



21世纪高等教育规划教材

C 语言程序设计

普措才仁 主编



电子科技大学出版社

21 世纪高等教育规划教材

C 语言程序设计

主 编 普措才仁

副主编 曹永春 蔡正琦

顿毅杰 马 俊

电子科技大学出版社

图书在版编目(CIP)数据

C语言程序设计/普措才仁主编. —成都:电子科技大学出版社, 2009.5

21世纪高等教育规划教材

ISBN 978-7-81114-980-7

I. C… II. 普… III. C语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 072852 号

内 容 简 介

本书系统介绍了 C 语言的基本概念、语法规则和编程方法。内容主要包括 C 语言的数据类型、算法描述、表达式等 C 语言基本知识;结构化程序设计中的顺序、选择、循环三种基本结构;组成 C 语言程序的基本单元——函数;数组、结构体、共用体等自定义数据类型;C 语言中重要的概念——指针;最后介绍了文件操作。

为配合读者学习本书,各章均配有丰富的习题和精心设计的上机实验题目,并给出了参考答案,供读者复习和检查学习效果时使用。

本书可作为高等学校本科、高职高专计算机及相关专业程序设计的入门教材,也可作为全国计算机等级考试的辅导教材,还可供从事软件开发以及相关领域的工程技术人员参考使用。

21 世纪高等教育规划教材

C 语言程序设计

主 编 普措才仁

副主编 曹永春 蔡正琦 顿毅杰 马 俊

出 版:电子科技大学出版社(成都市一环路东一段 159 号电子信息产业大厦 邮编:610051)

策划编辑:罗 雅

责任编辑:张 鹏

主 页:www.uestcp.com.cn

电子邮箱:uestcp@uestcp.com.cn

发 行:新华书店经销

印 刷:北京广达印刷有限公司

成品尺寸:185 mm×260 mm 印张 25 字数 656 千字

版 次:2009 年 5 月第一版

印 次:2009 年 5 月第一次印刷

书 号:ISBN 978-7-81114-980-7

定 价:35.80 元

■ 版权所有 侵权必究 ■

◆ 本社发行部电话:028-83202463;本社邮购电话:028-83208003。

◆ 本书如有缺页、破损、装订错误,请寄回印刷厂调换。

◆ 课件下载在我社主页“下载专区”。

前 言

在众多程序设计语言中,C语言有其独特之处。作为一种高级程序设计语言,C语言具有方便性、灵活性和通用性等特点。同时,它还向程序员提供了直接操作计算机硬件的功能,具备低级语言的特点,适合各种类型的软件开发。而且C语言是计算机学科重要的核心课程,也是其他理工科专业计算机基础知识的必修课。尤其是近年来,学习和掌握C语言的需求越来越多,特别是对进一步提高C语言程序设计能力的需求也越来越多。为此,我们组织编写了这部《C语言程序设计》教材。本书内容系2008年教育部人文社会科学基金一般项目规划基金(编号 08JA740034)的子课题。为了加强计算机专业学生更好地学习交流,我们开发了与教材内容配套的《C语言程序设计》课程教学系统。

由于C语言涉及的概念比较复杂,规则繁多,使用灵活,容易出错,不少初学者感到学习困难和难以理解。作者根据多年教学经验和对初学者学习、理解编程语言规律的了解,在各章节内容编排上由浅入深,循序渐进,概念清晰,层次分明。既注重主干基础知识的透彻讲解,又注重难点、重点知识的分解与综合运用,不仅表达了C语言的基本概念、基本规则,而且从软件工程的角度出发,阐述了分析问题、解决问题的思路和方法。

全书共分10章内容,第1、2章介绍了C语言基础知识。第3、4、5章介绍了C语言进行结构化程序设计的基本方法,包括算法的概念和描述方法以及结构化程序设计的顺序结构、选择结构、循环结构。第6章介绍了数组的概念及应用。第7章介绍了函数与编译预处理。第8章则对指针作了充分阐述。第9章对结构体、共用体和链表等内容作了较为详细的介绍。第10章对C语言文件及位运算操作做了详细的阐述。

本书系作者根据多年从事C语言课程教学实践经验和应用C语言的体会,在多次使用以及逐步完善的讲义基础上,并广泛参考有关资料编写而成的。在结构上符合学习逻辑,内容充实,选材上注重系统性和实用性。全书精选了大量典型例题,帮助读者加深理解和掌握所学内容。书中程序例题均经过Visual C++ 6.0或Turbo C 2.0运行环境的调试,可直接引用。为配合本书学习,各章均配有丰富的习题和精心设计的上机实验题目,并给出了参考答案,供读者复习和检查学习效果时使用。

本书由普措才仁主编,曹永春、蔡正琦、顿毅杰、马俊担任副主编。本书编者均系多年从事教学、研究的一线教师,他们有着丰富的教学实践经验,所编内容力求做到结构严谨,概念准确,教材内容组织合理,语言使用规范,符合教学规律。同时在

编写时兼顾了全国计算机等级考试的要求。

经审定,本书可作为高等学校本科、高职高专层次计算机及相关专业“程序设计”课程入门教材,也可作为全国计算机等级考试辅导教材,还可供从事软件开发以及相关领域的工程技术人员参考使用。

教材建设是一项长期艰巨的系统工程,尤其是计算机科学技术发展迅速,内容更新快。因此,为了使教材更新能跟上时代的发展,我们将密切关注计算机科学技术最新进展及时修订,以适应高校计算机课程教学的需要。

本书在编写过程中,得到诸多专家和领导的热情支持与指导,在此一并表示衷心的感谢。

由于编者水平有限,加之时间仓促,书中错误和不当之处在所难免,恳请同行和广大读者不吝批评指正。

编 者

2009 年 3 月

目 录

第 1 章 C 语言概述	(1)
1.1 计算机与程序设计语言	(1)
1.2 C 语言的发展和特点	(2)
1.3 C 程序结构	(3)
1.4 C 程序的编译过程	(5)
习题 1	(6)
第 2 章 数据类型及其运算	(8)
2.1 C 语言的数据类型	(8)
2.2 常量与变量	(9)
2.3 整型数据	(12)
2.4 实型数据	(17)
2.5 字符型数据	(19)
2.6 各类数值型数据之间的混合运算	(23)
2.7 算术运算符和算术表达式	(25)
2.8 赋值运算符和赋值表达式	(28)
2.9 逗号运算符和逗号表达式	(32)
习题 2	(33)
第 3 章 顺序程序设计	(36)
3.1 C 语句概述	(36)
3.2 赋值语句	(37)
3.3 数据的输入输出	(38)
3.4 字符数据的输入输出	(39)
3.5 格式输入与输出	(40)
3.6 关于算法	(50)
3.7 结构化程序设计方法	(58)
3.8 顺序结构程序设计举例	(59)
习题 3	(61)
第 4 章 选择结构程序设计	(66)
4.1 关系运算符和表达式	(66)
4.2 逻辑运算符和表达式	(67)
4.3 if 语句	(70)
4.4 switch 语句	(78)
4.5 程序举例	(81)
习题 4	(83)

第 5 章 循环结构程序设计	(93)
5.1 概述	(93)
5.2 goto 语句以及用 goto 语句构成循环	(93)
5.3 while 语句	(94)
5.4 do-while 语句	(96)
5.5 for 语句	(97)
5.6 循环的嵌套	(101)
5.7 几种循环的比较	(102)
5.8 break 和 continue 语句	(102)
5.9 程序举例	(104)
习题 5	(107)
第 6 章 数组	(116)
6.1 一维数组	(116)
6.2 二维数组	(123)
6.3 字符数组	(127)
6.4 程序举例	(134)
习题 6	(137)
第 7 章 函数与编译预处理	(149)
7.1 函数的概念	(149)
7.2 函数的定义和调用	(150)
7.3 函数的返回值	(151)
7.4 函数的参数及其传递方式	(155)
7.5 函数的嵌套调用和递归调用	(158)
7.6 变量的作用域及其存储类型	(162)
7.7 内部函数和外部函数	(169)
7.8 应用举例	(170)
7.9 编译预处理	(173)
7.10 宏定义	(173)
7.11 文件包含	(179)
7.12 条件编译	(180)
习题 7	(182)
第 8 章 指针	(195)
8.1 地址和指针的概念	(195)
8.2 指针与数组	(201)
8.3 指针与函数	(213)
习题 8	(218)
第 9 章 结构体与共用体	(227)
9.1 结构体	(227)
9.2 结构体类型数组	(234)
9.3 共用体类型	(235)

9.4 枚举类型	(241)
9.5 用 typedef 定义类型	(243)
9.6 用指针处理链表	(244)
习题 9	(252)
第 10 章 文件与位运算	(259)
10.1 文件概述	(259)
10.2 文件的打开与关闭	(263)
10.3 文件的顺序读写	(265)
10.4 文件的随机读写	(272)
10.5 位运算	(275)
习题 10	(279)
习题参考答案	(286)
实验部分	(314)
编译器介绍	(314)
实验 1 熟悉 C 语言程序开发环境	(330)
实验 2 数据类型及其运算	(332)
实验 3 顺序结构程序设计	(334)
实验 4 选择结构程序设计	(336)
实验 5 循环结构程序设计	(339)
实验 6 数组	(342)
实验 7 函数与编译预处理	(346)
实验 8 指针	(349)
实验 9 结构体与共用体	(353)
实验 10 文件与位运算	(356)
实验答案	(361)
附录	(383)
附录 A 常用 ASCII 字符表	(383)
附录 B 运算符和结合性	(384)
附录 C C 库函数	(386)
参考文献	(392)

第 1 章 C 语言概述

1.1 计算机与程序设计语言

人类语言是人与人之间交流信息的工具,通过自然语言实现人与人之间的沟通,使别人能够明白地按照语言的表达来办事。而计算机语言是人与计算机之间交流信息的工具,通过计算机语言实现人与计算机之间的沟通,使计算机能够明白地按照语言的表达来工作。用计算机解决问题时,人们必须首先将解决该问题的方法和步骤按一定序列和规则用计算机语言描述出来,形成计算机程序,执行程序才能让计算机产生相应的操作。程序就是按照计算机语言的语法规则、语句格式,编制成的一段能够让计算机理解并按照执行的语句的集合。

计算机语言与程序经历了以下三个阶段的发展。

1.1.1 机器语言

机器语言就是计算机可以直接识别和执行的二进制语言。我们知道,在计算机内部采用二进制表示信息,指挥计算机完成一个基本操作的指令也是由二进制代码构成的,称之为机器指令。每一条机器指令的格式和含义都是由设计者规定的,并按照这个规定设计和制造硬件。例如,加法指令为 1 0 0 0 0 0 0 0,减法指令为 1 0 0 1 0 0 0 0。一个计算机系统的全部机器指令的总和称为指令系统,也就是机器语言。机器语言的特点是计算机可以直接识别和执行,效率高,节省内存;但难以阅读和记忆。

1.1.2 汇编语言

汇编语言是一种符号语言,它将难以记忆的二进制指令代码用有意义的英文单词缩写来代替,英文单词缩写被称为助记符,每一个助记符代表一条机器指令。例如,用 ADD 表示加操作,用 SUB 表示减操作。汇编语言提高了程序的可读性和可写性,使编程者在编写程序时稍微轻松了一点。但汇编语言程序不能在计算机上直接运行,必须把它翻译成相应的机器语言程序才能运行。将汇编语言程序翻译成机器语言程序的过程叫做汇编,汇编过程是计算机运行汇编程序时自动完成的。

1.1.3 高级语言

高级语言是相对低级语言而言的,低级语言是计算机能直接识别的语言(机器语言)或符号化的机器语言(汇编语言),高级语言是一种通用的、面向用户的各类需要、与特定的机器相分离,并遵循一定的严格规定与形式的语言,其语言格式接近于自然语言或接近于数学函数形式,例如,用“+”和“-”来表示加减运算。编写高级语言程序时,不需要考虑具体的计算机硬件系统,但是,计算机无法直接执行高级语言程序,必须将高级语言程序翻译成机器语言程序才能执行。高级语言不仅易学易用,通用性强,而且具有良好的可移植性。

高级语言种类繁多(据统计有上千种),曾经引起广泛关注和使用的高级语言有 FORTRAN、BASIC、Pascal 和 C 等命令式语言(或称过程式语言);有 LISP、PROLOG 等陈述式语言;还有当

前流行的面向对象的程序设计语言,例如 C++、Java、Visual C++、Visual Basic、Delphi、PowerBuilder 等。

C 语言是目前国际上比较流行的高级编程语言之一,自 20 世纪 70 年代产生以来,因其语法简介、使用方便且具备强大的功能而受到编程人员的普遍青睐,在各类大、中、小型和微型计算机中得到了广泛的使用,并很快成为当代最优秀的程序设计语言之一。

1.2 C 语言的发展和特点

1.2.1 C 语言的发展

C 语言是在 B 语言的基础上发展起来的,它的根源可以追溯到 ALGOL 60。1960 年出现的 ALGOL 60 是一种面向问题的高级语言,它离硬件比较远,不宜用来编写系统程序。1963 年,英国的剑桥大学推出了 CPL (Combined Programming Language) 语言。CPL 语言在 ALGOL 60 的基础上接近硬件一些,但规模比较大,难以实现。1967 年,英国剑桥大学的 Martin Richards 对 CPL 语言作了简化,推出了 BCPL (Basic Combined Programming Language) 语言。1970 年,美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,又作了进一步简化,设计出了简单的 B 语言。最初的 UNIX 操作系统是采用汇编语言编写的,B 语言版本的 UNIX 是第一个用高级语言编写的 UNIX。C 语言是美国贝尔实验室的 Dennis M. Ritchie 在 B 语言版的 UNIX 基础上开发出来的,1972 年在 DEC PDP-11 计算机上实现了最初的 C 语言。当时开发 C 语言的目的是为了编写新版 UNIX 操作系统,新版 UNIX 操作系统中 90% 的代码由 C 语言编写,10% 的代码由汇编语言编写。随着 UNIX 操作系统的广泛应用,C 语言也被人们认识和接受。

20 世纪 80 年代前后,C 语言在各种计算机上的快速推广产生了许多 C 语言版本。这些版本虽然是类似的,但通常互不兼容。显然人们需要一个与开发平台和机器无关的标准 C 语言版本。以 1978 年发表的 UNIX 第 7 版中的 C 编译程序为基础,Brian W. Kernighan 和 Dennis M. Ritchie (合称 K&R) 合著了影响深远的名著“The C Programming Language”,这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础。1983 年,美国国家标准化协会(ANSI)根据 C 语言问世以来各种版本对 C 的发展和扩充,着手制定 C 语言的规范,到 1989 年完成了 C 标准的制定,该标准称为 ANSI C。Brian W. Kernighan 和 Dennis M. Ritchie 编著的“The C Programming Language—Second Edition”(1988 年,第 2 版)介绍了 ANSI C 的全部内容,该书被称为 C 语言的圣经(C Bible)。

C 语言的发展方向是 C++ 语言。在 C 语言的基础之上,为了满足管理程序的复杂性以及代码重用的需要,1980 年贝尔实验室的 Bjarne Stroustrup 博士及其同事对 C 语言进行了改进和扩充,最初的成果称为“带类的 C”,而后称为“新 C”。1983 年,由 Rick Mascitti 提议正式命名为 C++ (C Plus Plus)。因为在 C 语言中,运算符“++”是对变量进行增值运算,那么 C++ 的喻义是对 C 语言进行“增值”。1994 年制定了 ANSI C++ 草案。此后又经过不断完善,成为目前的 C++。C++ 语言仍然在不断的不断发展中。

1.2.2 C 语言的特点

C 语言既适用于开发大型的复杂系统软件,也可以开发一般的软件,许多著名的系统软件,如 UNIX、dBase、FoxProC 等都是由 C 语言编写的。C 语言具有如下特点:

(1) 语言简洁、紧凑,使用方便、灵活。C 语言只有 32 个关键字,程序书写形式自由。

(2)具有丰富的运算符和数据类型,并引入指针的概念,可使程序效率更高;C语言也具有强大的计算功能和逻辑判断功能。

(3)C语言可以直接访问内存地址、进行位操作,完成类似于汇编语言的操作,使其能够胜任开发系统软件的工作。有时C语言也被称为“中级语言”,其含义是它将高级语言的优点与汇编语言的“实用性”结合在一起。

(4)生成的目标代码质量高,程序运行效率高。

(5)可移植性好,即程序可以很容易地改写后运行在不同的计算机上。

但是,C语言也具有如下局限性:

(1)C语言的数据类型检查机制较弱,这使得程序中的一些错误不能在编译时被发现。

(2)C语言本身几乎没有支持代码重用的语言结构,因此,一个程序员精心设计的程序很难为其他程序所用。

(3)当程序达到一定规模时,程序员很难控制程序的复杂性。

1.3 C 程序结构

下面举一个简单的例子来说明C程序的基本结构。

例 1.1 一个简单的C程序,这也是世界上第一个C程序。

```
/* 输出一个字符串常量于屏幕上 */
#include <stdio.h>
main()
{
    printf("Hello,world! \n");
}
```

程序第一行“/* */”是C语言的注释符,其作用是对程序或语句加以说明,在编译时编译系统忽略注释。“#include <stdio.h>”是头文件包含命令,因为在程序第4行在使用的printf()函数是标准库函数,C语言规定,在使用标准函数库中的函数时,编译系统要求程序提供有关的信息(例如对这些输入输出函数的声明)。“#include <stdio.h>”的作用是用来提供这些信息的。

程序第2行“main()”是主函数,主函数的名称是固定的,只能是main,程序的执行总是从主函数开始和结束。函数是C程序的基本单位,C语言源程序由函数组成。一个C程序可以只有一个main()函数(又称主函数)组成,也可以由一个main()函数和若干个其他函数(自定义函数或库函数)结合而成的。

程序第4行“printf”是C编译系统提供的标准库函数中的输出函数,其功能是把双撇号内的字符串按原样输出。“\n”是换行符,即在输出“Hello,world!”后回车换行。语句的最后有一个分号。

例 1.2 一个由两个函数构成的C程序。

```
#include <stdio.h>
int sum(int x, int y)
{ int z;
  z=x+y;
  return z;
}
```

```

main()
{ int  a, b, c;                /* 定义变量 a 和 b */
scanf("%d,%d",&a,&b);          /* 给变量 a 和 b 赋值 */
    c = sum(a, b);              /* 求 a 与 b 之和, 结果赋给变量 c */
    printf("c = %d\n", c);      /* 输出变量 c 的值 */
}

```

本程序包括两个函数: 主函数 main 和被调用函数 sum。sum 函数的作用是将 x 和 y 的和赋给变量 z, return 语句将 z 的值返回主函数 main。返回值是通过函数名 sum 带回 main 函数中调用 sum 函数的位置。

main 函数中的 scanf 是“输入函数”的名字, 程序中 scanf 函数的作用是输入 a 和 b 的值。&a 和 &b 中的“&”的含义是“取地址”, scanf 函数的作用是将两个数据分别输入变量 a 和 b 的地址所标志的单元中, 也就是输入给变量 a 和 b。scanf 函数中双撇号括起来的“%d,%d”是输入输出的“格式字符串”, “%d”表示以十进制整数形式输入。

程序中“c = sum(a, b);”语句是 sum 函数的调用, 在调用时将实际参数 a 和 b 的值分别传送给 sum 函数中的形式参数 x 和 y。经过执行 sum 函数得到一个返回值, 这个值返回到调用 sum 函数的位置, 然后把这个值赋给变量 c。printf 函数输出变量 c 的值。在执行 printf 函数时, 将“c=”原样输出, “%d”将由 c 的值取代之, “\n”执行换行。程序的执行结果如下:

输入: 5,6<回车>

输出结果为: c=11

从本例看出, 一个 C 程序由若干函数构成, 程序的执行总是从 main() 开始, 当发生函数的调用及返回时, 程序的执行流程在函数间跳转。

通过以上两个例子, 可以看到:

(1) C 程序是由函数构成的。一个 C 程序由一个或多个源文件组成, 一个源文件(模块)由一个或几个函数组成, 一个源文件是一个编译单位。函数是 C 程序的基本单位。一个 C 源程序必须包含且只能包含一个 main 函数, 也可以包含一个 main 函数和若干个其他函数。一个 C 程序总是从 main 函数开始执行的, 而不论 main 函数在整个程序中的位置如何。程序中的全部工作都是由各个函数分别完成的。编写 C 程序就是编写一个个函数。C 的这种特点使得容易实现程序的模块化。

编写 C 程序就是编写一个 C 函数, C 程序中所有的函数都是平行的, 即在定义函数时是相互独立的。函数不能嵌套定义, 但可相互调用, 要遵循调用规则, 也可自己调用自己(递归调用), 其他函数不能调用 main 函数。

(2) 一个函数由两部分组成。

① 函数的首部, 即函数的第一行。包括函数名、函数类型、函数属性、函数参数(形参)名、参数类型。

例如, 例 1.2 中的 sum 函数的首部为:

int	sum	(int	x,	int	y)
↓	↓	↓	↓	↓	↓
函数类型	函数名	函数参数类型	函数参数名	函数参数类型	函数参数名

一个函数名后面必须跟一对圆括弧, 函数参数可以没有, 如 main()。

② 函数体, 即函数首部下面的大括弧“{……}”内的部分。如果一个函数内有多个大括弧, 则最外层的一对“{ }”为函数体的范围。

函数体一般包括：

◆声明部分：在这部分中定义所用到的变量，如例 1.2 中 main 函数中的“int a,b,c;”。

◆执行部分：由若干个语句组成。当然，在某些情况下也可以没有声明部分(如例 1.1)。甚至可以既无声明部分，也无执行部分，但这是合法的。

(3)每个语句和数据定义的最后必须有一个分号。分号是 C 语句的必要组成部分。

(4)C 语言输入和输出的操作是由库函数 scanf 和 printf 等函数来完成的。

(5)可以使用“/* */”对 C 程序中的任何部分作注释，使用注释时需要注意以下几点：注释可以单独占一行，也可以跟在语句后面；“/*”和“*/”必须成对使用，并且“/”和“*”以及“*”和“/”之间不能有空格，否则会出错；如果注释内容在一行写不下，可以另起一行继续写；注释中允许使用汉字。在非中文操作系统下，看到的是一串乱码，但不影响程序运行。

1.4 C 程序的编译过程

采用 C 语言编写的程序称为 C 源程序(文件扩展名为 c)。计算机硬件只能执行机器代码指令序列(称为可执行程序)，而不能直接运行 C 源程序。因此，必须先编译程序，将 C 源程序转换成对应的可执行程序。由 C 源程序转换成可执行程序的过程一般要经过下面 4 个步骤。

(1)编辑。启动 C 编译器，编辑(或修改)源程序，并存盘。

(2)编译。如果编译成功，则可进行下一步操作；否则，修改源程序再重新编译，直至编译成功。

(3)连接。如果连接成功，则可进行下一步操作；否则，根据系统的错误提示进行相应修改，再重新连接，直至连接成功。

(4)运行。通过观察程序运行结果，验证程序的正确性。如果出现逻辑错误，则必须再修改源程序，重新编译、连接和运行，直至程序正确。

运行一个 C 程序的编译过程可用图 1-1 所示直观地表示。

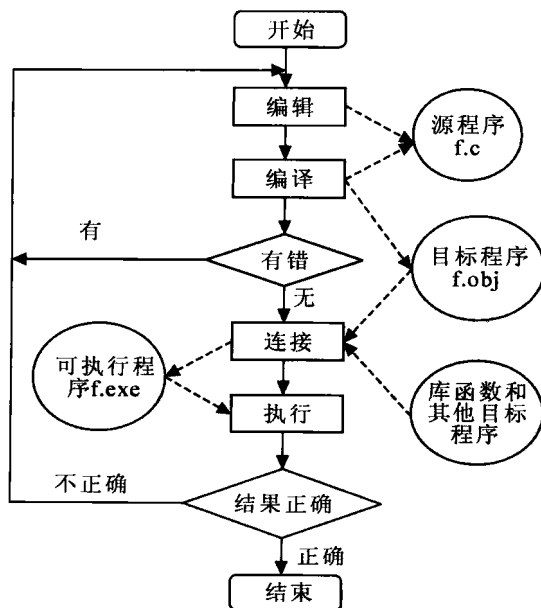


图 1-1 C 程序的编译过程

图中实线表示在上一步骤完成后直接进行下一步骤。虚线表示将本步骤的结果存储在文件中,以后再操作。编译程序将 C 源程序翻译成目标程序后保存在 .obj 文件中,目标程序还不能直接执行,因为其中还不包含类似于 printf() 等函数的可执行代码(这些可执行代码包含在 C 编译系统提供的函数库中),为此,将目标程序与函数库进行连接,生成最终的可执行程序保存在 .exe 文件中,该 .exe 文件可以脱离 C 编译系统直接运行。

习 题 1

一、选择题

1. 一个 C 程序的执行是从_____。
 - A. 本程序的 main 函数开始,到 main 函数结束
 - B. 本程序文件的第一个函数开始,到程序本文件的最后一个函数结束
 - C. 本程序的 main 函数开始,到本程序文件的最后一个函数结束
 - D. 本程序的第一个函数开始,到本程序 main 函数结束
2. 以下叙述不正确的是_____。
 - A. 一个 C 源程序可由一个或多个函数组成
 - B. 一个 C 源程序必须包含一个 main 函数
 - C. C 程序的基本组成单位是函数
 - D. 在 C 程序中,注释说明只能位于一条语句的后面
3. C 语言规定:在一个源程序中,main 函数的位置_____。
 - A. 必须在最开始
 - B. 必须在系统调用的库函数的后面
 - C. 可以任意
 - D. 必须在最后

二、填空题

1. 函数体一般包括_____和_____。
2. C 源程序的基本单位是_____。
3. 一个 C 源程序中至少应包含一个_____。
4. 在一个 C 源程序中,注释部分两侧的分界符分别是_____和_____。
5. 在 C 语言中,输入操作是由库函数_____完成的;输出操作是由库函数_____完成的。
6. 在 C 语言中,每个语句的结束符是_____。

三、编程题

1. 参照课本例 1.1,试编写一个 C 程序,输出如下信息。
“# \$ % ^ & * ~ | \”(提示:使用键盘上的一些符号)。
2. 试编写一个 C 程序,输出如下图案信息。

```
*  
*  *  
*  *  *
```

试试看,可以输出其他组合图案吗?

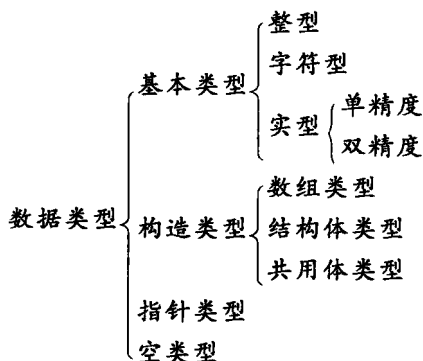
3. 编写一个求两个整数乘积的函数 `int mult(int x,int y)`,在 `main()` 函数中调用 `mult()` 计算两个整数乘积。

第2章 数据类型及其运算

2.1 C语言的数据类型

计算机中的数据不单是简单的数字,所有计算机处理的信息,包括文字、声音、图像等都是以一定的数据形式存储的。数据在内存中保存的情况是由数据类型决定的。针对不同类型的数据,计算机采取不同的存储方式并进行不同的处理。随着处理对象的复杂化,数据类型也变得更加丰富。数据类型的丰富程度直接反映了程序设计语言处理数据的能力。

C语言一个很重要的特点是数据类型十分丰富。因此,C语言程序的数据处理能力很强。在C语言中,数据类型可分为:基本数据类型、构造数据类型、指针类型和空类型4大类。



2.1.1 基本数据类型

基本数据类型最主要的特点是,其值不可以再分解为其他类型。也就是说,基本数据类型是自我说明的。利用基本数据类型可以构成更复杂的数据结构。

2.1.2 构造数据类型

构造数据类型是根据已定义的一个或多个数据类型用构造的方法来定义的。也就是说,一个构造类型的值可以分解成若干个“成员”或“元素”。每个“成员”都是一个基本数据类型或又是一个构造类型。在C语言中,构造类型有以下几种:

- ◆数组类型
- ◆结构体类型
- ◆共用体(联合)类型

2.1.3 指针类型

指针是一种特殊的,同时又是具有重要作用的数据类型。其值用来表示某个变量在内存存储器中的地址。虽然指针变量的取值类似于整型量,但这是两个类型完全不同的量,因此不能混为一谈。

2.1.4 空类型

在调用函数值时,通常应向调用者返回一个函数值。这个返回的函数值是具有一定的数据类型的,应在函数定义及函数说明中给以说明,例如在例题中给出的 max 函数定义中,函数头为: `int max(int a,int b)`;其中“int”类型说明符即表示该函数的返回值为整型量。又如在例题中,使用了库函数 sin,由于系统规定其函数返回值为双精度浮点型,因此在赋值语句 `s=sin(x)`;中,s 也必须是双精度浮点型,以便与 sin 函数的返回值一致。所以在说明部分,把 s 说明为双精度浮点型。但是,也有一类函数,调用后并不需要向调用者返回函数值,这种函数可以定义为“空类型”。其类型说明符为 void。在后面函数中还要详细介绍。

在本章中,我们先介绍基本数据类型中的整型、浮点型和字符型,其余类型在以后各章中陆续介绍。

2.2 常量与变量

对于基本数据类型量,按其取值是否可改变又分为常量和变量两种。在程序执行过程中,其值不发生改变的量称为常量,其值可变的量称为变量。它们可与数据类型结合起来分类。例如,可分为整型常量、整型变量、浮点常量、浮点变量、字符常量、字符变量、枚举常量、枚举变量。在程序中,常量是可以不经说明而直接引用的,而变量则必须先定义后使用。

2.2.1 常量和符号常量

在 C 语言中,常量有直接常量(或字面常量)和符号常量两种。直接常量即指 C 语言的数值常量和字符常量,直接常量可以从其字面形式上区分其数据类型。符号常量则实指 C 语言用标识符定义的常量。数值常量可以分为整型常量和实型常量。字符型常量可以分为字符常量和字符串常量。

1. 整型常量

合法的整型常量如 12、0、-3 等。

2. 实型常量

实型常量只能用十进制形式表示,如 4.6、-1.23 都是合法的实型常量。

3. 字符常量

'a'、'+'、'\n'、'\101'都是合法的字符常量。

4. 字符串常量

"gansu"、"university"、"++? ab"都是合法的字符串常量。

而'ab'既不是字符常量,也不是字符串常量。

5. 符号常量

在 C 语言中,可以用一个标识符来表示一个常量,称之为符号常量。

符号常量在使用之前必须在程序开头先定义,其一般形式为:

`#define 标识符 常量`

其中,“#define”也是一条预处理命令(预处理命令都以“#”开头),称为宏定义命令(在后面预