

BASIC语言 辅导教程

孙家启
周 煦 编著
吴国凤

机械工业出版社

BASIC语言辅导教程

孙家启 周 煦 吴国凤 编著

王德民 王杰臣 姜德森 主审

机械工业出版社

内 容 简 介

本书是结合《BASIC语言程序设计》修订本(孙家启主编),并参照其他BASIC语言教本编写的。全书共十二章,分别是:计算机概论、BASIC语言的基本概念、简单程序设计、分支程序设计、循环程序设计、数组、函数与子程序、字符和表格图形设计、数据文件、绘图、汉字BASIC、结构化程序设计和三个附录,分别是:程序设计中的几个问题、DOS的使用、字处理程序WORD STAR等。每章都按学习提要、目的要求、复习思考题、基本练习题和程序设计题分别加以概括总结应学习的基本概念、要点与程序设计技巧。

本书是在多年教学与辅导学生复习的经验基础上,为学习《BASIC语言程序设计》修订本一书的高等院校、函授大学、电视大学、职工大学师生及自学者编写的,是一本很实用的学习辅导书和参考书。

BASIC 语言辅导教程

孙家启 周 煦 吴国风 编著

责任编辑:李 敬

封面设计:田淑文

*

机械工业出版社出版(北京阜成门外百万庄南街一号)

(北京市书刊出版业营业许可证出字第117号)

通县建新印刷厂印刷

新华书店北京发行所发行·新华书店经售

*

开本787×1092 1/16·印张 9·字数212千字

1990年6月北京第一版·1990年6月北京第一次印刷

印数 0,001—8,250·定价:4.20元

*

ISBN 7-111-02433-8/TP·130

前 言

本书是为《BASIC语言程序设计》修订本(孙家启主编)而编写的辅导教程(即配套教材)。全书共十二章、和三个附录,一至十二章都和《BASIC语言程序设计》修订本一书各章相对应。各章都扼要阐述了该章学习提要、目的要求,大部分章内给出了一定量的复习思考题、基本练习题和程序设计题,《BASIC语言程序设计》修订本书中的习题都在本书相应各章中列出,每道题都有参考答案,程序都经IBM PC机运行通过,一般都附有运行结果。需要说明的是:本书的习题解答,只提供一解法;事实上,许多题目可以有多种不同的解法;而且本书的习题解答是按照循序渐进规律,适应读者学习至该章为止的内容,以及利用复习和加深理解刚学习过的内容而编写的,因此,不一定每题解答都是最佳的解法。

本书的编写,力求以通俗性、知识性和实用性为指导思想。适用于学习《BASIC语言程序设计》修订本一书的高等院校、函授大学、职工业余大学,电视大学师生以及各级工程技术人员、管理人员作为自学辅导材料或参考书。

孙家启编写了本书的第一、七、八、附录I、附录II、附录III,周煦编写了第二、九、十、十二章,吴国凤编写了、三、四、五、六、十一章。甘肃工业大学王德民,哈尔滨电工学院王杰臣、姜德森、对本书进行了主审,他们对我们的编写工作给予很多的指导和帮助。在编写出版过程中,机械电子工业部教育司、机械电子工业部计算机基础教育研究会理事长张尊成教授给予极大关怀,合肥工业大学、甘肃工业大学、哈尔滨电工学院等院校教材科、函授部给予了大力支持,参加《BASIC语言程序设计》修订本一书的编者提供不少有益的资料,特借此机会一并表示诚挚的谢意。

由于我们水平有限,时间仓促,因此,在本书中定会存在不少缺点和错误,恳请读者批评指正。

编 者

1990年3月

ABS 103 / 06
ABE 10 / 06

目 录

第一章 电子计算机概论	(1)
一、学习提要	(1)
二、目的要求	(1)
三、数制与码制	(2)
四、习题与解答	(7)
第二章 BASIC语言的基本概念	(10)
一、学习提要	(10)
二、目的要求	(11)
三、复习思考、练习题	(11)
四、习题与解答	(12)
第三章 简单程序设计	(14)
一、学习提要	(14)
二、目的要求	(14)
三、复习思考题	(15)
四、基本练习题	(15)
五、习题与解答	(16)
第四章 分支程序设计	(21)
一、学习提要.....	(21)
二、目的要求	(21)
三、复习思考题	(21)
四、基本练习题	(22)
五、程序设计题	(23)
六、习题与解答	(26)
第五章 循环程序设计	(30)
一、学习提要	(30)
二、目的要求	(31)
三、复习思考题	(31)
四、基本练习题	(31)
五、程序设计题	(34)
六、习题与解答	(36)
第六章 数组	(42)
一、学习提要	(42)
二、目的要求	(42)
三、复习思考题	(42)
四、基本练习题	(42)
五、程序设计题	(44)
六、习题与解答	(46)

第七章 函数与子程序	(54)
一、学习提要	(54)
二、目的要求	(55)
三、复习思考题	(55)
四、基本练习题	(55)
五、程序设计题	(56)
六、习题与解答	(58)
第八章 字符和表格图形设计	(65)
一、学习提要	(65)
二、目的要求	(65)
三、程序设计题	(65)
四、习题与解答	(69)
第九章 数据文件	(76)
一、学习提要	(76)
二、目的要求	(76)
三、程序设计题	(77)
四、习题与解答	(81)
第十章 绘图	(85)
一、学习提要	(85)
二、目的要求	(85)
三、程序设计题	(85)
四、习题与解答	(87)
第十一章 汉字BASIC	(90)
一、学习提要	(90)
二、目的要求	(91)
三、汉字系统的应用	(91)
四、习题与解答	(93)
第十二章 结构化程序设计	(100)
一、学习提要	(100)
二、目的要求	(100)
三、分析、编写一个结构化程序	(100)
四、习题与解答	(101)
附录 I 程序设计中的几个问题	(104)
一、如何提高程序的质量	(104)
二、IBM PC BASIC 语言的几个约定	(108)
三、在使用BASIC语言时最常见的错误有哪些?	(109)
四、BASIC 语言程序模块的连接	(112)
五、BASIC语言程序怎样直接读取dBASEⅢ数据库文件	(117)
附录 II DOS的使用	(120)
一、什么是DOS	(120)
二、DOS由哪几部分组成?功能是什么	(120)
三、磁盘与磁盘文件	(120)

四、启动DOS的含义是什么	(121)
五、怎样启动 DOS	(121)
六、什么情况下需要启动 DOS	(122)
七、哪几个驱动器能启动 DOS	(122)
八、DOS的常用命令	(122)
九、批处理文件	(126)
十、DOS常用命令一览表	(128)
附录Ⅲ 字处理程序 WORD STAR.....	(130)
一、编辑文件的过程	(130)
二、编辑控制键	(131)
三、编辑技巧	(131)
四、打印文件	(134)
五、编辑非文书文件	(135)
参考文献.....	(135)

第一章 电子计算机概论

一、学习提要

对于未曾接触过计算机的读者,在开始正式学习BASIC语言之前。对于计算机有一个粗浅的了解,是顺利学习BASIC语言所绝对必要的。为此,在本章,作为预备知识,我们介绍了与学习计算机程序设计语言密切相关的一些计算机知识。

本章主要讲述了以下四个方向的问题:

1. 计算机系统的组成、基本工作原理及计算机解决实际问题的大致过程。
2. 简单地介绍了计算机机器语言、汇编语言以及高级语言的基础知识。
3. 阐述IBM PC及其兼容机的硬件及其硬件的配置。
4. 简明扼要地阐述了软件开发的一般常识。

二、目的要求

通过对本章的学习,我们要了解计算机的基本结构,程序存贮和程序控制原理。要弄清楚计算机是怎样用机器语言、汇编语言或高级语言编写程序的。还需要掌握以下几个概念:

1. 数据

计算机有各种类型,可在各种领域中应用。但它能处理的对象却是统一的——数据。数据含义是凡是能被计算机接受的,并能处理的文字、数字、符号等各种不同的组合。

2. 计算机系统

构成计算机系统有两大部分:硬件和软件。

(1) 硬件:是计算机的物理设备,如主机、键盘、打印机、终端等设备。

(2) 软件:在计算机上所使用的所有程序及其有关文件、手册、资料、说明书等。

3. 语言

首先将各类(种)语言,从数学角度给予定义:语言是一个字符表(集)上的句子的任意集合。即由有穷字母、数字、符号组成句子,再由有一定语义的句子按一定规则的任意集合。

程序设计语言可分为两大类:低级语言和高级语言。

低级语言:①机器语言;②汇编语言。

高级语言根据翻译执行方式不同可分为:①解释方式,例如多数BASIC版本;②编译方式,例如FORTRAN、PASCAL等。又可根据高级语言的针对性将高级语言称为过程语言或算法语言。

4. 源程序与目标程序

凡是用除机器语言以外其他诸种语言,如用汇编语言,各种高级语言编写的程序统称为源程序。上机运行时必须经过机内已存放的翻译程序,译成机器能执行的机器语言程序。由源程序译成的机器语言程序称为目标程序或目的程序。

承担翻译工作的程序的命名法：①担任汇编语言源程序的翻译程序称之汇编程序；②担任解释方式源程序的翻译程序称之解释程序；③担任编译方式源程序的翻译程序称之编译程序。

5. 程序与程序设计

程序：是为解决某一问题，用计算机能接受的语言，对给定的数据及其处理过程的描述。

在广义上可把程序设计定义为：数据结构、算法、方法论的统一过程。

6. 算法

算法是求解问题要求计算机进行各类操作的具体描述。如输入/输出数据，作字符比较的逻辑运算，加减、乘除各类算术运算用语言进行描述都是算法。

三、数制与码制

在BASIC语言中，主要使用十进制数。但在特殊场合也允许使用八进制和十六进制数。在BASIC语言的逻辑运算中，还要用到二进制数的概念和补码的计算方法。因此，在这里我们准备非常简单地介绍一下各种不同进制数及其转换，数的补码概念及补码的计算方法。

1. 数制

在日常生活和工作中使用的数都是十进制数。我们知道，十进制数有如下三个特点：

(1) 任何一个十进制数都是用十进制系统中的数字(指Digit)字符表示的。十进制系统的数字字符为：0, 1, 2, 3, ..., 9。例如：53, 0.2, 3.14等都是十进制数。

(2) 一个数的大小是由数字字符在数中的位置决定的。设有一个十进制数。

$$d_n d_{n-1} \cdots d_0 d_{-1} d_{-2} \cdots d_{-m}$$

其中每个 $d_i (i=n, n-1, \dots, 0, -1, -2, \dots, -m)$ 都是一个数字字符(0, 1, ..., 9中的一个)。这个数可写成展开式为：

$$N = d_n \times 10^n + d_{n-1} \times 10^{n-1} + \cdots + d_0 \times 10^0 + d_{-1} \times 10^{-1} + \cdots + d_{-m} \times 10^{-m}$$

例如，945.236可写成：

$$9 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2} + 6 \times 10^{-3}$$

式中的“10”是十进制数的基数(Radix)。

(3) 十进制数的计数规则是“逢十进一”，“借一当十”。

上述三个特点，刻画了数制中最本质的内容，从这三个特点出发，不难了解其它数制。

在电子数字计算机中，数是以二进制数的形式存贮的。所谓二进制数就是以“2”为基数的数，它有两个数字字符：0和1，它的计数规则是：“逢二进一”，“借一当二”。一个九位二进制数101100101的展开式为：

$$1 \times 2^8 + 0 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

计算上面的展开式得357，这就是那个二进制数所表示的十进制数。

按照“逢二进一，借一当二”的计数规则，我们很容易进行二进制数的四则运算。

例1 计算两个二进制数相加：10011.01 + 100011.11

$$\begin{array}{r} 10011.01 \\ +) 100011.11 \\ \hline 110111.00 \end{array}$$

即 $10011.01 + 100011.11 = 110111.00$

例2 计算两个二进制数相减：10110.01 - 1100.1

$$\begin{array}{r} 10110.01 \\ - 1100.1 \\ \hline 1001.11 \end{array}$$

同样，根据上述数制的三个特点，可以描述计算机技术中广泛使用的八进制数和十六进制数。

八进制数是以“8”为基数的数，它有八个数字字符：0, 1, 2, ..., 7。八进制数的计数规则是：“逢八进一，借一当八”。其展开式为：

$$N = \sum_{i=-m}^m d_i \cdot 8^i$$

按照“逢八进一，借一当八”的规则，可以进行八进制数的四则运算。

例3 计算八进制数35与67的和。

$$\begin{array}{r} 35 \\ + 67 \\ \hline 124 \end{array}$$

即 $35 + 67 = 124$ 。

十六进制数是以“16”为基数的数。这种数的16个数字字符是0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F。计数规则是：“逢十六进一，借一当十六”。十六进制数5B E的展开式为：

$$3 \times 16^3 + 5 \times 16^2 + B \times 16^1 + E \times 16^0$$

把数字字符B看做十进制数的11，E看做是十进制数的14，上式经十进制计算后得13758。这就是十六进制数35BE的十进制表示。

按照“逢十六进一，借一当十六”的计数规则，可以进行十六进制数的四则运算。

由于使用了不同进位制的数，所以在写一个数时，必须声明它的进位制，例如：“二进制数11010”，“八进制数65”“十六进制数36BE”等。显然，这种方法不太方便。今后，当需要强调数的进位制时，就把数用圆括号括起来，在括号右下方用阿拉伯数字表示该数的进位基数。例如十六进制数36BE表示为 $(36BE)_{16}$ ，八进制数65表示为 $(65)_8$ ，二进制数11010表示为 $(11010)_2$ 。

2. 不同进制数之间的转换

在使用计算机进行各种运算时，常需要进行不同进制数之间的变换。

二进制数、八进制数和十六进制数转换为十进制数，可直接用展开式实现。例如：

$$(1011)_2 = 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0 = (11)_{10}$$

$$(35)_8 = 3 \times 8^1 + 5 \times 8^0 = (29)_{10}$$

$$(35)_{16} = 3 \times 16^1 + 5 \times 16^0 = (53)_{10}$$

十进制数转换为二进制、八进制和十六进制数，是上述转换的逆转换。因此，转换方法仍是以展开式为基础。

根据展开式，可以得到如下的转换方法。

十进制正整数N的P进制数的求法称为“除P取余法”。即用P去除N，得一整数商和一余数。此余数就是P进制数的最低位。用P再去除所得商，又得一整数商和一余数。此余数是p进制数的次低位。这样不断地除下去，直到商为零时为止。

例4 将十进制477转换为八进制数、十六进制数和二进制数。

除8取余：

$477 \div 8 = 59$	余5	↑
$59 \div 8 = 7$	余3	
$7 \div 8 = 0$	余7	

即 $(477)_{10} = (735)_8$

除16取余：

$477 \div 16 = 29$	余13(D)	↑
$29 \div 16 = 1$	余13(D)	
$1 \div 16 = 0$	余1	

即 $(477)_{10} = (1DD)_{16}$

除2取余：

$477 \div 2 = 238$	余1	↑
$238 \div 2 = 119$	余0	
$119 \div 2 = 59$	余1	
$59 \div 2 = 29$	余1	
$29 \div 2 = 14$	余1	
$14 \div 2 = 7$	余0	
$7 \div 2 = 3$	余1	
$3 \div 2 = 1$	余1	
$1 \div 2 = 0$	余1	

即 $(477)_{10} = (111011101)_2$

十进制小数 $M(0 < M < 1)$ 的P进制数的求法称为“乘P取整法”。就是用P去乘M，得一乘积，其整数部分就是P进制小数的最高位；用P再去乘上面所得积的小数部分，又得一乘积，其整数部分是P进制小数的次高位。这样不断地乘下去，直到乘积的小数部分为零时为止。

例5 求十进制数0.451171875的八进制数、十六进制数和二进制数。

乘8取整：

$0.451171875 \times 8 = 3.609375$	取3	↓
$0.609375 \times 8 = 4.875$	取4	
$0.875 \times 8 = 7.0$	取7	

即 $(0.451171875)_{10} = (0.347)_8$

乘16取整：

$0.451171875 \times 16 = 7.21875$	取7	↓
$0.21875 \times 16 = 3.5$	取3	
$0.5 \times 16 = 8.0$	取8	

即 $(0.451171875)_{10} = (0.738)_{16}$

乘2取整:

$0.451171875 \times 2 = 0.90234375$	取0	↑
$0.90234375 \times 2 = 1.8046875$	取1	
$0.8046875 \times 2 = 1.609375$	取1	
$0.609375 \times 2 = 1.21875$	取1	
$0.21875 \times 2 = 0.4375$	取0	
$0.4375 \times 2 = 0.875$	取0	
$0.875 \times 2 = 1.75$	取1	
$0.75 \times 2 = 1.5$	取1	
$0.5 \times 2 = 1.0$	取1	

即 $(0.451171875)_{10} = (0.011100111)_2$

值得注意的是, 不是任何十进制小数都能完全精确地化成二进制、八进制或十六进制小数。这表现在“乘P取整”时, 乘积的小数部分永远不会出现0。这时, 可根据题目要求的精度, 换算到一定位数即可结束。

一个既有整数部分又有小数部分的十进制数, 在转换为其它进制数时, 可按上述方法, 对整数部分和小数部分, 分别进行转换。

最后, 说明一下, 二进制、八进制和十六进制数之间的转换问题。

二进制数、八进制数和十六进制之间的转换比较简单。只须注意的是: 在把二进制数的各位, 按三位一组或四位一组进行分组时, 整数部分是由低位向高位分, 最后不足三位(或四位)时, 在最高位前补齐0, 小数部分则是由高位向低位分, 最后不足三位(或四位)时, 在最低位后补齐0。

例6 将二进制数 10110100111.10110111 用八进制数表示; 把八进制数 735461.14 用二进制表示。

10	110	100	111	·	101	101	110
←	←	←	←		→	→	→
↓	↓	↓	↓		↓	↓	↓
2	6	4	7	·	5	5	6

即 $(10110100111.10110111)_2 = (2647.556)_8$

7	3	5	4	6	1	·	1	4
↓	↓	↓	↓	↓	↓		↓	↓
111	011	101	100	110	001	·	001	100

即 $(235461.14)_8 = (111011101100110001.001100)_2$

例7 将二进制数 10110100111.10110111 表示成十六进制数; 将十六进制数 $32DF$ 转换成二进制数。

101	1010	0111	·	1011	0111
←	←	←		→	→
↓	↓	↓		↓	↓
5	A	7	·	B	7

即 $(10110100111.10110111)_2 = (5A7.B7)_{16}$

3	2	D	F
↓	↓	↓	↓
0011	0010	1101	1111

即 $(32DF)_{16} = (0011001011011111)_2$

3. 数的原码、补码和反码

码制是解决二进制数在计算机中的形式问题。数在计算机里面的表示形式称为机器数。

机器数首先要解决的一个问题是数的正负号的表示方法问题。在计算机中，正号用0表示，负号用1表示，并且通常放在数的最高位之前。例如： $(+7)_{10}$ 和 $(-7)_{10}$ 在机器中分别表示为（设机器字长为16位）：

+7 0000000000000111

-7 1000000000000111

数的这种表示方法称为原码。常用 $[7]_{\text{原}}$ 和 $[-7]_{\text{原}}$ 表示。

两个原码数进行相加时，首先要判别两数的符号是否相同。如果符号相同，就进行加法运算，其结果数的符号就是参加运算数的符号。如果两数的符号不同，就要进一步比较两数绝对值的大小，然后，绝对值大的数做被减数，另一个做减数，进行减法运算。被减数的符号为结果数的符号。

从上述计算加法的过程中可以看出，即使是一个很简单的加法运算，其过程也是相当复杂的，这必然导致运算器的复杂化和计算时间的增长。

既然如此，试问有没有办法，不管数是正还是负，符号和数做为一个整体参加一次加法运算，就能自然得到正确的结果呢？这种办法是有的。这就是下面要着重说明的补码问题。

目前，绝大多数的计算机，数都用补码表示。为了说明补码，现讲一下数的反码。

一个正数的反码，等于该数的原码。一个负数的反码，等于该数的原码的各二进制位的0变1、1变0、符号位不变所形成的结果。

例如，+7和-7的反码为：

$[+7]_{\text{反}} = 0000\ 0000\ 0000\ 0111$

$[-7]_{\text{反}} = 1111\ 1111\ 1111\ 1000$

由反码很容易得出补码。一个正数的补码等于该数的原码。负数的补码等于该数的反码在末位上加1。

$[+7]_{\text{补}} = 0000\ 0000\ 0000\ 0111$

$[-7]_{\text{补}} = 1111\ 1111\ 1111\ 1001$

数采用补码表示后，加法就变得非常简单。这时，进行两个数相加，无论其正负号如何，作一次它们的补码的加法就得到正确的结果。不过，这结果仍是补码的形式。上述运算可表示为：

$$[A]_{\text{补}} + [B]_{\text{补}} = [A+B]_{\text{补}}$$

例8 设机器字长为16位。已知 $A=9$ ， $B=-4$ ，用补码计算 $A+B$ 。

$[A]_{\text{补}} = [9]_{\text{补}} = 0000\ 0000\ 0000\ 1001$

$[B]_{\text{补}} = [-4]_{\text{补}} = 1111\ 1111\ 1111\ 1100$

$$\begin{array}{r} 0000\ 0000\ 0000\ 1001 \\ +) 1111\ 1111\ 1111\ 1100 \\ \hline 1\ 0000\ 0000\ 0000\ 0101 \quad (\text{符号位前面的1丢掉}) \end{array}$$

即 $[9]_{\text{补}} + [-4]_{\text{补}} = [9-4]_{\text{补}} = 0000000000000101$

由于结果是正数补码，可知计算结果为 $(101)_2 = (5)_{10}$

例9 设机器字长为16位，已知 $A=-9$ ， $B=4$ ，用补码计算 $A+B$ 。

$$[A]_{\text{补}} = [-9]_{\text{补}} = 1111 \ 1111 \ 1111 \ 0111$$

$$[B]_{\text{补}} = [4]_{\text{补}} = 0000 \ 0000 \ 0000 \ 0100$$

$$[A]_{\text{补}} + [B]_{\text{补}} = 1111 \ 1111 \ 1111 \ 1011$$

计算结果是负数的补码 $[A+B]_{\text{补}}$ 。

如果负数补码的各位（符号位除外）0变1，1变0，末位上再加1，就得到该数的原码：

$$[A+B]_{\text{原}} = 1000 \ 0000 \ 0000 \ 0101$$

$$A+B = (-101)_2 = (-5)_{10}$$

所以，现在绝大多数的电子计算机都采用补码运算。

四、习题与解答

1. 计算机系统是怎样组成的？

答：一个完整的计算机系统由硬件和软件两部分组成。

计算机硬件由运算器、控制器、存储器、输入设备、输出设备等组成。

运算器和控制器也合称为中央处理器，简称CPU。CPU和内存存储器合称为主机。输入、输出设备加上外存储器称为外部设备。终端设备、脱机设备、数/模转换器、模/数转换器、开关量输入输出器等则称为外围设备。

计算机硬件是可以看得见、摸得着的电子机械装置。

计算机软件包括系统软件和应用软件以及相应的文件。系统软件包括操作系统、程序设计语言、非程序设计语言和实用程序。计算机应用就是通过应用软件来实现。

2. 试述IBM PC机的硬件组成。

答：目前IBM PC机几乎成为微机的一个标准，国际上大都向IBM PC机靠拢，我国也生产了它的兼容机0520系列机，作为我国微型机的优选机种，并且配上了汉字，已在我国大量推广与普及。

IBM PC机最小的硬件配置只需要三个部分，即键盘、显示器和主机箱。键盘是计算机输入设备，显示器是计算机的输出设备。主机箱内安装了系统板，系统板上装有CPU和内存存储器。为了使显示器能工作，主机箱内安装了一块选件板，上面是IBM黑白显示器和并行打印机适配器。

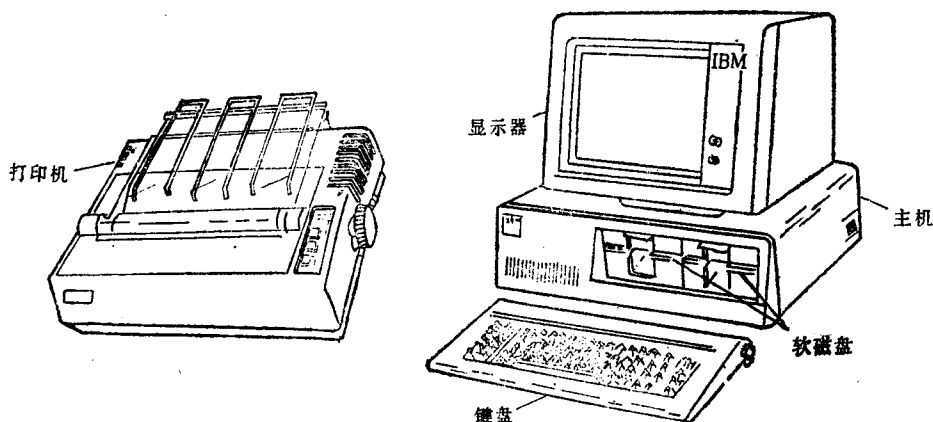


图 1-1

由于计算机厂家提供的常常是基本配置，它在最小配置的基础上，通过存贮器扩充板可以把内存扩至640kB，增加磁盘驱动适配器和两部软磁盘驱动器作为外存贮器（PC/XT配一软盘驱动器和一台温氏硬盘），增加打印机作为硬拷贝输出，打印的字符，可以放大、缩小、加黑及正常打印。打印机通过并行打印机接口与系统装置相连。IBM PC微型机系统设备的基本配置，见图1-1。

3. 使用Wordstar进行指法练习，测试你每分钟输入的字符数。

答：优秀打字员可达650~670字符/分。学生应达到不低于100字符/分。

4. 程序设计方法有哪些？

答：（1）自顶向下的设计方法；

（2）模块化程序设计；

（3）程序的分层结构设计；

（4）结构化程序设计；

（5）逐步求精的设计方法。

5. 用计算机解题分哪几步？

答：使用计算机解决实际问题，大体上有以下几个步骤：

（1）提出和分析问题——就是针对所研究的物理过程，分析和研究已知数据、所要求解的问题以及求解的精度等。

（2）构造数学模型——通过上述分析研究，把实际的物理过程用一系列数学算式表达出来，这就是构成数学模型的过程。例如，有些问题可归结为算式：

$$B = \sum_{k=0}^n a_k x^{n-k}; y = 2 \int_0^x \sin(x) dx \text{ 等，这些就是数学模型。}$$

（3）选择计算方法——对所建立的数学模型，选择适当的计算方法。有时建立起来的数学模型不一定适合于计算机直接进行计算。例如，计算机直接按 $B = \sum_{k=0}^n a_k x^{n-k}$ 求 B 值，

当 n 很大时，计算工作量就相当大，需要较长的计算时间。而用下式

$$B = ((\dots((a_0 x + a_1)x + a_2)x + a_3)x + \dots + a_{n-2})x + a_{n-1})x + a_n$$

计算 B 值，不仅可以减少计算工作量，而且还可以简化程序的编制。有些复杂的数学模型不能直接用计算机进行计算，例如， $y = 2 \int_0^x \sin x dx$ 。这时需用某些近似计算方法。对一些复杂问题，要求找出近似的计算方法和合适的算法并不是一件很简单的事。它是“数值计算方法”课程所要解决的问题。

（4）画流程图——流程图是算法的一种表现形式，也可以称为流程图语言，它往往比计算机程序设计语言描述算法更为直观，逻辑表达更为清晰。

（5）编写计算程序——用程序设计语言把计算过程描述出来，就是进行程序设计。简单的计算问题，可以根据算式，直接用程序设计语言把程序写出来。对于比较复杂的计算问题，在编写程序之前，还需用流程图把计算步骤直观地表达出来。然后，再按照流程图，用程序设计语言编写程序。

（6）上机计算——将程序和原始数据送入计算机进行计算。上机过程是调试和修改程

序的阶段，因为程序中难免会有些错误。这种调试和修改有时要反复进行多次，最后才能得到正确的程序和计算结果。

(7) 分析结果——计算机输出的结果是否正确，要细心分析。从实际物理过程到计算机输出结果，经过了许多过程。数学模型不合适，计算方法不正确或精度不够，程序中有错，都会得到不能令人满意的结果。这时应当重复上述某些步骤，直到获得正确结果。

(8) 写出程序说明书——在程序经过调试正常运行及分析结果正确后，写出程序说明书是必要的。

本书的中心任务是解决第五个步骤，即编制程序的问题。同时也简要地介绍一些常用的数值计算方法，即第三个步骤的问题。而第六个步骤上机计算，则是本课程的实践环节，读者应给以足够的重视，许多程序设计中的错误，是在上机运行时发现的。

6. 程序说明书应包括哪些内容？

答：程序说明书应包括：题目、任务要求、算法、流程图、程序清单、操作说明、设备的配置、变量注释、程序注释、输入数据的内容及形式、输出结果的内容及格式、不正确的操作可能产生的错误信息和出错后的处理办法等。

第二章 BASIC语言的基本概念

一、学习提要

本章主要介绍BASIC语言的程序构成及BASIC语言的基本成分和语法规则。

一个BASIC语言的程序，一般都由注释部分、输入部分、工作部分和输出部分构成，只是对于不同的程序，构成的语句不同而已。程序中的每个语句行又有：行号、语句定义符、语句体和语句终止符组成。行号是语句的标志。BASIC系统通常是按照行号的顺序（从小到大）来执行程序。语句定义符规定每条语句的具体功能。语句体规定每条语句具体执行的内容。语句终止符是每条语句行的结束标志。每当一个语句行送入计算机后，通常按“RETURN”键，以表示一个语句行的结束。

BASIC语言的基本成分包括：常量、变量、函数、表达式。

1. 常量 { 数值常量
字符常量
逻辑常量

数值常量（即常数）：它分三种类型：整型、单精度型、双精度型。整型数的取值范围从 $-32768 \sim 32767$ 之间所有的十进制整数；单、双精度数的取值范围，绝对值在 $2.9 \times 10^{-38} \sim 1.7 \times 10^{38}$ 之间所有的数。常数在书写格式上分：定点格式和浮点格式两种。所谓定点格式就是按照数学上十进制的日常书写格式，即在一个常数中，只出现正负号、小数点以及0~9十个数字符号；所谓浮点格式就是在定点格式的基础上，增加了以10为底的指数部分。不过，在BASIC语言中，底数10要用E（单精度符号）或D（双精度符号）来代替。

字符常量：是用一对双引号括起来的一串字符，系统能接受的字符（双引号除外）都可以作为字符常量的成分。一个字符常量的字符个数为0~255个。

逻辑常量只有两个：TRUE（真，一般用“1”表示）、FALSE（假，一般用“0”表示）。逻辑常量主要是用来表示关系运算或逻辑判断的结果。

2. 变量 { 数值变量
字符串变量

数值变量是用来存放数值，**字符串变量**是用来存放字符串。变量名的命名要遵循BASIC语言的规定，必须以字母开头的字母、数字、小数点组成的序列。不允许用希腊字母，如： α , β , γ , δ , π 等和语句定义符作为变量名。变量的类型一般也分为：整型、单精度型、双精度型。它们可以用类型说明符号或者用变量类型定义说句这两种形式来确定。

3. 函数

BASIC系统提供了两类函数：一类算术函数；一类是三角函数。它们统称标准函数，也叫内部函数或叫库函数。这些函数已编成了一个独立的子程序，当用户引用时，只要写出相应的函数名及其参数，计算机则会自动算出相应的函数值。