

高职高专规划教材



双高规划教材

C语言程序设计

教程

李辉 编



西北工业大学出版社

高职高专规划教材

C 语言程序设计教程

李辉 编

西北工业大学出版社

【内容提要】本书为高职高专规划教材之一。书中主要介绍了C语言程序设计的基本语法知识和用C语言编写程序的基本方法与技术,涵盖教育部考试中心最新全国计算机等级考试二级C语言程序设计考试大纲的主要内容,符合教育部“关于进一步加强高等学校计算机基础教学的意见”(白皮书)中有关C语言程序设计的教学基本要求,同时包括C++语言简介和上机实验指导等。

本书面向初学者,强调应用性。在内容组织上精心设计,由浅入深,循序渐进,所选程序实例通俗易懂,力求做到学习C语言知识与培养C程序设计开发能力的融会贯通。

图书在版编目(CIP)数据

C语言程序设计教程/李辉编. —西安:西北工业大学出版社, 2006.10
ISBN 7-5612-1753-6

I. C… II. 李… III. C语言—程序设计—教材 IV. TP312

中国版本图书馆CIP数据核字(2004)第016400号

出版发行:西北工业大学出版社

通信地址:西安市友谊西路127号 邮编:710072

电 话:029-88493844 88491757

网 址:www.nwpup.com

电子邮件:computer@nwpup.com

印 刷 者:陕西天元印务有限公司

开 本:787 mm×1 092 mm 1/16

印 张:19

字 数:499千字

版 次:2006年10月第1版

2006年10月第1次印刷

定 价:22.00元

前 言

大力发展职业教育，是推进我国工业化、现代化建设的迫切需要，也是完善现代国民教育体系的必然要求。国民经济的各行各业不但需要一大批科学家、工程师和经营管理人才，同时也迫切需要数以千万计的高技能人才和数以亿计的高素质劳动者。

只有培养出大量高素质的劳动者，才能把我国的人数优势转化为人力优势，提高全民族的竞争力。因此，我国近年来十分重视高等职业教育，把高等职业教育作为高等教育的重要组成部分，并以法律的形式加以约束与保证。高等职业教育从此进入了蓬勃发展时期，驶入了高速发展的快车道。

为了满足高职高专教学改革的需要，在编写教材大纲的过程中，根据高等职业教育的特点，广泛收集了高等职业院校的教学计划，对多个省、市高等职业教育的实际情况进行了调研，经过反复讨论和修改，以使编写大纲能最大限度地符合我国高等职业教育的要求，切合高职高专教育的实际。

本书内容选材的依据是：教育部高等学校计算机科学与技术教学指导委员会、非计算机专业计算机基础教学指导分委员会“关于进一步加强高等学校计算机基础教学的意见”（白皮书）中有关 C 语言程序设计教学基本要求和教育部考试中心最新全国计算机等级考试二级 C 语言程序设计考试大纲。本书共分为 13 章，第 1~11 章为 C 语言程序设计的基本内容，第 12 章为 C++ 语言简介，第 13 章为上机实验指导。考虑到教学层次和目标要求的不同，书中对第 9 章指针、第 10 章结构体与共用体、第 11 章文件、第 12 章 C++ 语言简介等内容加上了星号“*”，作为选学内容，在实际的教学组织中可以灵活取舍，既可以作为扩展内容，亦可作为自学内容。不过，目前我国许多行业开始实行劳动准入制度和职业资格制度，为此建议：对准备参加计算机等级考试和软件水平考试的读者，最好能够学完书中的全部内容。顺便推荐一个“集成学习环境（C 语言）”学习软件，里边的知识点总结和程序实例讲解都非常好。另外还有一个“C 语言学习系统”也不错。

C 语言是一种适用于结构化程序设计的高级语言。大多数程序员愿意选择它开发系统软件和应用软件，如著名的 UNIX 操作系统的大部分就是用 C 语言编写的。时下流行的很多新型语言如 C++，Java，C#，J#，等也是由 C 语言发展而来的。因此，C 语言程序设计是计算机基础教学的典型核心课程之一，是全国计算机等级考试二级的考试项目。作为一名老师，我深深体会到学习 C 语言对学生的重要性，衷心期盼本书能对广大读者朋友学习和掌握 C 语言程序设计有所裨益。

本书可作为高等职业院校、高等技术院校、高等专科院校的计算机教材，也可供优秀职高学校选做教材。

由于编者水平有限，书中难免存在不足，恳请各位专家、老师以及广大读者把使用本书的情况及各种意见、建议及时反馈回来，以便今后再版时不断地修改完善。

编 者

目 录

第 1 章 C 语言基础	1
1.1 绪论.....	1
1.1.1 计算机软件.....	1
1.1.2 程序设计语言.....	2
1.2 C 语言概述.....	3
1.2.1 C 语言的发展简史.....	3
1.2.2 C 语言的特点.....	3
1.3 C 语言源程序的基本结构.....	4
1.4 C 程序的编辑、编译、链接和运行.....	8
1.4.1 Turbo C 2.0 集成开发环境简介.....	8
1.4.2 上机调试 C 程序的一般步骤.....	9
本章小结.....	12
习题一.....	12
第 2 章 基本数据类型	14
2.1 C 语言的数据类型.....	14
2.2 标识符与关键字.....	15
2.2.1 标识符.....	15
2.2.2 关键字.....	15
2.3 常量.....	16
2.3.1 数值常量.....	16
2.3.2 字符常量.....	16
2.3.3 符号常量.....	17
2.4 变量.....	18
2.4.1 变量说明.....	18
2.4.2 整型变量.....	19
2.4.3 实型变量.....	21
2.4.4 字符变量.....	22
2.4.5 变量初始化.....	24
本章小结.....	25
习题二.....	25

第 3 章 运算符与表达式	26
3.1 运算符	26
3.1.1 运算符的种类.....	26
3.1.2 运算符的优先级别.....	27
3.1.3 运算符的结合性.....	27
3.2 算术运算符和算术表达式	27
3.2.1 算术运算符.....	27
3.2.2 算术表达式.....	28
3.2.3 算术运算符的优先级和结合性.....	30
3.3 赋值运算符和赋值表达式	31
3.3.1 赋值运算符.....	31
3.3.2 复合赋值运算符.....	31
3.3.3 赋值表达式.....	31
3.4 关系运算符和关系表达式	32
3.4.1 关系运算符.....	32
3.4.2 关系表达式.....	33
3.4.3 关系运算符的优先级和结合性.....	33
3.5 逻辑运算符和逻辑表达式	33
3.5.1 逻辑运算符.....	33
3.5.2 逻辑表达式.....	34
3.5.3 逻辑运算符的优先级和结合性.....	34
3.6 位运算符	35
3.6.1 按位与运算.....	35
3.6.2 按位或运算.....	35
3.6.3 按位异或运算.....	36
3.6.4 按位取反运算.....	36
3.6.5 左移位运算.....	36
3.6.6 右移位运算.....	36
3.7 其他运算符	37
3.7.1 条件运算符.....	37
3.7.2 逗号运算符.....	38
3.7.3 sizeof 运算符.....	39
3.7.4 强制类型转换运算符.....	40
3.8 不同类型数据间的转换	41
本章小结	42
习题三	42

第 4 章 顺序结构程序设计	44
4.1 结构化程序设计.....	44
4.1.1 结构化程序设计概述.....	44
4.1.2 结构化程序设计的基本结构及其特点.....	45
4.2 算法.....	46
4.2.1 算法的基本概念.....	46
4.2.2 算法的特性.....	47
4.2.3 算法的流程图表示法.....	48
4.2.4 基本算法.....	49
4.2.5 算法评价.....	50
4.3 C 语句概述.....	51
4.3.1 表达式语句.....	51
4.3.2 函数调用语句.....	52
4.3.3 空语句.....	52
4.3.4 复合语句.....	53
4.3.5 控制语句.....	53
4.4 数据输出.....	54
4.4.1 printf 函数（格式输出函数）.....	54
4.4.2 putchar 函数（字符输出函数）.....	59
4.5 数据输入.....	60
4.5.1 scanf 函数（格式输入函数）.....	60
4.5.2 getchar 函数（字符输入函数）.....	62
4.6 顺序结构程序设计举例.....	64
本章小结.....	66
习题四.....	67
第 5 章 分支结构程序设计	68
5.1 分支结构概述.....	68
5.2 if 语句.....	69
5.2.1 单分支结构.....	69
5.2.2 双分支结构.....	70
5.2.3 多分支结构.....	71
5.3 if 语句的嵌套.....	72
5.4 switch 语句.....	74
5.5 分支结构程序设计举例.....	79

本章小结.....	83
习题五.....	83
第 6 章 循环结构程序设计.....	85
6.1 用 goto 和 if 语句一起构成循环.....	85
6.2 while 循环结构.....	86
6.3 do···while 循环结构.....	88
6.4 for 循环结构.....	90
6.5 循环的嵌套.....	92
6.6 break 和 continue 语句.....	93
6.6.1 break 语句.....	93
6.6.2 continue 语句.....	94
6.7 循环结构程序设计举例.....	95
本章小结.....	101
习题六.....	101
第 7 章 数组.....	102
7.1 一维数组.....	102
7.1.1 数组的概念.....	102
7.1.2 一维数组的定义和引用.....	102
7.1.3 一维数组的初始化.....	104
7.1.4 一维数组应用举例.....	104
7.2 二维数组.....	107
7.2.1 二维数组的定义和引用.....	107
7.2.2 二维数组的初始化.....	108
7.2.3 二维数组应用举例.....	109
7.3 字符数组与字符串.....	112
7.3.1 字符数组的定义.....	112
7.3.2 字符数组的初始化.....	113
7.3.3 字符串与字符数组.....	114
7.3.4 字符数组的输入与输出.....	115
7.3.5 字符串处理函数.....	115
7.3.6 字符数组应用举例.....	117
本章小结.....	120
习题七.....	120

第 8 章 函数	121
8.1 函数的概念	121
8.2 函数的定义	123
8.2.1 无参函数的定义	123
8.2.2 有参函数的定义	124
8.3 函数参数与返回值	125
8.3.1 函数的参数	125
8.3.2 函数的返回值	126
8.4 函数的声明	127
8.5 函数的调用	130
8.5.1 函数调用的一般格式	130
8.5.2 函数调用的方式	131
8.5.3 函数的传值调用	132
8.5.4 函数的嵌套调用	133
8.5.5 函数的递归调用	136
8.6 变量存储类别	138
8.6.1 局部变量和全局变量	139
8.6.2 auto 变量	142
8.6.3 register 变量	143
8.6.4 static 变量	143
8.6.5 extern 变量	145
8.7 内部函数和外部函数	146
8.7.1 内部函数	147
8.7.2 外部函数	147
8.8 编译预处理命令	149
8.8.1 宏定义	149
8.8.2 #include 命令	153
8.9 函数应用举例	156
本章小结	158
习题八	159
*第 9 章 指针	160
9.1 指针的概念	160
9.2 指针与指针变量	160
9.2.1 指针变量的定义	160
9.2.2 指针变量的引用	161

9.2.3 指针变量作函数的参数	164
9.3 指针与数组	166
9.3.1 指向一维数组的指针	166
9.3.2 指向二维数组的指针	169
9.3.3 数组名作函数的参数	171
9.4 指针与字符串	175
9.4.1 字符串的表示形式与引用	176
9.4.2 字符指针变量与字符数组	177
9.4.3 字符串指针作函数的参数	178
9.5 指针与函数	181
9.5.1 指向函数的指针	181
9.5.2 指向函数的指针作函数的参数	183
9.5.3 返回指针的函数	185
9.6 指针数组与指向指针的指针	186
9.6.1 指针数组	186
9.6.2 指向指针的指针	187
9.7 指针应用举例	189
本章小结	192
习题九	193
*第 10 章 结构体与共用体	194
10.1 结构体	194
10.1.1 概述	194
10.1.2 结构体类型的定义	194
10.1.3 结构体变量的定义	196
10.1.4 结构体成员的引用	197
10.1.5 结构体变量的初始化	198
10.1.6 结构体数组	200
10.1.7 指向结构体变量的指针	203
10.1.8 指向结构体数组的指针	205
10.2 共用体	206
10.2.1 共用体类型的定义	206
10.2.2 共用体变量的定义	206
10.2.3 共用体成员的引用	208
10.3 枚举类型	211
10.3.1 枚举类型的定义	211
10.3.2 枚举元素的引用	212

10.4 用 typedef 定义类型.....	214
本章小结.....	216
习题十.....	217
*第 11 章 文件	218
11.1 文件概述.....	218
11.2 文件类型指针.....	219
11.3 文件的打开与关闭.....	219
11.3.1 文件的打开.....	219
11.3.2 文件的关闭.....	221
11.4 文件的读写.....	222
11.4.1 字符读写函数.....	222
11.4.2 字符串读写函数.....	226
11.4.3 数据块读写函数.....	229
11.4.4 格式化读写函数.....	232
11.5 文件的定位.....	234
11.5.1 rewind 函数.....	234
11.5.2 fseek 函数.....	235
本章小结.....	237
习题十一.....	238
*第 12 章 C++语言简介	239
12.1 面向对象程序设计.....	239
12.1.1 面向对象技术的特点.....	239
12.1.2 面向对象程序设计语言.....	240
12.2 C++语言概述.....	241
12.2.1 从 C 语言到 C++语言.....	241
12.2.2 C++语言的特点.....	242
12.3 类.....	242
12.3.1 类的定义.....	243
12.3.2 数据成员和成员函数.....	244
12.4 对象.....	245
12.4.1 对象的定义格式.....	245
12.4.2 对象成员的引用.....	247
12.4.3 对象赋值语句.....	248
本章小结.....	249

习题十二.....	250
第 13 章 上机实验指导	251
实验 1 熟悉 Turbo C 2.0 集成开发环境.....	251
实验 2 基本数据类型、运算符和表达式.....	252
实验 3 顺序结构程序设计.....	254
实验 4 选择结构程序设计.....	257
实验 5 循环结构程序设计.....	261
实验 6 数组.....	263
实验 7 函数.....	266
实验 8 指针.....	269
实验 9 结构体与共用体.....	272
实验 10 文件.....	273
附录	275
附录 I 常用字符与 ASCII 码对照表.....	275
附录 II ANSIC 规定的保留字.....	276
附录 III C 语言运算符的优先级和结合性.....	277
附录 IV 常用标准库函数.....	278
附录 V Turbo C 2.0 编译出错信息表.....	282
参考文献	291

第 1 章 C 语言基础

程序设计语言是编程人员与计算机之间交换信息的一种工具,在众多的计算机程序设计语言中 C 语言是比较实用的一种。另外,目前流行的很多新型语言如 C++, Java 等都衍生自 C 语言。掌握了 C 语言,今后再学习这些新型语言就比较轻松了。

本章教学基本要求:

- ◆ 了解 C 语言的特点
- ◆ 理解 C 程序的开发过程
- ◆ 掌握 C 语言源程序的基本结构和书写格式
- ◆ 掌握在 Turbo C 2.0 环境下调试程序的方法

1.1 绪 论

众所周知,计算机系统是以硬件为基础、以软件为平台呈现给用户的。我们使用计算机其实就是通过操作软件来驱动硬件工作。没有配置任何软件的计算机称为“裸机”。裸机是不会有目的地工作的,所以要使计算机俯首贴耳地工作,就必须配备相应的软件。

1.1.1 计算机软件

“软件”一词是在 1960 年巴黎的一次计算机学术会议上正式启用的。广义地讲软件就是运行、管理和维护计算机的各种程序以及运行这些程序所必须的手续、规则、资料等的总称。简单地说软件就是指计算机运行所需的各种程序。

从软件配置的角度,一般将软件分为两大类:系统软件和应用软件。

系统软件一般是计算机必备并由计算机厂商提供的。系统软件主要包括:

(1) 操作系统。操作系统是管理和控制计算机系统软、硬件资源的大型程序,是用户与计算机之间的接口,它提供了软件运行的环境。

(2) 语言处理程序。语言处理程序是指解释程序、编译程序和汇编程序等。

用汇编语言或高级语言编写的程序称为源程序。

用高级语言编写的源程序,不能被计算机直接识别与执行,只有被翻译成二进制机器代码后,才能在计算机上执行。这种翻译过程通常有两种方式,一种是“解释执行”方式,即翻译一句执行一句,这时称完成这种翻译工作的程序为解释程序;另一种是“编译执行”方式,即为提高运行效率和对源程序保密而一次性将源程序转换成可执行的机器代码的语言处理程序,这时称完成这种翻译工作的程序为编译程序,在采用“编译执行”方式时,由编译程序产生出来的二进制代码程序被称为相对于该源程序的“目标程序”。

一种将汇编语言编写的源程序翻译成等价的机器语言的特定程序称为汇编程序。

(3) 服务性程序。服务性程序是指为了帮助用户使用与维护计算机,提供服务性手段而编制的

一类程序。这类程序包含内容广泛，如程序的编辑、装配、调试和诊断等。

1) 编辑程序。编辑程序是经常使用的一种服务性程序，它为用户提供一个书写环境。用户一般通过键盘与显示屏幕，方便地输入并编辑自己的程序。编辑功能包括修改、删除、插入、移动、复制和粘贴某些内容等。

2) 装配程序。装配程序，也称链接程序，它的功能是将若干个目标模块和相应高级语言的库函数（程序）链接在一起，产生可执行的运行模块。

3) 调试程序。调试程序的功能是帮助用户检查程序中有什么错误，以及错在什么地方，以便于修改。

应用软件一般指用户在各自的应用领域中，为解决各类实际问题而用某种计算机语言编写出来的程序。应用软件不胜枚举，其中商用信息处理软件所占比例最大。工程与科学计算软件大多属于计算问题，控制类软件旨在实现生产的自动化。应用软件使得传统的产业部门面目一新，带给人们的是惊人的生产效率和巨大的经济效益。

1.1.2 程序设计语言

语言是人类最重要的交际工具之一，如自然语言（汉语、英语等）、数学语言（ $C=2\pi R$ ）和计算机语言等。要编写程序就必须使用计算机语言。计算机语言是根据预先定义的规则（语法）而写出的语句的集合，这些语句组成了程序。

从计算机诞生到今天，程序设计语言伴随着计算机硬件技术的进步而不断地在演化。目前世界上广泛使用的计算机语言大致有十几种。按照语言级别有低级语言和高级语言之分。低级语言包括机器语言和汇编语言。高级语言有许多种，其中较著名的有：BASIC, FORTRAN, PASCAL, C, C++和 Java 等。

(1) 机器语言。机器语言是一种能够被计算机直接接受并执行而无须中间处理的面向机器的语言。它与计算机硬件密切相关，每种机器都有自己的一套机器语言，即计算机指令系统。机器语言的主要特点是与特定的机器相关，运算效率高，但用户难于使用，烦琐，费时，且易出错。

(2) 汇编语言。汇编语言是一种符号化的机器语言，即用助记符号表示二进制代码。汇编语言只比机器语言进化了一点点，它与机器语言之间基本上是一一对应的，只是表示方法不同而已，但不能被机器直接识别与执行，需要“代真”。所谓“代真”就是用汇编程序将汇编语言转换或翻译成计算机能直接执行的机器语言的过程。汇编语言在一定程度上克服了机器语言的缺点，但对大多数用户来说，仍然是不便理解和使用的。

(3) 高级语言。高级语言是一种既类似于数学语言(如 $y=5*\cos(x)+1$)又接近于自然语言（如英语），且比低级语言更接近于待解问题的程序设计语言。其特点是在一定程度上与具体机器无关，易学、易用、易维护。高级语言的设计目标是：使程序员摆脱低级语言烦琐的细节。

近年来，高级语言的发展极为迅速，如 Visual Basic, Visual C, Delphi, Power Builder, Java, C#, J#等面向对象的可视化开发工具，大大提高了计算机语言解决问题的能力，并将逐步成为主流。

面向对象程序设计语言借鉴了 20 世纪 50 年代的人工智能语言 LISP, 成形于 70 年代的 Smalltalk, 开始流行于 90 年代。由 C 语言发展而来的 C++语言是迄今为止商业上最受欢迎的一种支持面向对象的程序设计语言。

其实，每一种语言都有其自身的特点和不足。面对不同的问题，应根据实际情况来选择程序设计语言，以便优质高效地解决相关问题。

1.2 C语言概述

C语言是继 BASIC 语言、FORTRAN 语言、COBOL 语言和 PASCAL 语言之后的一种国际上通用的计算机程序设计语言。它既可用于系统软件的开发，也可用于应用软件的编写。

1.2.1 C语言的发展简史

C语言是在 B语言的基础上发展起来的，其根源可以追溯到 1960 年出现的 ALGOL 60。ALGOL 60 离硬件比较远，不易用来编写系统程序。1963 年，英国剑桥大学推出了 CPL (Combined Programming Language) 语言。CPL 语言比 ALGOL 60 接近硬件，但规模较大，难以实现。1969 年，英国剑桥大学的 M.Richards 对 CPL 语言做了简化，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，设计出了简单而又接近硬件的 B语言 (取 BCPL 的第一个字母)，但 B语言过于简单，功能有限。1972~1973 年间，AT&T 公司 Bell 实验室的丹尼斯·里奇 (D.Ritchie) 在 B语言的基础上设计出了 C语言 (取 BCPL 的第二个字母)。C语言既保持了 BCPL 和 B语言的优点，又克服了它们的缺点。

最初的 C语言是为 UNIX 多用户、多任务计算机操作系统而设计的，依附于 UNIX 系统，主要在贝尔实验室内部使用。1983 年美国国家标准协会 ANSI 为了使 C语言得到更快的发展，制定了 C语言标准，称为 ANSI C，并于 1987 年公布了关于 C语言的新标准——87 ANSI C。现在使用的各种 C语言版本都是以此标准为基础的。

1.2.2 C语言的特点

C语言揉合了高级语言的一系列特点与汇编语言的效率，并且不专用于某一个特定的应用领域。另外，C语言限制少，通用性强，这使得它比一些公认的其他语言用起来更方便，效率更高。目前，C语言已从位于贝尔实验室的发源地传播到世界各地，成为全球程序员的公共语言，并由此诞生了几个新的主流语言 C++，Java 等，它们都建立在 C语言语法和基本结构的基础上。现在国际上的许多软件都是在 C语言及其衍生的各种语言的基础上开发出来的。

C语言具有以下特点：

(1) 语言简洁、紧凑，使用方便、灵活。它共有 32 个关键字，9 种控制语句，程序书写形式自由。

(2) 具有现代语言的各种数据结构。C语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型和共用体类型等，可以实现各种复杂的数据结构 (如链表、树和栈等) 运算，特别是指针类型数据，使用起来更为灵活、多样。

(3) 运算符丰富。ANSI C 共有 34 种运算符，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。

(4) 理想的结构化程序设计语言。具有如 if-else 语句、switch 语句、while 语句、do-while 语句和 for 语句等结构化控制语句，符合现代编程风格的要求。

(5) 具有双重性。允许直接访问物理地址，进行位 (bit) 操作，实现汇编语言的大部分功能，可以直接对硬件进行操作。因此它具有高级语言和低级语言的双重功能。

(6) 语法限制不太严格, 编程自由度大。一般高级语言语法检查比较严格, 几乎能检查出所有的语法错误。而 C 语言放宽对语法的检查, 允许编程人员有较大的自由度。

(7) 生成目标代码质量高, 程序可读性强, 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%。

(8) 程序可移植性好。C 语言具有汇编语言的一些功能, 涉及硬件操作, 由于这些操作都是通过调用函数来操作系统功能的, 因此, 能够很方便地在不同硬件的计算机之间进行移植。

1.3 C 语言源程序的基本结构

为了说明 C 语言源程序的基本结构, 下面介绍几个简单的 C 程序实例。这些实例一方面体现了 C 语言的特点, 另一方面可以帮助读者初步了解 C 语言源程序的基本结构和程序设计的格式。

【例 1.1】 在计算机屏幕上输出 A first simple C program. 字符串。

程序:

```
#include"stdio.h"          /*编译预处理命令, 指定头文件*/
main()                    /*主函数*/
{                          /*函数体开始标志*/
    printf("A first simple C program.\n"); /*调用格式输出函数 printf(), 输出字符序列*/
}                          /*函数体结束标志*/
```

程序运行结果:

A first simple C program.

分析:

这是一个简单的 C 程序。程序首行#include"stdio.h"是编译预处理命令, 用于通知编译器在本程序中包含标准输入/输出头文件 (stdio.h) 信息。main()表示主函数, 主函数体用一对花括号 "{}" 括起来, 其中左花括号 "{" 表示主函数体开始, 右花括号 "}" 表示主函数体结束; 主函数体中只有一个 printf()函数调用语句, 其功能是在计算机屏幕上按原样输出双撇号 "" 内的字符串 A first simple C program.; "\n" 是换行符, 表示在输出 A first simple C program.后自动换行; 分号 ";" 是 C 语句的结束标志。注释符 "/*" 和 "*/" 之间的内容表示对程序的解释, 以便于读者理解。

【例 1.2】 求两数之和。

程序:

```
#include"stdio.h"          /*编译预处理命令, 指定头文件*/
main()                    /*主函数*/
{                          /*函数体开始标志*/
    int m,n,sum;          /*说明 m, n, sum 为整型变量*/
    m=10;                 /*给变量 m 赋初值 10*/
    n=20;                 /*给变量 n 赋初值 20*/
    sum=m+n;             /*计算 m, n 的和并赋给变量 sum*/
```



```

    printf("Sum is %d\n",sum);    /*调用格式输出函数 printf(), 输出变量 sum 的值*/
}                                  /*函数体结束标志*/

```

程序运行结果:

Sum is 30

分析:

在程序中第2~9行是主函数。其中,第2行是主函数头;第3~9行是主函数体,函数体中包含两个部分:第4行是说明部分,说明变量 m , n 和 sum 为整型变量;第5~8行是语句部分,第8行是格式输出函数 `printf()`调用语句。`printf()`函数中双撇号“””内的 `Sum is` 原样输出, `%d` 是输出“格式字符串”,它表示输出时是以带符号的十进制形式输出 sum 的值, sum 是要输出的两数之和,其值为30。

【例 1.3】计算半径 r 为 2.0 时圆的面积。

程序:

```

#define PI 3.14159                /*编译预处理命令, 宏定义*/
#include"stdio.h"                  /*编译预处理命令, 指定头文件*/
#include"math.h"                  /*编译预处理命令, 指定头文件*/
main()                             /*主函数*/
{                                  /*函数体开始标志*/
    float r,s;                    /*说明 r, s 为单精度实型变量*/
    r=2.0;                        /*给变量 r 赋初值 2.0*/
    s=PI*pow(r,2);               /*调用数学函数 pow()计算 r^2*/
    printf("r=%f,s=%f\n",r,s);   /*调用格式输出函数 printf(), 输出结果*/
}                                  /*函数体结束标志*/

```

程序运行结果:

r=2.000000, s=12.566360

分析:

这是一个当半径 r 等于 2.0 时,编程计算圆面积的程序。程序中第1行是编译预处理命令中的宏定义,其作用是编译时,用宏体自动替换宏名。其中: PI 是宏名(一般用大写), 3.14159 是宏体(可以是一个字符串等)。程序中的 `pow(r,2)` 是常用的数学函数,功能是计算 r 的平方,其有关的数学函数包含在指定头文件 `math.h` (详见附录 IV) 中。

【例 1.4】任意从键盘上输入两个整数 m 和 n , 比较它们的大小, 并输出较大的数。

要求: 比较两数大小用自定义函数实现, 输入两个整数和输出较大的数在主函数中完成。

程序:

```

#include"stdio.h"                /*编译预处理命令, 指定头文件*/
main()                             /*主函数*/
{                                  /*主函数体开始标志*/
    int m,n,c;                    /*说明 m, n, c 为整型变量*/

```