

依据教育部考试中心 1998 年
制定考试大纲编写

最新计算机等级考试(二级)用书

QBASIC程序设计

朱玉文 龚圆明 刘万春 编

依据教育部考试中心 1998 年制定考试大纲编写

最新计算机等级考试(二级)用书

QBASIC 程序设计

朱玉文 龚圆明 刘万春 编

国防工业出版社

· 北京 ·

图书在版编目(CIP)数据

QBASIC 程序设计/朱玉文等编. —北京:国防工业出版社, 2000. 6

最新计算机等级考试(二级)用书

ISBN 7-118-02275-6

I. Q… II. 朱… III. QBASIC 语言-程序设计-水平
考试-自学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字 (2000) 第 14000 号

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京怀柔新华印刷厂印刷

新华书店经销

*

开本 787×1092 1/16 印张 20½ 468 千字

2000 年 6 月第 1 版 2000 年 6 月北京第 1 次印刷

印数:1—4000 册 定价:28.00 元

(本书如有印装错误,我社负责调换)

前 言

现代社会是信息化的社会,计算机是信息处理的工具,是高科技领域中不可缺少的核心,它直接渗透到经济、文化和社会的各个领域,迅速地改变着人们的观念、以及生活和社会的节奏,是发展知识经济的支撑之一。

在信息科学技术中,微电子是基础,计算机硬件及通信设施是载体,计算机软件是核心,没有软件就没有计算机的应用。软件产业已成为信息产业的核心和支柱。发展信息产业,会大大提高我国的总体实力,增强我国在国际社会的竞争地位。为了适应知识经济发展的需要,推动信息产业的发展,就需要在全民中普及计算机基本知识,广开渠道,培养和造就一批又一批能熟练运用计算机和软件技术的各类人材。1994年起国家教委推出了全国计算机等级考试,它是一种重视应试人员对计算机硬件和软件实际掌握能力的考试,不限制报考人员的学历背景,任何年龄阶段人员都可报考,这使得考试的目的和要求提到了一个统一和客观的程度,无疑为计算机的普及起到了极大的推动作用。

BASIC语言从60年代中期问世以来,随着计算机的发展,一直在不断地更新换代。早期的BASIC语言只有30条语句左右,只具有一些简单功能,是初学计算机程序设计人员的入门语言。90年代推出的QBASIC语言,在操作和功能上作了大量的改进,增加了很多功能。它包含90多条语句,采用菜单、窗口、对话框等屏幕操作技术,集程序的输入、输出、编辑、运行、调试、存储和打印等众多功能于一屏,使用极为简便灵活。

本书根据教育部考试中心制定的“全国计算机等级考试二级考试大纲”中关于“QBASIC语言程序设计”的考试要求编写,主要介绍QBASIC语言基础知识、QBASIC语言的语法规则及利用QBASIC语言进行程序设计的有关知识。在编制过程中作者根据编程经验及多年来辅导等级考试的教学经验,对读者易于忽略和较难理解的地方作了重点分析,各章节还提供大量的习题分析与实践练习题给读者,以满足考前练习的需要。

全书共分十二章:第一章QBASIC基础知识;第二章QBASIC程序初步;第三章选择结构;第四章循环结构;第五章数组;第六章过程;第七章字符串;第八章屏幕控制与作图;第九章文件;第十章QBASIC操作环境;第十一章上机操作;第十二章实践练习参考答案。最后附录中给出QBASIC语言考试大纲和有关全部QBASIC语言的关键字、错误信息表、语句、函数。语句、函数有些不属大纲要求,这里给读者作为学习中的参考。

计算机软件是一门应用科学,在注重基础理论的同时,实践练习同样非常重要。应考同学首先应掌握好基础知识,要将勤学、苦练、上机验证等多方面结合才能掌握好这门学科。我们希望本书的读者注重对问题的理解,能够多思考,并且举一反三,切不可死记硬背一个说明、一个程序,只有这样才能对任一问题或变换后的题型应用自如,正确解答。

本书编写过程中得到国防工业出版社的大力支持,在此编者表示诚挚的感谢。参加本书编写工作的还有:刘俐、王民、杨聪、徐一华、何鑫、夏军、郑红霞、裴明涛、杨云飞、谢玉清、龚小舟、顾伟、李小刚等同志。由于编者水平有限,书中难免存在错误和不妥之处,殷切希望广大读者批评指正。

编 者

第一章 QBASIC 基础知识

1.1 QBASIC 语言的历史及特点

1.1.1 BASIC 语言的历史回顾

最早的计算机程序设计高级语言 FORTRAN 设计于 1950 年,设计的目的是为了解决复杂的数学问题。对一般人来讲,运用最初的程序设计语言进行程序设计是比较困难的,但当时计算机应用领域小,使用计算机的都是一些从事计算机工作的专业人员,所以没将程序设计作为一个重要问题提出。随着计算机的发展,计算机的应用不再局限于数值计算,复杂的程序设计成了人们使用计算机的一种障碍,需要一种编程简单的程序设计语言显得尤其迫切。

由此,1964 年,美国的两位计算机科学家 G. Kemeny 和 Thomas E. Kurtz 创造了 BASIC 语言。BASIC 这个词是英文 Beginner's All Purpose Symbolic Instruction Code 的缩写,意思是初学者通用符号指令代码。BASIC 语言以它无可比拟的“容易学习”的特点,使它从本来供大学生使用的一种程序设计语言,很快跨出校园走向社会,并从美国传播到全世界,成为初学计算机的首选语言。几十年来,BASIC 语言随着计算机的发展,其本身也经历了以下几个发展阶段:

(1)原版本的 BASIC 语言是在 FORTRAN 基础上开发的一种非常简单的新语言,设计该语言的特殊意义是学习编程简单化,当时使用在一些小型机上,采用编译方式,功能相对较弱。但由于它的出现使程序设计语言从只能被专业人员使用发展到能为一般人使用,开辟了一个“大家都来学习计算机”的新阶段。

(2)微机 BASIC。1975 年美国麻省理工学院的 Altair 计算机的诞生迎来了微型计算机的一场革命。美国的比尔·盖茨(Bill Gates)和 Paul Allen 为 Altair 开发了一个 BASIC 版本,该版本将编译型的 BASIC 压缩成只有 4KB 的解释型 BASIC 语言,该版本在当时的 PC 机中是最流行的产品,基本上每一台微机都配置。

经过若干年的完善和发展,BASIC 的功能不断扩展,由小型、简单的学习语言发展成为功能丰富的实用语言,并先后出现 GW BASIC 和 M BASIC,它们能处理文件,具有绘图和音乐功能,其绘图甚至超过其它语言。

(3)结构化 BASIC。70 年代由于计算机硬件的发展,存储量的扩大,随之带来软件的高速发展,程序越来越复杂,引起了软件编制上的所谓“软件危机”。荷兰的计算机科学家 E. Q. Dijkstra 等人提出了结构化程序设计思想,解决软件由手工编制工程化的规范。结构化程序设计思想要求高级语言只有顺序、分支、循环三种基本结构,并且程序模块化,按功能划分模块。第二代 BASIC 语言不能满足结构化程序设计要求。

20 世纪 80 年代中期,美国国家标准化协会根据结构化程序设计的思想,提出了一个新的 BASIC 标准草案,并于 1987 年正式通过,称为“Full BASIC(X3.113-1987)”,意为“完全的 BASIC”。BASIC 语言的两位创始人 Kemeny 和 Kurtz 参与了这一标准的制定。在此前后,出现了一些结构化的 BASIC 语言,如 True BASIC、Quick BASIC、Turbo BASIC 等。

1984 年推出的 True BASIC 语言,它严格遵循美国国家标准化协会的规定,完全符合结构化和模块化程序设计的要求,并保留了 BASIC 易学、易懂、易编程和易调试的特点。此外,它还提供了解释和编译两种工作方式。True BASIC 在我国也有相当程度的普及。

Quick BASIC 是微软公司 1987 年推出的 BASIC 版本。它具有以下特点:提供了一个开发程序的集成环境,用户在编程、修改、编译、调试及运行时均可通过菜单操作,十分方便;与 GW BASIC 和 BASIC 高度兼容,提供了全局变量和局部变量;程序模块化;编译后产生一个可执行文件,提高了执行效率。

在 Quick BASIC 的基础上,微软公司又在几年前推出了 QBASIC,随 MS DOS5.0 版本一起问世。QBASIC 介于第二代到第三代之间的现代化 BASIC 语言,它包含 Quick BASIC 语言的主要功能,是一种结构化的语言。

第三代 BASIC 语言的特点是:完全符合结构化程序设计的要求,并具有编译工作方式。这一代 BASIC 语言的出现,大大提高了 BASIC 语言的档次与功能,受到了用户的欢迎。

(4)Windows 环境下的 BASIC。80 年代中期,微软公司推出的 Windows 提供了图形方式的用户界面,用窗口替代了 DOS 命令的提示行,用户不必再记忆枯燥无味的 DOS 命令,只需用鼠标在窗口或桌面上选择一个菜单项式图标,即可完成所需要的操作,使人们使用计算机变得更方便。1991 年微软公司推出第一个在 Windows 下的 BASIC 语言 Visual BASIC(即“可视化的 BASIC”)。Visual BASIC 与 Quick BASIC 兼容,Quick BASIC 编写的程序不作修改即可在 Visual BASIC 环境下运行。除 Visual BASIC 以外,还有 GFA-BASIC、CA-REAL12ER 等版本。

1.1.2 BASIC 语言的特点

(1)语法简单、易于掌握。基本 BASIC 只有 17 种语句,其中命令关键字、单词及运算符与英语词汇含义相当,易于理解和记忆。

(2)BASIC 语言是一种会话式语言,用户可通过键盘、显示器与计算机对话。计算机在解释执行源程序时,发现语法错误会暂停执行,并在屏幕上显示出相应信息,指出错误位置,等待用户修改程序。

(3)提供计算器功能,例如用语句:

```
PRINT 0.5×0
```

在屏幕上即可获得结果。

(4)既具有科学计算功能,又具有数据处理、绘制彩色图形、控制声音等功能。

(5)对于 QBASIC 语言,除了上述特点外,它还具有以下几方面的特点:

①扩充了前期版本变量和常量类型;改进了循环结构;提供新的选择结构;提供作为

单独模块的子程序和函数;不需要行号。

- ②提供良好的窗口编辑环境。
- ③提供联机“帮助”。
- ④提供“分步”、“追踪”等调试工具。

1.2 QBASIC 源程序的结构

下面是 QBASIC 源程序的一个例子。

【例 1.1】编写一个计算 $s=1!+2!+3!+\cdots+n!$ 的程序。

```

DECLARE SUB F(S,N)
CLS
INPUT "N=";N
FOR I= 1 TO N
    CALL F(S,I)
    SUM=SUM+S
NEXT I
PRINT "SUM=";SUM
END

SUB F(S,N)
A=1
FOR I=1 TO N
    A=A*I
NEXT I
S=A
END SUB
    
```

QBASIC 语言中输入源程序时,系统自动将程序格式作规范化处理:

- (1)将语句中关键字(语句定义、函数名字)全部改成大写;
- (2)在赋值号(“=”)及+、-、*、/等两侧各留空格;
- (3)每个语句行的行号后留一空格,当一行有多个语句时,用冒号(“:”)分隔,并在冒号后留一空格。

QBASIC 语言程序的一般结构为:

- (1)一个源程序可由一个或多个模块组成。
- (2)在一个程序中,必须有且只能有一个主模块。可以有一个或多个子程序模块,也可以只有一个主模块。
- (3)一个模块中有若干语句行构成,当同一行中有一个以上语句时,可用“:”号分隔,一行的总字符不得超过 255 个。
- (4)语句行可以没有行号或标号。行号为纯数字,由 1~40 组成,行号后面不加冒号;

标号可以是数字、英文字母或二者的组合,标号后必须加冒号。例如:

a:END

这里“a:”是标号。行号或标号的位置必须写在每一行的最左端,不能出现在一行中两个语句之间。行号和语句标号仅作为控制转移的目标,不决定程序执行的顺序。

(5)每一个模块有一个结束语句 END,主模块以“END”结束,子模块以“END SUB”或“END FUNCTION”结束。

1.3 QBASIC 字符集

QBASIC 用的是微软 BASIC 字符集,包括:字母(A~Z,a~z)、数字(0~9 和十六进制 A~F 或 a~f)、特殊字符 28 个。在 QBASIC 中特殊字符的含义如下:

1. 数据类型标识符

!	单精度	%	整型
#	双精度	&	长整型
\$	串		

2. 数学运算符

*	乘号	\	整除(反斜杠)
/	除号(斜杠)	=	关系运算符等于或赋值号
+	加号	>	大于
-	减号	<	小于
^	乘方(上箭头)	.	小数点

3. 特殊字符

'	注释行(单引号)
"	双撇号(串常量定界符)
;	控制 PRINT 语句和 INPUT 语句的输出
,	控制 PRINT 语句和 INPUT 语句的输出
(	左圆括号
)	右圆括号
␣	空格(即空白)
:	冒号用于一行上多个语句的分隔
?	INPUT 语句提示,问号
_	下划线(与其它 BASIC 版本兼容保留字符,但 QBASIC 不支持续行)
[	左方括号
]	右方括号
@	圈 A

注意:数学中常用的 α 、 β 、 γ 和 Π 不能使用,要用其它字符替代。

1.4 QBASIC 中的常量

QBASIC 语言程序设计处理的数据类型有两大类:数值型数据和字符型数据。

QBASIC 中运算的量有常量、变量、数组、数组元数、函数等。

常量是指在程序处理过程中其值始终保持不变的量。QBASIC 程序中常量可分为数值常量和字符串常量。

1.4.1 数值常量

数值常量就是数学中常用到的常数。数值常量有整型常量和实型常量两类。

1. 整型常量(即整数)

整数是由若干个数字组成的序列,可以有数值符号。如 -738 、 $+3879$ 、 -7468 、 $+0$ 、 -0 等都是合法的整数形式。

需要注意的是:不能在数字之间出现分位符号,如 $4,000$ 、 $-5,431$ 等都不是 QBASIC 合法的整数。

2. 实型常量(即实数)

QBASIC 中实数有两种记数方法:

(1)日常记数法(又称“定点实数”):即十进制小数形式,可以有小数点和数值符号。例如 0.145 、 0.0 、 $+4.12$ 、 -3.15 、 5.0 等均为实数形式。

(2)科学记数法:用指数形式表示一个实数。例如在数学上 6341.32 表示成 6.34132×10^3 , 35000 表示成 3.5×10^4 , -0.000124 表示成 -1.24×10^{-4} 。在计算机中用上下标表示非常麻烦,在程序设计中更为困难,因此约定用英文字母“E”表示以10为底的幂数。上面几个数就可以写成: $6.34132E+3$ 、 $3.5E+4$ 、 $-1.24E-4$ 。

除了常用普通型实数(单精度实数)外,QBASIC 还允许使用双精度实型,单精度最多提供7位有效数字,双精度可提供达16位有效数字。

QBASIC 语言中单精度末尾带“!”,由于常用数值都为单精度,由此,“!”符号可以省略不写。例如: $32.4!$ 与 32.4 是等价的。

双精度末尾带“#”号,指数形式用“D”替代单精度的“E”。例如: $35123.756\#$ 和 $12.345678D4$ 都是双精度实数。

对于 QBASIC 语言中出现的常数,区分它们类型的方法为:

(1)数值在 $-32768 \sim 32767$ 之间不带小数的常数为整数,QBASIC 语言用“%”作为整型数的符号。例如: 134 、 134% 、 -389 、 346% 均为整型数。

(2)超过 $-32768 \sim 32767$ 范围在 $-2147483648 \sim 2147483647$ 范围之内不带小数的常数为“长整型常量”。长整型常量末尾可加一个“&”符号。

(3)不超过7位带小数点的常数按实数单精度数表示。

(4)多于7位且带小数点的数,如果是带“#”的常数或是含有以“D”表示指数的实数,按双精度实数表示。

QBASIC 各种数值数据表示和有效位数、表示数的范围如表 1.1 所列。

表 1.1

类型	表示方法	占用字节	有效范围	有效位数	示 例
整数	(1)不带小数点、范围在-32768~32767之间的数； (2)在上述范围内的数字尾有“%”符号的数	2	-32768~32767	≤ 6	正确:200,-12345 错误:+55000(太大) -55000(太小) 23,000(含逗号)
长整数	(1)超过-32768~32767范围,而在-2147483648~2147483647之间的不带小数点的常数； (2)范围在-2147483648~2147483647之间且数字尾带“&.”符号的常数	4	-2147483648~2147483647	≤ 11	正确:-240000,666&. 错误:22345678987(超过范围) 100(不是长整数)
定点单精度实数	(1)含有小数点且不超过7位数字的常数； (2)末尾有“!”且不超过7位数字的小数	4	-9999999! ~ +9999999!	≤ 7	正确:134. 657 -4326! 错误:8765. 432(超过7位) -75432(是整数)
单精度浮点数	以指数形式表示的浮点数,以“E”或“e”表示指数部分	4	-3. 402823E38~ 3. 402823E38	≤ 13	正确:-3. 638E-22 24567. 0e2 错误:43. 45D+12(不是单精度)
双精度浮点数	以指数形式表示的浮点数,以“D”或“d”表示指数部分	8	-1. 79769134862315D+308~ 1. 79683134862315D+308	≤ 22	正确:8. 6548756D102 错误:2. 672D10 # (指数部分不应出现#号)
定点双精度实数	(1)多于7位且包含小数点的数； (2)末尾有“##”的常数	8	-9999999999999999~ 9999999999999999	≤ 16	正确:1234 # 123456. 789872 错误:123456789(长整数)

1.4.2 字符串常量

用一对双撇号(“ ”)括起来的若干个合法字符串称为字符串常量。例如:“NAME”、“ROOM”、“WINDOW”、“QBasic”等都为合法字符串常量。必须指出,双撇号只是字符串的定界符,表示字符串的起止限界,不属于字符串本身。

1.4.3 符号常量

符号常量是指使用一个符号代替常量在程序中应用。例如求圆的半径、周长、面积等都要用 $\pi=3.1416$,在一个程序中如多次求圆的面积或周长等重复使用 π 值时,QBASIC语言可以用符号代替常量,用符号代替常量的说明语句是:

CONST PI=3.1416

这样在以后的程序中 PI 代替常数 3.1416。

【例 1.2】符号常量定义形式和使用。

CONST PI=3.1416

r=6

s=2 * PI * r

area=PI * r * r

PRINT "length=";s,"area=";area

END

程序运行结果如下：

length=37.6992 area=113.0976

使用符号常量有以下优点：

- (1)可减少多次重复写同一个常量，一次修改该常量，则整个程序都被修改。
- (2)使语句可读性好。

1.5 变 量

在程序执行过程中，其值发生变化的量称为变量。在计算机科学中，变量是一个或者若干个存储单元地址的标志。变量的值是存储单元存放的信息。变量有三种属性：变量名、变量类型和变量值。

1.5.1 变量命名

在 QBASIC 程序中，变量名是用符号名表示的，在同一程序模块中，不能出现两个同名的量具有不同值的情况，即变量名必须唯一。QBASIC 对变量命名有如下规定：

- (1)变量名的第一个字符必须是字母。
- (2)变量名中第二个字符及其后的各字符可以是字母、数字或小数点。
- (3)变量名的长度范围为 1~40 个字符。
- (4)表示变量类型的符号(如%、!、&、#、\$)应作为最后一个字符。
- (5)变量名中不允许空格，并且大小写等效。

例如：a, B1, XY, Face 为合法变量名；而 4A, Liu—Li, Chang Lan 为不合法变量名。

1.5.2 变量的类型

在 QBASIC 中变量有 5 种类型，各种类型在计算机中占用不同长度的存储单元。例如：存放整型变量，在内存中占 2 个字节；存放单精度实数的变量，在内存中占 4 个字节；存放双精度实型数据的变量占内存单元 8 个字节；而长整型变量占 4 个字节；字符串变量则按其字符的多少，每个字符占用一个字节。

在程序设计中如何定义变量类型，使计算机能按要求分配相应的内存单元和确定数据的存放形式，有两种方法。

1. 在变量名末尾加类型定义字符

在变量末尾用%、!、&、#、\$ 作为整型、长整型、单精度实型、双精度型、字符串型变量的类型定义符。

2. 用 DIM 语句和类型名来指定变量类型

QBASIC 可使用如下类型定义语句：

```

DIM 变量名 AS INTEGER    (变量定义为整型)
DIM 变量名 AS LONG       (变量定义为长整型)
DIM 变量名 AS SINGLE     (变量定义为单精度型)
DIM 变量名 AS DOUBLE     (变量定义为双精度型)
DIM 变量名 AS STRING     (变量定义为字符串型)

```

使用 DIM 语句定义变量的类型后,该变量名末尾加上类型定义符或不加类型定义符都是合法的定义。

【例 1.3】DIM x AS INTEGER

```

x%=365
x=653
PRINT x%,x
END

```

程序运行结果如下：

```

653      653

```

在程序中对变量作类型定义后,不能再将该变量定义为其它类型,如对例 1.3 中的 x,如用 x&、x!、x# 等都不合法。

3. DEFtype 语句

DEFtype 语句的格式为：

```
DEF<类型><字母域>[,<字母域>]...
```

其中:<类型>可为 INT(整型)、LNG(长整型)、SNG(单精度型)、DBL(双精度型)、STR(变长字符串型)等;<字母域>可以是一个字母,也可以是用减号连接前后两个字母组成的一个字母区域,字母域中的字母大写、小写等效。例如:A—D、G、x—y 等都表示字母域。

使用 DEFtype 语句为模块中使用的变量提供缺省数据类型说明,有如下几点特性:

(1)使用 DEFtype 语句指定任一字母域中以任一个字母开头的变量如没有被类型说明符和 DIM 语句定义,则变量的类型就是 DEFtype 语句中类型关键字所规定的类型。

(2)关键字 DEF 和表示类型的关键字(INT、LNG、SNG、DBL、STR 等)之间不允许有空格。例如,DEFINT A—J 定义 A—J 中字母为整型量。

(3)DEFtype 语句的作用为当前所在模块。当 DEFtype 定义在主模块中时,由此主模块调用的任何 SUB 和 FUNCTION 过程,都自动将此语句复制到该过程中,使得过程模块中使用的变量同主模块中使用的变量相同。

1.5.3 关于变量类型定义的几点说明

(1)类型说明符和 DIM 语句优先于 DEFtype 语句。即符合 DEFtype 语句规定的变

量,还可以使用类型说明符或 DIM 语句重新指定变量类型。例如:

```
DEFINT A-X
DIM F2 AS SINGLE
LET X=4;LET P=12
LET F2=X/P
LET D2#=F2*P-3
PRINT X,P,F2,D2#
END
```

(2)在程序中没有带类型说明符,同时也未用 DIM 语句定义的变量,如属于 DEFtype 语句所规定范围,则为 DEFtype 语句所规定的类型,否则系统按单精度型变量处理。

(3)在程序中变量应遵循先定义后使用的原则。若要使用 DIM 语句或 DEFtype 语句定义变量类型,应把它们书写在相应模块的开始。

(4)对任何数值类型变量,未赋值前,其值均为 0;对变长字符串型变量,未赋值前,其值为空,长度为 0;对每个定长字符型变量,未赋值前,其值为 n 个空格字符,长度为 n。

(5)QBASIC 语言允许不同类型的数值进行运算,运算时系统作类型转换,转换成相同类型后再进行运算或赋值。

(6)对于字符串变量,允许定长与变长进行相互赋值。赋给定长字符串的变量的值若超过规定的长度,则后面多余字符被舍去;若小于规定长度,则后面字节用空格字符补齐。

【例 1.4】place1 和 place\$ 分别是定长和不定长字符串变量。

```
DIM place1 AS string * 5 .
place1="Chinese"
place$="People"
PRINT place1,place$
END
```

程序运行结果如下:

```
Chine      People
```

程序中 place1 是长度为 5 的字符串变量,只能放入 5 个字符“Chine”,place\$ 是可变长字符串变量,放置字符串按实际赋给的字符串长度。

1.6 函 数

QBASIC 语言提供了丰富的标准函数,同时也允许用户利用 FUNCTION 过程定义函数,这里只讨论标准函数。

标准函数分为数值函数和字符串函数。我们把自变量的值和函数的值为数值的一类函数称为数值函数,把自变量的值和函数的值中至少有一个为字符串型的一类函数称为字符串函数。表 1.2 为常用标准函数。

表 1.2

函 数	功 能	说 明
SIN(X)	$\sin x$	自变量单位为弧度
COS(X)	$\cos x$	自变量单位为弧度
TAN(X)	$\tan x$	自变量单位为弧度。注意: X 如接近 $\pi/2$ 或 $\pi/2$ 的奇数倍, 则 TAN(X) 可能会超过计算机允许的上限值而溢出
ATAN(X)	$\arctan x$	函数值的单位为弧度。 $-\pi/2 < \text{函数值} < \pi/2$
LOG(X)	$\log_e x$ (求以 e 为底数的对数值, 即求自然对数 $\ln x$)	X 值应为正数
EXP(X)	e^x	$e \approx 2.71828$ 。注意: X 的值不应使函数值超出计算机允许的范围
SQR(X)	x 的平方根(正根)	X 应大于或等于 0
ABS(X)	x 的绝对值 $ x $	例如 $\text{ABS}(-4)=4$, $\text{ABS}(4)=4$
INT(X)	得到一个不大于 x 的最大整数	例如 $\text{INT}(-8.6)=-9$, $\text{INT}(8.6)=8$
FIX(X)	对小数简单截断	例如 $\text{FIX}(8.6)=8$, $\text{FIX}(-8.6)=-8$
RND(X)	产生一个在 (0,1) 区间的随机数	RND 可不带参数, 例如 PRINT RND 可得到一个 (0,1) 间的随机数, RND(X) 作用与 RND 相同 (X > 0)。如果 X < 0, 不同的 X 值对应不同的随机数, RND(0) 得到上一次的随机数
SGN(X)	符号函数	$\text{SGN}(X) = \begin{cases} 1 & (X > 0) \\ 0 & (X = 0) \\ -1 & (X < 0) \end{cases}$
CINT(X)	整型转换函数	将 X 值四舍五入转换为一个整型, X 为实型数
CLNG(X)	将 x 转换为一个长整型数	将 X 值四舍五入转换成一个长整型数
CSNG(X)	将 x 转换为一个单精度型数	将 X 值转换为一个单精度型数
CDBL(X)	将 x 转换为一个双精度型数	将 X 值转换为一个双精度型数

说明:

(1) 函数的自变量(又称“形式参数”)可以是一个常量、变量或表达式。如 $\text{TAN}(-4 * 2.8)$ 、 $\text{SQR}(7.9 * 5.8/4.3)$ 等。自变量必须用括号括起来。

(2) 一个数学函数的值是一个数值, 它可以参加运算, 如:

$$2.7 * \text{INT}(2.7 * 3.9) * \text{LOG}(10.35)$$

但标准函数不能单独作为一个语句, 如下面的写法不是一个合法的 QBASIC 语句:

$$\text{COS}(4 * 3.14159/5)$$

它只是一个表达式。

(3) 有关字符型函数将在后面章节中介绍。

1.7 运算符和表达式

QBASIC 有三类运算符与三种表达式：

- (1) 算术运算符, 用来进行算术运算。
- (2) 关系运算符, 用来比较两个运算量的大小。
- (3) 逻辑运算符, 用来进行逻辑运算。

与运算符相对应有三种表达式, 即数值表达式、关系表达式和逻辑表达式。

1.7.1 算术运算符和数值表达式

1. 算术运算符

QBASIC 语言提供 7 个算术运算符, 如表 1.3 所列。

表 1.3

运算符	含义	举例	数学上表示
+	加	$a+b$	$a+b$
-	减	$a-b, -x$	$a-b, -x$
*	乘	$a*b$	$a \times b$
/	除	a/b	$a \div b$ 或 $\frac{a}{b}$
\	整除	$a \setminus b$	取 $a \div b$ 整数商
MOD	求余	$a \text{ MOD } b$	取 $a \div b$ 的余数
^	乘方	a^b	a^b

在 QBASIC 算术运算中, 加、减、乘、除和乘方运算与数学上相同, 这里只解释整除和取模(求余)运算。

整除: 要求前后两个操作数的值为整数, 若不为整数则自动按四舍五入取整, 整除运算的商取整数部分。

取模(求余): 被除数 a 除以除数 b 求得余数。

例如:

$$\begin{array}{ll}
 76 \setminus 12 = 6 & 76 \text{ MOD } 12 = 4 \\
 35 \setminus 49 = 0 & 35 \text{ MOD } 49 = 35 \\
 -65 \setminus 23 = -2 & -62 \text{ MOD } 23 = -16 \\
 33.8 \setminus 5 = 6 & 33.8 \text{ MOD } 5 = 4
 \end{array}$$

2. 算术运算的优先级

算术运算符的优先级由高到低的次序为:

() $\rightarrow$ 函数 $\rightarrow$ ^ $\rightarrow$ 取负 $\rightarrow$ *, / $\rightarrow$ \ $\rightarrow$ MOD $\rightarrow$ +, -

即圆括号优先级最高, 应先算括号内, 后算括号外; 函数运算次之; 乘方运算排在第三, 向后依次为取负、乘除、整除、取余以及加减运算。注意, 对于同一级运算, 按照从左向右的次序进行。

3. 数值表达式

数值表达式又称算术表达式,它是由算术运算符和圆括号将若干运算量(包括常量、变量、函数、数组元素等)连接起来的式子。算术表达式的值为数值型。

例如:求表达式 $a * (b - d) \setminus 3 * \text{ABS}(x + 5) / p^3 - 4$ 的运算顺序,已知 a, b, d, x, p 的值分别为 5, 7, 4, 6 和 2。

$$\begin{array}{l}
 a * (b - d) \setminus 3 * \text{ABS}(x + 5) / p^3 - 4 \\
 \textcircled{1}(b - d) = 3 \qquad \qquad \textcircled{2}(x + 5) = 11 \\
 \qquad \qquad \qquad \textcircled{3}\text{ABS}(11) = 11 \quad \textcircled{4}p^3 = 8 \\
 \textcircled{5}a * 3 = 15 \qquad \textcircled{6}3 * 11 = 33 \\
 \qquad \qquad \qquad \textcircled{7}33 / 8 = 4.125 \\
 \textcircled{8}15 \setminus 4.125 = 3 \\
 \qquad \qquad \qquad \textcircled{9}3 - 4 = -1
 \end{array}$$

计算中,①~⑨表示优先顺序,最后结果为-1。

明确数学表达式表示成 QBASIC 数值表达式的运算符的优先顺序后,就很容易将数学式表示成 QBASIC 的数值表达式。例如:

数学式	QBASIC 数值表达式
$\sqrt{a^2 - x^2}$	<code>SQR(a^2 - x^2)</code>
$\frac{e^x - e^{-x}}{2}$	<code>(EXP(x) - EXP(-x)) / 2</code>
$\frac{a+b}{a-b}$	<code>(a + b) / (a - b)</code>
$2e^x \cos(x + \frac{\pi}{3})$	<code>2 * EXP(x) * COS(x + 3.1416 / 3)</code>

4. 不同类型数据的混合运算

在数值表达式中,可以包含有不同类型的数据,如:

$$\begin{array}{ll}
 12 * 3.5 & \text{(整数与实型)} \\
 12345678 * 82 & \text{(长整型与常规整型)}
 \end{array}$$

对不同类型数据进行运算结果是何种类型, QBASIC 语言有如下规定:

(1) 同类型数据互相运算,结果值的类型不变。当两个整型相乘,不能让其积的范围超过整型的最大范围,否则出现溢出。

例如: $12345 * 23456$, 其结果超过 $-32768 \sim 32767$ 范围,系统提示“溢出”。

(2) 整型数据与实型数据互相运算,其结果为实型。一个长整型与单精度型或双精度型运算结果为双精度型。

(3) 整型与长整型运算,结果为长整型。

(4) 单精度与双精度运算,结果为双精度型。

实际上,两种不同类型的量进行运算,结果的类型为两者中占字节数多的那一种数据类型。

1.7.2 字符串运算符和字符串表达式

字符串运算符只有一个,即字符串连接符“+”,它的作用是把运算符两侧的字符串连

接为一个字符串。例如：

```
PRINT "I" + "Love" + "China"
```

结果输出一个字符串：

```
I Love China
```

用“+”将各字符串连接起来的式子，是最简单的“字符串表达式”。

字符串表达式是指由字符串型常量、变量、函数(函数值必须为字符串)通过连接号连接而成的式子。字符串表达式的值是一个字符串。

例如，对如下语句：

```
x$="abcd",r$="xyz"  
PRINT x$+y$,CHR$(48)+"-"+CHR$(57)
```

其结果为：

```
abcdxyz      0-9
```

这里 CHR\$(48)求 ASCII 码为 48 的字符(即 0)。

1. 7. 3 关系运算符和关系表达式

1. 关系运算符

QBASIC 语言提供的关系运算符有 6 种，如表 1. 4 所列。

关系运算符的优先级为同级，即在关系表达式中运算的顺序为从左到右进行运算。

表 1. 4

关系运算符	含义	相应数学符号
=	等于	=
>	大于	>
<	小于	<
<>	不等于	≠
>=	大于等于	≥
<=	小于等于	≤

2. 关系表达式

关系表达式是由两个数值表达式或两个字符串表达式通过关系运算符连接而成的式子。关系表达式的值是一个逻辑值(“真”或“假”)。当一个关系式成立，或者说条件满足时，其值为“真”，常记为 T 或 Y，否则为“假”，常记为 F 或 N。

在 QBASIC 中，没有专门设置逻辑型变量和逻辑型常量，而是用数值来代表逻辑值，规定以“零”代表“假”，以“非零”代表“真”。在关系表达式的运算中，如果得到的值为“假”，则此表达式值为 0，如果表达式值为“真”，则表达式是一个固定的非零值-1。

例如，设 x=26,y=8,a\$="abcd",b\$="abtd",则有：

关系式	取值	结果值
x>2*y	T	-1
x MOD y=0	F	0