高职高专计算机教育规划教材

C语言程序设计

曹玲焕 孙 萍 主编 朱 珍 余小燕 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高职高专计算机教育规划教材

C语言程序设计

曹玲焕　孙　萍　主　编
朱　珍　余小燕　副主编
骆金维　参　编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书以新标准对C语言的规定为依据，通过大量的实例，从计算机语言和程序设计的基本知识出发，系统地介绍了C语言程序设计中的各种数据类型及其运算、基本语句、选择结构程序设计、循环结构程序设计、数组的定义和引用、函数、指针、编译预处理、结构体与共用体、位运算、文件操作等。每个知识点配有相应的练习，方便边学边练、讲练结合的授课形式；每章配有相应的实训内容，重在培养学生的实际动手能力；精选的习题根据知识特点采用多种形式，作为对该章内容的巩固和延伸；将上机环境与全国计算机等级考试二级接轨，采用Visual C++ 6.0运行环境，内容涵盖C语言等级考试大纲，旨在帮助读者学好C语言的同时，顺利通过等级考试。

本书通俗易懂、逻辑性强、循序渐进，便于读者接受，适合作为高职高专院校C语言课程的教材，也可作为培训教材或读者自学的参考书。

图书在版编目（CIP）数据

C语方程序设计/曹玲焕，孙萍主编. —北京：中国铁道出版社，2009.7
（高职高专计算机教育规划教材）
ISBN 978-7-113-10116-9

Ⅰ.C… Ⅱ.①曹…②孙… 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122967号

书　　名：C语言程序设计
作　　者：曹玲焕　孙　萍　主编

策划编辑：秦绪好　王春霞
责任编辑：王占清　　**编辑部电话：**（010）63583215
编辑助理：何红艳
封面设计：付　巍　　**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：北京鑫正大印刷有限公司
版　　次：2009年9月第1版　　2009年9月第1次印刷
开　　本：787mm×1092mm　1/16　印张：15.75　字数：386千
印　　数：4 000册
书　　号：ISBN 978-7-113-10116-9/TP・3336
定　　价：24.00元

前言

FOREWORD

C 语言是一种被广泛学习、普遍使用的计算机程序设计语言。它的高级语言形式、低级语言功能具有特殊的魅力。由于 C 语言具有完整的编程语言特点，很多高职高专院校开设此程序设计课程。

本书充分考虑到高职高专院校教学的特色，注重理论联系实际，突出实用性，语言组织上通俗易懂。全书以培养学生结构化程序设计的基本能力为主线，采用由浅入深、循序渐进的教学策略，力求简洁、明确、清晰，使读者能够较快入门，并逐步学会自己动手编写程序。

全书共 13 章，将 C 语言的学习分为三大部分：第一部分（第 1、2 章）介绍了 C 程序的基本框架、C 程序的实现过程、数据类型及数据处理等程序设计的基础知识；第二部分（第 3～7 章）培养程序设计的基本能力，介绍了三种结构程序的设计、数组、函数等内容；第三部分（第 8～13 章）培养设计应用程序的能力，介绍了指针、结构体、共用体、文件、综合实用程序实训等内容。编排时充分考虑了教师组织教学的需要，各章均提供了丰富的例题、练习与思考、上机实训及习题。例题典型有趣，贴近学生的实际；习题方便进行过程考核，使学生在练习中体验成功，激发学习兴趣，适用于边讲边练、讲练结合的教学模式。每章配有相应的实训内容，精选实训案例，并将知识点融于案例中，引导学生在完成每个题目的过程中强化相应的知识点，学会相应的技能，提高学生的实际应用能力，便于学生上机实践。此外将上机环境与全国计算机等级考试二级接轨，采用 Visual C++ 6.0 系统运行环境。内容方面针对 C 语言等级考试，旨在帮助读者学好 C 语言的同时，顺利通过等级考试。本书适合作为高职高专院校的学习教材，也可作为培训教材或读者自学的参考书。

全书由高职高专院校富有教学与开发经验的教师编写。曹玲焕、孙萍任主编，朱珍、余小燕任副主编。孙萍编写了第 1～4 章，并对全书进行了排版；朱珍编写了第 6、11、12 章；余小燕编写了第 5、9 章；骆金维编写了第 13 章；曹玲焕编写了第 7、8、10 章，并对全书进行了修改、统稿。

在编写过程中得到了张涛等诸位老师的大力支持，在此一并表示感谢。

由于时间仓促，加之编者水平和经验有限，书中难免有不足之处，恳请广大读者批评指正。

编者邮箱：cao_ling_huan@163.com

编 者

2009 年 6 月

目 录

CONTENTS

第 1 章 程序设计基础 1
1.1 C 语言程序设计概述 1
1.1.1 C 语言简介 1
1.1.2 C 程序与程序设计 1
1.2 算法及结构化程序 2
1.2.1 算法 2
1.2.2 结构化程序 3
1.3 简单 C 语言程序的构成和格式 5
1.3.1 程序举例 5
1.3.2 小结 6
小结 8
实训 认识 C 语言程序 8
习题 10
第 2 章 基本数据类型和基本运算 12
2.1 常量 12
2.1.1 数值常量 12
2.1.2 字符常量 13
2.1.3 转义字符常量 14
2.1.4 字符串常量 14
2.1.5 符号常量 15
2.2 变量及其数据类型 15
2.2.1 标识符 15
2.2.2 变量的数据类型 16
2.2.3 变量的初始化 19
2.3 算术运算符和算术表达式 19
2.3.1 算术运算符 19
2.3.2 运算符的优先级、结合性和算术表达式 20
2.3.3 类型转换 21
2.3.4 求字节数运算符 21
2.4 赋值运算符和赋值表达式 22
2.5 自增、自减运算符和逗号运算符 22
2.5.1 自增、自减运算符 22
2.5.2 逗号运算符和逗号表达式 23
小结 23
实训 数据类型、运算符与表达式 24
习题 25

第 3 章 顺序结构程序设计 29
3.1 C 语句概述 29
3.1.1 C 语句的分类 29
3.1.2 赋值语句 30
3.2 格式输出和输入 31
3.2.1 格式输出函数（printf()函数） 31
3.2.2 格式输入函数（scanf()函数） 34
3.3 字符的输出和输入 36
3.3.1 字符输出函数（putchar()函数） 36
3.3.2 字符输入函数（getchar()函数） 36
3.4 顺序结构程序设计举例 37
小结 39
实训 顺序结构程序设计 39
习题 40
第 4 章 选择结构程序设计 43
4.1 关系运算和逻辑运算 43
4.1.1 关系运算符和关系表达式 43
4.1.2 逻辑运算符和逻辑表达式 44
4.2 if 语句 46
4.2.1 if 语句的三种形式 46
4.2.2 if 语句的嵌套 50
4.2.3 条件运算符和条件表达式 51
4.3 switch 语句 52
4.4 选择结构程序设计举例 54
小结 56
实训 选择结构程序设计 56
习题 57
第 5 章 循环结构程序设计 61
5.1 while 循环语句 61
5.2 do...while 循环语句 63
5.3 for 循环语句 64
5.4 break 语句和 continue 语句 67
5.4.1 break 语句 67
5.4.2 continue 语句 68
5.5 循环结构的嵌套 68
5.6 循环结构程序设计举例 69
小结 71
实训 循环结构程序设计 71
习题 72

第 6 章 数组 76
6.1 一维数组 76
6.1.1 一维数组的定义 77
6.1.2 访问数组元素 77
6.1.3 初始化数组元素 77
6.1.4 数组元素的输出 79
6.1.5 一维数组的应用 79
6.2 二维数组 83
6.2.1 二维数组的定义 84
6.2.2 访问二维数组元素 85
6.2.3 初始化二维数组 86
6.2.4 二维数组的应用 87
6.3 字符数组与字符串 88
6.3.1 字符数组的说明和引用 88
6.3.2 字符数组的输入/输出 89
6.3.3 字符串处理函数与字符串数组 90
6.4 数组应用举例 93
小结 95
实训 数组 95
习题 95
第 7 章 函数 98
7.1 函数概述 98
7.2 函数的定义和返回值 99
7.2.1 函数定义的语法 99
7.2.2 函数的返回值 101
7.3 函数的调用 101
7.3.1 函数的两种调用方式 101
7.3.2 函数的参数传递 102
7.3.3 函数调用的语法要求 103
7.4 函数的说明 104
7.4.1 函数说明的形式 104
7.4.2 函数说明的位置 105
7.5 局部变量和全局变量 106
7.5.1 函数中的局部变量和全局变量 106
7.5.2 利用全局变量传递数据 108
7.6 变量的存储类别 109
7.7 函数的嵌套调用和递归调用 114
7.7.1 函数的嵌套调用 114
7.7.2 函数的递归调用 115

7.8 函数的存储类型 118
7.8.1 外部函数 118
7.8.2 静态函数 119
7.9 函数程序举例 119
小结 122
实训 函数 122
习题 125
第 8 章 指针 129
8.1 指针及其定义 129
8.1.1 指针 129
8.1.2 指针的定义 130
8.1.3 指针的初始化 131
8.2 指针的运算 131
8.2.1 引用运算 131
8.2.2 指针的算术运算 133
8.2.3 指针的赋值运算 134
8.2.4 指针的关系运算 135
8.3 指针和变量 136
8.3.1 利用指针处理简单数据 136
8.3.2 指针作函数参数 137
8.4 指针和数组 139
8.4.1 指针和一维数组 140
8.4.2 指针和二维数组 145
8.5 指针和字符串 151
8.5.1 使指针指向字符串 151
8.5.2 利用指针输入和输出字符串 152
8.5.3 字符串数组 153
8.5.4 程序举例 153
8.6 指针和函数 154
8.6.1 指向函数的指针 154
8.6.2 指向函数的指针变量作函数参数 156
8.6.3 指针函数 156
小结 157
实训 指针 157
习题 158
第 9 章 编译预处理 162
9.1 宏定义 162
9.1.1 不带参数的宏定义 162
9.1.2 带参数的宏定义 163

9.2 文件包含 164
9.3 条件编译 165
小结 166
实训 编译预处理 167
习题 167
第 10 章 结构体和共用体 170
10.1 结构体类型 170
10.1.1 结构体类型的说明 170
10.1.2 结构体类型变量的定义 171
10.1.3 结构体变量的初始化 173
10.1.4 结构体成员引用 173
10.1.5 结构体变量的赋值运算 173
10.2 结构体数组 174
10.2.1 结构体数组的定义 174
10.2.2 结构体数组初始化 174
10.2.3 结构体数组的应用举例 175
10.3 结构体与函数 176
10.3.1 结构体变量作函数参数 176
10.3.2 结构体型函数 179
10.4 指针和结构体 179
10.4.1 结构体指针及其定义 179
10.4.2 通过指针引用结构体成员 180
10.4.3 结构体指针作函数参数 181
10.4.4 结构体指针函数 183
10.5 链表 183
10.5.1 链表概述 183
10.5.2 链表的操作 184
10.6 共用体类型 190
10.6.1 共用体类型的说明 190
10.6.2 共用体变量的引用和初始化 191
10.7 自定义类型 193
10.7.1 自定义类型的形式和功能 193
10.7.2 自定义类型的应用 193
10.8 程序举例 194
小结 196
实训 结构体和共用体 197
习题 198
第 11 章 位运算 203
11.1 位运算符 203

11.2 位运算符的功能 203
11.2.1 按位与运算 203
11.2.2 按位或运算 204
11.2.3 按位异或运算 204
11.2.4 求反运算 204
11.2.5 左移运算 205
11.2.6 右移运算 205
11.2.7 位运算举例 206
小结 207
实训 位运算 207
习题 209
第 12 章 文件 **211**
12.1 文件概述 211
12.2 文件类型指针与文件操作 212
12.3 文件打开函数 212
12.4 文件关闭函数 213
12.5 文件读/写函数 213
12.5.1 单个字符读/写函数 213
12.5.2 字符串读/写函数 214
12.5.3 数据块读/写函数 215
12.5.4 格式化读/写函数 216
12.6 文件定位函数 217
12.6.1 文件的定位 217
12.6.2 文件位置指针的检测 219
12.7 文件出错的检测 219
12.7.1 文件出错检测函数 ferror() 219
12.7.2 出错标志置零函数 clearerr() 220
12.8 应用举例 220
小结 222
实训 文件 223
习题 224
第 13 章 综合实训 **227**
附录 A C 语言中的关键字 **235**
附录 B 运算符的优先级和结合性 **236**
附录 C 常用字符与 ASCII 码对照表 **237**
附录 D 标准库函数 **238**
参考文献 **242**

第1章 程序设计基础

如果想要让计算机完成指定的任务，就必须告诉计算机怎么做。其中，操作的步骤即完成任务的步骤就是算法，根据任务采用的某种程序设计语言进行编写程序的过程就是程序设计。

1.1 C语言程序设计概述

1.1.1 C语言简介

1. C语言的诞生和发展

C语言诞生以前，系统软件主要用汇编语言编写。由于汇编语言程序依赖于计算机硬件，其可读性和可移植性都很差；但一般的高级语言又难以实现对计算机硬件的直接操作（这正是汇编语言的优势），于是人们希望有一种计算机语言既能具有高级语言的优点，又能具有低级语言的功能。

C语言是贝尔实验室于20世纪70年代初研制出来的，后来又被多次改进，并出现了多种版本。20世纪80年代初，美国国家标准化学会（ANSI），根据C语言问世以来各种版本对C语言的发展和扩充，制定了ANSI C标准（1989年再次进行了修订）。目前，在微型计算机上广泛使用的C语言编译系统有Microsoft C，Turbo C，Borland C，Visual C++等。虽然它们的基本部分都相同，但还是有一些差异，所以还需注意自己所使用的C语言编译系统的特点和规定。为了易于以后的C++过渡，本课程上机采用Visual C++ 6.0运行环境。

2. C语言的特点

C语言同时具有汇编语言和高级语言的优势。

① 语言简洁、紧凑，使用方便、灵活。

② 运算符极其丰富。

③ 生成的目标代码质量高，程序执行效率高。

④ 可移植性好（较之汇编语言）。

⑤ 可以直接操纵硬件。

1.1.2 C程序与程序设计

1. C程序

目前，社会上使用的程序设计语言有很多种，其中大多数被称为计算机的“高级语言”，

如 Visual Basic、C++、Java 以及本书将要介绍的 C 语言等。这些语言采用接近人们习惯的自然语言和数学语言作为表达式，使人们学习和操作起来十分方便。

但是，计算机只能接受和处理由 0 和 1 的代码构成的二进制指令或数据，即“机器语言”。人们将由高级语言编写的程序称为“源程序”，将由二进制代码表示的程序称为“目标程序”。需要将“源程序”翻译成计算机所能直接识别并执行的“目标程序”，这种具有翻译功能的软件称为“编译程序”。C 语言自带编译功能。C 源程序经过 C 编译后生成一个扩展名为.OBJ 的二进制文件（称为目标文件），然后由称为“连接程序”的软件，将此.OBJ 文件与 C 语言提供的各种库函数连接起来生成一个扩展名为.EXE 的可执行文件，如图 1-1 所示。在操作系统环境下，只需单击或输入此文件的名称（可不必输入扩展名.EXE），就可运行该可执行文件。

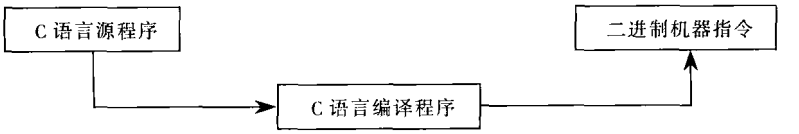


图 1-1　C 语言编译程序功能示意图

2. 程序设计

简单的程序设计一般包含以下几个部分。

① 确定数据结构。确定输入、输出的数据及存放数据的数据结构。

② 确定算法。解决问题的步骤。

③ 编码。根据确定的数据结构和算法，使用选定的计算机语言编写程序代码。

④ 上机调试。消除程序中的错误。

⑤ 整理并归纳文档资料。

1.2　算法及结构化程序

1.2.1　算法

学习计算机程序设计语言的目的是要用语言作为工具，设计出可供计算机运行的程序。在拿到一个需要求解的问题后，怎样才能编写出程序呢？除了选定合理的数据结构外，一般来说，关键的一步就是算法的设计，可以根据算法用任何一种计算机高级语言将其转换为程序。那么什么是算法？算法的特性又是什么？算法如何表示？下面将逐一进行介绍。

1. 算法的定义

算法是指为解决某个特定的问题而采取的、确定的、有限的步骤。

2. 算法的特性

① 有穷性。一个算法总是在执行了有限的操作步骤后停止。

② 确定性。算法中的每个步骤必须有确切的含义，不能有二义性。

③ 可行性。算法中指定的操作，都可以通过已经验证过可以实现的基本运算执行有限次后实现。

④ 有零个或多个输入。在算法开始之前对算法最初赋予的量，这些输入取自特定的数据

对象集合，即算法用来处理数据对象，而这些数据对象需要通过输入得到。

⑤ 有一个或多个输出。算法的目的是为了“求解”，这些结果只有通过输出才能得到。

3. 算法的表示

算法的常用表示方法有自然语言、伪代码、流程图、N-S 图。这里重点介绍流程图和 N-S 图。

（1）流程图

一般的流程图由图 1-2 中所示的几种基本图形组成。

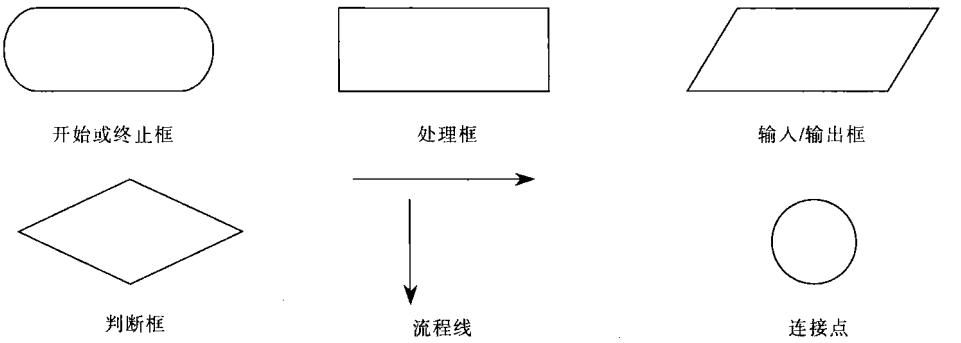


图 1-2　一般的流程图所用的几种基本图形

（2）N-S 图

将流程图中的流程线去掉，算法的每一步都用一个矩形框描述，将一个个矩形框按执行的次序连接起来即为一个完整的算法描述。

1.2.2　结构化程序

结构化程序由三种基本结构组成，即顺序结构、选择结构、循环结构。

1. 顺序结构

按语句在程序中的先后顺序逐条执行，如图 1-3 所示，其中图 1-3（a）为流程图，图 1-3（b）为 N-S 图。

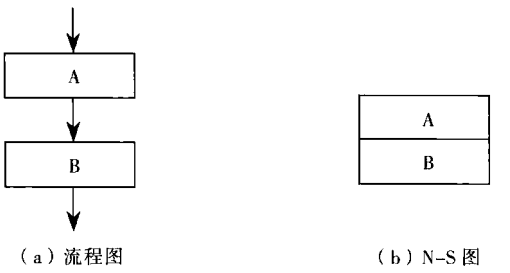
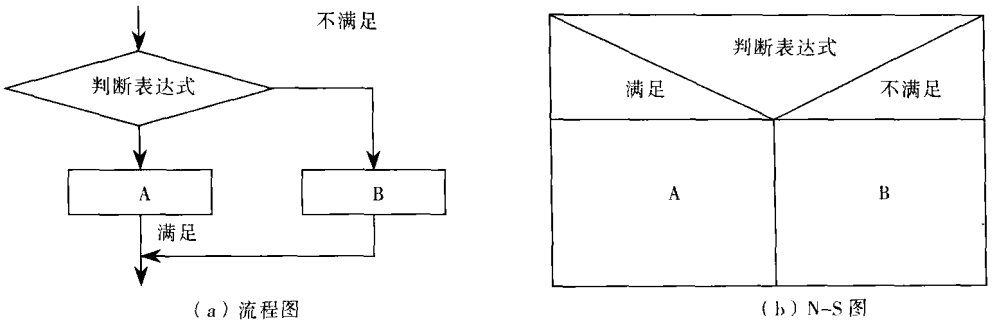


图 1-3　顺序结构流程图和 N-S 图

2. 选择结构

根据不同的条件选取执行不同分支中的语句，如图 1-4 所示，其中图 1-4（a）为流程图，图 1-4（b）为 N-S 图。

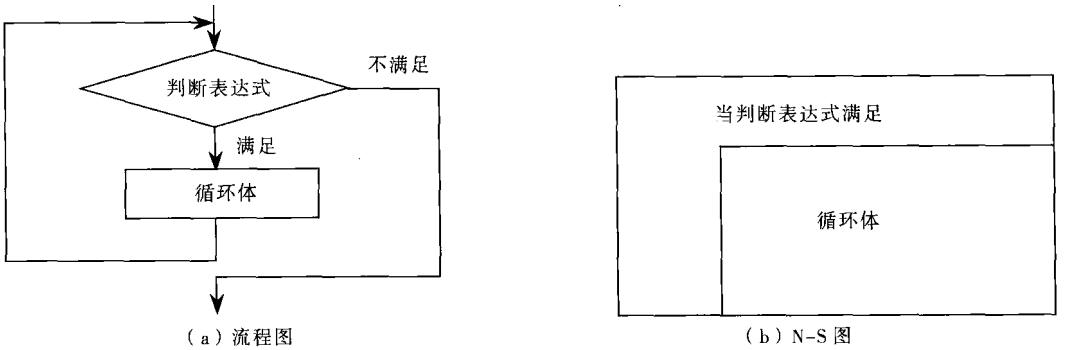


（a）流程图　　（b）N-S图

图 1-4　选择结构流程图和 N-S 图

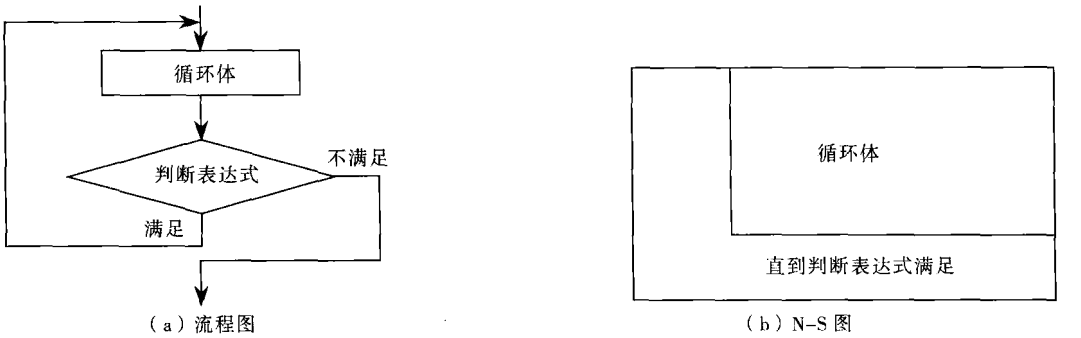
3. 循环结构

根据各自的条件，使同一组语句重复执行多次或一次也不执行，如图 1-5、图 1-6 所示，其中图 1-5（a）、图 1-6（a）为流程图，图 1-5（b）、图 1-6（b）为 N-S 图。图 1-5 所示为当型循环流程图，特点是：当指定的条件成立时，就执行循环体，否则就不执行。图 1-6 所示为直到型循环流程图，特点是：先执行循环体，再判断指定的条件是否成立，若成立则继续执行循环体，否则退出循环。



（a）流程图　　（b）N-S图

图 1-5　当型循环结构流程图和 N-S 图



（a）流程图　　（b）N-S图

图 1-6　直到型循环结构流程图和 N-S 图

这三种基本结构组成的算法可以解决任何复杂的问题。由三种基本结构所构成的算法称为结构化算法；由三种基本结构所构成的程序称为结构化程序。

1.3　简单 C 语言程序的构成和格式

1.3.1　程序举例

不同的高级语言程序的组成和格式不同，本节先给出几个 C 语言程序的例子，以对 C 语言程序的结构有一个初步的认识。

【例 1.1】编写在屏幕上显示字符串“How are you!”的程序。

```
#include<stdio.h>
void main()
{
   printf("How are you!\n");
}
```

程序运行结果如下：

```
How are you!
```

【说明】

该程序的组成和格式介绍如下：

① 该程序由一个主函数 main()组成。C 语言中的任何一个程序都必须有 main()函数。函数名必须是 *main* 且其后的一对括号不能少，括号内可以为空。

② 一对花括号{ }所括起的部分是 main()函数的函数体。“{”表示函数体的开始，“}”表示函数体的结束。函数体内可以有多个语句，本例中只有一个输出语句。

③ printf()函数是系统提供的标准库函数中的输出函数。C 语言程序的输出是由输出函数完成。“How are you !”是要输出的内容。“\ n”表示换行。它是由“\”和“n”两个字符构成，属转义字符。在 printf()函数中输出的内容和转义字符都要用双引号括起来。有关 printf()函数和转义字符在后续章节中做详细介绍。

④ 在 C 语言中，分号“;”作为语句的结束符，如本例中在 printf()函数后面加了一个“;”，printf()变成了语句。

⑤ # include 是预编译程序命令，它将头文件“stdio.h”的内容展开在# include <stdio.h >。所在的行位置处。所以，在每个使用标准库函数的程序中，都必须带有# include 命令行。在该命令行中使用的一对尖括号< >，也可使用一对双引号，即# *include* "stdio.h "。

【例 1.2】编写当给出圆的半径时，计算圆的周长的程序。

【分析】由数学知识可知：圆的周长 c=2πr。程序中先给变量 r 赋值，再用此公式进行计算，最后输出结果。

```
#include<stdio.h>
void main()
{
    int  r;                          /*定义变量 r，说明其为整型*/
    float  c;                        /*定义变量 c，说明其为实型*/
    r=5;                             /*给 r 赋值*/
    c=2*3.14159*r;                   /*计算圆周长，将值赋给 c*/
    printf("c=%f\n",c);              /*输出圆周长*/
}
```

程序运行结果如下：

```
c=31.415900
```

【说明】

① 以上程序中的第四行和第五行是变量的定义部分，在这里，对程序中用到的每一个变量进行定义并且说明其类型。r 被说明为整型变量。所以，赋给 r 的值为整型值（int）。c 是实型变量（float），赋给它的值为带小数的实型值。其中，int、float 是关键字，关键字的概念在后面做具体的介绍。

② 第六行是赋值语句，将整型数据 5 赋给了整型变量 r，使 r 的值为 5，第七行也是赋值语句，在这里，先计算表达式 2*3.14159*r，然后将计算所得到的值赋给变量 c 。

③ 第八行中一对双引号内的%f 是输出的格式控制说明，用来指定输出时的数据类型和格式，%f 表示以标准格式输出一个实型数据，当输出时，在%f 所在的位置上显示这个实数。此行中一对双引号内的 c =是原样照印的字符串，双引号外的 c 是将要输出的变量值的变量名。

【例 1.3】编写求两个整数中较小数的程序。

【分析】 本题目编写两个函数，即 min()函数和 main()函数。min()函数的功能是比较两个整数的大小，并返回其中较小的数。main()函数用来输入这两个整数，调用 min()函数及输出结果。

```
#include<stdio.h>
int  min(int  x,int  y)                    /*定义 min()函数,函数值为整型*/
{  int  z;                                 /*定义变量 z*/
   if(x<y)  z=x;                           /*找两个数中较小的数*/
   else   z=y;
   return  z;                              /*将 z 的值返回,通过 min()函数带回到调用处*/
}
void main()
{  int  a,b,c;                             /*定义变量*/
   printf("Input  a  and  b :");           /*输出提示信息*/
   scanf("%d%d",&a,&b);                    /*输入变量 a 和 b 的值*/
   c=min(a,b);                             /*调用 min()函数, 并把函数的返回值赋给 c*/
   printf("min=%d\n",c);                   /*输出较小的数*/
}
```

程序运行结果如下：

```
Input a and b : 8 5↙
min=5
```

【说明】

① 以上程序不仅包括主函数 main()，还包括 min()函数，min()函数用来求两个整数中的较小值。有关函数的问题将在第 7 章介绍。在这里，只是说明一个 C 语言程序可以由多个函数构成及给出函数的定义形式。

② 主函数中的“scanf("% d %d",&a , &b);”是输入语句，scanf()是格式化输入函数，它是一个由系统提供的标准库函数，其后括号内双引号括起的%d 为格式说明符，%d 表示输入带符号的十进制整型数，执行该语句时，变量 a 和 b 的值必须以十进制整数形式由键盘输入。

1.3.2 小结

由以上三个例子可以看出：

① C 语言程序由一个或多个函数组成。

其中必须有一个名为 main 的主函数。无论 main()函数在整个程序中的位置如何，程序的执行总是从主函数开始。主函数中的所有语句执行完，则程序执行结束。其他函数都是在执行 main()

函数时，通过调用或嵌套调用而得以执行。这些被调用的函数可以是系统提供的库函数，也可以是程序设计者根据需要自己编制的函数。

② C 语言程序中定义的函数包括函数说明和函数体两部分。

函数说明指明函数类型、函数的属性、函数名、函数参数名和形式参数类型。函数名是一个标识符（标识符的规定在第 2 章介绍），可以用除关键字和 C 语言库函数名以外的名称。函数类型是指函数返回值的类型，可以是 C 语言中任何一种简单或构造数据类型。函数属性是指函数的内部或外部属性。函数参数名是指形式参数名，即可以向函数传递的变量代表名，也是标识符。形式参数类型就是形式参数的数据类型。在 C 语言中，函数的形式参数类型可以放在函数说明部分的圆括号外面，也可以放在圆括号里面。函数体则是函数说明部分下面的从“{”开始、到“}”结束的部分，函数体包括对数据对象描述的变量的定义和对算法描述的语句序列。对函数的详细介绍将在第 7 章讲解。

③ 习惯用小写英文字母书写 C 程序，大写英文字母作为常量的宏定义和其他特殊用途。

④ C 程序书写格式自由，一行内可以写多条语句，一条语句也可分写在多行上。但考虑程序的清晰度，最好在一行内写一条语句。

⑤ 每个语句和变量的定义的最后必须有一个分号，分号是语句的组成部分。

⑥ 函数定义之外可以包含说明部分，此说明部分可称为外部说明，它可包括预编译命令，如上面例子中的 #include 和外部变量的说明等。

⑦ 函数中允许有一对以上的花括号。这是由于 C 语言中有些控制语句如 if、while、do、for、else、switch 等经常是由若干语句组成的复合语句，复合语句需要用花括号括起来。这样一个 C 语言程序中可能出现多层花括号，最外层的花括号表示函数体的开始和结束。内层的花括号表示复合语句的开始和结束。实际上，花括号也就表示了程序结构层次的范围。

⑧ C 语言程序中也可以使用注释。

注释部分的格式为：

```
/*注释内容*/
```

如例 1.3 中所有语句都加了注释，注释在程序的编译过程中不产生任何执行代码，只是在程序中起到解释、说明的作用。注意“注释”不可以嵌套，即成对出现。例如：

```
printf("a=%d\n",a); /*输出/*a 的*/值*/
```

错，/*a 的*/”不需要用注释符。

⑨ C 语言中没有输入/输出语句，程序的输入和输出是通过调用标准库函数中的输入函数和输出函数实现。

将下面程序中有错误的地方改正。

```
#include"stdio.h"
main()
{
  int  a,b;                  /*定义两个整型变量*/
  a=4;b=a*2;
  printf("b=%d\n",b);    /*输出 b 的/*值*/
```