



普通高等教育“十三五”规划教材

C语言 程序设计基础

彭旭东◎主 编



科学出版社

普通高等教育“十三五”规划教材

C 语言程序设计基础

彭旭东 主 编

王成霞 万 红 王海燕

李 可 樊丽珍 李 婧

副主编

贵州师范学院内部使用

科 学 出 版 社

北 京

内 容 简 介

“C 语言程序设计”是目前我国高校理工科专业普遍开设的一门基础课。本书针对该课程的特点与教学改革的需求,系统介绍了 C 语言程序设计的相关知识。本书分入门篇、中级篇、高级篇,从介绍 C 语言程序设计、本课程的学习方法等内容开始,详细介绍了程序的整体结构及运行方法,常用数据类型、常量、变量、运算符、程序流程控制、顺序结构、选择结构、循环结构、函数、数组、字符串、复合数据类型及指针等方面的知识。此外,还介绍了设计程序的部分知识。

本书可作为高校各专业“C 语言程序设计”课程的教学用书,也可供自学 C 语言程序设计的人员参考。

图书在版编目(CIP)数据

C 语言程序设计基础/彭旭东主编. —北京:科学出版社, 2019.10
(普通高等教育“十三五”规划教材)

ISBN 978-7-03-061589-3

I. ①C… II. ①彭… III. ①C 语言-程序设计-高等学校-教材
IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 113478 号

责任编辑:刘 刚 吴超莉 / 责任校对:陶丽荣
责任印制:吕春珉 / 封面设计:东方人华平面设计部

科学出版社 出版

北京东黄城根北街 16 号
邮政编码: 100717
<http://www.sciencep.com>

三河市骏杰印刷有限公司印刷
科学出版社发行 各地新华书店经销

*

2019 年 10 月第 一 版 开本: 787×1092 1/16
2019 年 10 月第一次印刷 印张: 17 1/2

字数: 413 000

定价: 44.00 元

(如有印装质量问题,我社负责调换〈骏杰〉)

销售部电话 010-62136230 编辑部电话 010-62138978-2029

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

前 言

“C 语言程序设计”是目前我国高校理工科专业普遍开设的一门基础课，同时也是学习程序设计的入门课程。本书介绍了利用 C 语言进行程序设计的基本概念和基础知识，力求让初学者能够逐步掌握有关 C 语言的知识，以及设计程序的基础知识。

本书分入门篇、中级篇和高级篇，共 13 章。入门篇包括第 1~7 章。这部分首先介绍了什么是 C 语言程序设计，以及学习建议；然后从一个完整的小程序开始，逐步介绍数据类型、常量与变量、运算符、选择结构和循环结构。学习完本部分内容，读者应具备初步的程序设计能力，能够独立设计、开发和调试相对简单的小程序，解决学习或工作中的简单计算问题。中级篇包括第 8~10 章。这部分重点介绍了函数、数组和字符串等方面的内容。在学习了本部分内容后，读者应该具备如下能力：利用函数设计与实现规模比较大的程序；利用数组处理更多的数据；利用字符和字符串处理非数值型数据等。高级篇包括第 11~13 章。这部分首先介绍了设计复杂程序的知识，然后以结构体为重点，介绍了 C 语言支持的常用复合数据类型，最后介绍了指针方面的基础知识。读者在学习了本部分内容后，应该对 C 语言及程序设计有比较深刻的理解。

本书适合于不同学时的“C 语言程序设计”课程教学。对于 48 学时的课程，教师可选讲第 1~12 章的主要内容；对于 64 学时的课程，教师可以讲完全部内容。另外，本书有配套的《C 语言程序设计基础学习指导》。如果与本书配合使用，将达到更好的学习效果。本书可供自学 C 语言的人员参考。读者在阅读时，建议先重点学习第 1~7 章，然后再根据自己的实际需求和兴趣来学习余下的章节。

本书由彭旭东任主编，王成霞、万红、王海燕、李可、樊丽珍、李婧任副主编。其中，第 1 章由李可编写，第 2、3、5 章由樊丽珍编写，第 4、8、11 章由王成霞编写，第 6、7 章由万红编写，第 9、10 章由王海燕编写，第 12 章和第 13 章的 13.6 节由李婧编写，第 13 章中的 13.1~13.5、13.7 节由彭旭东编写。第 1~7 章由王成霞审阅，第 8~13 章由万红审阅。李可和李婧审阅了书中的习题。全书由彭旭东统稿。

在编写本书的过程中，李洁和孟繁红提出了宝贵的意见和建议，在此一并表示感谢。

由于编者水平有限，书中难免存在不足之处，恳请各位读者提供意见和建议（邮箱 cp3319@sina.com）。

彭旭东

2019 年 6 月

目 录

人 门 篇

第 1 章 概述	1
1.1 C 语言的发展	1
1.2 为什么学习 C 语言程序设计	2
1.2.1 C 语言的主要特点	2
1.2.2 C 语言的应用范围	3
1.2.3 C 语言在编程语言中的排行	3
1.3 怎样才能学好 C 语言程序设计	4
习题	4
第 2 章 入门小程序	5
2.1 第一个小程序	5
2.2 深度剖析	7
2.2.1 C 语言程序的组成部分	7
2.2.2 标识符	10
2.3 运行程序	11
习题	13
第 3 章 数值型数据	14
3.1 数据类型	14
3.1.1 数据类型的分类	14
3.1.2 整型数据和实型数据	15
3.2 常量	16
3.2.1 整型常量	16
3.2.2 实型常量	17
3.3 变量	18
3.4 常用运算符	19
3.4.1 赋值运算符和赋值表达式	19
3.4.2 算术运算符和算术表达式	20
3.4.3 求字节数运算符	21
3.5 如何选择数据类型	21
习题	22

第 4 章 设计简单程序	24
4.1 问题、算法与程序	24
4.2 算法	26
4.2.1 算法的概念	26
4.2.2 算法的表示方法	29
4.3 程序设计风格	30
4.4 人机交互	30
4.4.1 用 printf() 函数输出数据	31
4.4.2 用 scanf() 函数输入数据	34
4.4.3 界面友好性	37
4.5 程序中的 3 个小问题	38
4.5.1 转换数据类型	38
4.5.2 复合赋值运算	40
4.5.3 常用的数学库函数	41
习题	42
第 5 章 程序流程控制基础与顺序结构	45
5.1 自增与自减运算	45
5.2 关系运算与逻辑运算	46
5.2.1 关系运算	46
5.2.2 逻辑运算	47
5.3 程序流程控制的基本概念	48
5.4 顺序结构	50
5.4.1 语句分类	50
5.4.2 顺序结构举例	51
习题	52
第 6 章 选择结构	55
6.1 简单的分支结构	55
6.1.1 使用 if 语句实现简单的分支结构	55
6.1.2 条件运算符和条件表达式	57
6.2 多分支选择结构	58
6.2.1 用于多分支结构的 if 语句	58
6.2.2 嵌套的 if 语句	60
6.2.3 switch 语句	63
6.2.4 if 语句和 switch 语句的比较	66
6.3 综合实例	69
习题	71
第 7 章 循环结构	77
7.1 基本概念	77

7.1.1 循环问题的分类	77
7.1.2 逗号运算	77
7.2 for 语句	78
7.3 while 语句	81
7.4 do-while 语句	84
7.5 嵌套的循环结构	88
7.6 特殊的流程控制语句	90
7.6.1 break 语句	90
7.6.2 continue 语句	91
7.7 综合实例	92
习题	94

中 级 篇

第 8 章 函数	99
8.1 基本概念	99
8.2 定义函数	102
8.2.1 函数的参数	102
8.2.2 return 与返回	103
8.2.3 函数体	104
8.3 声明函数	105
8.4 调用函数	106
8.5 函数调用的嵌套与递归	110
8.5.1 函数的嵌套调用	110
8.5.2 函数的递归调用	112
8.6 综合实例	116
习题	118
第 9 章 数组	125
9.1 基本概念	125
9.2 一维数组	126
9.2.1 定义一维数组	126
9.2.2 引用一维数组元素	127
9.2.3 初始化一维数组	129
9.2.4 向函数传递一维数组	132
9.2.5 一维数组应用实例	134
9.3 二维数组	138
9.3.1 定义和引用二维数组	139
9.3.2 初始化二维数组	140
9.3.3 向函数传递二维数组	142

9.3.4 二维数组应用实例	146
习题	149
第 10 章 字符与字符串	156
10.1 字符类型	156
10.1.1 字符常量	156
10.1.2 字符变量	157
10.2 字符处理	158
10.2.1 字符的输入/输出	158
10.2.2 字符库函数	162
10.3 字符串	165
10.3.1 字符串常量	165
10.3.2 使用字符型数组处理字符串	165
10.3.3 输入/输出字符串	167
10.4 综合实例	170
习题	172

高级篇

第 11 章 设计复杂程序	177
11.1 分治与模块化程序设计	177
11.2 变量的定义位置	180
11.2.1 局部变量	180
11.2.2 全局变量	182
11.3 存储类型	183
11.3.1 局部变量的存储类型	184
11.3.2 全局变量的存储类型	186
11.3.3 函数的存储类型	189
11.4 const 限定词	190
11.5 重命名数据类型	191
习题	193
第 12 章 结构与复合数据类型	199
12.1 结构体	199
12.1.1 结构体类型	199
12.1.2 结构体变量的定义和初始化	201
12.1.3 结构体变量的引用	202
12.1.4 嵌套的结构体	204
12.2 结构体数组	205
12.2.1 结构体数组的定义和访问	205
12.2.2 结构体数组的初始化	206

12.2.3 结构体数组的应用	206
12.3 结构体与函数	208
12.3.1 结构体变量的成员作为函数的实际参数	208
12.3.2 结构体变量作为函数的参数	210
12.3.3 结构体数组作为函数的参数	211
12.3.4 函数返回值为结构体	214
习题	215
第 13 章 指针	220
13.1 基本概念	220
13.1.1 指针与地址	220
13.1.2 获取地址与访问	221
13.2 指针与变量	223
13.3 指针与一维数组	227
13.3.1 数组名与指针	227
13.3.2 加减整数	227
13.3.3 自增与自减	230
13.3.4 关系运算	232
13.4 指针与二维数组	233
13.4.1 行指针与二维数组	234
13.4.2 指针变量与二维数组	235
13.4.3 指针数组与二维数组	238
13.5 指针与字符串	240
13.5.1 基本概念	240
13.5.2 处理字符串的常用库函数	243
13.5.3 综合实例	250
13.6 指针与结构体	252
13.6.1 指向结构体变量的指针	252
13.6.2 指向结构体数组的指针	254
13.6.3 结构体指针作为函数的参数	256
13.7 返回地址	257
习题	262
附录	266
附录 1 保留字	266
附录 2 运算符	266
附录 3 ASCII 编码表	268
参考文献	269

入门篇

第1章 概述

本章是对 C 语言（C programming language）的初步介绍。首先介绍了 C 语言的发展；然后讲解了 C 语言的特点、应用范围等；最后探讨了 C 语言程序设计的学习方法。在开始学习 C 语言程序设计之前，首先要弄清楚计算机程序、程序设计和程序设计语言 3 个概念。

计算机程序（computer program），简称程序，是指为了得到某种结果而编写的一组指示计算机动作的指令序列。

程序设计（program design），就是寻求解决问题的方法，并将其实现步骤编写成计算机可执行程序的过程。

程序设计语言（program language），泛指一切用于编写计算机程序的语言。程序设计语言包括机器语言、汇编语言、高级语言 3 种。C 语言属于高级语言。

1.1 C 语言的发展

C 语言的发展如图 1.1 所示。



图 1.1 C 语言的发展

1972 年 11 月, C 语言在贝尔实验室问世。1973 年初, C 语言的主体完成。丹尼斯·里奇和肯·汤普逊用它完全重写了 UNIX 操作系统。1978 年 11 月, 美国电话电报公司(AT&T)贝尔实验室正式发布 C 语言。1983 年, 为了避免各开发厂商用的 C 语言语法产生差异, 美国国家标准局(American National Standards Institute, ANSI)制定了一套完整的 C 语言标准。1989 年 12 月, C 语言标准被批准, 被称为 ANSI X3.159-1989 “Programming Language C”。这个版本的 C 语言标准通常被称为 ANSI C。由于这个版本是 1989 年完成制定的, 因此也简称为 C89。1990 年 9 月, ANSI C 被 ISO(International Standards Organization)采纳为国际标准。1999 年 8 月, ISO 发布了 C99。2011 年 12 月, ISO 发布了 C11。

目前流行的 C 语言编译系统大多是以 ANSI C 为基础进行开发的, 但不同版本的 C 语言编译系统实现的语言功能和语法规则略有差异。读者可选择 Microsoft Visual C++、Dev-C++、Code::Blocks 或者其他开发工具完成本书的学习。

1.2 为什么学习 C 语言程序设计

C 语言作为一种计算机程序设计语言, 既具有高级语言的特点, 又具有低级语言的特点。它可以作为系统设计语言编写系统程序, 也可以作为应用程序设计语言编写不依赖计算机硬件的应用程序, 范围非常广泛。

1.2.1 C 语言的主要特点

C 语言是一种结构化语言。其层次清晰, 便于按模块化方式组织程序, 易于调试和维护。C 语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型, 便于实现各类复杂的数据结构; 还可以直接访问内存的物理地址, 进行位(bit)一级的操作。

具体来讲, C 语言的特点如下。

1. 简洁紧凑、灵活方便

C 语言(1990 年发布的 C90)共有 32 个保留字, 9 种控制语句, 程序格式书写比较自由。

2. 运算符丰富

C 语言共有 34 个运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理。

3. 数据结构丰富

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、共用体类型等, 能用来实现各种复杂的数据类型的运算, 具有较强的计算功能、逻辑判断功能。其引入了指针概念, 使程序运行效率更高。另外, C 语言具有强大的图形功能, 可以编写 3D 游戏、做数据库等。

4. C语言是结构化语言

作为结构化语言的C语言，其主要结构成分是函数。在C语言中，函数是完成程序功能的基本结构。一个程序的各个任务可以用函数分别定义和编码，使程序模块化。

5. C语言允许直接访问物理地址，可以直接对硬件进行操作

C语言能像汇编语言一样对位、字节和地址进行操作。

6. C语言适用范围广，可移植性好

C语言适合于多种操作系统，如DOS、UNIX，也适用于多种机型。

当然，C语言也有自身的不足，如语法限制不太严格，对变量的类型约束不严格，影响程序的安全性，对数组下标越界不做检查等。从应用的角度看，C语言比其他高级语言更难掌握。

1.2.2 C语言的应用范围

C语言的应用范围非常广泛，不仅仅应用于软件开发，各类科研项目也常用到。下面列举了C语言的一些常见应用领域。

1) 应用软件。Linux操作系统中的应用软件都是使用C语言编写的，因此这样的应用软件的安全性非常高。

2) 对性能要求严格的领域。一般对性能有严格要求的地方都是用C语言编写的，如网络程序的底层和网络服务器端底层、地图查询等。

3) 系统软件和图形处理。C语言具有很强的绘图能力和可移植性，并且具备很强的数据处理能力，可用来编写系统软件、制作动画、绘制二维图形和三维图形等。

4) 数字计算。相对于其他编程语言，C语言是数字计算能力较强的高级语言。

5) 嵌入式设备开发。手机、PDA等消费类电子产品内部的应用软件、游戏等很多都是采用C语言进行嵌入式开发的。

6) 游戏软件开发。利用C语言可以开发很多游戏，如推箱子、贪吃蛇等。

1.2.3 C语言在编程语言中的排行

TIOBE编程语言社区排行榜是编程语言流行趋势的一个指标，每月更新。这份排行榜排名基于互联网上有经验的程序员、课程和第三方厂商的数量。排名使用著名的搜索引擎（如Google、MSN、Yahoo!等）进行计算。虽然这个排行榜不能说明一门编程语言好不好，或者一门语言所编写的代码数量多少，但至少能反映某个编程语言的热门程度。在TIOBE编程语言社区排行榜中，C语言、Java和C++霸占了前三名，拥有绝对优势，如表1.1所示。

表 1.1 TIOBE 编程语言社区排行榜

排名	2016 年 4 月榜单		2017 年 10 月榜单		2018 年 1 月榜单	
	语言	占有率	语言	占有率	语言	占有率
1	Java	20.846%	Java	12.431%	Java	14.215%
2	C	13.905%	C	8.374%	C	11.037%
3	C++	5.918%	C++	5.007%	C++	5.603%
4	C#	3.796%	C#	3.858%	Python	4.678%
5	Python	3.330%	Python	3.803%	C#	3.754%
6	PHP	2.994%	JavaScript	3.010%	JavaScript	3.465%
7	JavaScript	2.566%	PHP	2.790%	Visual Basic .NET	3.261%
8	Perl	2.524%	Visual Basic .NET	2.735%	R	2.549%
9	Ruby	2.345%	Assembly language	2.374%	PHP	2.532%
10	Visual Basic .NET	2.273%	Ruby	2.324%	Perl	2.419%

1.3 怎样才能学好 C 语言程序设计

建议初学者在理解概念的基础之上采用如下方式学习 C 语言程序设计。

1) 依照猫画虎。先掌握 C 语言编程软件的基本使用方法,按照书中示例程序原样输入并运行程序,看是否有错误,熟练掌握示例程序的编程语句,不要求快,一定要输入准确,养成良好的编程习惯。

2) 尝试性修改。在第一阶段的基础上对示例程序有一定认识,明白这个程序为什么这样编写,在示例程序上进行尝试性修改,观看程序会发生什么变化,为什么会这样变化?跟自己预想的变化一样吗?如果不看书,你能正确输入并运行程序吗?不能的话,你能改正吗?分析原因,加深理解。

3) 尝试程序调试。将正确的程序按照错误集类型,每次只改错一个地方,记录错误提示信息,再调试正确,反复修改调试,找到 C 语言程序出现错误的原因及解决办法,累积调试经验。

4) 进行典型程序研究。研究经典的 C 语言程序,知道这个程序解决了哪些实际问题,怎么解决的。解决流程中有什么好的方法、技巧值得学习?有其他更好的解决方案吗?

5) 进行自主源程序设计。C 语言程序设计的目的是让计算机能明白我们解决问题的方法,执行解决实际问题的方案。通过以上学习,自己尝试着用 C 语言去解决实际中的问题,在解决问题的过程中提高 C 语言程序设计、调试的能力,为进一步学习打好基础。

习题

1. 什么是计算机程序?
2. 程序设计语言包括几种语言?

第2章 入门小程序

通过第1章的学习，我们已经了解了什么是C语言程序设计。本章首先通过一个简单的C语言程序，介绍C语言程序的基本格式、组成结构等内容；然后详细介绍了C语言程序从编写到运行的全过程，以及运行开发环境的应用。通过本章内容的学习，读者可初步了解C语言程序的基本结构和运行方法，为以后的学习打下基础。

2.1 第一个小程序

下面我们就从一个简单的C语言程序开始学起。这里我们先不急于知道程序代码中每一段符号的具体含义、遵循的语法规则，以及C语言程序的运行调试和执行过程。只是让大家先看看到底C语言程序是通过哪些步骤解决实际问题的。

【例 2.1】通过键盘输入一个圆的半径，计算出这个圆的周长，并将结果显示在屏幕上。

要完成这个程序设计，需要以下4个步骤。

1. 问题分析

圆周长的数学公式： $C=2\pi r$ ，其中 π 为圆周率，取值为3.1415926； r 为圆的半径； C 为圆的周长。

2. 解决问题的方法

- 1) 输入半径值 r 。
- 2) 根据数学公式，计算圆的周长。
- 3) 输出结果。

3. 编写程序

【程序清单】

```
1:  /* e02-01.c:输入圆的半径，求周长 */
2:  #include<stdio.h>
3:  #define PI 3.1415926
4:
5:  int main(void)
6:  {
```

```
7:    double r,C;  
8:    scanf("%lf",&r);  
9:    C=2*PI*r;           //计算圆的周长  
10:   printf("输出周长值为: %f",C);  
11:   return 0;  
12: }
```

注意

程序清单中每行前面的行号与冒号只用于解释程序，在抄录源程序时不能录入。

这个程序并不复杂，我们逐行解释一下。

第 1 行是一条注释信息，用来说明与程序相关的内容。注释信息放在`/**/`这对符号中间，而且其中可以包含多行注释信息。

第 2 行是预处理指令，包含一个文件 `stdio.h`。

第 3 行也是预处理指令，定义用符号 `PI` 代表 3.1415926。

第 5 行中的 `main` 是函数的名字，表示主函数，第 6~12 行花括号括起来的部分是主函数的函数体，即主函数的执行操作部分。`main` 前面的 `int` 代表主函数执行完毕会返回一个整数。

第 7 行代码，程序中会用到两个变量 `r` 和 `C`，并且它们的值为实数，按照相应格式为其声明。

第 8 行代码，调用了 `scanf()` 函数，`scanf` 是函数的名字，执行输入操作，把输入的值存入变量 `r`。

第 9 行代码，完成了类似数学公式 $C=2\times\text{PI}\times r$ 的计算。用另外一种形式给出注释内容，即`//`符号后面的内容为注释信息。

第 10 行代码，调用了 `printf()` 函数，`printf` 是输出函数的名字，执行输出操作，输出内容到屏幕上。

第 11 行代码，返回一个 0 值，程序结束。

4. 编译运行

将此程序代码通过 Microsoft Visual Studio 2010 Express（简称 VC 2010）进行输入、编译，最后进行运行，得到运行结果如下：

2.3

输出周长值为: 14.451326

这个 C 语言小程序，为我们解决了一个数学方面的计算问题。通过这个程序，我们大致了解了编写一个简单 C 语言程序的思路，以及 C 语言程序的结构形式，并且对代码部分执行相关操作有了一个简单的认识。接下来，我们将深入研究组成 C 语言程序的各个部分。

2.2 深度剖析

通过【例 2.1】，我们已经对 C 语言程序有了初步的认识。下面以这个程序为依托，深度剖析 C 语言程序中各个部分的作用和意义，从而有利于读者宏观上对 C 语言程序的把握。在此基础上读者通过对后续课程的学习，对这些内容将会有更深刻的理解，经过反复练习和不断积累经验，就可以灵活运用这些内容，从而逐步掌握编写程序的方法。

2.2.1 C 语言程序的组成部分

1. C 语言程序的基本结构

C 语言程序是一个函数结构，即程序是由函数组成的。

函数的基本格式如下：

函数的返回值类型 函数名（参数列表）

```
{  
    函数体  
}
```

这种格式包括函数头部和函数体。函数体必须用一对 {} 括起来，函数体部分主要是实现程序功能的语句系列（关于语句的概念在第 5 章进行讲解），这些语句由声明部分和执行部分两部分组成，语句是 C 语言程序执行的最小单位，后面必须要有一个分号。

C 语言采用模块化的程序设计思想，模块通过函数来实现。当要解决复杂问题时将其划分成若干模块，每个模块编写成一个函数，通过主函数调用其他函数来组织运行过程（关于函数的具体内容在第 8 章做详细介绍）。

【例 2.1】中的程序第 5~12 行是主函数的定义部分。每个 C 语言程序必须有且仅有一个主函数，任何一个 C 语言程序都是从主函数的第一条语句开始执行，到主函数的最后一条语句结束的。主函数的函数名是 `main`，程序第 5 行是主函数的函数头部分，程序第 7~11 行是主函数的函数体部分。

函数头部分的 `int` 代表执行完主函数之后系统会返回一个整型量。

程序第 7 行属于声明部分，声明了两个变量（关于变量的概念将在第 3 章做详细讲解），`double` 说明这两个变量在程序运行期间值为小数，我们在编写程序时就是按照这样的格式来声明程序中所需要的变量的，位置一般写在函数体的开始部分。

程序第 8 行属于函数调用语句，调用了标准库中的输入函数 `scanf`，用来从键盘接收一个值给变量 `r`，我们在编写程序需要给变量赋值时可以按这种函数的格式来书写。`%lf` 对应变量的数据类型 `double`（相关概念在第 3 章、第 4 章做说明），`&r` 代表变量的地址（相关概念在第 13 章做说明），`r` 是用来接收数据的变量名。

程序第 9 行是一个表达式语句，执行了一个表达式操作 `C=2*PI*r`，先执行 `2*PI*r` 的操作（这里的 `*` 代表数学中的乘号），得到一个值，接下来通过 `=`（赋值运算符）将

此值赋予变量 C。

程序第 10 行是一个函数调用语句，调用了标准库中的输出函数 `printf`，输出一些信息到输出设备上。我们在编写程序时如果需要设定关于输出的操作步骤，可以通过调用 `printf` 函数来实现。

程序第 11 行，返回一个值 0，程序结束。

2. 注释

C 语言中，注释（comment）是用来向用户提供有关程序内容的说明信息。它可以说明程序的功能（如【例 2.1】中的第 1 行），也可以用来解释代码段的含义（如【例 2.1】中的第 9 行）。但注释并不属于程序的必备部分，程序可以没有注释。

【例 2.1】中包含了一行注释信息 “/* e02-01.c:输入圆的半径，求周长 */”，用来说明此程序文件名为 e02-01.c，程序功能是输入圆的半径求圆的周长。

C 语言还支持另一种注释方法：单行注释，即以 `//` 开头，注释内容放于其后，直到本行末尾（不能换行）。

【例 2.1】第 9 行通过这种方法进行注释：“`C=2*PI*r; //计算圆的周长`”，`//` 后面的注释内容用来说明前面语句的功能是求圆的周长。

3. 预处理指令

C 语言中通过符号 `#` 引导的指令叫预处理指令。预处理指令由预处理器在编译阶段执行（编译的概念在 2.3 节进行说明）。

`#include` 属于预处理指令中的文件包含指令，其一般格式为

```
#include <文件名>
```

或

```
#include "文件名"
```

该指令指示预处理器将文件名对应的文件内容插入到当前代码位置处。文件包含指令中符号 `<>` 和 `" "` 的区别在于，“`"`”除了包含头文件之外，还可以包含自己编写的文件。

【例 2.1】中用到预处理指令 `#include <stdio.h>`，文件名为 `stdio.h`，这是一个头文件。至于为什么要包含 `stdio.h`，那是因为主函数的函数体部分调用了库函数 `scanf()` 和 `printf()`，这两个函数属于 C 语言标准库中的输入/输出函数，而关于它们的原型就包含在文件 `stdio.h` 中。换句话说，如果程序调用了包含在其他头文件中的库函数，那么此处的文件名就应该换成别的文件名。

`#define` 属于预处理指令中的宏定义，宏定义的一般格式为

```
#define 标识符 字符串
```

此处的标识符也被称为宏名（关于标识符的命名规则后续内容将会说明）。可以认为宏名是代表字符串的一个名字，字符串可以是数值型数据，也可以是一个表达式，还可以是若干字符组成的普通字符串等，总之该指令是通过预处理器将程序中出现宏名的地方用字符串中的内容进行替换。

【例 2.1】中用到的预处理指令 `#define PI 3.1415926`，宏名定义为 `PI`，字符串为