

高 等 学 校 计 算 机 课 程 规 划 教 材

C语言程序设计

李爱玲 主 编
钟家民 罗 颖 副主编
姬秀荔 主 审



清华大学出版社



高等学校计算机课程规划教材

C语言程序设计

李爱玲 主 编
钟家民 罗 颖 副主编

清华大学出版社
北京

内 容 简 介

本书对知识点的讲解由浅入深,强调算法设计,突出编程思路,注重实例讲解和对学生动手能力的培养。

本书共分13章,内容主要包括C语言概述,数据类型、运算符、表达式,顺序结构程序设计,选择结构程序设计,循环结构程序设计,数组、函数、预处理,指针,结构体、枚举类型与共用体,位运算,文件操作,综合实例程序设计。

本书适合作为高校计算机程序设计基础教材,也适合作为社会各类人士的自学参考书。本书配有《C语言程序设计实验指导》(ISBN 978-7-302-27934-1)供读者学习使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C语言程序设计/李爱玲主编. —北京:清华大学出版社,2012.2

(高等学校计算机课程规划教材)

ISBN 978-7-302-27544-2

I. ①C… II. ①李… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第246975号

责任编辑:汪汉友

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦A座

邮 编: 100084

社总机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者: 北京富博印刷有限公司

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 20.5

字 数: 498千字

版 次: 2012年2月第1版

印 次: 2012年2月第1次印刷

印 数: 1~4000

定 价: 29.50元

产品编号: 041161-01

出版说明

信息时代早已显现其诱人魅力,当前几乎每个人随身都携有多个媒体、信息和通信设备,享受其带来的快乐和便宜。

我国高等教育早已进入大众化教育时代。而且计算机技术发展很快,知识更新速度也在快速增长,社会对计算机专业学生的专业能力要求也在不断翻新。这就使得我国目前的计算机教育面临严峻挑战。我们必须更新教育观念——弱化知识培养目的,强化对学生兴趣的培养,加强培养学生理论学习、快速学习的能力,强调培养学生的实践能力、动手能力、研究能力和创新能力。

教育观念的更新,必然伴随教材的更新。一流的计算机人才需要一流的名师指导,而一流的名师需要精品教材的辅助,而精品教材也将有助于催生更多一流名师。名师们在长期的一线教学改革实践中,总结出了一整套面向学生的独特的教法、经验、教学内容等。本套丛书的目的就是推广他们的经验,并促使广大教育工作者更新教育观念。

在教育部相关教学指导委员会专家的帮助和指导下,在各大学计算机院系领导的协助下,清华大学出版社规划并出版了本系列教材,以满足计算机课程群建设和课程教学的需要,并将各重点大学的优势专业学科的教育优势充分发挥出来。

本系列教材行文注重趣味性,立足课程改革和教材创新,广纳全国高校计算机优秀一线专业名师参与,从中精选出佳作予以出版。

本系列教材具有以下特点。

1. 有的放矢

针对计算机专业学生并站在计算机课程群建设、技术市场需求、创新人才培养的高度,规划相关课程群内各门课程的教学关系,以达到教学内容互相衔接、补充、相互贯穿和相互促进的目的。各门课程功能定位明确,并去掉课程中相互重复的部分,使学生既能够掌握这些课程的实质部分,又能节约一些课时,为开设社会需求的新技术课程准备条件。

2. 内容趣味性强

按照教学需求组织教学材料,注重教学内容的趣味性,在培养学习观念、学习兴趣的同时,注重创新教育,加强“创新思维”,“创新能力”的培养、训练;强调实践,案例选题注重实际和兴趣度,大部分课程各模块的内容分为基本、加深和拓宽内容3个层次。

3. 名师精品多

广罗名师参与,对于名师精品,予以重点扶持,教辅、教参、教案、PPT、实验大纲和实验指导等配套齐全,资源丰富。同一门课程,不同名师分出多个版本,方便选用。

4. 一线教师亲力

专家咨询指导,一线教师亲力;内容组织以教学需求为线索;注重理论知识学习,注重学习能力培养,强调案例分析,注重工程技术能力锻炼。

经济要发展,国力要增强,教育必须先行。教育要靠教师和教材,因此建立一支高水平的教材编写队伍是社会发展的关键,特希望有志于教材建设的教师能够加入到本团队。通过本系列教材的辐射,培养一批热心为读者奉献的编写教师团队。

清华大学出版社

前 言

“C 语言程序设计”是计算机专业和非计算机专业的一门基础课程。通过本课程的学习,可以使学生更好地了解和应用计算机,培养学生应用计算机独立解决问题的能力,为今后进一步的学习奠定良好的、扎实的计算机语言基础。

本书是学习 C 语言程序设计的基础教材,由教学经验丰富的第一线教师精心组织了教材的内容。对 C 语言的精华部分作了较为详细的介绍;较难的题目给出编程思路;还针对学生学完 C 语言后普遍感觉提高和综合应用难的问题,在最后一章安排了综合实例设计与分析,方便学生对全书内容的综合理解和应用;考虑到 C 语言程序设计是一门实践性比较强的课程,本教材出版有配套实验指导书,方便学生学习使用。

在教材的前两章,介绍了 C 语言的基本概念、各种数据类型;第 3 章~第 5 章介绍了 C 语言的基本程序设计技术、C 语言函数的特点;第 6 章、第 7 章介绍数组及函数的相互调用及变量的特性;第 8 章、第 9 章详细地介绍了指针的特点和灵活性还有预处理的内容;第 10 章~第 12 章介绍了结构体、链表技术、位运算和文件的操作方法;第 13 章是综合实例设计。附录给出了算法知识、常见编译错误信息、常用的库函数等。列举的例题都是作者的精心设计,并全部在 Visual C++ 6.0 环境下调试通过。

本书的讲述深入浅出,配合典型例题,通俗易懂,实用性强,可作为高等院校计算机专业和非计算机专业本、专科学生的 C 语言教材,也可以作为自学者的参考用书。可免费给读者提供由本书作者开发的 C 语言 for Windows 集成实验与学习环境、书中源码和课件。

本书第 1 章~第 3 章由罗颖编写,第 4 章~第 7 章由闫怀平、桑书娟编写,第 8 章~第 10 章由李爱玲编写,第 11 章~第 13 章由钟家民编写,全书由姬秀荔主审。

由于作者水平有限,书中难免会有不足和错误,希望读者和专家提出宝贵意见,以帮助我们将此教材进一步完善。

编 者

2012 年 1 月

目 录

第 1 章 C 语言概述	1
1.1 C 程序设计语言发展	1
1.2 C 语言特点	2
1.3 简单的 C 程序开发	4
1.4 C 语言的运行环境	7
1.5 完整的 C 程序运行过程	14
1.6 算法	18
本章小结	21
习题 1	21
第 2 章 数据类型、运算符和表达式	24
2.1 C 语言的数据类型及作用	24
2.1.1 整型数据	25
2.1.2 字符类型	27
2.1.3 浮点类型	28
2.2 常量、变量和标识符	29
2.2.1 常量	29
2.2.2 变量	30
2.2.3 标识符	32
2.3 运算符与表达式	32
2.3.1 算术运算符	33
2.3.2 关系和逻辑运算符	35
2.3.3 赋值运算符	36
2.3.4 逗号运算符	38
2.3.5 条件运算符	39
2.3.6 sizeof 运算符	40
2.3.7 地址运算符	42
2.3.8 运算符的优先级及结合性	42
2.4 表达式	43
2.5 数据类型的转换	47
2.6 综合举例	49
本章小结	50

习题 2	51
第 3 章 顺序结构程序设计	53
3.1 程序基本结构	53
3.2 输入输出语句	54
3.2.1 scanf() 函数	55
3.2.2 printf() 函数	57
3.2.3 getchar() 函数与 putchar() 函数	60
3.3 综合举例	61
本章小结	65
习题 3	65
第 4 章 选择结构程序设计	70
4.1 if 语句	70
4.1.1 if 选择结构	70
4.1.2 if...else 选择结构	71
4.1.3 if...else...if 选择结构	71
4.1.4 if 语句的嵌套	73
4.2 switch 语句	76
4.3 选择结构程序举例	79
本章小结	84
习题 4	84
第 5 章 循环结构程序设计	89
5.1 while 循环语句	89
5.2 do...while 循环语句	90
5.3 for 循环语句	92
5.4 三种循环语句的比较	94
5.5 循环结构程序举例	94
5.5.1 单重循环应用举例	94
5.5.2 嵌套循环及其应用举例	96
5.5.3 break 语句和 continue 语句	98
5.6 程序排错	102
5.6.1 程序中常见的出错原因	102
5.6.2 程序调试与排错基本方法	103
本章小结	104
习题 5	104

第 6 章 数组	109
6.1 一维数组	109
6.1.1 一维数组的定义	109
6.1.2 一维数组的引用	110
6.1.3 一维数组的初始化	110
6.1.4 一维数组应用举例	111
6.2 二维数组与多维数组	113
6.2.1 二维数组的定义	114
6.2.2 二维数组元素的引用	114
6.2.3 二维数组的初始化	115
6.3 字符数组与字符串	118
6.3.1 字符数组的定义	118
6.3.2 字符数组的初始化	118
6.3.3 字符数组的引用	119
6.3.4 字符串和字符串结束标志	119
6.3.5 字符数组的输入输出	120
6.3.6 字符串处理函数	120
6.4 数组程序设计举例	124
本章小结	127
习题 6	127
第 7 章 函数	130
7.1 函数概述	130
7.2 库函数简介	131
7.3 函数定义	133
7.3.1 无参函数的定义	133
7.3.2 有参函数的定义	133
7.4 函数调用	134
7.4.1 形式参数和实际参数	134
7.4.2 函数的返回值	135
7.4.3 函数的调用	136
7.5 函数的嵌套和递归	139
7.5.1 函数的嵌套	139
7.5.2 函数的递归调用	140
7.6 变量的作用域及存储类型	144
7.6.1 静态、动态	144
7.6.2 变量的作用域	145
7.7 外部、内部函数	147
7.8 应用举例	148

本章小结·····	151
习题 7 ·····	152
第 8 章 预处理 ·····	157
8.1 宏定义 ·····	157
8.1.1 不带参数的宏定义·····	157
8.1.2 带参的宏·····	159
8.2 文件包含 ·····	161
8.3 条件编译 ·····	163
本章小结·····	165
习题 8 ·····	165
第 9 章 指针 ·····	168
9.1 指针概述 ·····	168
9.1.1 指针的概念·····	168
9.1.2 指针变量的定义和使用·····	169
9.2 指针运算符与指针表达式 ·····	171
9.2.1 指针运算符与指针表达式概述·····	171
9.2.2 指针变量作函数的参数·····	173
9.3 指针与数组 ·····	174
9.3.1 指针与一维数组·····	174
9.3.2 指针与二维数组·····	178
9.3.3 数组指针作函数的参数·····	181
9.3.4 指针与字符数组·····	186
9.3.5 指针数组·····	188
9.4 指针与函数 ·····	190
9.4.1 指向函数的指针变量的定义及使用·····	190
9.4.2 用指针类型数据作函数参数·····	191
9.4.3 带参的主函数·····	194
9.4.4 返回指针的函数·····	194
9.5 指向指针的指针 ·····	196
9.6 指针应用举例 ·····	197
本章小结·····	201
习题 9 ·····	202
第 10 章 结构体、联合体与枚举类型 ·····	205
10.1 结构体类型变量的定义和引用·····	205
10.1.1 结构体类型变量的定义·····	206
10.1.2 结构体类型变量的引用·····	206

10.1.3	结构体类型变量的初始化	207
10.2	结构体数组的定义和引用	208
10.3	结构体指针的定义和引用	214
10.3.1	指向结构体类型变量的使用	214
10.3.2	指向结构体类型数组的指针的使用	216
10.4	链表	218
10.4.1	链表的概述	218
10.4.2	链表操作	219
10.5	联合	224
10.5.1	联合体的定义	224
10.5.2	联合体变量的引用	225
10.6	枚举类型	228
10.6.1	枚举类型的定义和枚举变量的说明	228
10.6.2	枚举类型变量的赋值和使用	228
10.7	综合应用举例	230
	本章小结	234
	习题 10	234
第 11 章	位运算	239
11.1	位运算概述	239
11.2	位运算符和位运算	239
11.3	位运算综合举例	243
	本章小结	246
	习题 11	246
第 12 章	文件	249
12.1	C 文件概述	249
12.2	文件的打开与关闭	250
12.2.1	文件类型指针	250
12.2.2	文件的打开函数	250
12.2.3	文件的关闭函数	252
12.3	文件的读写	252
12.3.1	文件读写函数	252
12.3.2	文件读写举例	255
12.4	文件定位	257
12.4.1	文件定位函数——fseek()函数	257
12.4.2	出错检测函数——ferror()函数	259
12.5	文件综合举例	259
	本章小结	263

习题 12	263
第 13 章 综合实例程序设计	266
13.1 程序设计的基本过程.....	266
13.2 综合程序设计实例.....	267
13.2.1 题目的内容要求.....	267
13.2.2 程序的功能设计.....	268
13.2.3 程序的数据设计.....	269
13.2.4 程序的函数设计.....	270
13.2.5 函数编程及调试.....	271
13.2.6 整体调试.....	290
13.2.7 程序维护.....	290
本章小结.....	290
习题 13	290
 附录 A 常用 ASCII 码字符对照表.....	 291
 附录 B 编译错误信息	 293
 附录 C 常用库函数	 306
 参考文献.....	 314

第 1 章 C 语言概述

C 语言是近年来国际上广泛推行的、很有发展前途的计算机高级语言。它适合于作为系统描述语言,既可用于写系统软件,也可用来写应用软件。

1.1 C 程序设计语言发展

以前的操作系统等系统软件主要是用汇编语言编写的(包括 UNIX 操作系统在内)。由于汇编语言依赖于计算机的硬件,程序的可读性和可移植性都比较差。为了提高可读性和可移植性,最好改用高级语言,但是一般的高级语言难以实现汇编语言的某些功能(汇编语言可以直接对硬件进行操作,例如,对内存地址的操作、位操作等)。人们设想能否找到一种既具有一般高级语言特性,又具有低级语言特性的语言,并集它们的优点于一身,于是 C 语言就应运而生了。

C 语言的根源可以追溯到 ALGOL 60。ALGOL 60 由一个国际委员会于 19 世纪 60 年代早期设计的,它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。CPL(Combined Programming Language,混合编程语言)由剑桥和伦敦大学于 1963 年开发而成,比 ALGOL 60 接近硬件一些,但是规模比较大,难以实现。BCPL(Basic Combined Programming Language,基础混合编程语言)由剑桥大学的 Martin Richards 于 1967 年发明。B 语言由贝尔实验室的 Ken Thompson 于 1970 年设计出来,并用 B 语言写了第一个 UNIX 操作系统,在 PDP-7 上实现。1971 年在 PDP-11/20 上实现了 B 语言,并写了操作系统,但是由于 B 语言过于简单,因此功能有限。C 语言由贝尔实验室的 Dennis Ritchie 于 1972 年设计出来。C 语言既保持了 BCPL 和 B 语言的优点(精练,接近硬件),又克服了它们的缺点(过于简单,数据无类型等)。1975 年 UNIX 第 6 版公布后,C 语言的突出优点才引起人们的普遍注意,1978 年以后,C 语言已经先后移植到大、中、小、微型计算机上,已独立于 UNIX 和 PDP。现在 C 语言以其独有的特点已经成为目前应用最广泛的语言之一,并且广泛地用在教学上。

20 世纪 80 年代初期,许多工程项目都令结构化方法到达了极限,为解决这个问题,一种称为面向对象编程的新编程方法诞生了。1979 年当 Bjarne Stroustrup 在新泽西州的 Murray Hill 实验室工作时,发明了 C++,最初称为“带类的 C”,1983 年改名为 C++,它包括了 C 语言所有的特征属性和优点,发明 C++ 不是企图创造一种全新的编程语言,而是对 C 这个高度成功语言的改进。

20 世纪 90 年代初,Sun 公司在以 C 语言为核心的基础上产生了一种称为 Oak 的语言,1994 年 Oak 语言被重新命名为 Java,并重新定位为一种因特网语言,这种语言应能轻便地使用于许多平台。1999 年 Sun 公司在 Java 的基础上开发出 Java 2 Enterprise Edition 作为服务器语言并强力推荐,2000 年 J2EE 开始得到广泛应用。与此同时。微软公布了 .NET,几乎包括了 J2EE 的所有特性,但没有 J2EE 昂贵的价格。.NET 包括了一个新的 C 家族语

言——C#,读做 C-pound。2001 年员工们认为应该将其叫为 C sharp。2002 年 C# 成为 .NET 发行版本的一部分,具有自动清理内存的特色。

C 语言是一种灵活性很强,应用非常广泛的面向过程编程的语言。但是 C++ 是一种多范型程序设计语言,它不仅仅是一种面向对象的语言,同时也是一种面向过程的语言,它是在 C 的基础上发展起来的,引入了类的概念。VC 是 Visual C++ 的简称,是可视化开发语言的一种,和 VB(Visual Basic)、VJ(Visual Java)、VF(Visual Foxpro)相似,VC 提供了 C++ 的集成化开发环境(IDE),可以说 VC 是一种开发工具,本质上是利用 C++ 语言编程的。

Java 是一种跨平台,适合于分布式计算环境的面向对象编程语言。具体来说,它具有如下特性:简单性、面向对象、分布式、解释型、可靠、安全、平台无关、可移植、高性能、多线程、动态性等。

C# 不允许直接内存操作,C# 没有指针,C# 的关键字更加明了;C# 中的每种类型都可以当作是对象,C# 只允许单继承,也没有全局变量和全局常数,所有的一切都必须封装在一个类中,使代码具有更好的可读性,减少了命名冲突的可能;网络服务看起来就像是 C# 的本地对象,C# 组件能够方便地为 Web 服务,并允许它们通过 Internet 被运行在任务操作系统上的任何语言调用;C# 中不能使用未初始化的变量,对象的成员变量由编译器负责置 0,C# 不支持不安全的指向,不能将整数指向引用类型,C# 还提供了边界检查和溢出检查功能;C# 在语言中内置了版本控制功能,保证复杂的软件可以方便地开发和升级。

在对操作系统和系统实用程序以及需要对硬件进行操作的场合,C 语言明显地优越于其他高级语言。C 语言是面向过程的一种结构化语言,且描述能力强,同样适应于教学和其他广泛的应用领域,因此更有生命力。

同步训练:

思考 C 语言较其他诸如 C++、C#、Java 等的最大优点是什么?

1.2 C 语言特点

C 语言之所以能存在和发展,并具有生命力,总是有其不同于(或优于)其他语言的特点。正是因为 C 语言具有强大的功能,许多著名的操作系统软件,如 UNIX、Linux 等都是由它编写的。C 语言的主要特点如下。

(1) C 语言的标识符要求区分大小写。C 语言以英文小写字母为基础,符合人们日常阅读和书写的习惯。同一个单词的大小写代表不同含义的标识符。

(2) 语言简洁、紧凑,使用方便、灵活。C 语言一共只有 32 个关键字,9 种控制语句,程序书写形式自由,压缩了一切不必要的成分。因此 C 程序比较简练,使人易写易懂,源程序短,输入程序时工作量少。

(3) 程序由函数组合而成,因此程序功能结构比较清楚。而且每个函数都是独立的,可以单独编译,对设计一个大型程序来说,有利于分工编译和调试。

(4) 运算符丰富。C 的运算符包含的范围很广泛,共有 34 种运算符。C 把括号、赋值、强制类型转换等都作为运算符处理,从而使 C 的运算类型极其丰富,表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

(5) 数据结构丰富,具有现代化语言的各种数据结构。C 的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。能用来实现各种复杂的数据结构(如链表、树、栈等)的运算,尤其是指针类型数据,使用起来比 Pascal 更为灵活、多样。

(6) 具有结构化的控制语句(如 if...else 语句、while 语句、do...while 语句、switch 语句、for 语句)。用函数作为程序的模块单位,便于实现程序的模块化。C 是理想的结构化语言,符合现代编程风格的要求。

(7) 语法限制不太严格,程序设计自由度大。例如对数组下标越界不做检查,由程序编写者自己保证程序的正确。对变量的类型使用比较灵活,例如,整型量与字符型数据以及逻辑型数据可以通用。

(8) C 语言允许直接访问物理地址,能进行位(bit)操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。因此,C 既具有高级语言的功能,又具有低级语言的许多功能,可用来写系统软件。C 语言的这种双重性,使它既是成功的系统描述语言,又是通用的程序设计语言。有人把 C 语言称为“高级语言中的低级语言”或“中级语言”,意为兼有高级和低级语言的特点。

按此观点可将各语言分类如下。

高级语言:如 BASIC、FORTRAN、COBOL、Pascal、Ada、Modula-2。

中级语言:如 C、FORTH、宏汇编。

低级语言:如汇编语言。

一般仍习惯将 C 语言称为高级语言,因为 C 程序也要通过编译、连接才能得到可执行的目标程序,这是和其他高级语言相同的。

(9) 生成的目标代码质量高,程序执行效率高。一般只比汇编程序生成的目标代码效率低 10%~20%。

(10) 用 C 语言写的程序可移植性好(与汇编语言相比)。基本上不做修改就能用于各种操作系统,如 DOS、Windows、UNIX 等,以及各种型号的计算机。

上面只介绍了 C 语言的最容易理解的一般特点,至于 C 语言内部的其他特点将结合以后各章的内容作介绍。由于 C 语言的这些优点,使 C 语言应用面很广。许多大的软件都用 C 语言编写,这主要是由于 C 语言的可移植性好和硬件控制能力高,表达和运算能力强。许多以前只能用汇编语言处理的问题,现在可以改用 C 语言来处理了。

C 语言也有一定的弱点,如,非强制类型:语法限制不严格,这使得编程者无法过多地依赖 C 编译程序去查错;缺少实时检查:数组越界等。

下面,从应用的角度出发,对 C 语言和其他传统的高级语言作一下简单比较。从掌握语言的难易程度来看,C 语言比其他语言难一些。BASIC 是初学者入门的较好的语言,FORTRAN 也比较好掌握。对科学计算多用 FORTRAN 或 PL/I;对商业和管理等数据处理领域,用 COBOL 为宜。C 语言虽然也可用于科学计算和管理领域,但并不理想,C 语言的特长不在这里。对操作系统和系统实用程序以及需要对硬件进行操作的场合,用 C 语言明显地优越于其他高级语言,有的大型应用软件也用 C 语言编写。从教学角度,由于 Pascal 是世界上第一个结构化语言,曾被认为是计算机专业的比较理想的教学语言,前几年在数据结构等课程的教学一般用 Pascal 语言举例。由于 Pascal 语言难以推广到各实际应用领域,到目前为止基本上只是教学语言。C 语言也是理想的结构化语言,且描述能力强,同样

适于教学。C 语言除了能用于教学外,还有广泛的应用领域,因此更有生命力。Pascal 和其他高级工语言的设计目标是通过严格的语法定义和检查来保证程序和正确性,而 C 语言则是强调灵活性,使程序设计人员能有较大的自由度,以适应宽广的应用面。

总之,C 语言对程序员的要求较高。程序员使用 C 语言编写程序会感到限制少,灵活性强,功能强,可以编写出任何类型的程序。现在,C 语言已不仅仅用来编写系统软件,还可以用来编写应用软件,学习和使用 C 语言的人已越来越多。

同步训练:

思考 C 语言有哪些特点?

1.3 简单的 C 程序开发

下面,从最简单的程序例子来分析 C 语言的程序构成及运行情况。

【例 1-1】 下面的代码是一个完整的可以运行的程序,它的功能是输出一个句子:

```
/* -----  
           example1.c  
----- */  
#include "stdio.h"  
main()  
{  
    printf ("It is a first C program.\n");          /* 输出语句 */  
}
```

运行结果:

It is a first C program.

【例 1-2】 下面程序的功能是求两数和并且输出。

```
/* -----  
           example2.c  
----- */  
#include "stdio.h"  
main()  
{  
    int a,b,sum;          /* 定义变量 */  
    a=234;                /* 赋值语句 */  
    b=567;                /* 赋值语句 */  
    sum=a+b;              /* 求和语句 */  
    printf("sum is %d\n",sum) ; /* 输出语句 */  
}
```

运行结果:

sum is 801

【例 1-3】 下面程序的功能是比较两数的大小,并且输出较大的值。

```

#include<stdio.h>
main()                /* 主函数 */
{
    int a,b,c;          /* 声明部分,定义变量 */
    int max(int x,int y);
    scanf("%d,%d",&a,&b) ; /* 输入变量 a 和 b 的值 */
    c=max(a,b);          /* 调用 max 函数,将得到的值赋给 c */
    printf("max=%d\n",c) ; /* 输出 c 的值 */
}

int max(int x,int y)    /* 定义 max 函数,函数值为整型,形式参数 x,y 为整型 */
{
    int z;              /* max 函数中的声明部分,定义本函数中用到的变量 z 为整型 */
    if(x>y) z=x;         /* 条件判断语句 */
    else z=y;
    return(z) ;          /* 将 z 的值返回,通过 max 带回调用处 */
}

```

运行结果:

```

5 17↵
max=17

```

从上面 3 个 C 程序可以看出: 一个 C 程序是由 3 个部分组成:

- 注释;
- 编译预处理;
- 程序主体。

C 语言的注释为“/*”与“*/”之间的内容,它可以占多行。也可以用“//”之后的内容,但是只能注释一行,换行后即为程序主体内容。

例如:

```

//*****
//this is a simplest program.
//*****

```

注释是程序员为读者做的说明,是提高程序可读性的一种手段。一般可将其分为两种: 序言注释和注解性注释。前者用于程序开头,说明程序或文件的名称、用途、编写时间、编写人以及输入输出说明等,后者用于程序中难懂的地方。

每个以符号“#”开头的行,称为编译预处理行。如 #include 称为文件包含预处理命令。有关内容将在以后的章节中介绍。

#include "stdio.h"的作用是在编译之前将文件"stdio.h"的内容增加(包含)到程序 example1.c 中,以作为其一部分。stdio.h 是系统定义的一个“头文件”,它设置了 C 语言的 I/O 相关环境,定义输入输出函数 printf 和 scanf 等。printf 和 scanf 的使用方法将在后面章节中介绍。

main()表示主函数,每一个 C 语言程序都必须有一个 main()函数。main()作为程序的入口,无论 main 函数处在程序中的哪个位置,程序总是从 main 函数开始。函数体用花括号