



21世纪全国高职高专计算机类专业通用教材

C语言程序设计

主编 王艳娟 张倩 隆岩



山东科学技术出版社
www.lkj.com.cn

21 世纪全国高职高专计算机类专业通用教材

C 语言程序设计

主编 王艳娟 张 倩 隆 岩
主审 许文宪

 山东科学技术出版社

图书在版编目 (CIP) 数据

C 语言程序设计/王艳娟, 张倩, 隆岩主编. —济南:
山东科学技术出版社, 2015

21 世纪全国高职高专计算机类专业通用教材

ISBN 978 - 7 - 5331 - 7642 - 6

I . ①C … II . ①王 … ②张 … ③隆 … III . ①C 语
言—程序设计—高等职业教育—教材 IV . ①TP312

中国版本图书馆 CIP 数据核字(2014) 第 289932 号

21 世纪全国高职高专计算机类专业通用教材

C 语言程序设计

主编 王艳娟 张 倩 隆 岩

出版者: 山东科学技术出版社

地址: 济南市玉函路 16 号

邮编: 250002 电话: (0531) 82098082

网址: www.lkj.com.cn

电子邮件: sdkj@sdpress.com.cn

发行者: 山东科学技术出版社

地址: 济南市玉函路 16 号

邮编: 250002 电话: (0531) 82098071

印刷者: 山东新华印务有限责任公司

地址: 济南市世纪大道 2366 号

邮编: 250104 电话: (0531) 82079112

开本: 787mm × 1092mm 1/16

印张: 21.25

版次: 2015 年 1 月第 1 版第 1 次印刷

ISBN 978 - 7 - 5331 - 7642 - 6

定价: 39.00 元

主 审 许文宪

主 编 王艳娟 张 倩 隆 岩

副主编 张 婵 崔 敏

内容提要

本书是为以 C 语言作为程序设计入门课程的读者编写的教材,注重教材的可读性和适用性,全书共 8 大模块: C 语言基础知识、控制语句、数组、函数、编译预处理、指针、结构体以及文件。

每一模块开头给出了“学习目标”,使初学者能提前了解所需掌握的知识要点。在“具体知识点讲解”中对关键知识点进行详细说明,同时配以丰富的、典型的实例题目,帮助读者消化知识点;每一个“实例题目”均从初学者的角度详细分析了程序的设计、实现过程,用以培养初学者的编程思维,分析问题和解决问题的能力;在“关键技术总结中”指出了在编程中需要注意的问题;“能力拓展”中采用典型样例程序,由浅入深,强化知识点、算法、编程方法和技巧;在每一模块末的“小结”中总结了该模块的关键知识点;“课后习题”包含丰富的选择题、分析程序题目、实训题目,在提高学生编程能力的同时,注重理论知识的掌握;“综合实例”由易到难,循序渐进,着重培养读者的程序设计能力及良好的程序设计风格 and 习惯。

本书可以作为高等学校大学本科、高职高专学生《C 语言程序设计》课程教学用书。



前 言

C 语言是国内外广泛流行的程序设计语言,它功能强大,具有丰富的数据类型和运算符,使用灵活,既具有高级语言的优点,又兼具低级语言的许多特点,通用性强,已成为软件开发中的主流。

C 语言不仅适用于系统软件的设计,还适用于应用软件的设计。目前,许多软件开发工具如: Visual C++、Visual C++.NET、JAVA 等都遵循着标准的 C 语言的基本语法,许多嵌入式的系统均来用 C 语言来开发。C 语言是程序开发人员必须掌握的程序设计语言,是进一步学习其他高级语言和应用计算机的基础。

由于 C 语言语法规则多,知识点零散、使用灵活、容易出错,不少初学者学习起来感到困难,希望有一本易于入门、易于学习、实用性强的书。济南职业学院计算机系于 2012 年成立“C 语言标准化建设”课程组,组织 C 语言课程教师、后续课程教师、课程专家等开展《C 语言程序设计》课程标准化建设工作。课程组秉承“理论够用、重在实践、例题典型”的原则,系统整理知识点,将 C 语言精简为 C 语言基础知识、C 语言控制结构、数组、函数、编译预处理、指针、结构体与共用体、文件这八大知识模块。在知识模块后为了固化学生的编程能力和编程思想实现了两个综合实例,系统的介绍了小型程序的设计、实现流程。

每一模块包含“学习目标”“具体知识点讲解”“实例题目”“技术关键点总结”“能力拓展”“小结”“课后习题”等内容。“学习目标”部分明确学习任务,给出每一模块应该掌握的内容以及达到的目的,以便读者进行针对性学习;“具体知识点讲解”部分对关键知识点进行详细介绍,并配以丰富的图表进行说明,直观感受,易于理解;“实例题目”部分采用典型例题,从初学者的角度详细分析了程序的设计、实现过程,用以培养初学者的编程思维,分析问题和解决问题的能力;“技术关键点总结”部分给出初学者在编程初期容易忽视以及需要注意的问题;“能力拓展”部分采用由浅入深的样例程序,强化知识点、算



法、编程方法和技巧。“小结”部分归纳本模块的要点以及重点;“课后习题”部分包含有丰富的“选择题”、“分析程序运行结果”、“实训题目”,“提高题目”,在强化学生编程能力的同时,注重理论知识的掌握。

本书特色:

1. “案例与任务”驱动

首先明确每模块的教学目标,针对性强,从实例入手,由浅入深的讲解 C 语言程序设计的内容。

2. 内容整合,例题典型

传统的 C 语言教材分为 11 章节,很多知识点分散在各个章节,对于初学者来讲不易于更加系统的学习,为此本书对其进行了整理,将相关的知识点放在同一模块中。同时在每一模块中依据课题组的讨论以及后续课程的需要,采用丰富的典型例题,实现了与后续课程的平稳过渡。

3. “递进式”的讲解思路,设计独特

本书采取由浅入深的递进式讲解思路,每一模块中详实的介绍关键知识点,采用典型的案例题目,注重程序设计方法和思路的讲解,使读者在掌握必备理论知识的同时,又培养了读者良好的编程思想。

2

本书是济南职业学院 2013 年教学改革项目的研究成果,凝聚了课题组教师多年的教学、科研的经验与体会,适合作为高职高专院校的程序设计课程的教材,也适合计算机爱好者自学和参考用书。本书由许文宪主审,王艳娟、张倩、隆岩担任主编,张婵、崔敏担任副主编。其中王艳娟编写第一模块、第二模块、第六模块、综合实例 1 以及附录,张倩编写第四模块、第五模块、第七模块,隆岩编写第三模块、第八模块,崔敏和张婵编写综合实例 2。

由于编者水平有限,书中难免有不少缺点和不足,热切期望得到专家和读者的批评指正。

编 者

2014 年 10 月



目 录

模块一 C 语言基础知识	1
1.1 C 语言的构成	2
1.1.1 简单 C 程序举例	2
1.1.2 C 语言程序的结构	5
1.1.3 C 语言程序的特点	5
1.1.4 算法的描述	6
1.2 C 语言数据表现形式	9
1.2.1 常量	9
1.2.2 变量	13
1.3 C 语言的基本数据类型	13
1.3.1 整型数据	14
1.3.2 字符型数据(char)	17
1.3.3 实型数据	19
1.4 C 语言的常用运算符和表达式	20
1.4.1 算术运算符	21
1.4.2 赋值运算符	25
1.4.3 关系运算符	28
1.4.4 逻辑运算符	30
1.4.5 条件运算符	33
1.4.6 逗号运算符	35
1.4.7 不同类型数据间的混合运算	36
1.4.8 运算符小结	37
1.5 数据输入与输出	39
1.5.1 数据输入输出基本概念	39
1.5.2 字符数据的输入输出函数	39
1.5.3 格式输入与输出	41
模块二 C 语言控制结构	53
2.1 结构化程序设计的三种基本结构	53
2.1.1 顺序结构	53
2.1.2 顺序结构程序举例	54



2.1.3 能力拓展	55
2.2 选择结构	57
2.3 if、switch 选择语句	58
2.3.1 单分支选择语句	58
2.3.2 双分支选择语句	61
2.3.3 if 语句的嵌套	64
2.3.4 多分支选择结构	67
2.3.5 选择结构能力拓展	73
2.4 循环结构	78
2.5 while 语句	79
2.6 do while 语句	83
2.7 for 语句	86
2.8 跳转语句	91
2.8.1 break 语句	92
2.8.2 continue 语句	93
2.9 循环嵌套	94
2.10 循环结构能力拓展	96
模块三 数组	107
3.1 一维数组	108
3.1.1 一维数组的定义	108
3.1.2 一维数组元素的引用	109
3.1.3 一维数组的初始化	111
3.1.4 一维数组实例题目	112
3.2 二维数组	118
3.2.1 二维数组的定义	118
3.2.2 二维数组元素的引用	119
3.2.3 二维数组元素的初始化	119
3.2.4 实例题目	121
3.3 字符数组	124
3.3.1 字符数组的定义	124
3.3.2 字符数组的初始化	124
3.3.3 字符数组的引用	125
3.3.4 字符串和字符串结束标志	126
3.3.5 字符数组的输入输出	126
3.3.6 字符串处理函数	129
3.3.7 实例题目	133
3.4 能力拓展	135



模块四 函数	143
4.1 定义函数	143
4.1.1 无参函数的定义	144
4.1.2 有参函数的定义	146
4.2 调用函数	147
4.2.1 函数的调用	147
4.2.2 形式参数和实际参数	149
4.2.3 函数的返回值	151
4.2.4 被调用函数的声明	153
4.3 数组作为函数参数	156
4.3.1 数组元素作函数实参	156
4.3.2 数组名作为函数参数	157
4.4 函数的嵌套调用	160
4.5 函数的递归调用	161
4.6 局部变量和全局变量	164
4.6.1 局部变量	164
4.6.2 全局变量	166
4.7 变量的存储类别	168
4.7.1 动态存储方式与静态存储方式	168
4.7.2 局部变量的存储类别	168
模块五 预处理命令	176
5.1 宏定义	176
5.1.1 无参宏定义	177
5.1.2 有参宏定义	179
5.2 文件包含	181
5.3 条件编译	182
模块六 指针	186
6.1 指针变量的定义与使用	186
6.1.1 指针变量的定义	187
6.1.2 指针变量的引用	187
6.1.3 实例题目	188
6.2 指针变量作函数参数	192
6.3 指针引用一维数组	195
6.3.1 指向数组元素的指针	195
6.3.2 通过指针引用数组元素	196



6.3.3 实例题目	197
6.3.4 数组与指针变量的运算	201
6.4 数组名做函数参数	202
6.4.1 数组元素做函数参数	202
6.4.2 数组名做函数参数	203
6.5 指针引用多维数组	209
6.5.1 指针指向二维数组	209
6.5.2 二维数组的行指针和列指针	211
6.5.3 实例题目	212
6.5.4 指向由 m 个元素组成的一维数组的指针变量(数组指针)	213
6.6 指针引用字符串	214
6.7 指向函数的指针	219
6.8 指针数组和指向指针的指针	222
6.8.1 指针数组	222
6.8.2 指向指针的指针	223
6.9 能力拓展	225
模块七 结构体和共用体	238
7.1 结构体	238
7.1.1 结构体的定义	238
7.1.2 定义结构类型变量	239
7.1.3 结构体变量成员的引用方法	241
7.1.4 结构体变量的初始化	242
7.2 结构体数组	244
7.3 结构体指针变量的说明和引用	249
7.3.1 指向结构体变量的指针	249
7.3.2 指向结构体数组的指针	250
7.3.3 结构体指针变量作函数参数	251
7.4 用指针处理链表	252
7.4.1 链表的定义	252
7.4.2 单链表的基本操作	253
7.5 共用体	262
7.5.1 共用体的概念	262
7.5.2 共用体变量的赋值和使用	263
7.6 枚举类型	265
7.6.1 枚举的概念	265
7.6.2 枚举类型变量的赋值和使用	266



模块八 文件	271
8.1 文件的分类	271
8.2 文件的打开与关闭	272
8.2.1 文件指针	272
8.2.2 文件的打开 [fopen() 函数]	272
8.2.3 文件的关闭 [fclose() 函数]	274
8.3 字符读写 fgetc() 函数和 fputc() 函数	275
8.3.1 读字符 fgetc() 函数	275
8.3.2 写字符 fputc() 函数	276
8.4 字符串读写 fgets() 函数和 fputs() 函数	278
8.4.1 读字符串 fgets() 函数	278
8.4.2 写字符串 fputs() 函数	279
8.5 数据块读写 fread() 函数和 fwrite() 函数	280
8.5.1 写数据块 fwrite() 函数	280
8.5.2 读数据块 fread() 函数	281
8.6 格式化读写 fscanf() 函数和 fprintf() 函数	283
8.6.1 格式化写 fprintf() 函数	283
8.6.2 格式化读 fscanf() 函数	284
8.7 文件的随机读写	285
8.7.1 文件定位	285
8.7.2 文件的随机读写	286
8.8 文件检测函数	287
 综合实例一 模拟银行存取款业务	292
综合实例二 工程师管理系统	296
附录 A C 程序调试与运行	313
附录 B ASCII 码值对照表	320
附录 C 运算符和结合性	322
附录 D 位运算	324
参考文献	327



模块一 C语言基础知识

◆ 学习目标

通过本模块的学习,读者将能够

- ✓ 了解 C 程序的构成
- ✓ 设计简单的 C 程序
- ✓ 掌握 C 语言中常量和变量的概念
- ✓ 了解 C 语言的基本数据类型
- ✓ 掌握各种数据之间的转换
- ✓ 掌握 C 语言中的运算符运算规则及其表达式
- ✓ 掌握数据的输入/输出函数

人们相互交流需要语言,人和计算机交流信息,也需要“语言”,为此需要创造一种计算机和人能够识别的“语言”,这就是计算机语言。计算机语言经历了以下几个发展阶段。

机器语言 计算机的工作是基于二进制的,从根本上来说,计算机只能识别和接受由 0 和 1 组成的计算机指令。这种计算机能直接识别和接受的二进制代码称为机器指令,而机器指令的集合就是该计算机的机器语言。机器语言与人们语言习惯差别太大,难学、难记、难写、难检查、难修改,因此难以推广和使用。

符号语言 为了克服机器语言的缺点,人们创造出了符号语言。它是用一些英文字母和数字来表示一个指令。例如:ADD 代表“加”,SUB 代表“减”。但是计算机并不能直接识别和执行符号语言的指令,需要用一种称为汇编语言的软件,把符号语言的指令转换成机器语言。为此,符号语言又称为**符号汇编语言**或者是**汇编语言**。虽然汇编语言比机器语言较接近于人们的语言习惯,但是也很难普及,只在专业人员中使用,而且不同型号计算机的机器语言与汇编语言是互不通用的,机器语言和汇编语言是完全依赖于具体机器特性的面向机器的语言,称为**计算机低级语言**。

高级语言 为了克服低级语言的缺点,20 世纪 50 年代创造出了第一个计算机高级语言——FORTRAN 语言,它很接近于人们习惯使用的自然语言和数学语言。这种语言功能很强,且不依赖于单一机器,用它写出的程序对任何型号的计算机都适用,称为**高级语言**。C 语言是国际上广泛流行的一种计算机高级语言。

C 语言是在 20 世纪 70 年代初问世的,1978 年由美国电话电报公司(AT&T)贝尔实



验室正式发表了 C 语言。同时由 B. W. Kernighan 和 D. M. Ritchie 合著了著名的《The C Programming Language》一书,通常简称为《K&R》,也有人称之为《K&R》标准。美国国家标准协会(American National Standards Institute)在此基础上制定了一个 C 语言标准,于 1983 年发表,通常称之为 ANSI C,目前已成为计算机程序设计的主流语言。

1.1 C 语言的构成

1.1.1 简单 C 程序举例

下面通过 3 个典型的 C 程序来介绍 C 语言的构成。

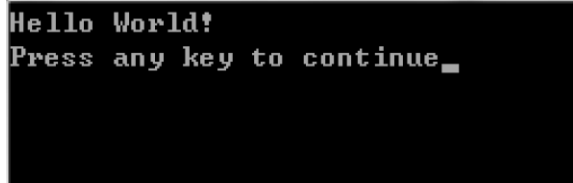
【例 1.1】要求屏幕上输出下面一行信息:

Hello World!

程序实现:

```
#include<stdio.h>    //这是编译预处理命令
void main()           //定义主函数
{                     //函数开始的标志
printf("Hello World! \n"); //输出 Hello World! 并运用"\n"实现换行
}                     //函数结束的标志
```

程序运行结果:



以上运行结果是在 Visual C++ 6.0 环境下运行程序时屏幕上得到的显示结果。其中第 1 行是程序运行时输出的结果,第 2 行是 Visual C++ 6.0 系统在输出完运行结果后自动输出的一行信息,意思是“如果想继续进行下一步,请按任意键”。当用户按任意键后,屏幕上不再显示运行结果,而是返回程序窗口,以便进行下一步操作。

程序分析:C 语言中不提供输入函数和输出函数,为此在使用库函数进行基本的输入输出时,编译系统要求程序提供有关此函数的信息(如对这些输入输出函数的声明、宏定义、全局变量的定义等),程序第 1 行“#include<stdio.h>”的作用就是提供这些信息的。“stdio.h”是系统提供的一个文件名,stdio 是“Standard Input and Output”的缩写,而文件后缀“.h”是头文件“header file”的缩写。“stdio.h”是编译系统中提供的标准输入输出头文件,当用户用到标准输入输出函数时,就要调用这个头文件。

程序各行的右侧,如果有“//”,则表示从此到本行结束是程序的注释,注释中的文本不会被编译系统编译,不产生任何目标代码,注释对运行不起任何作用。注释是给其他人看的,而不是让计算机执行的。C 语言有两种注释方式:



(1)以“//”开始的单行注释,如上程序中的注释,这种注释可以独占一行,也可以出现在一行中其他内容的右侧。该注释的范围从“//”开始,以换行符结束,不能跨行。

(2)以“/*”开始到“*/”结束的多行注释(也称为块注释),这种注释包含有多行,编译系统在发现一个“/*”后,会开始找注释结束符“*/”,把两者之间的全部内容作为注释。

程序第2行中 main 是函数的名字,表示“主函数”,main 前面加上 void 表示该函数无返回值。

程序第3行“{”与第5行“}”相对应,“{ }”是构成函数的基本部分,其作用是将函数括起来,而且是成对出现的。

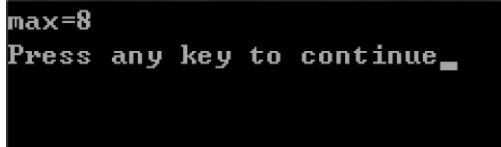
程序第4行“printf(“Hello World! \n”);”是一个输出语句,printf()函数是C编译系统提供的库函数中的输出函数。作用是将 printf()函数中双撇号内的字符串“Hello World!”按照原样输出,而“\n”是转义字符,其基本含义是实现换行,即在输出“Hello World!”之后,显示屏上的光标位置向下移动到下一行的开头。

每条语句最后都有一个分号,表示语句结束。

【例 1.2】求两个数的最大数。

```
#include<stdio. h>      //这是编译预处理命令
void main()              //定义主函数
{                          //函数开始的标志
int a=5,b=8,max;         //程序的声明部分,定义 a,b,max 为整型变量
if(a>b) max=a;           //若 a>b 成立,将 a 的值赋予 max
else max=b;              //否则(即 a>b 不成立),将 b 的值赋予 max
printf("max=%d\n",max); //按照"max=%d\n"格式输出结果
}                          //函数结束的标志
```

程序运行结果:



```
max=8
Press any key to continue_
```

程序分析:本程序的作用是求 a 和 b 两个整数中的最大值,并将最大值赋予变量 max。程序第4行是函数中的声明部分,C 程序中要求在函数中所运用到的变量要先声明后使用,为此在函数的开头就是变量的声明部分,这里定义 a、b、max 为整型变量,同时 a 的初始值为 5,b 的初始值为 8。

程序第5行是条件判断语句,含义是如果满足“a>b”,则将 a 的值赋给变量 max。

程序第6行也是条件判断语句,它与程序第5行相对应,含义:反之(即 a>b 不成立),则将 b 的值赋给变量 max。

程序第7行“printf(“max=%d\n”,max);”是程序的输出语句,按照字符串输出格式将 max 的值进行输出,在输出时将“%d”的位置用 max 来取代,“\n”是换行符。

【例 1.3】定义求两个整数和的函数 sum,并在主函数中进行测试。



```
#include<stdio.h> //定义 sum 函数,函数值为整型,形式参数 x 和 y 为整型
int sum(int x,int y) //定义 sum 函数
{
    int isum; //定义变量 isum
    isum=x+y; //isum 存放 x,y 两个数的和
    return isum; //将 isum 的值作为 sum 的返回值,返回调用 sum 函数的位置
}

void main() //定义主函数
{
    int a,b,c; //定义整型变量 a,b,c
    scanf("%d,%d",&a,&b); //输入变量 a 和 b 的值,输入时用","分隔
    c=sum(a,b); //调用 sum 函数,并将函数的值赋给 c
    printf("sum=%d\n",c); //按照格式输出两个数的和
}
```

程序运行结果:

```
5,8
sum=13
Press any key to continue_
```

4

程序分析:本程序包含两个函数:主函数 main()和被调用的函数 sum()。

程序第 3 行是自定义 sum()函数的首部,其主要组成部分有:函数返回值类型、函数名、圆括号、参数列表。

sum 函数的作用是将形式参数 x、y 进行求和运算,将两者的和赋给变量 isum,程序第 7 行 return 语句将 isum 的值作为 sum()函数的返回值,返回给调用 sum()函数的主函数 main(),返回值是通过函数名 sum 带回到 main()函数中。

程序第 11 行是定义 main()函数中的变量 a、b、c,第 12 行 scanf 是输入函数的名字,该函数的作用是通过终端设备(通常是键盘)输入变量 a 和 b 的值。scanf()函数包含两部分内容:一是双撇号中的内容,它制定输入的数据按照什么格式输入,“%d”的含义是十进制整数形式;二是输入的数据项,即数据放到哪里,赋给哪个变量,为此变量名称前面要加上“&”。在 C 语言中“&”是地址符,“&a”表示“变量 a 的地址”。执行 scanf()函数,从键盘输入两个整数,送到变量 a 和 b 的地址处。

程序第 13 行“c=sum(a,b);”调用 sum()函数,在调用时将 a 和 b 作为 sum()函数的参数(实际参数),实参 a 和 b 分别将值传递给 sum()函数中的形式参数 x 和 y(x 的值是 a,y 的值是 b),然后执行 sum 函数的函数体(程序的第 5~7 行),使 sum()函数得到返回值,并将返回值带到程序中的第 14 行“=”右侧,取代 sum(a,b),然后把返回值赋给变量 c。

程序的第 14 行按照输出格式输出结果。



1.1.2 C 语言程序的结构

通过以上几个程序举例,可以看到一个 C 语言程序的结构有以下特点:

(1)一个完整的 C 程序是由一个或者多个源程序文件组成。一个规模比较小的程序,往往只包括一个源程序文件,如例 1.1、例 1.2 以及例 1.3 均是一个源程序文件,其中例 1.1、例 1.2 的源文件中只有一个函数[main()函数],例 1.3 源文件中有两个函数[main()函数和 sum()函数]。

(2)函数是构成 C 程序的基本单位。程序中几乎全部的工作均是由函数完成的,函数是构成 C 程序的基本单位,一个 C 语言程序是由一个或者多个函数组成的,其中必须包含一个 main()函数[有且只能有一个 main()函数]。

(3)函数包含两部分:函数首部和函数体。函数的第一行(包括函数名,函数类型,函数参数名和参数类型)即为函数首部。函数体是程序中“{ }”内的部分。

例如,例 1.3 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号,括号内写函数的参数类型以及参数名。

如果函数没有参数,则圆括号也不能省略,是一对空括号,如:main()。

(4)程序总是从 main()函数开始执行的。不论 main()函数在整个程序中的位置如何[main()函数可以放在程序的最前,也可以放在程序的最后,或者在一些函数之前,另一些函数之后],C 程序总是以 main()函数的“{”开始,以“}”结束。

(5)程序中对计算机的操作是由函数中的语句完成的。语句是构成函数的基本单位,一个函数通常包含一条或者多条语句。C 程序的书写格式比较自由,一行内可以写几条语句,一条语句可以分在多行上,但为了条理清晰,习惯是一行写一条语句。

(6)在每个数据声明和语句的最后必须有一个分号。分号是构成 C 程序语句的基本单位,只有一个“;”的语句称之为空语句。

(7)C 程序本身并不提供输入输出语句,输入输出的操作是由库函数 scanf()函数和 printf()函数等来完成的,需要进行编译预处理 #include<stdio.h>。

(8)在 C 程序中应当包含必要的注释。一个好的、有使用价值的程序都应当加上注释,以增加程序的可读性,“//”是单行注释,也可以用“/*……*/”实现多行注释(也称为块注释)。

1.1.3 C 语言程序的特点

(1)C 语言简洁、紧凑,使用起来方便灵活。

(2)C 语言具有丰富的数据类型和运算符。C 语言提供的数据类型包括:整型、实型、字符型、数组类型、指针类型、结构体类型和共用体类型等。C 语言包含有多种运算符(详