

BASIC

BASIC BA

上海交通大学出版社

白英彩 张玉轩 编
冯大本 审

BASIC语言及其应用

12
X / 1

730.1
247x/1

BASIC 语言及其应用

张玉轩 白英彩 等编

冯大本 审



上海交通大学出版社

019827

内 容 提 要

本书主要是针对目前国内普遍使用的个人计算机而编写的。全书分三大部分：第一部分为基础部分，介绍 BASIC语言的基础知识；第二部分介绍数值运算程序；第三部分介绍非数值运算程序。本书系统完整，通俗易懂，例题丰富，既是一本 BASIC语言的入门参考书，又是工程技术人员的编程参考书。本书可供各高等及中等学校作教材，也适合自学者使用，并可作工程技术人员参考书。

JS264/04

BASIC 语言及应用

出 版：上海交通大学出版社

（淮海中路1984弄19号）

发 行：新华书店上海发行所

排 版：浙江上虞科技外文印刷厂

印 刷：江苏太仓印刷厂

开 本：787×1092（毫米）1/16

印 张：15.5

字 数：384000

版 次：1990年10月 第一版

印 次：1990年11月 第一次

印 数：1—3500

科 目：233—301

ISBN7—313—00734—5/TP·39

定 价：3.10 元

前 言

本书主要针对目前国内普遍使用的个人计算机而编写的。编写时主要考虑以下两个方面：一是考虑到目前全国已拥有几十万台 IBM PC/XT、AT 及其兼容机，应用十分广泛，初学者首先都要学习 BASIC 语言，而 BASIC 语言方面的手册类资料不少，但尚少专门的适合于微机特点的 BASIC 教材；二是现有的 BASIC 教材，限于当时的历史条件，已不敷读者所需。

BASIC 语言既能应用于各种数值计算，又能进行事务管理，同时又是一种十分普及的会话式语言，特别适合初学计算机的同志。

本书共分三大部分：第一部分（1~9 章）为基础部分，介绍了 BASIC 语言的基础知识，是本书的必读部分，为后续的编程打下良好的基础。这部分的最后一章例举了一些 BASIC 语言的应用实例，基本上代表了 BASIC 语言的一些应用方面。第二部分（10~12 章）介绍数值运算程序，该部分的内容主要有一般数值计算、自动控制中的计算及一些常用简单的工程计算。这一部分与下一部分主要是结合实践应用来介绍 BASIC 的一些编程技巧。第三部分（13~14 章）介绍非数值运算程序，主要是介绍排序、检索、数据文件。

本书作为教材编写的，具有通俗易懂、系统性强的特点。本书还适合作为读者使用，也适合具有高中以上文化水平的科技人员、教师及其他管理人员使用。

本书不是针对某一具体的机型而编写的，书中的内容虽适用于个人计算机，但读者在实际使用中，应结合具体的机器及有

本书中，张玉轩副教授编写第七章至第九章和第十一章至第十四章；白英彩教授编写第四章至第六章；黄惠民讲师编写第一章至第三章；陆惠恩讲师编写第十章。冯大本高级工程师主审了本书；章仁龙等同志为本书的编写提出了不少宝贵的意见，在此一并致谢。

由于作者水平有限，书中难免有不少缺点和错误，恳请读者批评指正。

编 者

1990.5

第一部分 BASIC 语言

第一章 电子计算机简介	1
§ 1.1 电子计算机系统的组成和基本工作原理	1
§ 1.2 电子计算机的主要用途	2
§ 1.3 计算机语言	2
第二章 BASIC 语言基本概念	4
§ 2.1 BASIC 语言特点	4
§ 2.2 BASIC 语言的基本概念	4
第三章 BASIC 的输入、输出语句	8
§ 3.1 LET 语句	8
§ 3.2 INPUT 语句	10
§ 3.3 READ/DATA 语句	11
§ 3.4 RESTORE 语句	12
§ 3.5 三种输入语句的比较	13
§ 3.6 PRINT 语句	13
§ 3.7 TAB 函数	15
§ 3.8 PRINT USING 语句	16
§ 3.9 自定义函数	17
第四章 BASIC 程序编制	20
§ 4.1 STOP 和 END 语句	20
§ 4.2 REM 语句	21
§ 4.3 BASIC 程序编制	21
第五章 分支和循环	22
§ 5.1 无条件转移语句	22
§ 5.2 条件转移语句	22
§ 5.3 流程图	25
§ 5.4 循环语句	26
§ 5.5 常用程序结构	28
§ 5.6 双重循环	33
第六章 数组处理功能	35
§ 6.1 下标变量	35
§ 6.2 数组说明语句	35
§ 6.3 双下标变量	37
第七章 子程序和开关语句	40
§ 7.1 子程序	40

§ 7.2 开关语句	45
第八章 字符串处理	50
§ 8.1 字符串变量	50
§ 8.2 子字符串函数	52
§ 8.3 字符串连接合并	54
§ 8.4 其他字符串函数	54
§ 8.5 字符串比较	59
第九章 综合举例	61
§ 9.1 多项式计算	61
§ 9.2 逐次逼近法求微分	62
§ 9.3 级数求和	63
§ 9.4 矩阵转置	64
§ 9.5 矩阵减法	64
§ 9.6 矩阵与向量乘积	66
§ 9.7 因数分解	67
§ 9.8 牛顿迭代法求一元方程的根	68
§ 9.9 工资统计	70
§ 9.10 财务分类帐	72
§ 9.11 数据处理	73
§ 9.12 人事统计	75
§ 9.13 编码和译码	78
§ 9.14 字符串检索	79
§ 9.15 票证统计	80
第一部分小结	81

第二部分 数值与工程计算程序

第十章 数值计算	83
§ 10.1 一元方程求根	83
§ 10.2 线性方程组求解	87
§ 10.3 求微分	96
§ 10.4 数值积分	98
§ 10.5 常微分方程数值解法	101
§ 10.6 矩阵运算	103
§ 10.7 差分方程求解	110
第十一章 BASIC 在自动控制中的应用	113
§ 11.1 由传递函数 $G(S)$ 计算频率响应 $G(j\omega)$	113
§ 11.2 由频率响应测量数据拟合传递函数	120
§ 11.3 APPLESOFT BASIC 控制功能	125

§ 11.4	TRS80 LEVEL II BASIC 控制功能	128
§ 11.5	IBM PC BASIC 控制功能	129
§ 11.6	单板机的 BASIC	131
§ 11.7	编译 BASIC 与控制	135
§ 11.8	参数最优化	136
第十二章	BASIC 工程计算	145
§ 12.1	图形面积计算	145
§ 12.2	图心计算	146
§ 12.3	二次力矩计算	147
§ 12.4	飞轮转动惯量计算	149
§ 12.5	简支梁计算	150
§ 12.6	梁挠度计算(集中分布载荷)	151
§ 12.7	梁挠度计算(均匀分布载荷)	152
§ 12.8	梁的剪切应力计算	153
§ 12.9	平面应力计算	154
§ 12.10	轴应力计算	156
§ 12.11	承受不同方向集中载荷的轴计算	157
§ 12.12	齿轮计算	159
§ 12.13	串联谐振电路计算	161
§ 12.14	并联谐振电路计算	161
§ 12.15	滤波器计算	162
§ 12.16	回路电流法解直流电路	163

第三部分 非数值运算程序

第十三章	非数值运算基本方法和技术	168
§ 13.1	排序	168
§ 13.2	检索	177
第十四章	数据文件设计	179
§ 14.1	ON ERR GOTO 语句	179
§ 14.2	RESUME 语句	180
§ 14.3	POKE 和 PEEK 的用途	181
§ 14.4	文件概念简介	182
§ 14.5	数据录入与检错程序	183
§ 14.6	顺序文件程序设计	190
§ 14.7	随机文件程序设计	220
§ 14.8	IBM PC 数据文件设计简介	232

第一部分 BASIC 语言

第一章 电子计算机简介

电子计算机是一种能够自动、高速、精确地进行计算和信息处理的现代化设备。从世界上第一台程序控制的电子管计算机——ENIAC 问世以来,已经历了电子管、晶体管、集成电路和大规模集成电路等发展过程,现在以超大规模集成电路和人工智能为主要特征的第五代计算机正在研制之中。

§ 1.1 电子计算机系统的组成和基本工作原理

为了说明电子计算机的组成和工作原理,我们先看看用算盘进行计算的步骤。

计算 $(38 + 25) \times 24 = ?$

其步骤为:

第一步:根据给定题目,确定计算方法,并把计算公式,解题步骤和原始数据写在纸上。

第二步:在算盘上计算 $38 + 25$,得结果 63,将此中间结果记在纸上,然后再用算盘做 63×24 ,得最后结果 1512。

第三步:将最后结果 1512 记在纸上,计算完毕。

根据以上计算步骤,可知要完成一个计算必须具备以下条件:

- (1) 类似于算盘的计算装置。
- (2) 用来存放计算题目,计算步骤,原始数据,中间结果和最后结果的记忆装置。
- (3) 控制整个计算过程的控制装置。

尽管电子计算机是现代化的设备,它比算盘不知要先进多少倍,但它并不神秘,它是模拟人的解题思维过程进行计算的,也是由类似算盘、纸、大脑等功能的运算器、存储器、控制器和输入输出设备等部分组成。

现对各个组成部分分别加以介绍如下:

1. 运算器 用来进行各种算术运算和逻辑运算。
2. 控制器 用来对计算机各个组成部分的动作进行控制的装置,它能按人们预先规定的计算顺序向有关部件发出控制命令,指挥它们协调地动作。
3. 主存储器 用来存储程序和近期要用的数据。其存储速度快,但容量有限。
4. 辅助存储器 用来存放暂时不用的程序和数据,它的存储器容量大,但存储速度慢。常用的有磁带、硬盘、软盘等。
5. 输入输出设备 用来输入程序和原始数据,输出计算结果。常用的输入设备有:键盘、盒式磁带、硬盘、软盘、纸带输入机、卡片输入机等;常用的输出设备有:打印机、终端显示器

等。

由以上五大部分组成了计算机**硬件**。其中运算器、控制器和主存储器合称为主机，运算器和控制器合称为中央处理器(CPU)，除主机外的输入输出设备和辅助存储器合称为外部设备。

计算机的硬件构成了计算机的物质基础，但要使计算机能按人们的意图完成计算任务，除了硬件外，还必须有软件的支持。

计算机软件是为了运行、管理和维护计算机而编制的各种程序的总和。软件可分为以下三大类：

1. 系统软件 由机器设计者提供的为了管理和使用计算机的软件。它包括：

(1) 各种语言的翻译程序，如汇编程序、解释程序和编译程序；

(2) 机器的监控程序或操作系统；

(3) 程序库，即是为了扩大计算机功能，便于用户使用而设置的各种标准子程序的总和。

2. 应用软件 用户利用计算机及其附属的各种系统程序而编制的用来解决实际问题的应用程序。

3. 数据库管理系统 用于信息处理和情报检索。该系统将大量的数据按一定规律组织起来，存放在计算机中，用户可根据需要查询、检索、修改、更新和删除数据库中的信息。

总之，一个完整的电子计算机系统应该包括硬件和软件两大部分。

§ 1.2 电子计算机的主要用途

电子计算机不仅用于科学计算，而且它已渗透到各行各业。除了进行数值计算外，计算机还可以用来处理各种字符、图形信息。

计算机的应用范围主要有以下几个方面：

1. 科学计算 由于计算机具有高速、精确、容量大的特点，过去人工需要几个月、甚至几年的工作量，用计算机仅需几天、几小时甚至几秒钟就可完成。例如，18世纪英国数学家用手算花了20年时间，才把圆周率计算到707位小数，而今天在高速电子计算机上，一秒钟就可以达到这个精度。

2. 信息管理 利用计算机进行信息管理可对大量数据进行采集、加工、合并、分类、传送、存储和检索等综合分析工作。数据管理已广泛应用于各个行业。据统计，目前在计算机应用中，信息管理所占的比重最大。世界上工业发达国家在企业管理中都使用了计算机信息管理系统，其中办公室自动化能实现文件的输入、存储、编辑、检索、打印和显示，使工作质量和效率大为提高。

3. 自动控制 电子计算机借助于传感器、A/D、D/A转换器组成计算机控制系统，可实现模拟信息的测量、控制、巡回检测和自动记录等，如机床数控、交通管理、机器人等。计算机不仅提管了控制水平、产品的质量和劳动生产率，而且开辟了许多新的控制领域。

§ 1.3 计算机语言

人类使用计算机的最基本的方法就是人向计算机发出各种命令或指令，让计算机按人的

意图去做。如何使计算机理解人的意图呢？只有通过一种语言来表达人的意志。当然我们不能用普通的自然语言来指挥计算机，而必须用一种为计算机能理解和识别的语言来和计算机交谈，这样一种语言就称为计算机语言，也称为程序设计语言。

计算机语言主要有以下几类：

1. 机器语言 即用计算机指令编制的程序，为二进制代码。这种语言较繁琐，直观性差，不易互相通用，一般称低级语言。

2. 汇编语言 用一种特定的助记符来表示机器指令的二进制代码，如指令 `ADD A, B` 即表示累加器 `A` 的内容和寄存器 `B` 的内容相加后送入 `A` 中。这种语言比机器语言简单、直观。但它仍是一种面向计算机的语言。

3. 高级语言 是面向问题的语言。它和自然语言接近，是一种能为计算机接受的语言。这种语言一般用英文表示，运算式和运算符号也用接近于人们常用的代数表达式或符号来表示。

高级语言把繁重的翻译工作交给了计算机来完成，从而大大简化了人的工作。

编译程序充当了高级语言和低级语言之间的翻译角色。它是作为计算机的系统软件放在计算机中的。

本书主要介绍一种高级语言，即目前国内外普遍流行且简单易学的 `BASIC` 语言。学完本书将会在 `IBM PC`、`CROMEMCO SYSTEM III`、`TRS 80`、`APPLE II`、`PC 1500` 等机器上运行 `BASIC` 程序，至于其他机型，可参考有关手册。

第二章 BASIC 语言基本概念

§ 2.1 BASIC语言特点

BASIC 是 BEGINNERS ALL-PURPOSE SYMBOLIC INSTRUCTION CODE (初学者通用符号指令代码)的缩写。它具有以下特点:

1. “人机会话”功能强

BASIC 语言是采用解释程序将语句逐句地翻译成机器代码,因而计算机能逐句地检查程序中的语句是否出现语法错误,如有错误,计算机会在显示屏或打印纸上显示出错误信息,用户可立即通过键盘修改程序,而不必把程序全部输入到机器后一起“算总帐”,这有利于程序的调试。计算机还可随时向用户询问一些信息,要求用户输入程序运行中必需的参数,用户亦可根据机器运行的情况及自己的意图随时改变输入给计算机的参数。通过终端设备与计算机进行交互式会话,使用户使用时十分灵活方便,尤其对于初学者更为适宜。

2. 简单易学

BASIC 语言的语法规则比较简单,基本语句只有十几种,且语句中涉及的英文字母、数字、运算符都和人们习惯使用的英文字符及数学运算符号差不多,易于理解和记忆。例如:“PRINT”在英语中的意思是“打印”,在 BASIC 语言中也代表“打印输出”。数学符号 $+$ 、 $-$ 、 $\times$ 、 $\div$,在 BASIC 语言中相应地表示为 $+$ 、 $-$ 、 $*$ 、 $/$ 。BASIC 语法规则和程序结构简单,没有复杂的层次结构,且尽可能地排除了语句中容易出现的二义性,学习和使用比较方便。此外 BASIC 语言还具有其他高级语言的一般特点。

§ 2.2 BASIC语言的基本概念

1. BASIC 语言的基本字符集

语言由句子组成,而单词和字是组成句子的基本成份。同样, BASIC 程序也是由一条条语句组成的,而语句由 BASIC 基本字符组成,所有基本字符的集合就构成了 BASIC 基本字符集。

BASIC 语言的基本字符集包括:

- (1) 数字集: 0~9 共 10 个数字。
- (2) 字母集: A~Z 共 26 个字母。
- (3) 运算符集: $+$ (加)、 $-$ (减)、 $*$ (乘) / (除)、 $\wedge$ 或 $\uparrow$ (乘幂)。
- (4) 标点符号集: $\cdot$ (小数点)、 $,$ (分号)、 $,$ (逗号)、 $:$ (冒号)、 $?$ (问号)、 $(($ (左圆括号)、 $)$ (右圆括号)、 $"$ (字符串括号)。
- (5) 比较符号集: $=$ (等号)、 $>$ (大于)、 $<$ (小于)、 $<>$ (不等于)、 $\geq$ (大于等于)、 $\leq$ (小于等于)。
- (6) 专用符号集: $\$$ (字符串符号)、 $\square$ (空格号)、 $\#$ (格式符号)。

2. 数值的表示及范围

BASIC 的数值采用十进制,在程序中有二种表示方法:

(1) 实数型形式 只有数字、小数点和正负号的数的表示形式称为实数型表示形式,整数表示亦属此类。例如: +25, 123.15, 123.14159, -289.54, -0.87924, 它们都是实数型形式。

(2) 指数型形式(科学记数法) 由尾数、E 和阶码三部分组成,其中尾数表示数字的有效部分,E 表示幂的底数 10,阶码表示幂的指数。其中阶码和尾数的正号可以省略,而负号则不能省略。例如: $+23E+7$ (23×10^7) $-12.345E+5$ (-1234500), $9.87654E-6$ (0.00000987654)。

指数形式一般用来表示特别大(小)的数。

应用数值编制程序时应注意以下几点:

(1) 不同机器的数的表示范围是不同的。

APPLESOFT 规定,整数的范围为 $-32768 \sim 32767$,指数范围为 $-1E38 \sim 1E38$ 。在做加法时,数字可以大到 $1.7E38$,如超过此数则为溢出。当实数的绝对值小于 $2.9388E-39$ 时,就作为机器零处理。

IBM PC 机规定,整数范围为 $-32768 \sim 32767$,实数范围为 $10E-38 \sim 10E38$ 。而 PC-1500 机,实数可表示 10 位有效数字,超过 10 位则作舍尾处理。对于指数型,数的范围为 $10E-99 \sim 10E99$,例如 12345.6789123 在机器中经舍尾处理后为 12345.67891 (保留 10 位有效数字)。123456789123 经机器处理为 $1.234567891E11$ (保留 10 位有效数字,如整数部分超过 10 位,则化为指数型)。

(2) 一个数内不能用逗号隔开。

例如: $A = 831520$ 是正确的;

$A = 831,520$ 是错误的。

(3) 指数形式的阶码只能是带符号的 1 位或 2 位整数。

例如: $1-4E121$, $1.4E6.2$, $68E(4+3)$ 等表示形式均是错误的。

(4) 计算机中数的有效位是有一定限制的,当实数型超过允许的有效位时,机器自动处理为指数型。

APPLESOFT BASIC 规定实数的有效位数不得超过 9 位,PC-1500 规定不得超过 10 位,IBM PC 机对实数有两种表示方法,即单精度和双精度。单精度表示有效数值最大为 7 位。

在 IBM PC 机中单精度实数型的例子有: 48.6, 46.81

单精度指数型用 E 表示。如 $-1.09E-06$ 。

双精度实数型用 # 号表示。如 3489.0 #。

双精度指数型用 D 表示,有效数字不得超过 17 位。

例如: 1234567890.123456789 机器处理为 $1.2345678901234568D+09$ 。

例如 $1/3$ 这个数,在数学上表示 $0.3333\cdots$;在 APPLESOFT BASIC 中为 0.333333333 (9 位有效数字);在 PC 1500 机中处理为 0.3333333333 (10 位有效数字)。在 IBM PC 机单精度处理为 0.3333333! (7 位有效数字);双精度处理为 0.3333333333333333 # (17 位有效数字)。

3. 函数

(1) BASIC 基本函数

BASIC 语言提供了一些基本常用函数, 现将 APPLESOFT BASIC 基本函数列于表 2.1。

表 2.1 BASIC 基本函数

标准函数名	功 能	备 注
SIN(X)	$\sin x$	根据 x 弧度值算出正弦值
COS(X)	$\cos x$	根据 x 弧度值算出余弦值
TAN(X)	$\operatorname{tg} x$	根据 x 弧度值算出正切值
ATN(X)	$\operatorname{arctg} x$	根据 x 弧度值算出反正切值
INT(X)	不大于 x 的最大整数	例 $\operatorname{INT}(-8.9) = -9$
RND(X)	大于等于 0 而不小于 1 的随机数	
SGN(X)	$\operatorname{SGN}(x) = \begin{cases} 1 & (x > 0) \\ 0 & (x = 0) \\ -1 & (x < 0) \end{cases}$	符 号 函 数
ABS(X)	$\operatorname{ABS}(x) = X = \begin{cases} x & x > 0 \\ -x & x < 0 \end{cases}$	取 绝 对 值
SQR(X)	$\sqrt{x}$	求 平 方 根
EXP(X)	e^x	$e = 2.718289$ 取小数点后 6 位
LOG(X)	$\ln x$	求 x 的自然对数, x 值为正

(2) 随机函数

因各机规定不一, 分别叙述如下。

IBM PC 机的随机函数:

格式 1: RND

功能: 产生 0~1 之间的均匀分布的随机数

格式 2: RND(<变量>或<数>)

功能: 当变量 > 0 , 产生 0~1 之间均匀分布的随机数。

当变量 < 0 , 重复产生以前产生过的随机数。

当变量 $= 0$, 重复产生以前产生过的最后一个随机数。

APPLESOFT BASIC 的随机函数, 同上述格式 2。

PC 1500 机的随机函数:

格式: RND(I), 其中 I 可为数、变量、算术表达式。

功能: 产生一个不大于 I 的均匀分布的随机数。

如 $I > 0$, 则产生一个不大于 1 的正整数。

如 $I \leq 0$, 则产生一个小于 1 的正小数。

4. 变量、运算符和表达式

(1) 变量

在数学中, 变量是其值变化的量。BASIC 中的变量与数学中的变量不同。在 BASIC 中, 变量是用文字符号表示的内存单元。变量的名字为变量名。变量名可由一个字母组成, 或由一个字母后跟若干个字母或数字组成。变量名的长度各计算机规定不一, 如 IBM PC/XT, 变量名可为任意长, 还可允许使用小数点, 但只有前 40 个字符有效。APPLESOFT 规定变量的长度 ≤ 238 个字符, 但只取前面 2 个字符有效。例如变量名分别为 GOODAN 和 GODDRUSH 变量, 机器把它们当作同一个变量 GO。PC 1500 规定, 变量长度 ≤ 2 , 例如 A, AB, A0, B1, DD 均可作为变量名, 而 ABC, A10, X-Y, B(2) 不可作为变量名。

BASIC 中有三种变量, 即简单变量、下标变量和字符串变量。这里先介绍简单变量, 下标变量和字符串变量在后面章节再介绍。

简单变量是用来存放数值的存储单元。上述的变量名均可作为简单变量名, 简单变量具有值, 且该值为一个数。

(2) 运算符

① 算术运算符 + (加)、- (减)、* (乘)、/ (除)、() (括号)、^ (乘幂)。

② 逻辑运算符 AND (逻辑与)、OR (逻辑或)、NOT (逻辑非)。

③ 比较运算符 $>=$ (大于等于) $<=$ (小于等于)、 $<$ $>$ 或 $><$ (不等)。

(3) 运算顺序

对于算术运算来讲, 其运算顺序为: $() \rightarrow \wedge \rightarrow *, / \rightarrow +, -$ 。

对于逻辑运算来讲, 其运算顺序为: NOT $\rightarrow$ AND $\rightarrow$ OR。

同一式中的同级运算则按顺序先左后右进行。

(4) 表达式

在 BASIC 中, 把符合 BASIC 规定的, 用运算符和括号将数、变量和函数连接起来的数学式称为 BASIC 表达式。例如:

$5 * X \wedge 2 - 3 * X - 2 * \sin(A) / 3$ 为 BASIC 的表达式, 它表示数学中的

$$5x^2 - 3x - 2 \sin A / 3。$$

在 BASIC 中规定, 表达式书写时每个字符占一格, 所有的符号都必须一个一个地排在同一横线上, 不能写成 x^2, x_3 等。字母一定要大写。下面列举一些代数式和 BASIC 表达式, 试加以对照。

代数式

BASIC 表达式

$$v_0 t + \frac{1}{2} a t^2$$

$$V_0 * T + A * T \wedge 2 / 2$$

$$P(1 + r^2)$$

$$P * (1 + R \wedge 2)$$

$$\sqrt{s(s-a)(s-b)(s-c)}$$

$$\text{SQR}(S * (S - A) * (S - B) * (S - C))$$

$$e^2 \ln x$$

$$\text{EXP}(2) * \text{LOG}(X)$$

第三章 BASIC 的输入、输出语句

§ 3.1 LET 语句

语句名称:赋值语句

格式: LET {<变量>=<表达式>}

功能:将表达式的值赋给某个变量。

例 3.1: 求二次方程 $2x^2 + 3x + 6 = 0$ 的一个根。

```
10 LET A = 2
20 LET B = 3
30 LET C = 6
40 LET X = (-B + SQR(B^2 - 4 * A * C)) / (2 * A)
50 PRINT X
60 END
```

其中 10, 20, 30 均为赋值语句,其作用是分别将 2, 3, 6 赋值给变量 A, B, C。40 是将运算后求得的值赋给变量 x 。

使用赋值语句需注意以下几点:

(1) LET 是赋值之意,表示计算机的一个动作,是将等号右边表达式的值存入左边变量所代表的内存单元中,而不能与数字上的“等号”混同。

在数学上 $x = 5 \times 3$ 亦可写成 $3 \times 5 = x$,但在 BASIC 中不允许写 LET $5 * 3 = X$ 。又例如 BASIC 程序中常常使用 LET $N = N + 1$ 表示 N 单元的内容加 1 后再送回 N 单元,而在数学中 $N = N + 1$ 显然不可能成立。

(2) 程序中可以给一个变量多次赋值,每一次赋值后即可将前一次赋的值冲掉,因而该变量的值应为最后一次赋的值。注意这里不要理解为值的叠加。

例 3.2

```
10 LET X = 5
20 LET X = 6
30 PRINT X
40 END
```

RUN ↵ 后 x 的值为 6,而不是 5,也不是 11。

(3) 不允许给一个表达式赋值。赋值语句的左边只能是一个变量,而不能是一个表达式。其原因是变量代表一个内存单元,而表达式不能代表一个内存单元。

如 LET $X + Y = 5$ 是错误的。

为了加强对赋值语句的理解,试编写将 A、B 两个变量的值交换的程序。

如果程序为:

```
10 LET A = 1
```

```

20 LET B = 2
30 LET A = B
40 LET B = A
50 PRINT A,B
60 END

```

RUN↵ 显示结果为 $A = 2, B = 2$ 。运算结果没有达到交换的目的,因为在执行 30 语句时,A 原来的值被冲掉,40 语句中赋给 B 的不是 A 原来的值了。

正确的程序为:

```

10 LET A = 1
20 LET B = 2
30 LET C = A
40 LET A = B
50 LET B = C
60 PRINT A,B
70 END

```

RUN↵ 显示结果为 A 等于 2,B 等于 1,达到了交换的目的。

(4) LET 语句还具有计算功能,BASIC 程序的运算处理,主要靠 LET 语句来完成,请看例 3-1 中的 40 语句,把赋值号右边的表达式计算出来再赋给变量 X。

(5) 表达式中的变量必须在该语句前赋过值,否则,在执行此语句时自动视该变量为零。

例 3.3: 计算 $x = B + C$, 已知 $B = 3, C = 5$ 。

若写成: 10 LET $X = B + C$ 则因 B 和 C 在 10 语句前没有赋值,故在执行 10 语句时,X 中内容为零。

```

若改写成: 10 LET B = 3
           20 LET C = 5
           30 LET X = B + C

```

则因 10,20 语句分别对 B 和 C 赋值,故在执行 30 语句时,将 B 与 C 之和 8 赋给变量 X。

(6) 由于赋值语句具有计算和赋值的功能。所以它是构成程序的基本语句,在程序中将被多次被使用。为使操作简便,大多数计算机中,赋值语句中的 LET 可以省略不写,如上述的交换 A、B 值的程序亦可写成:

```

10 A = 1
20 B = 2
30 C = A
40 A = B
50 B = C
60 PRINT A,B
70 END

```