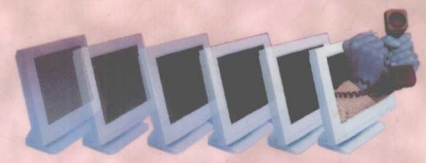
安徽省教育厅组编



计算机基础教育系列教材
JISUANJI JICHU JIAOYU XILIE JIAOCAI

C语言程序设计教程

吴国凤 周恒忠 胡宏智 冯崇岭 孙家启/编著

C YUYAN CHENGXU SHEJI JIAOCHENG

安徽大学出版社

图书在版编目(CIP)数据

C语言程序设计教程/吴国凤,周恒忠编著. -合肥:
安徽大学出版社,2003.1
安徽省计算机基础教育教材
ISBN 7-81052-632-4

Ⅰ.C… Ⅱ.①吴… ②周… Ⅲ.C语言-程序设计
高等学校-教材 Ⅳ.TP312

中国版本图书馆CIP数据核字(2002)第109561号

C语言程序设计教程 吴国凤 周恒忠 胡宏智 冯崇岭 孙家启 编著

出版发行	安徽大学出版社	**印　刷**	合肥中德印刷培训中心印刷厂
	(合肥市肥西路3号　邮编230039)	**开　本**	787×1092 1/16
联系电话	编辑室0551-5106428	**印　张**	17.5
	发行部0551-5107784	**字　数**	426千
责任编辑	李　虹	**版　次**	2003年1月第3版
封面设计	孟献辉	**印　次**	2003年1月第1次印刷
经　　销	新华书店		

ISBN 7-81052-632-4/T·86　　定价　21.80元

内容简介

本书是安徽省教育厅组编的计算机基础教育系列教材之一。根据国家教育部计算机基础课程教学指导委员会颁布的“C语言程序设计”教学要求，全面地、系统地叙述C语言的语法知识及其程序设计方法和技能。

本书兼顾了全国高等学校(安徽考区)计算机等级考试需要，其内容覆盖了二级C语言程序设计教学(考试)大纲的要求，所以它又是一本考试指导书——全国高等学校计算机等级(水平)考试系列教材之一。本书另配套有《C语言程序设计题解与实验》。

全书共分12章：C语言概述，数据类型和运算，输入和输出，语句和流程控制，数组，函数，指针，结构、联合与枚举，编译预处理，位运算，文件，C++基础等。内容丰富、系统性强、深入浅出，各章均有程序举例、要点及例题分析，还附有针对性习题，便于读者学习和自测，对备考极具参考价值。

本书特别适用于做普通高校本、专科非计算机专业的教材或成教、夜大、函大计算机专业的教材，也可供参加全国高等学校计算机等级(水平)考试考点学校使用，还可供广大计算机自学者、工程技术人员参考。

编委会名单

编 写 说 明

根据安徽省教育厅指示，为了推动计算机基础教育改革与建设，促进计算机基础课程教学与水平考试向纵深发展，我们按照计算机文化基础教育、技术基础教育和应用基础教育三个层次，组织编写了计算机基础教育系列教材。这套教材囊括了计算机文化基础、高级语言（QBasic，Visual Basic，C，Visual C++，PASCAL，FORTRAN77，FORTRAN90，FoxBASE⁺，FoxPro 2.5b For Windows，Visual FoxPro 等）程序设计、软件技术基础、微型计算机原理及应用、计算机网络微型组装与维护等方面内容，涵盖全国高校计算机水平考试的一、二、四级（全国等级考试的一、二、三级），因而具有广泛的适应性。这套教材所具有的突出特点是：紧扣计算机基础教育大纲（即计算机水平考试大纲），兼具普通教材与考试辅导材料的双重功能；立意创新，内容简练，大量针对性极强的习题和典型例题分析是其他教材所少见；编写人员都是教学、科研第一线有着丰富教学与实践经验的教师，他们深谙相关知识点的张弛取舍。我们还聘请了三位知名专家担任高级顾问，以确保本系列教材的编写质量。

本系列教材的先期版本现已问世，第一辑各册已于 2000 年底全部出齐。由于计算机技术的发展比人们想像的还要快，所以本系列教材在使用过程中，根据计算机技术发展及教学要求，进行了多次修改，增加了不少新内容，今后我们还将不断调整、更新教材内容、平台和版本，使之与当时发展相适应，以便教材以更新更好的面目呈现在读者面前。

本系列教材编写目的明确，它特别适合于作为普通高校非计算机专业的本、专科教学用教材或成教、夜大、函大计算机专业的教材，也可供各地计算机水平考试考点使用，还可供广大计算机自学者、工程技术人员参考。

编写委员会

2000 年 5 月

前　言

C语言是近年来国内外得到最迅速推广使用的一种现代编译型程序设计语言，它兼顾了多种高级语言的特点，具备汇编语言的功能。C语言程序处理功能强、运算速度快、目标效率高，具有完善的模块程序结构，可移植性强，而且可以直接实现对系统硬件及外部设备接口的控制，具有较强的系统处理能力。在当今世界技术先进国家中，使用C语言进行程序设计已成为软件开发的一个主流。1983年在C语言的基础上推出的面向对象的程序设计语言C++，在90年代得到迅猛发展。C++语言今后将成为最流行的一种计算机主流语言。掌握好C/C++语言已成为当今软件工作者的必备条件之一。

为了在我国高等学校更快地推广和普及C语言，编者根据国家教育部全国高等学校计算机基础课程教学指导委员会制订的C语言程序设计教学要求，兼顾全国高等学校（安徽考区）计算机等级（二级C语言程序设计）教学（考试）大纲编写此教材的。由于我国目前高等学校广泛使用的PC系列微型机上的C编译系统TurboC 2.0为实现版本，全面地、系统地讨论了C语言语法知识以及程序设计方法和技巧。书中各章均有要点及例题分析，还附有针对性的习题便于读者学习和自测，对参加全国高校计算机等级（二级C语言程序设计）考试极具参考价值。根据广大读者要求，本书第二次修订时，将书中各章内容作了充实并增加程序举例一节，又将原书第6、7两章合并成为第6章，增加了第12章C++基础，书中大部分章节习题内容也作了充实。本书还配套有《C语言程序设计题解与实验》。

本书由吴国凤、周恒忠、胡宏智、冯崇岭、孙家启合作编写。全书由孙家启修改定稿。在成书过程中，得到了省高校同行专家们的大力支持，在此表示感谢。

由于编写时间仓促，加之水平有限，难免有疏漏、错误之处，恳请读者批评指正。

编者

2002年10月

目 次

语言使用的+,-,*,/四则运算及与(and)、或(or)、非(not)等逻辑运算功能外,还可以实现以二进制位(bit)为单位的位与(&)、位或(|)、位非(~)、位异或(^)以及移位(>>,<<)等位运算,并且具有如a++,b--等单项运算和+=,-=,*=,/=等复合运算功能。

(5)C语言的数据类型丰富,它的数据类型有:整型、实型、字符型、数组类型、指针类型、结构体类型、联合体类型和枚举类型等,能用来实现各种复杂的数据结构。因此,C语言具有较强的数据处理能力。

(6)C语言程序中可以使用如#define、#include等编译预处理语句,能进行字符串或特定参数的宏定义,以及实现对外部文本文件的读取和合并。同时还具有#if、#else等条件编译预处理语句。这些功能的使用提高了软件开发的工作效率,并为程序的组织和编译提供了便利。

(7)C语言程序可移植性强,C语言程序本身并不依存于机器硬件系统,从而便于在硬件结构不同的机种间和各种操作系统实现程序的移植。

由于C语言具有上述众多特点,近年来迅速地得到普及和应用。许多大型的软件系统都用C语言编写,许多以前只能用汇编语言处理的问题现在可以改用C语言来处理。C语言被称为"高级汇编语言"。

C语言的优点很多,但和其他程序设计语言一样,它也有弱点,如运算符的优先级较多,有些还与常规约定不同,不便记忆;某些语法部分不易用形式化方法进行描述;各种C语言版本之间略有差别,缺乏统一的标准;C语言不是强类型的语言,它强调灵活、高效的同时,在一定程度上牺牲了某些安全性,如类型检验太弱,转换比较随便等。因此,C语言对程序设计人员提出了较高的要求,尤其在使用C语言的某些高级手段时更是如此。但是,C语言的优点远远超过了它的弱点,这些优点使C语言具有强大的吸引力。实际经验表明,程序设计人员一旦接触了这种语言,并且有一定程序设计经验后,就会对它爱不释手。

1.3 C语言的基本词法

任何程序设计语言如同自然语言一样,都具有自己一套对字符、单词及一些特定符号的使用规定,也有对语句、语法符方面的使用规则。在C语言中,所涉及的规定很多,其中主要有:字符集、保留字、标识符、语句和标准库函数等。这些规定构成了C程序的最小的语法单位。

1.3.1 字符集

一个C程序是C语言基本字符构成的一个序列。C语言的字符集就是ASCII字符集,主要分为下列几类:

(1)大小写英文字母(52个)。

(2)数字(10个)。

(3)键盘符号(33个),如表1-1所示。

表 1-1 键盘符号表

~	波浪号	)	右圆括号	:	冒号
、	重音号	_	下划线号	;	分号
!	惊叹号	-	减号	"	双引号
@	a圈号	+	加号	'	单引号
#	井号	=	等号	<	小于号
$	美元号	|	或符号	>	大于号
%	百分号	\	反斜杠	,	逗号
^	异或号	{	左花括号	.	小数点
&	与符号	}	右花括号	?	问号
*	星号	[	左方括号	/	(正)斜杠
(	左圆括号	]	右方括号		空格符号

(4)转义字符。

转义字符如表 1-2 所示,是由"反斜杠字符(\)"开始后跟单个字符或若干个字符组成的,通常用来表示键盘上的控制代码或特殊符号,例如回车换行符、响铃符号等。

表 1-2 转义字符表

✓	\n	回车换行符号	\a	响铃符号
✓	\t	Tab符号	\"	双引号
	\v	垂直制表符号	\'	单引号
	\b	左退一格符号	\\	反斜杠
	\r	回车符号	\ddd	1~3位8进制数ddd对应的符号
	\f	换页符号	\xhh	1~2位16进制数hh对应的符号

1.3.2 标识符

标识符是用户自定义的一种字符序列,通常用来表示程序中需要辨认的对象名称,比如符号常量、变量、数组、函数等对象的名字。

C语言规定,标识符是由字母或下划线开头的字母、数字、下划线组成的一串符号,ANSI C规定标识符长度不得大于32个字符,而PC机中通常是前8个字符有效。

下面给出一些正确和错误的标识符。

正确的标识符:name　i　b4　b_4　_b4　_b_4

错误的标识符:4b　(非字母或下划线开头)

b?　(含有非字母、数字、下划线的字符:?)

c.g　(含有非字母、数字、下划线的字符:.)

b-2　(含有非字母、数字、下划线的字符:-)

由于标识符主要用来命名,所以用户应选取有意义的标识符,以便在程序中能从标识符看出所标识的对象。

说明:

(1)在C语言中,标识符中大小写字母是有区别的。程序中基本上都采用小写字母表示各种标识符,如变量名、数组名、函数名等。书写的各种语句也均用小写字母,而大写字母只用来定义宏名。

(2)C语言规定,用户选取的标识符不是C语言规定的保留字。

(3)在C程序中,标识符的使用很多,使用时要注意语言规则。定义标识符时,一般应做到见名知意,以提高程序的可读性。

1.3.3 保留字

在C语言的程序中有特殊含义的英语单词称为"保留字",主要用于构成语句,进行存储类型和数据类型定义。这些特定的保留字不允许用户作为自定义的标识符使用。

C语言的保留字如下:

auto	自动	extem	外部	sizeof	计算字节数
break	中止	float	浮点	staic	静态
case	情况	for	对于	struct	结构
char	字符	goto	转向	switch	开关
continue	继续	int	整	typedef	类型定义
const	常量	if	如果	union	共用
default	缺省	long	长	unsigned	无符号
do	做	register	寄存器	void	空
double	双	return	返回	voltile	可变的
else	否则	short	短	while	当
enum	枚举	signed	带符号		

1.3.4 C语言的词类

C语言的词类主要分为:

(1)常量。在程序运行中其值是不变的数据。

(2)变量。用来存放程序运行中变化的数据,例如输入的原始数据、中间结果、最终结果等。

(3)运算符。用来表示简单加工计算的符号,如+(加)、-(减)、*(乘)、/(除)等。

(4)函数调用。形如"函数名(实际参数表)"的式子,它代表调用指定函数后获得的结果。

(5)表达式。用常量、变量、函数调用、运算符组成的式子,用来表示简单的加工计算。

(6)保留字。在程序或语句中,用来表示特定语法含义的英语单词。

上述词类的构成规则将在以后章节中介绍。

1.3.5 语句

语句是组成程序的基本单位,用以完成特定的操作,语句的有机组合序列能实现指定的计算处理功能。C语言中的语句分为:

(1)数据定义语句。用来定义程序中使用的各种能存放数据的对象的名称和特性。

(2)赋值语句。形如"变量=表达式"的语句,功能是计算表达式的值并赋予变量。

(3)函数调用语句。形如"函数名(实际参数表)"的语句,功能是调用指定函数。

(4)表达式语句。由任何表达式组成的语句。在C语言中赋值和函数调用都是表达式,所以赋值语句和函数调用语句也是一种特殊的表达式语句。

(5)流程控制语句。用来控制程序执行过程的语句,如选择控制语句、循环控制语句、中

止语句、继续循环语句、返回语句、无条件转移语句等。

(6)复合语句。用花括号括住的若干个任意语句。

(7)空语句。无任何操作的语句。

(8)其他语句。包括编译预处理命令、类型定义语句等。

上述语句的形式和功能将在以后的章节中陆续介绍。

1.4 C程序的基本结构

任何一种计算机程序设计语言,都具有特定的语法规则、语义和一定的表现形式。程序的书写格式和程序的构成规则是程序语言表现形式的一个重要方面。按照规定的书写格式和构成规则书写程序,不仅可以使程序设计人员和使用程序的人容易理解,更重要的是把程序输入给计算机时,计算机能够充分认识,从而能够正确执行它。

先从一个最简单的程序看起,该程序的功能是在屏幕上显示:Hello!

【例1.1】打印一个语句。

```
main()
{
    printf("Hello!"\n);
}
```

任何一个C语言源程序,都必须经过编译、连接后方可执行,不同种类的C语言,其编译、连接方式也不同。由于当前大部分的教学和考试都是以Turbo C 2.0为主,故本书以此作为主要考虑对象。要想运行该程序,按下Ctrl+F9,系统对程序进行编译、连接,若程序没有什么错误,则在屏幕上输出Hello!,然后回到程序编辑状态,想看程序运行结果,可按Alt+F5键。

现在,对程序本身作一解释。一个C语言程序,无论其大小,皆由一个或多个"函数"组成,而"函数"完成要做的操作。上例中,main()就是一函数。但main是一个特殊名,任何程序有且仅有一个main,且程序总是从main的开头执行。main的具体结构为:

```
main( )          /*函数名*/
{                /*函数开始*/
   :             /*函数体*/
}                /*函数结束*/
```

一个函数可调用另一个函数。函数间也可用参数来交换数据。函数名后的一对括号内括着一个参数表。例子中的main函数没有使用参数,故为一对空括号(不能省)。花括号{ }用以包围构成函数的语句(即函数体)。例子中,函数体是由一条语句组成的。

```
printf("Hello!    \n");
```

它是一个函数调用,它调用一个名叫printf的函数,带有参数"Hello! \n"。printf是一个屏幕上输出的库函数。它打印输出作为其参数的字符串(由括号所括的字符序列)。

串中的序列"\n"是C语言中用作换行符的记号,当遇到它时,屏幕上的光标前进到下一行的最左边。因此,如漏掉了"\n",会发现本语句输出结束时并未换行,若多次调用后,得到一个长的输出行。

例如：

```
printf("Hello,");
printf("This is a test.");
printf("\n");
```

其效果等于 printf("Hello, This is a test. \n");

它们都是在屏幕上输出：Hello, This is a test.

【例 1.2】求两数中的大值。

程序：

```
main()
{ int a,b,c;                    /*定义变量*/
  scanf("%d,%d",&a,&b);         /*输入*/
  c=max(a,b);                   /*调用函数*/
  printf("max=%d\n",c);         /*输出*/
}
int max(int x,int y)            /*定义函数*/
{ int z;                        /*定义调位变量*/
  if (x>y) z=x;
  else     z=y;
  return(z);                    /*返回大值*/
}
```

本程序包括两个函数：主函数 main 和子函数 max。

max 函数的作用是将 x 和 y 中较大者的值则赋给变量 z，并通过 return 语句将 z 的值返回。变量 x，y 是形式参数，在调用 max 函数时，由实参 a，b 分别对应传送给 x，y。变量 z 是局部变量，仅在 max 函数内有效，z 的值通过函数名 max 返回给主调函数，并赋给变量 c。

main 是主函数，在主函数中，将 a，b，c 定义为整型变量。a，b，c 也是局部变量，仅在主函数内有效。Scanf 是标准的输入函数，Printf 是标准的输出函数。

在程序中各行后面的"/*…*/"是注释部分，它可以出现在任何位置，但必须配对使用。注释的内容可以是英文或中文，主要是增加程序可读性。

通过上述两个 C 程序的例子，可以看出：

(1) C 程序是由函数构成的。每个 C 程序有且仅有一个主函数，该主函数的函数名规定写 main，也可以包含一个 main 函数和若干个其他函数。

(2)每个函数(包括主函数)定义分为两部分：函数的首部和函数体。

函数的首部形式：

```
类型  函数名(形式参数 1,形式参数 2,……)
形式参数的说明
```

函数体的形式：

```
{ 变量定义部分
  函数执行部分
}
```

(3)C 程序的书定格式自由，一行内可以写几个语句，一个语句可以分写在多行上，每个

语句最后总是以"}",";"作为语句的结束。

(4)C程序的执行总是从主函数开始,并在主函数中结束。主函数的位置是任意的,可以在程序的开头,可以在程序的结尾,也可以在两个函数之间。其他函数总是通过函数调用语句来执行的。

(5)主函数可以调用任何非主函数,任何非主函数都可以相互调用,但是不能调用主函数。

(6)C语言本身没有输入输出语句。输入和输出操作是由调用系统提供的标准输入输出函数(如scanf/printf)来完成的。

(7)可以用/*……*/对C程序中的任何部分作注释,以增加程序的可读性。

1.5 算 法

1.5.1 算法概念

要计算机帮助我们解决问题,首先要编写出计算机程序。众所周知,计算机程序是许多指令的集合,每一条指令让计算机执行完成一个具体的操作,一个程序所规定的动作全部执行完成后,就能产生计算结果。因此,编写出正确的程序就成为让计算机帮助人们解决问题的关键。编制正确的程序必须具备两个基本条件:一是要掌握一门计算机高级语言的语法规则;二是要掌握解题的方法和步骤。

计算机语言只是一种工具。读者只学习语言的语法规则是不够的,最重要的是学会针对各种类型的问题,拟定出有效的解题方法和步骤即算法。算法的研究是计算机科学的核心,而算法概念本身也是计算机程序设计中最基本的概念之一。

正确的算法必须满足下列3个条件:

①每个逻辑块(模块)必须由可以实现的语句来完成;

②每个模块间的关系应该是惟一的(即块与块之间的关系一定要确定);

③算法要能终止(不能造成死循环)。

下列过程就不是一个正确的算法。计算过程:

第1步:令n等于0;

第2步:n加1;

第3步:转向第2步。

如果你用计算机执行此过程,从理论上讲,计算机将永远执行下去(死循环)。

而下列过程就是一个正确的算法,它能结束:

第1步:设n等于0;

第2步:n加1;

第3步:如果n小于100,则转向第2步,否则停止。

这里不要把"计算方法"(Computational method)和"算法"(Algorithm)这两个词混淆。前者指的是求数值解的近似方法,后者是指解决问题的一步一步过程。

1.5.2 算法简例

【例 1.3】求 $1\times2\times3\times\cdots\times20$。

此例如果用原始的方法：$1\times2\Rightarrow2$，$2\times3\Rightarrow6$，$6\times4\Rightarrow24$，…，一步一步去写，算法虽然是正确的，但太繁琐，它要写 19 个步骤才能完成，因而这种方法不可取。在实际使用中，一般对这种例题，采用如下一种通用的算法。

设置两个变量：一个变量代表被乘数，一个变量代表乘数。然后再把每次乘积的结果放在被乘数变量中，并用一种循环方式来表示算法。设 t 为被乘数，i 为乘数。算法：

①1⇒t

②2⇒i

③t×i⇒t

④i+1⇒i

⑤若 i≤20，返回③；否则，结束。

在这个算法中③到⑤组成一个循环，在实现算法中反复多次执行③，④，⑤步骤，直到 i 值大于 20 为止，算法结束，最后一次变量 t 的值就是所求的结果。

如果题目改为 $1\times3\times5\times7\times9\times11\times13\times15\times17\times19\times21$。

算法只需稍作改动即可。算法：

①1⇒t

②3⇒i

③t×i⇒t

④i+2⇒i

⑤若 i≤21，返回③；否则，结束。

可以看出这种方法表示的算法具有通用性、灵活性。

【例 1.4】求两个正整数 m 和 n 的最大公因子。算法：

①输入 m，n 的值；

②用求余数%，求 m 除以 n 的余数，然后把余数赋给变量 r，即 m%n⇒r；

③判断 r 是否为 0；

④如果 r≠0，则把 n⇒m，r⇒n，返回②；

⑤若 r=0，则输出最大公因子，结束。

此种算法表示的是求两个整数 m 和 n 最大因子的欧几里德算法。

【例 1.5】某班有 32 个学生，试将其中成绩最高的打印出来。算法：

①g_1（表示第一个学生成绩）⇒m；

②2⇒i；

③ 当 m＞g_i（表示第 i 个学生成绩）时，则执行④。
　否则，g_i⇒m；

④i+1⇒i；

⑤ 当 i≤32 时，返回③执行。
　否则，计算结束。

这种算法称之选极值递推结构，它的作用是将一批数中最大或最小的值挑选出来。上

述算法是选最大值，若将③改为：

当 $m<g_i$ 时，则执行④。
否则，$g_i \Rightarrow m$。

则上述算法即为选最小值。

1.5.3 算法的特征

从以上所例举的试题中，不难看出一般计算机的算法应具有以下5个特征。

1. 有穷性

一个算法必须在有限次执行后完成。

上面的例1.2表明，当执行到第三轮 r=0 时，求得28和16的最大公因子为4时算法结束。显然它是满足了有穷性的条件。这是因为在第①步输入正整数 m 和 n 以后，接着按第②步执行，在第②步后，r 的值一定小于 n。当 r≠0 时，下一轮再执行第②步时，n 的值将减少。而正整数的递减序列必然导致最后的终止。当然，选择较大的 m 和 n 值可能增加第一步骤的执行次数。

2. 确定性

一个算法中的每一个步骤必须有明确的定义。计算机和自然语言不大相同，一切都要在程序中予以安排，不能有语意不明的地方。

这就是说，要执行的动作应该是非常明确的。它没有语义上解释的困难。诸如执行“x加6或者加7赋给变量y”这类的步骤是含糊不清的。因为它没有明确x应该加6或7中的哪一个数，在算法中类似这类的含糊不清的步骤是不允许的。在上面的实例中，第①步到第⑤步所表明的各个步骤其含义都是十分明确的。

3. 输入

算法总是要施加到运算对象上，是提供运算对象的初始情况，是算法的起点，所以一个算法应有0个或多个输入。

在例1.2中应有两个输入数据，它们是正整数 m 和 n。

4. 输出

一个算法要有一个或多个输出。在例1.2中，它有一个输出，就是最后一轮在第③步中判别 r=0 后输出的结果 n，若无输出，人们是无法知道答案的。

5. 可行性

一个算法的可行性是指所有待实现的运算必须是相当基本的，至少在原则上人们可以用纸和笔做有限次操作即可完成的。

在例1.2中，第①步、第⑤步完成两个输入的整数和输出这两个整数的最大公因子的操作。第②步到第④步所示的一些步骤中用到了两个整数的除法运算，判别一个整数是否为零以及一个变量赋值给另一个变量的运算。这些运算都是一些基本的可行的运算。

算法实质上反映的是解决问题的思路。许多问题，只要仔细分析对象数据，就容易找到处理方法，对于定义明确的任务往往是这样。

1.5.4 怎样表示一种算法

为了表示一个算法。可以用不同的方法。常用的有：自然语言；伪代码(Pseudo code)；

N—S结构化流程图;PAD图(Problem Analysis Diagram意为问题分析图);程序流程图。用计算机语言(包括低级语言和高级语言)写的程序也是算法的表示形式。下面介绍程序设计中常用的3种算法。

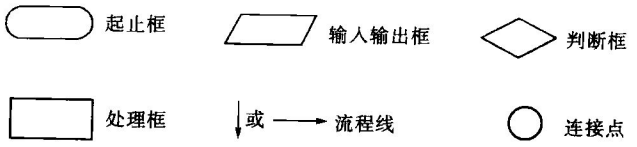


图1-1

1.用流程图表示算法

流程图是用一些图框来表示各种操作。用图形表示算法,直观形象,易于理解。美国国家标准化协会ANSI(American National Standard Institute)规定了一些常用的流程图符号,如图1-1所示,已为世界各国程序工作者普遍采用。

图1-1中菱形框的作用是对一个给定的条件进行判断,根据给定的条件是否成立来决定如何执行其后的操作。它有一个入口,两个出口。如图1-2所示。

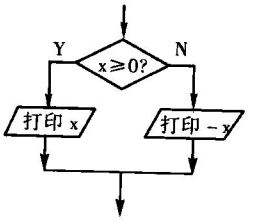


图1-2

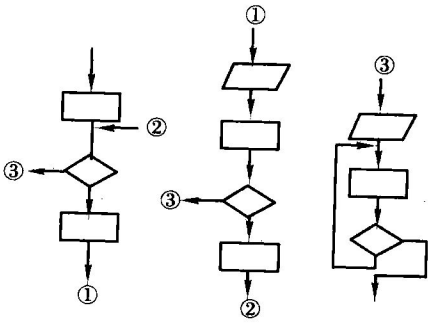


图1-3

连接点(小圆圈)是用于将画在不同地方的流程线连接起来。如图1-3所示中有两个以○为标志的连接点(在连接点圈中写上"1"),它表示这两个点是互相连接在一起的。实际上它们是同一个点,只是画不下才分开来画。用连接点,可以避免流程线的交叉或过长,使流程图清晰。

【例1.6】将例1.3中求20!的算法用流程图表示,流程图如图1-4所示。

菱形框两侧的"Y"和"N"代表"是"(yes)和"否"(no)。如果需要将最后结果打印出来,可以在菱形框的下面再加一个输出框,如图1-5所示。

2.用伪代码表示算法

流程图可表示一个算法,但在设计算法过程中使用不是很理想(尤其是当算法比较复杂、需要反复修改时)。为了设计算法时方便,常用一种称为伪代码(Pseudo code)的工具。

伪代码是用于自然语言和计算机语言之间的文字和符号来描述算法。它如同一篇文章,自上而下地写下来。每一行(或几行)表示一个基本操作。它不用图形符号,因此书写方便、格式紧凑,也比较好懂,便于向计算机语言算法(即程序)过渡。

例如,"打印x的绝对值"的算法可以用伪代码表示如下: