“十一五”重点规划教材

高等学校计算机及其应用系列

C语言程序设计

主　编/宁　慧　崔玉文　魏传宝

哈尔滨工程大学出版社
Harbin Engineering University Press

“十一五”重点规划教材

高等学校计算机及其应用系列

C语言程序设计

主　编/宁　慧　崔玉文　魏传宝

副主编/高　伟　郭江鸿　丛晓红

哈尔滨工程大学出版社

Harbin Engineering University Press

内容简介

本书根据教育部高等学校非计算机专业计算机基础课程教学指导分委员会的《关于进一步加强高校计算机基础教学的意见》的指导思想编写而成。全书共分12章，对C语言的基本运算、基本语法、基本程序设计方法进行了详细介绍，并配有大量的例题与练习题。本书提倡“轻语法，重实践”，即用少量篇幅介绍语法，重点讲解如何编程，有助于提高学生的自主学习能力，提高学生的创新性思维，为学生将来开发应用程序奠定基础。

本书语言流畅、逻辑性强，易学易懂，可以作为高等学校各专业的计算机基础课程的教材，也可供计算机等级考试者及其他学习者使用和参考。

图书在版编目(CIP)数据

C语言程序设计/宁慧，崔玉文，魏传宝主编. —哈尔滨：哈尔滨工程大学出版社，2009.2
ISBN 978-7-81133-369-5

Ⅰ.C… Ⅱ.①宁…②崔…③魏… 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(2009)第019697号

出版发行 哈尔滨工程大学出版社
社　　址 哈尔滨市南岗区东大直街124号
邮政编码 150001
发行电话 0451-82519328
传　　真 0451-82519699
经　　销 新华书店
印　　刷 哈尔滨工业大学印刷厂
开　　本 787mm×1 092mm 1/16
印　　张 21.25
字　　数 516千字
版　　次 2009年2月第1版
印　　次 2009年2月第1次印刷
定　　价 28.00元

http://press.hrbeu.edu.cn
E-mail:heupress@hrbeu.edu.cn

前　言

C语言功能丰富、使用灵活、可移植性好，既具有高级语言的优点，又具有低级语言的许多特点，不仅可以用来编写系统软件，还可以用于编写应用软件。因此，它的应用范围非常广泛。

C语言是我国高校普遍开设的一门重要的计算机基础课程，同时也是计算机专业学生学习程序设计语言的必修课程，是教育部提出的高等学校非计算机专业计算机课程“1 + X”模式中的重要组成部分。

本书是根据教育部高等学校非计算机专业计算机基础课程教学指导分委员会的《关于进一步加强高校计算机基础教学的意见》的指导思想编写的。全书共分12章，全面介绍了C语言的基本语法及C语言程序的设计方法。本书对C语言的基本运算、基本语法、基本程序设计方法进行了详细介绍，并配有大量的例题与练习题，使学生加深和巩固所学知识，提高学生的编程能力。

本书在编写过程中力求体系结构合理、重点突出、难度适中，提倡“轻语法，重实践”，即用少量篇幅介绍语法，重点讲解如何编程，有助于提高学生的自主学习能力，提高学生的创新性思维。为学生将来开发应用程序奠定基础。

全书共分为12章，由多年从事计算机基础教学的教师编写。第1,3,7章由宁慧编写，第2章由丛晓红编写，第4,5章由崔玉文编写，第6,8,10章由高伟编写，第9章及附录由魏传宝编写，第11,12章由郭江鸿编写，全书由宁慧统稿。

本书在编写过程中得到了哈尔滨工程大学计算机学院其他教师的帮助，他们对本书提出了许多宝贵意见，在此向他们表示衷心感谢。

由于编写水平有限，加之时间仓促，书中难免有错误和不妥之处，恳请广大读者批评指正！

编者

2009年1月

目　录

第1章 C语言概述

1.1 C语言的发展及特点

1.1.1 C语言的发展

C语言是目前国际上流行的一种结构化的程序设计语言。它既可以用来编写系统软件,也可以用来编写应用软件,因此深受广大程序设计者的欢迎。

以前的操作系统主要是用汇编语言编写的,但是由于汇编语言依赖于特定的计算机硬件,程序的可读性与可移植性都很差,于是人们希望用一种高级语言来编写系统软件,但是一般的高级语言难以实现汇编语言的某些功能,C语言正是在这种背景下被设计出来的。

C语言是随着UNIX操作系统的开发而诞生的,是在B语言的基础上发展起来的,在此之前人们设计了几种高级语言,但是离硬件都比较远,无法用来编写系统软件。1970年,美国贝尔实验室的Ken Thompson在其他语言的基础上,设计出了很简单而且很接近硬件的B语言,并且用B语言编写了第一个UNIX操作系统。1972年至1973年间,贝尔实验室的D.M.Ritchie在B语言的基础上设计出了C语言,并与Ken Thompson合作用C语言改写了UNIX操作系统。随着UNIX的广泛应用,C语言也迅速得到推广。

1.1.2 C语言的特点

C语言能被世界计算机界广泛接受是由于其自身的特点,C语言主要有以下几个特点。

(1)语言简洁、紧凑,书写形式自由。学习时入门相对容易,知道很少东西就可以开始编写程序。C语言关键字简练,源程序短,输入的工作量较少。

(2)C语言运算符丰富。共有34种运算符,使源程序书写精练,生成的代码质量高,运行速度快。

(3)数据类型丰富。具有现代语言的各种数据结构,能实现各种复杂的运算,尤其是指针类型数据,可使程序更加灵活,程序效率更高。

(4)语法限制不是很严格,程序设计自由度大。例如,C语言对数组下标越界不作检查,由程序编写者自己保证程序的正确性。因此,程序编写者应当仔细检查程序,保证其正确,而不能过分依赖C语言编译程序去查错。另外,C语言对变量类型的使用比较灵活,例如整型数据与字符型数据以及逻辑型数据可以通用等。

(5)C语言是一种模块化的程序设计语言,采用自顶向下、逐步求精的结构化程序设计方法,各模块功能独立,以函数形式编制,通过函数之间的相互调用和数据传递,实现系统整体的功能要求。

(6)C语言可以直接访问物理地址,能进行位操作,能实现汇编语言的很多功能,可以直接对硬件进行操作。因此,C语言既具有高级语言编写简单方便、便于理解的优点,又具有低级语言与硬件结合紧密的优点。

(7)用C语言编写的程序可移植性好,一般不作修改或作少量的修改就可以应用于不同的计算机上,也可以在多种操作系统下使用。

(8)C语言具有很好的通用性,既可以用于编写应用软件,也适合编写系统软件。例如,UNIX操作系统的源代码就是用C语言编写的。

1.2 C语言程序设计简介

当我们需要利用计算机来解决一个实际问题时,必须依靠一定的工具(人机界面)编写一个指令(语句)序列,一旦指令(语句)序列提交给计算机,计算机就可以执行这些指令(语句),产生结果。程序就是计算机可以执行的指令序列或语句序列。而设计、编制、调试程序的过程就称为程序设计。编写程序所使用的语言即为程序设计语言,用程序设计语言所编写出的程序必须严格遵守它的语法规则,这样编写出的程序才能被计算机所接受、运行,才能产生预期结果。下面认识一下C程序。

1.2.1 简单的C程序介绍

用C语言编写的程序,称为C语言源程序,简称C程序。C程序一般由一个或多个函数组成,这些函数可以放在同一个文件中,也可以分别放在几个文件中。下面先介绍一个简单的例子,认识一下C程序。

例1.1 计算长方形面积的C程序,假设长、宽值均为整数。

```
# include < stdio.h >
void main()
{
  int a,b, area;                    /* 定义了 a,b,area 为整型变量 */
  a = 5;                            /* 把 5 赋给变量 a */
  b = 3;                            /* 把 3 赋给变量 b */
  area = a * b;                     /* 计算长方形面积 */
  printf("area = %d \n",area);      /* 在屏幕上显示面积的值 */
}
```

程序运行结果如图1-1所示。

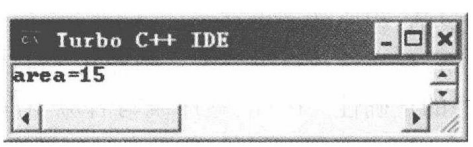


图1-1 例1.1的运行结果

(1)"/*"与"*/"之间的内容是注释。本例中用了大量的注释,主要是为了提高程序的可读性,便于理解。在实际应用中,一个程序只需在一些较难处加注释。注释不影响程序的执行,也不被编译。

(2)"#include"命令用来实现"文件包含"的操作。C语言的程序一般要有头文件,头文件在程序的开始用"#include"作出说明。对此读者不必深究,在后面的第8章中有详细的

介绍,在此只需记住。头文件可以是对程序中所用变量的说明,也可以是引用的库函数。在使用标准函数库中的输入输出函数时,编译系统要求程序提供有关的信息,程序中第 1 行的作用就是用来提供这些信息的,“stdio.h”是 C 编译系统提供的一个文件名,stdio 是“standard input & output”的缩写,即有关标准输入输出的信息。在程序中用到系统提供的标准函数库中的输入输出函数时,应在程序的开头写上如下所示的一行命令:

```
# include < stdio.h >
```

(3)main 是函数的名字,表示“主函数”,main 前面的 void 表示此函数是“空类型”,void 是“空”的意思,即执行此函数后不产生一个函数值。每一个 C 语言函数都必须有一个 main 函数。函数体由花括号{ }括起来。

(4)函数体中的每条语句都要以“;”号结束。

(5)“int a,b,area;”是声明部分,定义变量 a,b 和 area,指定 a,b,area 为整型变量。

(6)第 5,6 行是赋值语句,使 a,b 值分别为 5,3。

(7)第 7 行也是赋值语句,用来计算长方形的面积,并将计算结果赋给变量 area。

(8)第 8 行中的“%d”是格式说明,“%d”表示“以十进制整数类型”,用来指定输入输出时的数据类型和格式(详见第 3 章)。

(9)printf 是编译系统提供的标准函数库中的输出函数(详见第 3 章),在执行输出时,双撇号括起来的“area = ”按原样输出,在“%d”的位置上代以一个十进制整数值,printf 函数中括号内逗号右端的 area 是要输出的变量,它的值为 15,应出现在“%d”的位置上,“\n”是换行符,实现换行。因此程序运行时输出以下信息:

```
area = 15
```

例 1.2 在屏幕上输出“Beijing welcomes you.”。

```
# include < stdio.h >
void main()
{
  printf("Beijing welcomes you.\n");
}
```

程序运行结果如图 1-2 所示。

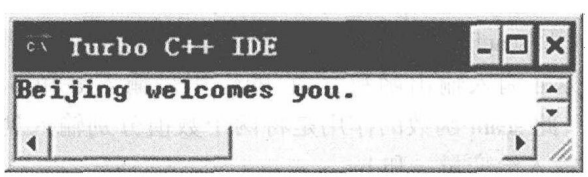


图 1-2 例 1.2 的运行结果

本程序的功能是输出一行信息。

例 1.3 输入两个整数,求两者中较大的数,并输出。

```
# include < stdio.h >
void main()                                  /* 主函数 */
{                                            /*  main()函数体开始 */
  int max(int x,int y);                      /* 对被调用函数 max 的声明 */
  int a,b,c;                                 /* 声明部分,定义变量 a,b,c */
```

```
    printf("input a,b:");
    scanf("%d,%d",&a,&b);                    /* 输入变量 a 和 b 的值 */
    c = max(a,b);                            /* 调用 max 函数,将调用结果赋给 c */
    printf("max = %d \n",c);
}                                            /*  main()函数体结束 */

int max(int x,int y)                         /* 求两个数中较大数的函数 */
{                                            /* max 函数体开始 */
    int z;                                   /* 声明部分,定义变量 z */
    if (x > y) z = x;
    else z = y;
    return z;                                /* 将 z 值返回,通过 max 带回调用处 */
}                                            /* max 函数体结束 */
```

程序运行结果如图 1 - 3 所示。

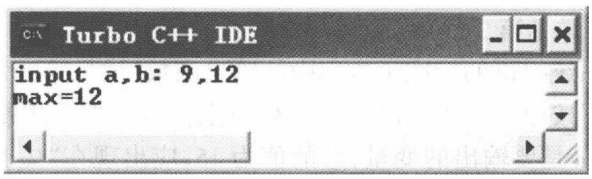


图 1 - 3　例 1.3 的运行结果

(1)程序包括两个函数,其中主函数 main 是整个程序执行的起点和终点,函数 max 的作用是比较得到两个数中较大的数,并将该数值赋给变量 z。

(2)return 语句将 z 值返回给主调函数 main。返回值是通过函数名 max 带回到 main 函数中的调用 max 函数的位置。

(3)程序第 4 行是对被调用函数 max 的声明,由于在 main 函数中要调用 max 函数,而 max 函数的位置在 main 函数之后,为了使编译系统能够正确识别和调用 max 函数,必须在调用 max 函数之前对 max 函数进行声明。在此读者只需初步了解,在第 7 章中有详细介绍。

(4)主函数 main 调用 scanf 函数获得两个整数,存入 a,b 两个变量。scanf 是输入函数(scanf 和 printf 是 C 的标准输入输出函数)。它的作用是输入 a 和 b 的值。&a 和 &b 中的"&"的含义是"取地址",此 scanf 函数的作用是将两个数值分别输入到变量 a 和 b 的地址所标志的单元中,也就是输入给变量 a 和 b。

(5)主函数 main 调用函数 max 获得两个数值中较大的值,并赋给变量 c,最后输出变量 c 的值。

(6)int max(int x,int y)是函数 max 的函数头,表明此函数从调用函数获得两个整数,并返回一个整数到调用函数。

(7)函数 max 也要用{ }将函数体括起来。从参数表获得数据,在函数体中进行比较处理,得到结果 z,然后将 z 返回调用函数 main。

1.2.2　C 程序结构

通过以上几个例子,我们对 C 语言程序结构有了一定的了解。

(1)C语言程序由函数构成。

①一个C源程序必须包含一个main函数,也可以包含一个main函数和若干个其他函数。函数是C程序的基本单位。

②被调用的函数可以是系统提供的库函数,也可以是用户根据需要自己编写的函数。

(2)main函数是每个程序执行的起始点。一个C程序总是从main函数开始执行,而不论main函数在程序中的位置如何。

(3)一个函数由函数首部和函数体两部分组成。

①函数首部,一个函数的第一行。格式如下:

函数类型 函数名([参数类型1 参数名1],…,[参数类型n 参数名n])

[]中的内容是可选项,函数可以没有参数,但是后面的()不能省略。

②函数体,函数首部下面{ }内的部分。如果一个函数有多个{ },则最外层的一对{ }为函数体的范围。函数体一般包括两部分。

```
{
  [声明部分]:在这部分中定义所用到的变量和对所调用函数作声明。
  [执行部分]: 由若干语句组成。
}
```

在某些情况下也可以没有声明部分,甚至可以既无声明部分也无执行部分。例如:

```
void dump()
{ }
```

这是一个空函数,什么也不执行,但它也是一个合法的函数。

(4)C程序书写格式自由,一行内可以写一个语句,也可以写几个语句,一个语句也可以分写在多行上。

(5)每个语句和数据声明的最后必须有一个分号,分号是C语句的组成部分。

(6)C语言本身没有输入输出语句,输入输出操作是通过调用库函数来完成的。

(7)C语言区分大小写,因此在使用变量时要注意。

(8)C语言的程序总是从主函数开始执行,并且终止于主函数。

1.3 C语言程序的组成成分

C语言程序是由语句组成的,而语句是由词汇构成的,每个词汇由字符构成。

1.C语言的字符集

字符是组成语言的最基本元素。C语言字符集由字母、数字、空格、标点和特殊字符组成。在字符常量、字符串常量和注释中还可以使用汉字或其他可显示的图形符号。

2.C语言词汇

在C语言中使用的词汇分为六类:标识符、关键字、运算符、分隔符、常量、注释符等。

【标识符】 在程序中使用的变量名、函数名、标号等统称为标识符。简单地说,标识符就是一个名字。除库函数的函数名由系统定义外,其余都由用户自定义。C语言规定,标识符只能是由字母(A~Z,a~z)、数字(0~9)、下划线组成的字符串,并且第一个字符必须是字母或下划线。例如,下面列出的标识符是合法的:

```
sum,average,_total,x,y,z5,b_c,ab
```

下面的标识符是不合法的：

3x(以数字开头)，a * b(出现非法字符 *)，a > b(出现非法字符 >)，a - b(出现非法字符 -)

在使用标识符时必须注意以下几点：

①编译系统将大写字母和小写字母区分开，认为是两个不同的字符。因此，sum 和 SUM 是两个不同的变量名。一般变量名用小写字母表示，以增加程序的可读性。

②在选择变量名时，应该做到“见名知义”，如 sum，average，length，count，area 等，以增加程序的可读性。本书在一些简单的例子中，为方便起见，用了单个字符的变量名(例如 a，b，c 等)，注意不要在所有程序中都这样定义变量名。

③标准 C 没有规定标识符的长度，但各种 C 编译系统都有自己的限制。例如有的系统规定标识符前 8 位有效，当两个标识符前 8 位相同时，则被认为是同一个标识符。而 Turbo C 允许标识符有 32 个字符。因此，在编写程序时应了解所用系统对标识符长度的规定，以免造成错误。一般情况下，标识符长度不要超过 8 个字符。

【关键字】 关键字是 C 语言规定的具有特定意义的字符串，通常也称为保留字。用户定义的标识符不应与关键字相同。C 语言的关键字分为以下几类：

①类型说明符。用于定义、说明变量、函数或其他数据结构的类型。例如前面例题中用到的 int，此外还有 long，float，double，short，auto，char，static，extern，register，struct 等。

②语句定义符。用于表示一个语句的功能。例如 if…else 就是条件语句的语句定义符，此外还有循环语句的语句定义符 while，for，goto 等。

③预处理命令字。用于表示一个预处理命令。如前面各例中用到的 include，此外还有 define，ifdef 等。使用时前面要加“ # ”。

【运算符】 C 语言中含有相当丰富的运算符。例如算术运算符、关系运算符、逻辑运算符、位运算符、赋值运算符、条件运算符等 13 类运算符。运算符、变量与函数一起组成表达式，表示各种运算功能。运算符由一个或多个字符组成。

【分隔符】 在 C 语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中，用以分隔各个变量。空格多用于语句各单词之间，作间隔符。在关键字、标识符之间必须要有一个以上的空格符作间隔，否则将会出现语法错误，例如把“int a;”写成“inta;”，C 编译系统会把“inta”当成一个标识符处理，其结果必然出错。

【常量】 C 语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种。在第 2 章中将专门给予介绍。

【注释符】 C 语言的注释符是以“/ * ”开头并以“ * /”结尾的串。在“/ * ”和“ * /”之间的即为注释。程序编译时不对注释作任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来，使编译跳过不作处理，待调试结束后再去掉注释符。

3.程序语句

C 程序是由若干条语句组成的，语句是程序的基本书写单位和执行单位。C 语句可分为表达式语句、函数调用语句、声明语句、空语句、复合语句及流程控制语句六种。

【表达式语句】 在一个表达式后加分号，就构成了表达式语句。执行一个表达式语句就是对表达式求值。最典型的是由赋值表达式加分号构成一个赋值语句。表达式能构成语句是 C 语言的一个重要特色。例如：

```
a = 3;
```

【函数调用语句】 由一个函数调用加一个分号构成,例如:

```
printf ("Beijing welcomes you.\n");
```

【声明语句】 在C语言中,一个名字(标识符)在使用之前必须先声明,按声明的对象可分为变量声明和函数声明。

【空语句】 只有一个分号的语句叫做空语句。有时用来作流程的转向点。

【复合语句】 在C语言中可以用一对花括号把一些语句括起来构成复合语句。在语法上它相当于一条语句。注意,在花括号外不再写分号。例如:

```
{
  z = x + y;
  t = z / 100;
  printf("%f", t);
}
```

【流程控制语句】 一般来说,C程序按主函数中语句的书写顺序执行。流程控制语句可以改变这种执行顺序,使程序中语句的执行顺序与书写顺序不一致。流程控制语句主要有以下两种形式:

①流程在两个函数之间转移,即函数调用或返回等流程转向语句。

②呈现某种控制结构,主要是选择结构和循环结构。

C语言有9种控制语句,具体如下:

if ( ) … else …	(条件语句)
for ( ) …	(循环语句)
while ( ) …	(循环语句)
do … while ( )	(循环语句)
continue	(结束本次循环语句)
break	(中止执行 switch 或循环语句)
switch	(多分支选择语句)
goto	(转向语句)
return	(从函数返回语句)

上面9种语句中的括号( )表示其中是一个条件,…表示内嵌的语句。

1.4 算法及表示

初学程序设计时,人们往往会有一种感觉,就是读别人编写的程序比较容易,而自己编写程序就难了,总是不知从何下手,这是为什么呢? 其中一个重要的原因就是没有掌握解决问题的算法。事实上,在生活中每做一件事情,都要遵循一定的步骤。例如,你去外地上大学,你新到一座城市,虽然有公共汽车,但是你不知道按照怎样的路线走才能找到你要去的学校,当别人告诉你一条路线,如先乘什么车,在什么站下车,再换乘什么车,等等,这就像告诉了你一个解决问题的算法,于是你就可以沿着这条路线到达目的地了。下次再来时,你就不会感到为难了。所以一定要重视算法的设计,多了解、掌握和积累一些计算机常用算法,养成编写程序前先设计好算法的习惯。

1.4.1 算法

一个程序应包括对数据的描述和对数据处理的描述。

1.对数据的描述。在程序中要指定数据的类型和数据的组织形式,即数据结构。

2.对数据处理的描述,即操作步骤,也就是算法。算法是为了解决一个问题而采取的方法和步骤,是程序的灵魂。为此,著名计算机科学家沃思(Nikiklaus Wirth)提出一个公式:

数据结构 + 算法 = 程序

正如前文所述,知道乘车路线是找到目的地的关键,编写一个程序的关键就是合理地组织数据和设计算法。显然去一个地方可能会有多条路线,同样地,解决一个问题也会有多种算法。例如,求 1 + 2 + 3 + … + 100,有的人可能先求 1 + 2,再加 3,一直加到 100;而有的人可能采用 100 + (1 + 99) + … + (49 + 51) + 50 = 5050;还有的人用其他的方法。又比如,排序算法也有很多,有冒泡法、交换法、选择法,等等。因此,为了有效地解题,不仅需要保证算法正确,还要考虑算法的质量,选择合适的算法。

设计好了算法,就可以将它用具体的语言进行描述,最终转化为解决问题的计算机程序。程序设计语言好比是交通工具,例如汽车,它仅仅是实现算法(到达目的地)的工具。到达目的地,可以利用各种交通工具,同理对于同一个算法,可以利用各种程序设计语言来实现,比如用 C 语言、FORTRAN 语言、汇编语言,等等。用不同的算法以及不同的程序设计语言解决同一个问题,只是速度和时间效率上的不同而已。因此,要想开发出高质量、高效率的程序,除了要熟练掌握程序设计语言这种工具以外,更重要的是要多了解、多积累并逐步学会自己设计一些好的算法。那么,究竟什么是算法呢?

所谓算法,就是为解决一个具体问题而采取的确定的有限的操作步骤。当然,这里所说的算法仅指计算机算法,即计算机能够执行的算法。

1.4.2 算法的特性

算法具备以下 5 个特性。

1.有穷性

一个算法应包含有限的操作步骤,而不能是无限的。并且在可以接受的时间内完成其执行过程。

2.确定性

算法的每一步操作都必须有确切的含义,不能有二义性。

3.输入

所谓输入就是算法在执行时需要从外界获得必要的信息。一个算法可以没有输入,也可以有多个输入。

4.输出

没有输出的算法是无意义的。一个算法可以有一个或多个输出。

5.可行性

算法中的每一步骤都应该能有效地执行,并得到确定的结果。例如把零作除数就是无效的步骤。

1.4.3 算法的表示

算法可以使用各种不同的方法描述。常用的表示算法的方法有自然语言、传统流程图、N - S 结构化流程图、伪码等。

1.用自然语言表示算法

所谓自然语言,就是人们日常使用的语言,可以是汉语、英语或其他语言。用自然语言描述的算法通俗易懂,但文字冗长,容易出现二义性。

例 1.4 求 5!,用自然语言表示算法。

Step1:计算 1×2,得到结果 2。

Step2:将第 1 步的计算结果 2 乘以 3,得到结果 6。

Step3:将第 2 步的计算结果 6 乘以 4,得到结果 24。

Step4:将第 3 步的计算结果 24 乘以 5,得到结果 120。

这样的算法虽然正确,但太繁琐,在乘数较少时尚可,若要计算 100!,则要写出 99 个步骤,显然不可取。可以用一种简便算法来描述。

设两个变量,一个变量代表被乘数,一个变量代表乘数,并且将每一步骤的乘积放在被乘数变量中,可以将算法改写如下。

Step1:1⇒p　　(将 1 存入变量 p 中)

Step2:2⇒q　　(将 2 存入变量 q 中)

Step3:p×q⇒p (p×q 的乘积仍放在变量 p 中)

Step4:q+1⇒q (使 q 的值加 1)

Step5:若 q≤5,返回执行第 3~5 步,否则,算法结束。最后得到 p 的值就是 5! 的结果。

如果题目改成求:2×4×6×8×10×12,算法只需作较少的改动。

Step1:2⇒p

Step2:4⇒q

Step3:p×q⇒p

Step4:q+2⇒q (使 q 的值加 2)

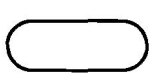
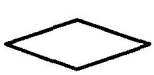
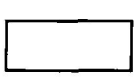
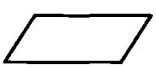
Step5:若 q≤12,返回执行第 3~5 步,否则,算法结束。最后得到 p 的值就是所求的结果。

上述算法不仅正确,而且是计算机能实现的较好的算法。若要计算 100!,改写上述算法,请读者自己完成。

2.用传统流程图表示算法

流程图是一种用图形来表示算法的方法。它用规定的图形符号、流程线和文字说明来描述各步骤的操作和执行过程。常用的流程图符号如表 1-1 所示。

表 1-1 流程图符号

符号	作用	符号	作用
	起止框		判断框
	处理框		流程线
	输入输出框		连接点

例 1.5 求 5!,用流程图表示算法(图 1－4)。

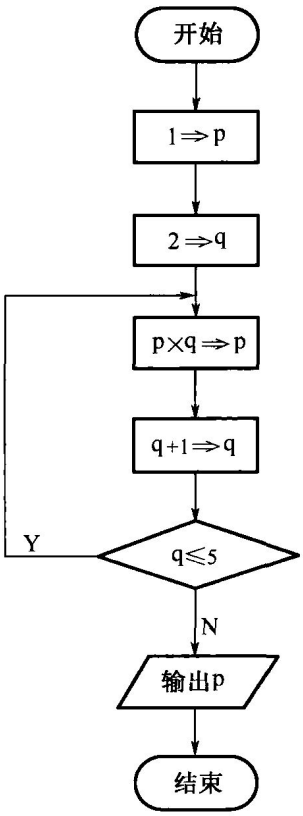


图 1－4

用传统流程图表示算法直观、形象,算法的逻辑流程一目了然,便于理解,但画法麻烦,由于允许使用流程线,用户可以随心所欲,使流程随意地转向,流程图变得毫无规律,从而造成阅读和修改上的困难。

为了克服上述的弊端,提出了结构化的程序设计方法,在结构化程序设计方法中,流程图只包括 3 种基本程序结构。

(1)顺序结构

顺序结构是最简单的一种基本结构,它是由一系列顺序的操作(语句)组成的,顺序结构的流程图如图 1－5 所示,表示执行 A 框所指定的操作后,再接着执行 B 框所指定的操作。

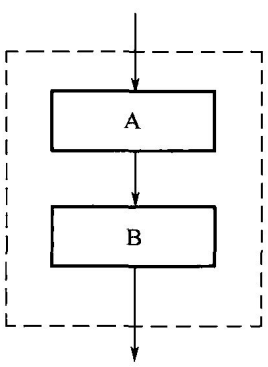


图 1－5 顺序结构

(2)选择结构

在选择结构中,根据给定的条件是否成立,选择执行 A 框或 B 框。无论条件是否成立,只能执行 A 框或 B 框之一,然后脱离本选择结构。A 框或 B 框可以有一个为空。选择结构的流程图如图 1－6 和图 1－7 所示。

(3)循环结构

循环结构是根据一定的条件,对某些语句重复执行的结