



普通高等教育“十三五”规划教材 计算机系列

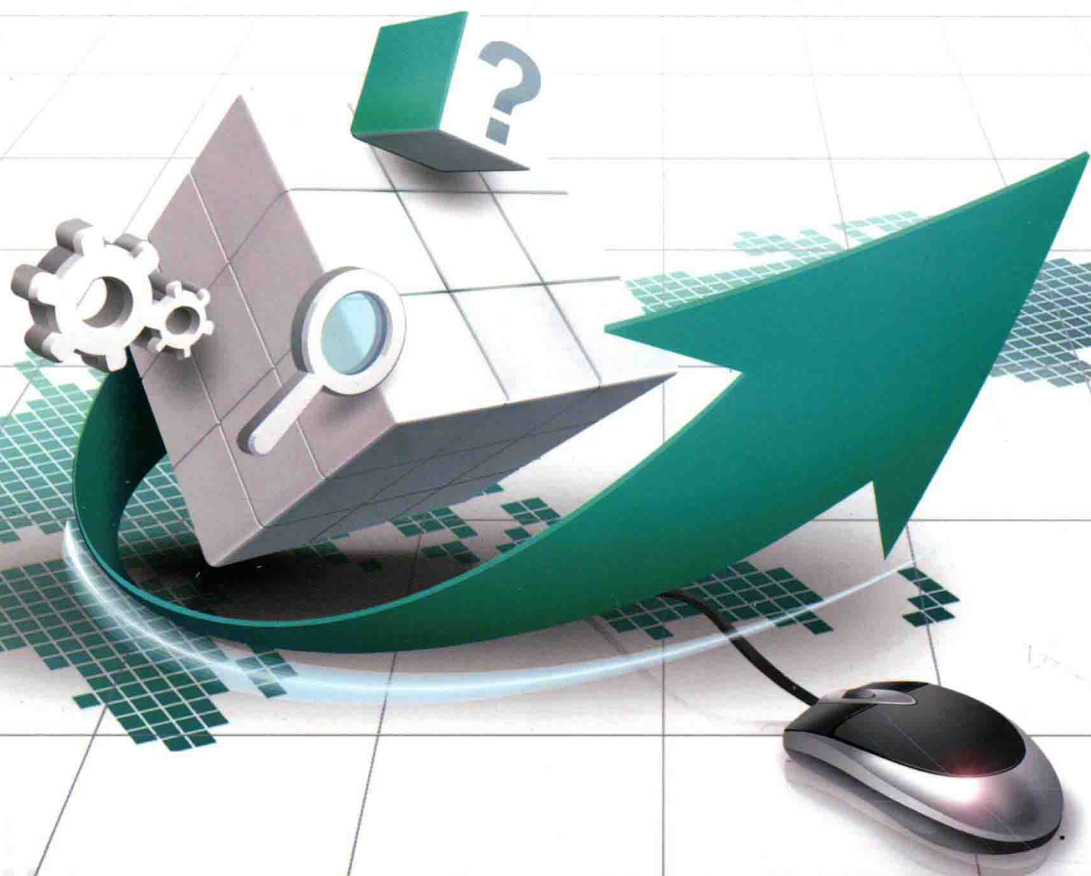
101010

10101001010111010101110

10101010

C语言程序设计 案例教程

孙 强 宋 丽 刘志宇 主 编
宋 阳 磨 然



科学出版社

普通高等教育“十三五”规划教材 计算机系列

C 语言程序设计案例教程

孙 强 宋 丽 刘志宇 主 编
宋 阳 磨 然
董会权 副主编

本书共分为12章。第1章介绍C语言的发展、特点、应用及安装；第2章介绍C语言的数据类型、运算符和表达式；第3章介绍C语言的程序结构；第4章介绍C语言的数组；第5章介绍C语言的指针；第6章介绍C语言的字符串；第7章介绍C语言的函数；第8章介绍C语言的预处理；第9章介绍C语言的位运算；第10章介绍C语言的编译；第11章介绍C语言的库函数；第12章介绍C语言的综合应用。

本书可作为高等院校计算机专业及相关专业的教材，也可供从事计算机工作的工程技术人员参考。

本书得到了牡丹江师范学院优秀青年骨干教师培养计划项目(MOP201406)和牡丹江师范学院教改项目(15-JG-001)的支持。在编写本书的过程中，参考了有关文献资料，在此表示衷心的感谢。

科学出版社

13201121303

科学出版社

内 容 简 介

本书基于 Visual C++ 6.0 集成环境,以案例教学为主线,根据编者多年的教学经验编写而成。本书分为 12 章,从分析 C 语言程序的基本结构开始,介绍了常量、变量、表达式和常用输入/输出库函数、流程控制、数组和字符串处理、函数、指针、编译预处理命令、结构体和共用体等知识。本书案例题型丰富,一些题目来自实际生活之中,有利于培养学生编程解决实际问题的意识和能力,提高学生对 C 语言的综合实践能力。每一章的示例程序都可在 Visual C++ 6.0 环境下编译运行,每一章后面均附有习题,读者通过完成习题可检验和巩固学过的内容。

本书既可作为高等院校本、专科学生学习 C 语言程序设计课程的教材和教学参考书,也可供各类 C 语言学习对象阅读参考。

图书在版编目(CIP)数据

C 语言程序设计案例教程/孙强等主编. —北京:科学出版社,2017
(普通高等教育“十三五”规划教材 计算机系列)

ISBN 978-7-03-051218-5

I. ①C… II. ①孙… III. ①C 语言-程序设计-高等学校-教材

IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2016)第 321303 号

责任编辑:戴薇 刘 扬 / 责任校对:张 曼
责任印制:吕春珉 / 封面设计:东方人华平面设计部

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

三河市良远印务有限公司印刷

科学出版社发行 各地新华书店经销

*

2017 年 7 月第 一 版 开本:787×1092 1/16

2017 年 12 月第二次印刷 印张:16 1/4

字数:368 000

定价:39.00 元

(如有印装质量问题,我社负责调换〈良远印务〉)

销售部电话 010-62136230 编辑部电话 010-62135927-2032

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

前 言

C 语言是一种广泛应用的结构化程序设计的高级语言。由于该语言具有简练灵活的特点,深受广大编程者的喜爱。C 语言具有功能丰富、表达力强、使用灵活方便、便于大型程序开发、便于编写、可移植性好等优点,又由于 C 语言引入了反映计算机硬件特性的机制,使 C 语言也能编写直接控制计算机硬件设备的程序,所以 C 语言是一门既适合编写系统软件,又适合设计应用软件的高级语言。在编排程序设计的内容方面,本书与配套教材《C 语言程序设计实训教程》(蔡春华、司巧梅、樊持杰、许艳惠、肖湘主编,科学出版社)的课程体系相吻合,做到循序渐进、系统学习、广泛实践,便于学生接受。案例精选了部分典型例题及一些实际生活中的题目,把 C 语言程序设计的方法融入实践环节,用大量的案例及同步练习帮助学生掌握程序设计的思想,学会程序设计的方法,训练程序设计的能力,达到初步解决程序设计实际问题的要求。

本书共分为 12 章,内容包括 C 语言概述、C 语言程序设计的基本知识、顺序结构、选择结构、循环结构、编译预处理、函数、数组、指针、结构体及共用体、文件管理、C 语言高级程序设计。本书是编者在多年来从事 C 语言程序设计教学工作的基础上,结合教学实践中的经验编写而成的。书中列举了学生容易出现问题的典型例题及实际生活中的题目,便于学生深入掌握重点内容,提高实践操作技能。本书力求体系结构安排合理、重点突出、难度适中;在语言叙述上注重概念清晰、通俗易懂,适应计算机教学实际需要。

本书由牡丹江师范学院孙强、宋丽、刘志宇、磨然和吉林工程技术师范学院宋阳担任主编,由吉林工业大学人文学院董会权担任副主编,具体编写分工如下:第 1 章、第 8 章、第 9 章由宋丽和董会权共同编写,第 2 章由磨然编写,第 3 章~第 7 章、前言、附录由孙强编写,第 10~第 12 章由宋阳和刘志宇共同编写。

本书得到了牡丹江师范学院优秀青年骨干教师培养计划项目(MQP201406)和牡丹江师范学院教改项目(16-JG18049 和 16-JG18046)的资助。

在编写本书的过程中,编者参阅了一些专家和同行的科研资料、专著、教材,在此一并表示感谢!

由于编者水平有限,加之时间仓促,书中的疏漏和不当之处在所难免,敬请广大读者批评指正。

编 者

2016 年 11 月

目 录

第 1 章 C 语言概述	1
1.1 程序设计语言及其发展	1
1.2 C 语言的标准	2
1.3 C 语言的程序结构	3
1.3.1 简单的 C 语言程序案例	3
1.3.2 C 语言程序的基本结构	6
1.3.3 集成开发环境	7
习题	7
第 2 章 C 语言程序设计的基本知识	9
2.1 数据类型	9
2.2 标识符、常量和变量	9
2.2.1 标识符	9
2.2.2 常量和变量	10
2.3 运算符和表达式	18
2.3.1 算术运算符和表达式	19
2.3.2 赋值运算符和表达式	21
2.3.3 关系运算符和表达式	22
2.3.4 逻辑运算符和表达式	23
2.3.5 其他运算符和表达式	25
习题	25
第 3 章 顺序结构	27
3.1 C 语言的语句	27
3.2 输入/输出函数	28
3.2.1 格式输出函数 printf()	28
3.2.2 格式输入函数 scanf()	33
3.2.3 字符输出函数 putchar	35
3.2.4 字符输入函数 getchar	36
3.3 顺序结构的应用	36
习题	39



第 4 章 选择结构	41
4.1 用 if 语句实现选择结构	41
4.1.1 if 语句的基本形式	41
4.1.2 if 语句的嵌套	44
4.2 switch 语句	46
习题	49
第 5 章 循环结构	55
5.1 while 循环结构	55
5.2 do...while 循环结构	57
5.3 for 循环结构	60
5.4 循环嵌套	62
5.5 break 语句和 continue 语句	64
5.5.1 break 语句	64
5.5.2 continue 语句	65
习题	66
第 6 章 编译预处理	75
6.1 宏定义	76
6.1.1 无参数的宏定义	76
6.1.2 带参数的宏定义	78
6.2 文件包含	83
习题	84
第 7 章 函数	86
7.1 函数的定义	87
7.2 函数调用	89
7.3 函数的返回值	94
7.4 函数的说明	94
习题	98
第 8 章 数组	101
8.1 一维数组	101
8.1.1 一维数组的定义	101
8.1.2 一维数组元素的引用	103
8.1.3 一维数组的初始化	104

8.1.4 一维数组的应用	105
8.2 二维数组	108
8.2.1 二维数组的定义	108
8.2.2 二维数组元素的引用	109
8.2.3 二维数组的初始化	110
8.2.4 二维数组的应用	111
8.3 字符串与字符数组	112
8.3.1 字符串	112
8.3.2 字符数组	113
8.3.3 用于字符处理的函数	117
8.3.4 字符串处理程序设计示例	119
8.3.5 二维字符数组	119
习题	121
第9章 指针	126
9.1 变量的地址与指针	126
9.2 指针变量的定义和基本类型	129
9.3 指针变量的赋值	129
9.4 指针变量的引用	131
9.5 指针变量作为函数参数	137
9.6 指向一维数组的指针变量	139
9.6.1 一维数组指针	139
9.6.2 数组元素地址作为实参	143
9.6.3 数组名作为函数参数	144
9.6.4 函数的指针形参和函数体中数组的区别	146
9.7 指向二维数组的指针变量	147
9.7.1 二维数组地址	147
9.7.2 引用二维数组元素	151
9.7.3 二维数组名作为实参	151
9.8 字符串与字符串指针变量	154
9.9 指针数组	158
9.9.1 指针数组的定义及引用	158
9.9.2 指针数组作为实参	160
9.10 指向指针的指针	160
9.11 指针数组作为 main 函数的形参	162
9.12 指针与函数	164



9.12.1 指向函数的指针变量	164
9.12.2 返回指针值的函数	165
习题	166
第 10 章 结构体及共用体	170
10.1 结构体	170
10.1.1 结构体类型的定义和变量的说明	171
10.1.2 结构体变量的赋值、初始化及引用	175
10.1.3 结构体数组的定义、初始化及引用	178
10.1.4 结构体指针变量的说明和使用	180
10.1.5 结构体变量作为函数参数	184
10.2 共用体	186
10.2.1 共用体类型的定义和变量的说明	186
10.2.2 共用体变量的引用	187
习题	188
第 11 章 文件管理	193
11.1 概述	193
11.2 文件的打开与关闭	194
11.2.1 文件指针	194
11.2.2 文本文件与二进制文件	195
11.2.3 文件的打开	196
11.2.4 文件的关闭	197
11.3 文件的读/写	198
11.3.1 fputc 函数和 fgetc 函数	198
11.3.2 fread 函数和 fwrite 函数	201
11.3.3 fscanf 函数和 fprintf 函数	205
11.3.4 fgets 函数和 fputs 函数	208
11.4 文件的定位	209
习题	212
第 12 章 C 语言高级程序设计	215
12.1 位运算	215
12.1.1 位运算符	216
12.1.2 按位与运算	216
12.1.3 按位或运算	217

12.1.4 按位异或运算	217
12.1.5 按位取反运算	218
12.1.6 左移运算符	218
12.1.7 右移运算符	219
12.1.8 程序举例	219
12.2 动态存储分配	220
12.3 链表	226
12.3.1 利用结构体变量构成链表	226
12.3.2 动态链表	228
12.3.3 链表的建立	229
12.3.4 单向链表的遍历	230
12.3.5 查找数据项	230
12.3.6 插入结点	231
12.3.7 删除结点	233
12.3.8 清空链表	234
习题	243
附录 1 C 语言中的关键字	245
附录 2 运算符与结合性	246
附录 3 ASCII 码字符表	247
参考文献	248

第1章 C语言概述

1.1 程序设计语言及其发展

自1946年世界上第一台电子计算机问世以来,计算机科学及其应用的发展十分迅猛,计算机被广泛地应用于人类生产、生活的各个领域,推动了社会的进步与发展。特别是随着国际互联网(Internet)的日益深入,传统的信息收集、传输及交换方式正被革命性地改变着,我们已经难以摆脱对计算机的依赖,计算机已将人类带入了一个新的时代——信息时代。

新的时代对我们的基本要求之一是自觉地、主动地学习和掌握计算机的基本知识和基本技能,并把它作为自己应该具备的基本素质。要充分认识到,缺乏计算机知识,就是信息时代的“文盲”。

一台计算机是由硬件系统和软件系统两大部分构成的,硬件是物质基础,而软件可以说是计算机的灵魂。没有软件,计算机就是一台“裸机”,是什么也不能做的,有了软件,计算机才能灵动起来,成为一台真正的“电脑”。所有的软件,都是用计算机语言编写的。计算机程序设计语言的发展,经历了从机器语言、汇编语言到高级语言的历程。

1. 机器语言

电子计算机所使用的是由“0”和“1”组成的二进制数,二进制是计算机语言的基础。计算机发明之初,人们只能用计算机的语言去命令计算机操作,即写出一串串由“0”和“1”组成的指令序列交由计算机执行,这种语言就是机器语言。使用机器语言是十分烦琐的,特别是在程序有错需要修改时。而且,由于每台计算机的指令系统往往不相同,所以,在一台计算机上执行的程序,要想在另一台计算机上执行,必须另编程序,这样就造成了重复工作。但由于机器语言是针对特定型号计算机的语言,故运算效率是所有语言中最高的。机器语言是第一代计算机语言。

2. 汇编语言

为了减轻使用机器语言编程的痛苦,人们进行了一种有益的改进:用一些简洁的英文字母、符号串来替代一个特定指令的二进制串。例如,用“ADD”代表加法,用“MOV”代表数据传递等,这样,人们很容易读懂并理解程序在干什么,纠错及维护都变得方便了,这种程序设计语言称为汇编语言,即第二代计算机语言。然而,计算机是不认识这些符号的,这就需要一个专门的程序,专门负责将这些符号翻译成二进制数的机器语言,这种翻译程序称为汇编程序。



汇编语言同样十分依赖于机器硬件，移植性不好，但效率十分高，针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件的功能和特长，程序精炼而质量高，所以至今仍是一种常用而强有力的软件开发工具。

3. 高级语言

从最初与计算机交流的痛苦经历中，人们意识到，应该设计一种这样的语言，这种语言接近于数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有机器上通用。经过努力，1954 年，第一个完全脱离机器硬件的高级语言——FORTRAN 问世了。40 多年来，共有几百种高级语言出现，有重要意义的有几十种，影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、SNOBOL、PL/1、Pascal、C、Prolog、Ada、C++、VC、VB、Delphi、Java 等。

高级语言的发展也经历了从早期语言到结构化程序设计语言，从面向过程到非过程化程序语言的过程。相应地，软件的开发也由最初的个体手工作坊式的封闭式生产，发展为产业化、流水线式的工业化生产。

20 世纪 60 年代中后期，软件越来越多，规模越来越大，而软件的生产基本上是“人自为战”，缺乏科学规范的系统规划与测试、评估标准，其恶果是大批耗费巨资建立起来的软件系统，由于含有错误而无法使用，甚至带来巨大损失，软件给人的感觉是越来越不可靠，以致几乎没有不出错的软件。这一切极大地震动了计算机界，史称“软件危机”。人们认识到：大型程序的编制不同于写小程序，它应该是一项新的技术，应该像处理工程一样处理软件研制的全过程。程序的设计应易于保证正确性，也便于验证正确性。1969 年，人们提出了结构化程序设计方法。1970 年，第一个结构化程序设计语言——Pascal 语言出现，标志着结构化程序设计时期的开始。

20 世纪 80 年代初，在软件设计思想上，又发生了一次革命，其成果就是面向对象的程序设计。在此之前的高级语言，几乎都是面向过程的，程序的执行是流水线式的，在一个模块被执行完成前，人们不能做其他事，也无法动态地改变程序的执行方向。这和人们日常处理事物的方式是不一致的，对于人而言，希望发生一件事就处理一件事，也就是说，不能面向过程，而应是面向具体的应用功能，也就是对象。其方法就是软件的集成化，如同硬件的集成电路一样，生产一些通用的、封装紧密的功能模块，称之为软件集成块，它与具体应用无关，但能相互组合，完成具体的应用功能，同时又能重复使用。对于使用者来说，只关心它的接口（输入量、输出量）及能实现的功能，至于是如何实现的，那是它内部的事情，使用者完全不用关心，C++、VB、Delphi 就是典型代表。

高级语言的下一个发展目标是面向应用，也就是说，只需要告诉程序自己要干什么，程序就能自动生成算法，自动进行处理，这就是非过程化的程序语言。

1.2 C 语言的标准

起初，C 语言没有官方标准。1978 年，美国电话电报公司贝尔实验室正式发表了 C

语言。布莱恩·柯林汉和丹尼斯·里奇出版了一本书——*The C Programming Language*。这本书被C语言开发者们称为K&R，很多年来被当作C语言的非正式的标准说明。人们称这个版本的C语言为K&R C。

1. ANSI C89 标准

20世纪七八十年代，C语言被广泛应用，从大型主机到小型微机，也衍生了C语言的很多不同版本。

1983年，美国国家标准协会(ANSI)成立了一个委员会——X3J11，来制定C语言标准。

1989年，美国国家标准协会通过了C语言标准——ANSI X3.159: 1989。因为这个标准是1989年通过的，所以一般简称为C89标准。有些人也简称为ANSI C，因为这个标准是美国国家标准协会发布的。

1990年，国际标准化组织(ISO)和国际电工委员会(IEC)把C89标准定为C语言的国际标准，即ISO/IEC 9899:1990。因为此标准是在1990年发布的，所以有些人把它称作C90标准。不过大多数人依然称之为C89标准，因为此标准与ANSI C89标准完全等同。

1994年，国际标准化组织和国际电工委员会发布了修订版，即ISO/IEC 9899:1990/Cor 1:1994，有时也简称为C94标准。

1995年，国际标准化组织和国际电工委员会再次发布修订版，即ISO/IEC 9899:1990/Amd 1:1995，有时也简称为C95标准。

2. C99 标准

1999年1月，国际标准化组织和国际电工委员会发布了C语言的新标准，即ISO/IEC 9899:1999，简称为C99标准。这是C语言的第二个官方标准。

3. C11 标准

2011年12月8日，国际标准化组织和国际电工委员会再次发布了C语言的新标准，即ISO/IEC 9899:2011-Information technology—Programming languages-C[9]，简称为C11标准，原名C1X。这是C语言的第三个官方标准，也是C语言的最新标准。

1.3 C语言的程序结构

1.3.1 简单的C语言程序案例

下面先介绍几个简单C语言程序，然后从中分析C语言程序的基本结构及所涵盖的基础语法知识，从而对C语言程序进行初步认识。

【例 1-1】要求在屏幕上输出以下信息。

Welcome to C world!

程序代码如下：

```
#include <stdio.h>                /*预处理命令*/
void main()                        /*主函数*/
{
    printf("Welcome to C world! \n");    /*格式输出函数*/
}
```

程序说明：

(1) 该程序是一个最简单的 C 语言程序，在程序的第 2 行中 main 是 C 语言程序中的“主函数”的名称。main 前面的 void 表示此函数是“空类型”，即执行此函数后不产生函数值。C 语言中，每个程序都必须有一个 main 函数，它是每一个 C 语言程序的执行起始点（入口点），也是程序执行的结束点。

(2) 用 {} 括起来的是 main 函数的函数体。main 函数中的所有操作（或语句）都在这一对 {} 之间。也就是说，main 函数的所有操作都在 main 函数的函数体中。

(3) 本例主函数中只有一条语句，printf 是 C 语言库函数中的格式输出函数，功能是把该语句中双引号内的字符原样输出（显示在屏幕上），“\n”是换行符，即在输出“Welcome to C world!”后按回车符换行。语句最后必须有一个分号。

(4) 在 C 语言中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信息（即对这些输入/输出函数的说明），程序第 1 行“#include <stdio.h>”就是用来提供这些信息的，“stdio.h”是 C 语言编译系统提供的一个标准输入/输出的库文件。include 是文件包含命令，相关内容会在第 5 章中详细介绍，这里不再赘述。在此只需记住，在程序中会用到系统提供的标准函数库中的输入/输出函数，即本例中的 printf 函数，因此在程序的开头应写上这样一行：

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

(5) “/* */”表示注释部分，注释可以用汉字或英文字符表示，可以出现在语句的后面，也可以单独成为一行。注释部分在程序编译和运行过程中不起作用，只是便于其他人读懂程序。

【例 1-2】求两数中的最小值。

程序代码如下：

```
#include<stdio.h>
void main()                /*主函数*/
{
    int a, b, c;            /*定义变量*/
    scanf("%d, %d", &a, &b);    /*输入数*/
}
```

```

c=min(a, b);          /*调用函数*/
printf("min=%d\n", c); /*输出*/
}

int min(int x, int y)  /*定义函数*/
{
    int z;             /*定义局部变量*/
    if (x>y)
        z=y;
    else
        z=x;
    return(z);         /*返回 z 值*/
}

```

经编译后运行结果:

```

14, 4 ✓
min=4

```

再次运行:

```

6, 29 ✓
min=6

```

程序说明:

(1) 本程序包括两个函数: 主函数 `main()` 和子函数 `min()`。`min` 函数的作用是将 `x`、`y` 中较小的数传送给变量 `z`, `return` 语句的作用是将 `z` 的值返回 `main` 主函数。返回值通过函数名 `min` 带回到 `main` 函数的调用处。

(2) `main` 函数中的 `scanf()` 是系统给出的标准输入函数, 通过它输入 `a` 和 `b` 的值, `&a`、`&b` 中的 “&” 是取地址符, 其功能是将输入的值分别送入变量 `a` 和 `b` 的地址所标示的存储单元, 即输入给变量 `a` 和 `b`。“`%d,%d`” 是输入/输出的格式字符串, 用来指定输入/输出数据的类型和格式。

(3) `main` 函数中第 4 行是调用 `min` 函数, 在调用时将实际参数 `a`、`b` 的值送给 `min` 函数中的形参 `x` 和 `y`。在 `min` 函数后得到一个返回值 (即 `min` 函数中的变量 `z` 的值), 把这个值送给变量 `c`, 然后输出变量 `c` 的值。

通过上述两个例子, 可知以下几点。

(1) C 语言程序是由函数构成的。一个程序至少含有一个 `main` 函数。函数是 C 语言的基本单位。被调用的子函数可以是系统给出的库函数, 也可以是自己定义的函数。

(2) 一个函数由两部分组成: 函数的首部 and 函数体。

① 函数首部: 函数的第一行, 包含函数类型、函数名、函数属性、函数参数名 (形式参数, 简称形参)、参数类型等。

以例 1-2 中的 `min` 函数来介绍, 可以看到



函数参数可以没有，但是函数后必须带一对小括号，即

```
void main()
```

② 函数体：函数首部下面大括号中的内容，当一个函数内有多个大括号时，最外层的一对大括号所包括的内容为函数体的范围。

函数体一般包含说明和执行两部分。

说明部分用于定义函数中所使用的变量和对所调用函数的说明，如例 1-2 中 `main` 函数体中的 `int a,b,c;`。

执行部分由若干条语句组成。

在某些情况下也可以没有说明部分，甚至没有执行部分。例如：

```
void main()  
{ }
```

这也是一个合法的 C 语言程序。

(3) `main` 函数通常位于程序之首，实际上它位于程序的开头、最后及函数与函数之间均是合法的，但不管在什么位置，一个 C 语言程序总是从 `main` 函数开始执行。

(4) C 语言程序书写格式自由，一行内可以写几条语句，一条语句可以分写在多行上。例如：

```
int a,b;  
a=5; b=8; /*两条语句在一行*/  
printf("a=%d,b=%d\n",  
      a,b); /*一条语句在两行*/
```

(5) C 语言本身没有输入/输出语句，输入/输出都由函数实现，因此，在程序的开始必须有头文件 `"include <stdio.h>"`。

(6) 可以用 `/* ..... */` 对 C 语言程序中的任何部分做注释，程序中加入必要的注释可增加程序的可读性。

(7) C 语言程序中每一条语句结束后都必须加一个分号。分号是 C 语言语句的组成部分。

1.3.2 C 语言程序的基本结构

(1) C 语言程序是由函数组成的。一个完整的 C 语言程序可以由一个或多个函数组成，其中 `main` 函数必不可少，且只有一个。C 语言程序执行时，总是从 `main` 函数开始，与 `main` 函数在整个程序中的位置无关，其他函数都是为 `main` 函数服务的。函数是 C 语

言程序的基本单位,用函数来实现特定的功能,所以说C语言是函数式的语言。C语言的函数包括系统提供的库函数(如printf函数),以及用户根据实际问题编制设计的函数。

(2) 源程序中可以有预处理命令,预处理命令通常放在源文件或源程序的最前面。

(3) 每一条语句都必须以分号结尾,但预处理命令、函数头和右大括号“}”之后不加分号。

(4) 注释不是程序的必需部分,在程序执行时注释不起任何作用。注释的作用是增加程序的可读性,因此,适当地在程序中加入注释,是一种良好的程序设计风格。C语言有块注释和行注释两种注释方法。

① 块注释。

形式:

```
/*
```

```
.....
```

```
注释内容
```

```
.....
```

```
*/
```

这是跨多行的注释语句,适用于注释多行,“/*”和“*/”之间为注释内容。

② 行注释。

形式:

```
/*...注释内容...*/
```

行注释放在一行上,通常放在语句之后。

(5) 在C语言中,虽然一行可写多条语句,一条语句也可占多行,但建议一行只写一条语句。

(6) 一般采用缩进格式书写程序,以提高程序的可读性和清晰性。

1.3.3 集成开发环境

集成开发环境提供编程时所必需的工具,这些工具有编辑器、编译器和调试器,它们集成在一个软件包内供程序员使用。在早前的DOS环境下,主要的C语言集成开发环境有TC 2.0、Turbo C++ 3.0和Borland C++。在Windows环境下,主要使用Visual C++ 6.0。Visual C++ 6.0既可以对C++进行编译,又可以对C进行编译。本书的示例程序均在Visual C++ 6.0集成环境下进行编辑、编译和运行。

习 题

一、填空题

1. C语言中,_____是程序的基本组成部分。



2. C 语言源程序的基本单位是_____。
3. C 语言中，一个函数由函数首部和_____两部分组成。
4. 一个 C 语言程序总是从_____开始执行。
5. 一个 C 源程序中至少应包括一个_____函数。

二、选择题

1. 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
2. 关于建立函数的目的，以下的说法正确的是（ ）。
A. 提高程序的执行效率 B. 提高程序的可读性
C. 减少程序的篇幅 D. 减少程序文件所占内存
3. 以下对 C 语言函数的描述中，正确的是（ ）。
A. C 语言程序必须由一个或一个以上的函数组成
B. C 函数既可以嵌套定义，又可以递归调用
C. 函数必须有返回值，否则不能使用函数
D. C 语言程序中有调用关系的所有函数必须放在同一个程序文件中

三、编程题

1. 参照本章例题，编写一个 C 语言程序，输出以下信息。

```
*****
Hello World!
*****
```

2. 参照本章例题，编写一个 C 语言程序，求任意两个整数的差并输出结果。
3. 指出下面程序中的错误。

```
#include (stdio.h)
main()
{
    int x , y , s ;
    x=10
    y=20;
    s=x + y ;
    printf("s = %d\n , s );
}
```