



安徽省高等学校“十一五”省级规划教材
高职高专计算机系列规划教材

C语言程序设计

方少卿 主编 汪伟 陈吉祥 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



安徽省高等学校“十一五”省级规划教材
高职高专计算机系列规划教材

C 语言程序设计

主 编 方少卿

副主编 汪 伟 陈吉祥

参 编 查 艳 李 婷
伍丽惠 汪广舟

内 容 简 介

本书是安徽省高等学校“十一五”省级规划教材,全书本着“案例导入,循序渐进;按章小结,复习巩固”的原则编写,第2~10章以一个通俗实例开始引入,同时全书配以相当数量的例题与习题,以便于读者学习C语言。

全书共分11章,主要内容包括C语言概述,数据类型、变量、运算符与表达式,程序设计算法基础,基本控制结构,数组,函数,指针,结构体、共用体与枚举,位运算,文件以及Visual C++ 6.0简介。本书注重应用性和实践性,通过一些典型例题的解题分析及C程序实现以帮助读者进一步加强对C语言的理解学习,每章内容后有对本章的复习指南,以帮助读者更好地掌握本章内容,并配有一定数量的习题供读者练习。为了给读者提供参考和强化读者对C语言的操作技能,还同时配套编写了《C语言程序设计实训指导与习题解答》供读者借鉴和参考。

本书适合作为高职高专各专业“C语言程序设计”课程的教材,也可作为电大、成人院校、各类培训教材用书,还可供准备参加计算机等级考试和自学C语言的读者阅读参考。

图书在版编目(CIP)数据

C语言程序设计/方少卿主编. —北京:中国铁道出版社, 2009.3

(高职高专计算机系列规划教材)

ISBN 978-7-113-09836-0

I. C… II. 方… III. C语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2009)第038260号

书 名: C语言程序设计

作 者: 方少卿 主编

策划编辑: 严晓舟 秦绪好

责任编辑: 翟玉峰 黄园园

编辑部电话: (010) 63583215

封面设计: 付 巍

封面制作: 白 雪

责任印制: 李 佳

出版发行: 中国铁道出版社(北京市宣武区右安门西街8号 邮政编码: 100054)

印 刷: 河北省遵化市胶印厂

版 次: 2009年5月第1版

2009年5月第1次印刷

开 本: 787mm×1092mm 1/16 印张: 21.25 字数: 496千

印 数: 4 000册

书 号: ISBN 978-7-113-09836-0/TP·3185

定 价: 30.00元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签,无标签者不得销售

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

21 世纪,信息技术高速发展并广泛应用于各个行业,计算机已经成为人们日常工作学习的常用工具,每一个公民都应当学习一些计算机知识。作为在校高职学生,学习计算机知识更是必然的选择,尤其学习一些计算机程序设计知识,将有助于今后更好地学习和工作。

掌握程序设计的前提是掌握程序设计语言,C 语言是目前广泛流行的通用程序设计语言,是高职院校普遍开设的一门计算机专业基础课程,也是计算机爱好者学习程序设计语言的首选。在众多的程序设计语言中,C 语言以其灵活性和实用性受到了广大计算机专业人员和应用人员的喜爱。C 语言是既被美国国家标准化协会(ANSI)认可又为工业界广泛支持的计算机语言之一,几乎任何一种机型、任何一种操作系统都支持 C 语言开发;C 语言在巩固其原有应用领域的同时,还拓展新的应用领域,支持大型数据库开发和 Internet 应用。读者一旦掌握了 C 语言,就可以较为轻松地学习其他任何一种程序设计语言,为后续的面向对象程序设计、Windows 程序设计、Java 程序设计等程序设计语言的学习打下基础。

本书以程序设计为主线,采用“案例导入,循序渐进;按章小结,复习巩固”的原则,以程序设计的需求引入 C 语言知识的学习,每章以一个典型而浅显的引例引入,系统介绍了 C 语言程序设计的基本知识、程序设计的算法基础、基本数据类型和数据运算、程序控制结构、数组、指针、函数、编译预处理、结构体和共用体、文件等,并通过丰富的程序设计实例,加强程序设计思维方法和实际编程的训练,最后介绍了 Visual C++ 6.0 集成开发环境。每章后面都配有本章的复习指南,同时安排了大量的练习题,以帮助读者对 C 语言的学习和巩固。

为了配合读者学习 C 语言知识,同时为了强化技能,提高读者对 C 语言学习的掌握程度及阅读程序、调试程序和编制程序的实际操作能力,编者编写了配套的实训教材,并与全书习题的参考答案结集出版了配套的《C 语言程序设计实训指导与习题解答》,供读者学习和借鉴。

本书是安徽省高等学校“十一五”省级规划教材,全书共分 11 章,其中第 1、2 章和附录 A~C 由铜陵职业技术学院查艳编写,第 3 章和第 9 章由铜陵职业技术学院伍丽惠编写,第 6 章的 6.0~6.7 节由铜陵职业技术学院李婷编写,第 4 章、第 6 章的 6.8 节和附录 D、附录 E 由铜陵职业技术学院方少卿编写,第 6 章的 6.9 节及习题由铜陵职业技术学院方少卿、李婷合编,第 7 章和第 8 章由安徽商贸职业技术学院汪伟编写,第 5 章和第 10 章由安徽工业经济职业技术学院陈吉祥编写,第 11 章由铜陵职业技术学院汪广舟编写。方少卿负责全书的规划与统稿。

需要本书配套课件和源程序的读者,请访问 <http://edu.tqbooks.net> 网站下载或与作者联系(E-mail: ahfsq@126.com)。

本书编写过程中得到了铜陵职业技术学院领导和信息工程系领导的大力支持,其中邹和平、胡登峰、王宗涛、高良诚、韩传稳对本书的编写提出了许多宝贵意见,李超、杨世琴、沈雅平、王玉霞、谢丽丽、马里艳、乔陈亮验证了全书的大部分源代码。在此向他们一并表示感谢!

本书编写过程中,参考了国内外同类教材,在此对这些教材的编者表示感谢!同时在编写过程中还得到了兄弟院校领导和老师的大力支持;中国铁道出版社的编辑为本书的出版付出了辛勤汗水,做了大量工作,在此对他们一并表示感谢!

由于编者水平有限,难免存在疏漏和不足之处,敬请读者和各位同仁不吝赐教。

编 者

2009 年 3 月

第 1 章 C 语言概述	1
1.1 C 语言的演变及特点	1
1.1.1 C 语言的发展过程	1
1.1.2 C 语言版本	1
1.1.3 C 语言的特点	1
1.2 C 源程序的结构特点	2
1.2.1 简单的 C 程序介绍	2
1.2.2 C 源程序的结构特点	2
1.2.3 书写程序时应遵循的规则	4
1.3 数据输入和输出的常用方法	4
1.3.1 格式化输出函数 printf	4
1.3.2 格式化输入函数 scanf	5
1.4 C 语言的字符集与 C 语言词汇	5
1.4.1 C 语言的字符集	5
1.4.2 C 语言词汇	6
1.5 Turbo C 集成开发环境	7
1.5.1 Turbo C 2.0 安装与启动	8
1.5.2 Turbo C 2.0 菜单介绍	8
1.6 复习指南	18
1.6.1 本章主要内容	18
1.6.2 本章学习提示	18
习题一	18
第 2 章 数据类型、运算符与表达式	20
2.0 引例	20
2.1 C 语言的数据类型	20
2.2 常量与变量	21
2.2.1 常量和符号常量	21
2.2.2 变量	22
2.3 整型数据	23
2.3.1 整型常量	23
2.3.2 整型变量	23
2.4 实型数据	26
2.4.1 实型常量	26
2.4.2 实型变量	27
2.5 字符型数据	28

2.5.1	字符常量	28
2.5.2	转义字符	28
2.5.3	字符变量	29
2.5.4	字符数据在内存中的存储形式及使用方法	29
2.5.5	字符串常量	31
2.6	C 运算符及表达式	31
2.6.1	表达式	31
2.6.2	C 运算符	31
2.6.3	算术运算符和算术表达式	32
2.6.4	赋值运算符和赋值表达式	34
2.6.5	逗号运算符和逗号表达式	35
2.6.6	关系运算符和关系表达式	36
2.6.7	逻辑运算符和逻辑表达式	36
2.6.8	条件运算符和条件表达式	37
2.6.9	运算符优先级和结合性	37
2.7	变量赋初值	38
2.8	数据类型转换	38
2.8.1	自动类型转换	38
2.8.2	强制类型转换	39
2.9	复习指南	40
2.9.1	本章主要内容	40
2.9.2	本章学习提示	42
习题二		42
第 3 章	程序设计算法基础	45
3.0	引例	45
3.1	算法概述	45
3.1.1	算法的概念	46
3.1.2	算法举例	46
3.2	算法的特性	47
3.3	算法的表示	47
3.3.1	用自然语言表示算法	48
3.3.2	用流程图表示算法	48
3.3.3	3 种基本控制结构和改进的流程图	50
3.3.4	用 N-S 图表示算法	53
3.3.5	用伪代码表示算法	53
3.3.6	用 PAD 图描述算法	54
3.3.7	用计算机语言表示算法	55
3.4	结构化程序设计方法	55
3.5	复习指南	58

3.5.1	本章主要内容	58
3.5.2	本章学习提示	58
习题三		59
第 4 章	基本控制结构	60
4.0	引例	60
4.1	C 语句概述	61
4.2	赋值语句	62
4.3	字符的输入与输出函数	63
4.3.1	字符输出函数 putchar	63
4.3.2	字符输入函数 getchar	64
4.4	格式化输入与输出函数	65
4.4.1	格式化输出函数 printf	65
4.4.2	格式化输入函数 scanf	68
4.5	顺序结构程序设计	70
4.5.1	顺序结构程序设计概念	70
4.5.2	顺序结构程序设计举例	71
4.6	选择结构程序设计	72
4.6.1	选择结构程序引例	72
4.6.2	选择结构程序概述	72
4.6.3	选择结构控制语句种类	73
4.6.4	if 语句	73
4.6.5	switch 语句	78
4.6.6	选择结构程序举例	81
4.7	循环结构程序	83
4.7.1	循环结构程序引例	83
4.7.2	循环结构程序概述	85
4.7.3	while 语句	85
4.7.4	do...while 语句	87
4.7.5	for 语句	90
4.7.6	goto 语句以及用 goto 语句构成循环	95
4.7.7	break 语句和 continue 语句	97
4.7.8	几种循环的比较	98
4.7.9	循环的嵌套	99
4.7.10	循环结构程序举例	103
4.8	3 种基本结构典型程序举例	104
4.9	复习指南	107
4.9.1	本章主要内容	107
4.9.2	本章学习提示	108
习题四		109

第 5 章 数组	118
5.0 引例	118
5.1 一维数组	119
5.1.1 一维数组的定义	119
5.1.2 一维数组元素的引用	120
5.1.3 一维数组的初始化	121
5.1.4 一维数组程序举例	122
5.2 二维数组	123
5.2.1 二维数组的定义	123
5.2.2 二维数组元素的引用	124
5.2.3 二维数组的初始化	124
5.2.4 二维数组程序举例	125
5.3 字符数组	127
5.3.1 字符数组的定义	128
5.3.2 字符数组的初始化	128
5.3.3 字符数组的引用	128
5.3.4 字符数组与字符串	129
5.3.5 字符数组的输入/输出	130
5.3.6 字符串处理函数	131
5.4 程序举例	134
5.5 复习指南	138
5.5.1 本章主要内容	138
5.5.2 本章学习提示	140
习题五	140
第 6 章 函数	146
6.0 引例	146
6.1 函数概述	147
6.2 函数的定义	149
6.3 函数的调用	150
6.3.1 函数调用的一般形式	150
6.3.2 函数的参数和函数值	151
6.3.3 函数的调用举例	154
6.3.4 函数的嵌套调用	155
6.3.5 函数的递归调用	156
6.4 函数声明和函数原型	160
6.4.1 函数声明	160
6.4.2 函数原型	161
6.5 局部变量和全局变量	163
6.5.1 局部变量	163

6.5.2	全局变量	165
6.5.3	用 extern 声明外部变量	166
6.6	变量的存储类型	168
6.6.1	静态存储方式与动态存储方式	169
6.6.2	用 auto 声明的局部变量	169
6.6.3	用 static 声明的局部变量	170
6.6.4	用 register 声明的局部变量	171
6.7	预处理命令	172
6.7.1	宏定义	172
6.7.2	文件包含	175
6.7.3	条件编译	176
6.7.4	库函数简介	178
6.8	复习指南	179
6.8.1	本章主要内容	179
6.8.2	本章学习提示	181
	习题六	182
第 7 章	指针	191
7.0	引例	191
7.1	指针的基本概念	193
7.2	变量的指针和指向变量的指针变量	195
7.2.1	指针变量的定义	195
7.2.2	指针变量的引用	195
7.2.3	指针变量作为函数参数	196
7.3	数组与指针	197
7.3.1	指针与一维数组	197
7.3.2	数组名作为函数参数	200
7.3.3	字符串与指针	201
7.3.4	指向多维数组的指针和指针变量	203
7.4	指针数组和指向指针的指针	206
7.4.1	指针数组的概念	206
7.4.2	指向指针的指针	210
7.4.3	main 函数的参数	212
7.5	函数的指针和指向函数的指针变量	213
7.6	返回指针值的函数	214
7.7	复习指南	216
7.7.1	本章主要内容	216
7.7.2	本章学习提示	216
	习题七	218

第 8 章 结构体、共用体与枚举	223
8.0 引例	223
8.1 结构体类型的定义	224
8.2 结构体类型变量的定义	225
8.3 结构体变量成员的引用方法	226
8.4 结构体变量的初始化	226
8.5 结构体数组的定义	228
8.6 结构体指针变量的定义和使用	230
8.6.1 指向结构体变量的指针	230
8.6.2 指向结构体数组的指针	232
8.6.3 结构体指针变量作为函数参数	233
8.7 动态存储分配	236
8.7.1 malloc 函数	237
8.7.2 calloc 函数	237
8.7.3 free 函数	238
8.7.4 realloc 函数	238
8.8 链表	239
8.8.1 链表概述	239
8.8.2 创建并输出单链表	240
8.8.3 单链表的删除和插入	241
8.9 共用体类型	246
8.10 枚举类型	248
8.11 类型定义符 typedef	250
8.12 复习指南	252
8.12.1 本章主要内容	252
8.12.2 本章学习提示	254
习题八	254
第 9 章 位运算	259
9.0 引例	259
9.1 位运算概述	259
9.1.1 位运算	259
9.1.2 位运算符	260
9.2 位运算	261
9.2.1 按位与运算	261
9.2.2 按位或运算	262
9.2.3 按位异或运算	263
9.2.4 求反运算	263
9.2.5 左移运算	264
9.2.6 右移运算	264

9.3	位域	264
9.4	复习指南	266
9.4.1	本章主要内容	266
9.4.2	本章学习提示	267
	习题九	267
第 10 章	文件	270
10.0	引例	270
10.1	C 文件概述	270
10.2	文件指针	271
10.3	文件的打开与关闭	272
10.3.1	文件的打开函数 fopen	272
10.3.2	文件的关闭函数 fclose	274
10.4	文件的读/写	275
10.4.1	字符读/写函数 fgetc 和 fputc	275
10.4.2	字符串读/写函数 fgets 和 fputs	276
10.4.3	数据块读/写函数 fread 和 fwrite	278
10.4.4	格式化读/写函数 fscanf 和 fprintf	281
10.5	文件的随机读/写	282
10.5.1	位置指针复位函数 rewind	283
10.5.2	位置指针随机定位函数 fseek	283
10.5.3	检测当前位置指针的位置函数 ftell	285
10.5.4	文件随机读/写应用举例	285
10.6	文件检测函数	287
10.7	程序举例	288
10.8	复习指南	289
10.8.1	本章主要内容	289
10.8.2	本章学习提示	291
	习题十	291
第 11 章	Visual C++ 6.0 简介	295
11.1	Visual C++ 6.0 概述	295
11.1.1	Visual C++ 6.0 的起源	295
11.1.2	Visual C++ 6.0 的版本	295
11.1.3	Visual C++ 6.0 的新特性	296
11.1.4	Visual C++ 6.0 与 Turbo C 的区别	296
11.2	Visual C++ 6.0 集成开发环境	297
11.2.1	Visual C++ 6.0 的启动	297
11.2.2	Visual C++ 6.0 菜单功能	297
11.2.3	Visual C++ 6.0 工具栏	301
11.2.4	项目和项目工作区	301

11.3 Visual C++ 6.0 创建应用程序	303
11.4 复习指南	306
11.4.1 本章的主要内容	306
11.4.2 本章学习提示	307
习题十一	307
附录 A ASCII 码表	308
附录 B C 语言的关键字	310
附录 C 运算符优先级和结合性	311
附录 D C 语言库函数	313
附录 E C 语言出错信息速查	322
参考文献	326

第 1 章 // C 语言概述

C 语言是国际上广泛流行的计算机编程语言，它既可用于编写系统软件，又可用于编写应用软件，本章先来了解 C 语言的一些基本知识。

1.1 C 语言的演变及特点

1.1.1 C 语言的发展过程

C 语言是 20 世纪 70 年代初美国贝尔（Bell）实验室 Dennis M.Ritchie（里奇）设计的一种程序设计语言，正式发表于 1978 年。它是在一种称为 B 语言的基础上进行重新设计的一种语言，由于是 B 语言的后继语言，故称为 C 语言。1978 年，Brian W.Kernighan 和 Dennis M.Ritchie（合称 K&R）合著了 *The C Programming Language* 一书，称为标准 C，成为后来广泛使用的 C 语言版本的基础，对 C 语言的发展产生了深远的影响。

1.1.2 C 语言版本

1983 年，美国国家标准协会（American National Standards Institute）在标准 C 基础上制定了 C 语言新标准，于 1983 年发表，称为 ANSI C，它比标准 C 有了更大的发展。目前广泛流行的 C 语言版本有 Microsoft C、Turbo C、Quick C 等，其编译系统基本相同，但也略有差异。

1.1.3 C 语言的特点

C 语言是一种通用的程序设计语言，是处于汇编语言和高级语言之间的一种中间型程序设计语言，常被称为“中级语言”。它既具有高级语言面向用户、可读性强、容易编程和维护等特点，又具有汇编语言面向硬件和系统，可以直接访问硬件的功能，在程序运行效率方面可以与汇编语言媲美。C 语言具有以下特点。

1. C 语言吸取了汇编语言的精华

汇编语言是一种面向机器的低级语言，尽管它的编程要比高级语言麻烦得多，但目标程序质量高，运行快，所以在工业自动化控制系统等领域仍然被广泛使用，显现出强大的生命力。

- C 语言提供了对位、字节及地址的操作，使程序可以直接访问硬件。
- C 语言吸取了宏汇编技术中的一些灵活处理方式，提供了宏替换命令 `#define` 和文件包含的预处理命令 `#include`。

- C 语言程序能与汇编语言程序实现无缝连接，可以在 C 语言程序中方便地引用汇编语言程序。
- C 语言编译生成的目标程序代码质量高，执行效率高，运行速度快。与汇编程序生成的目标程序代码执行效率相比，只低 10%~20%，这是其他高级语言所无法达到的。

2. C 语言继承和发扬了高级语言的优势

- 继承了 Pascal 语言具有丰富数据类型的特点，并具有完备的数据结构。
- 吸取了 FORTRAN 语言中模块结构的思想，C 语言中每个函数都是独立的，允许单独进行编译。这有利于大程序的分工协作和调试。
- 允许递归调用，使有些算法的实现简明、清晰。
- 发扬了高级语言面向用户、可读性强、容易编程和维护等特点，使 C 语言易学、易读、易懂、易编程、易维护。
- 具有良好的可移植性，它没有依赖于硬件的输入/输出语句，便于在不同硬件结构的计算机之间移植。

1.2 C 源程序的结构特点

1.2.1 简单的 C 程序介绍

下面的例子是一个最简单的 C 语言程序，一个无任何实际执行动作的程序，既无输入操作，也无输出操作。

```
/*最简单的 C 语言程序*/  
main()  
{  
}
```

程序分析：

上面的程序仅由一个函数体为空的主函数构成，运行时无任何动作。当然，这样的程序只是用来演示说明之用，没有任何实际意义。

程序的第①行是注释内容，并非程序有效的执行代码，C 语言中的注释内容用一对“/*”和“*/”括起来。

程序的第②行是 C 语言主函数的开始，main 是主函数名，这是一个特殊的函数，每个 C 语言程序都必须有且只有一个主函数，它是 C 程序运行的起点。main 后面的“()”是函数的参数部分，可以为空，但括号不能省略，一定要有。

程序的第③行、第④行是对应的一对花括号“{}”，表示函数体的开始和结束，当然函数体也允许为空（在后面的学习中，还会了解到花括号可以用来构成复合语句等）。

1.2.2 C 源程序的结构特点

一个实现某种特定功能的 C 语言程序（一个或多个文件）应包含若干个函数，每个函数又是由若干条语句组成的。同其他的高级语言一样，C 语言的语句用来向计算机系统发出操作指令。

因此，C语言程序的结构形式如图1-1所示。

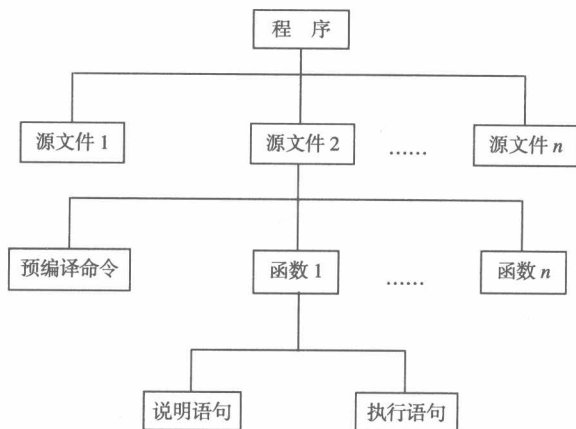


图 1-1 C语言程序结构形式图

说明：

- ① 一个C语言源程序可以由一个或多个源文件组成。
- ② 每个源文件可由一个或多个函数组成。
- ③ 一个源程序不论由多少个文件组成，都有且只能有一个 main 函数，即主函数。
- ④ 源程序中可以有预处理命令（#include 命令仅为其中的一种），预处理命令通常应放在源文件或源程序的最前面。

【例 1.1】编写程序，输出“Welcome!” 欢迎信息。

```

#include <stdio.h>
main()
{
    printf("Welcome!");          /*在屏幕上输出"Welcome!"*/
}
  
```

运行结果：

Welcome!

程序分析：

该程序的主函数体包含了标准的输出函数，实现输出功能。

程序的第一行是一个函数库包含，也称头文件包含，用来指出 C 语言程序调用函数的来源，格式如下：

```
#include <库函数名>
```

程序的第④行是一个标准输出函数调用，完成输出功能。

在 C 语言中，函数分为两类，一类是标准函数，是系统本身提供的库函数，例如，函数 printf，一般情况下要在主函数 main 之前加上相应的函数库包含，指明其来源；另一类是自定义函数，是用户根据自己需要，自行设计的一段程序完成特定功能，这也使 C 语言的模块程序设计思想得到了充分体现。

1.2.3 书写程序时应遵循的规则

【例 1.2】用函数 putchar() 完成字符串输出。

```
#include <stdio.h>
main()
{
    char msg[ ]="You are welcome!";          /*定义一个数组来存放字符串*/
    int i=0;
    clrscr();                                  /*清屏函数，将显示区域清空*/
    while(msg[i])
    {
        putchar(msg[i]);                      /*用 putchar 函数进行字符输出*/
        i++;
    }
}
```

以上程序是一个简单运用 putchar 函数完成字符串输出的程序，该程序严格遵循了程序的书写规则，这样的程序书写清晰，便于阅读、理解和维护。对书写程序的规则归纳如下：

① 一个语句或一个说明占一行，且要以分号结尾，但预处理命令，函数头和花括号“{}”等之后不能加分号。

② 用{}括起来的部分，通常表示了程序的某一层结构。{}一般与该结构语句的第一个字母对齐，并单独占一行。

③ 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写。以便看起来更加清晰，增加程序的可读性。

在编程时应力求遵循这些规则，以养成良好的编程习惯。

1.3 数据输入和输出的常用方法

程序一般都与外部有数据交换，这就涉及在程序中数据的输入和输出问题。在 C 语言中输入/输出是以标准函数形式提供的，通常源程序在开头部分要包含#include <stdio.h>这一行。在这一节中先简单介绍 C 语言中的两个常用的格式化输入/输出函数，即 printf 函数和 scanf 函数，C 语言中其余的输入/输出函数将在后面的章节进行更详细的介绍。

1.3.1 格式化输出函数 printf

格式：

printf(控制字符串, 参数 1, 参数 2, ..., 参数 n);

功能：按照控制字符串格式，将参数进行转换，然后在标准输出设备上输出。

控制字符串中有两种字符，一种是普通字符，将按原样输出来；另一种是格式符，C 语言中规定以“%”开头紧跟格式字符，最常用的格式字符如下：

%d——将参数按十进制形式输出。

%c——将参数看做单个字符输出。

%f——将参数按浮点数形式输出。

%s——将参数以字符串输出（空格为终止符）。

【例 1.3】用%d 格式符进行整型十进制数形式的输出。

```
...
a=12;
b=21;
printf("a=%d b=%d \n",a,b);          /*将 a、b 按十进制整数形式输出*/
...
```

在 printf 语句中“a=”及“b=”和回车换行“\n”都是普通字符，应按原样输出；两个格式符 %d 依次说明 a、b 应按十进制整数形式输出。因此该程序最终的输出结果如下：

```
a=12 b=21
```

1.3.2 格式化输入函数 scanf

格式：

```
scanf(控制字符串, 参数 1, 参数 2, ..., 参数 n);
```

功能：实现从标准输入设备（通常指键盘）上按规定格式输入数值或字符，并将输入内容存放在参数所指定的存储单元中。

例如：

```
...
scanf("%d %d",&a,&b);
/*在 printf 中参数是指要输出值的变量名，而 scanf 中参数是指要接受数据的变量存储单元地址，
  所以变量名前要加上“&”*/
...
```

该语句表示从键盘输入两个十进制整数，分别赋给变量 a 和 b，&a、&b 表示变量 a、b 的地址，输入的两个十进制整数中间用空格隔开。

1.4 C 语言的字符集与 C 语言词汇

日常口头交流时所使用的语言，比如英语，是由英文字符和基本词汇构成的语言体系，作为与计算机进行交流的语言，C 语言也有着类似的语法结构。这一节将介绍 C 语言的字符集与 C 语言词汇。

1.4.1 C 语言的字符集

C 语言的字符集由字母、数字、空格、标点和一些特殊字符组成。在某些情况下还可以使用汉字或其他可表示的图形符号。

1. 字母

小写字母 a~z 共 26 个，大写字母 A~Z 共 26 个。

2. 数字

0~9 共 10 个。

3. 空白符

空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中有意义。在其他地方出现时，只起到间隔作用，编译程序会对它们忽略不计。因此，在程序中使用空白符与否，对程序的编译不发生影响，但在程序中适当的地方使用空白符进行间隔将增加程序的清晰度和可读性。