1) &= -1 & \text{INT}(-4.2) &= -5 \\ \text{INT}(20/3) &= 6 & \text{INT}(-8/3) &= -3 \end{aligned}$$

(8) RND 函数叫做随机函数,它的值由计算机随机产生,与它的自变量值无关,但自变量值必须大于 0 (一般为了方便起见,用 1 表示自变量)。当计算机在执行 BASIC 程序的过程中碰到 RND 函数时,将自动产生出一个在 0 和 1 之间 (不包括 0 和 1 在内) 的具有六位 (或七位) 有效数字的随机小数作为 RND 的函数值。如:

```
10 PRINT RND(1),RND(1)
20 PRINT RND(1),RND(1)
30 END
RND
.1213501          .651861
.8688611          .7297625
```

程序输入并运行后,执行 10 语句时打印出第一行的两个数据,执行 20 语句时打印出第二行的两个数据。从打印结果可以看出:RND 函数的值是由计算机随机产生的,与自变量值无关。

在 BASIC 程序中利用 $\text{INT}(\text{RND}(1) * 10)$ 能够产生出 0~9 中的一个随机整数,这很容易得到证明:

$$\begin{aligned} \therefore 0 < \text{RND}(1) < 1 \\ \therefore 0 < \text{RND}(1) * 10 < 10 \\ \therefore 0 \leq \text{INT}(\text{RND}(1) * 10) < 10 \end{aligned}$$

因为 INT 函数的值是一个整数,所以 $\text{INT}(\text{RND}(1) * 10)$ 的值是 0~9 中的一个随机整数。

同理可证:利用 $\text{INT}(\text{RND}(1) * 90 + 10)$ 能够产生出 10~99 中的一个随机整数 (即两位正整数)。

更一般地情况,利用 $\text{INT}(\text{RND}(1) * A + B)$ 能够产生出 $[B, A + B - 1]$ 区间内的一个随机整数,其中 A 和 B 的值必须为正整数。

在含有随机函数的程序中，每次运行时将产生出相同的随机数序列，为了使每次运行时产生出不同的随机数序列，需要在含有随机函数的语句之前增加如下一条语句：

```
RANDOMIZE VAL(MID$(TIME$, 7, 2))
```

由于该语句涉及到许多知识，在此不作具体解释。

2. 字符串函数

字符串函数简称串函数。所谓字符串函数是指自变量的值或函数的值为字符串的一类函数。

在讨论字符串函数之前，首先假定自变量 X\$ 代表一个字符串、一个字符串变量、一个结果为串值的字符串函数或者一个字符串表达式，自变量 X 代表一个常数、一个数值变量、一个数值函数或者一个算术表达式，自变量 p 和 n 只取 0~255 之间的正整数。

(1) LEN(X\$) 函数

该函数叫做求字符串长度函数，它的值等于 X\$ 值中字符的个数（即串的长度）。例如：

```
LEN("ABCDEF")=6      LEN("YES  | OR  | ON")=9
LEN("08:42:16")=8    LEN("-12.403")=7
```

(2) STR\$(X) 函数

该函数叫做把数值转换成串值的函数，它的值是由自变量的值转化而成的字符串。注意，当自变量的值为正时，函数值的第一个字符为空格；当自变量的值为负时，函数值的第一个字符为负号。例如（假定变量 A 的值为 -3）：

```
STR$(34.26)="  | 34.26"      STR$(-100)="-100"
STR$(2*15+4)="  | 34"        STR$(A↑3)="-27"
```

(3) VAL(X\$) 函数

该函数叫做把串值转换成数值的函数，其功能与 STR\$(X) 函数正好相反，它把自变量的串值直接看作为一个数值。但是，若串值是以字母开头的，则该函数值为零；若串值是以数字开头的数字字母串，则取前面连续的数字（直到碰到字母为止）为其值。例如（设 D\$ = " -35"）：

```
VAL("480")=480          VAL("2.5E3")=2500
VAL("AS25")=0          VAL("25ASD")=25
VAL(D$)=-35            VAL("B2+C3")=0
```

(4) LEFT\$(X\$, n) 函数

该函数叫做求左子串函数，它的值等于 X\$ 的串值中左边 n 个字符所组成的串，若 n 大于串值的长度，则得到整个串值。例如：

```
LEFT$("WORKER", 4)="WORK"
LEFT$("STRING  | 20", 1)="S"
```

(5) RIGHT\$(X\$, n) 函数

该函数叫做求右子串函数，它的值等于 X\$ 的串值中最后 n 个字符所组成的串，如果 n 大于等于串值的长度，则得到整个串值。例如：

```
RIGHT$("KEY", 1)="Y"
RIGHT$("-27.4325", 3)="325"
```

(6) MID\$(X\$, p, n) 函数

该函数叫做求中间子串函数，它的值等于从 X\$ 值中第 p 个字符开始的、连续 n 个字符所组成的串，若 n 项被省略时，函数值等于从第 p 个字符开始的、后面所有的字符组成的串。

例如：

MID\$ (" 08/24/95", 4, 2) =" 24"

MID\$ (" A+B=280", 5) =" 280"

上面共介绍了六个字符串函数，当然在 BASIC 语言中还有其它字符串函数，这里就不一一介绍了。对于上面介绍的字符串函数，若从函数值的类型来分，LEN(X\$) 函数和 VAL(X\$) 函数的值是一个数值，所以把它们叫做结果为数值的字符串函数，这些函数可以同数值函数一样，参加各种数值运算；其余四个函数的值是一个字符串，所以把它们叫做结果为串值的字符串函数。

六、BASIC 表达式

BASIC 表达式包括算式表达式、字符串表达式和逻辑表达式三种类型。

1. 算术表达式

BASIC 算术表达式又称作 BASIC 数值表达式，它是用算术运算符和圆括号把常数、数值变量、数值函数连接起来的式子。算术表达式的值是一个数值。以后我们把单个常数、数值变量和数值函数也看作为算术表达式，因为它们的值也都是数值。

算术运算符有：

+ (加)、- (减)、* (乘)、/ (除)、↑ (乘方)

运算的优先次序为：

() → 函数 → ↑ → *、/ → +、-

由此可见，求 BASIC 算术表达式值的运算次序同数学上进行算术运算的次序完全一致。即先计算括号内，后计算括号外；先求函数值、再乘方、后乘除、最后加减；同一级运算要从左向右依次进行。

算术表达式运算顺序举例：

A + B * SIN(4 * X) ↑ 3 / (C + D)

⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮

⑦ ④ ② ① ③ ⑥ ⑤

在这个例子中，每个运算符下面都注有数字标号，计算机在求这个算术表达式值的时候，将严格按照标号的顺序进行，即①所指的运算最优先，②所指的运算次之（可把函数名看作为一种特殊的运算符），…，⑦所指的运算最后进行。

明确了 BASIC 语言中的算术运算符和运算的优先次序之后，就很容易把数学上的算式表示成 BASIC 算术表达式。

数学算式表示成 BASIC 算术表达式举例：

数学算式

$(\frac{7}{12} - \frac{7}{8}) \times 16.25$

$e^x \sin(x + \frac{\pi}{4})$

BASIC 算术表达式

$(7/12 - 7/8) * 16.25$

$EXP(X) * SIN(X + 3.14159/4)$