



普通高等教育“十一五”国家级规划教材

全国高校出版社优秀畅销书一等奖

中国高等院校计算机基础教育课程体系规划教材

丛书主编 谭浩强

# C 程序设计教程（第3版）学习辅导

谭浩强 编著

- 《C程序设计》（发行逾1400万册）的姊妹篇
- 内容更精练，重点更突出
- 使C语言更容易学习
- 紧扣最基本的教学要求

清华大学出版社





普通高等教育“十一五”国家级规划教材

全国高校出版社优秀畅销书一等奖

中国高等院校计算机基础教育课程体系规划教材

丛书主编 谭浩强

# C程序设计教程 (第3版) 学习辅导

谭浩强 编著

清华大学出版社  
北京



## 内 容 简 介

本书是与谭浩强所著的《C 程序设计教程(第3版)》(清华大学出版社出版)配合使用的参考书,全书可分4个部分。第一部分为《C 程序设计教程(第3版)》习题与参考解答,包括了该书各章的全部习题,对全部编程习题都给出了参考解答,包括114个程序;第二部分为常见错误分析和程序调试;第三部分为C语言上机指南,详细介绍了利用 Visual C++ 6.0 集成环境和 Visual Studio 2010 编辑、编译、调试和运行程序的方法;第四部分为上机实验,提供了学习本课程应当进行的12个实验。

本书内容丰富,实用性强,是学习C语言的一本好参考书,可作为《C 程序设计教程(第3版)》和其他C语言教材的参考书,既适合高等学校师生使用,也可供报考计算机等级考试者和其他自学者参考。

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 13501256678 13801310933

### 图书在版编目(CIP)数据

C 程序设计教程(第3版)学习辅导/谭浩强编著. —北京:清华大学出版社,2018  
(中国高等院校计算机基础教育课程体系规划教材)  
ISBN 978-7-302-50383-5

I. ①C… II. ①谭… III. ①C语言—程序设计—高等学校—教学参考资料 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第122969号

责任编辑:张民  
封面设计:何凤霞  
责任校对:李建庄  
责任印制:丛怀宇

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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

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

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:16

字 数:381千字

版 次:2018年8月第1版

印 次:2018年8月第1次印刷

定 价:39.00元

产品编号:078355-01

从20世纪70年代末、80年代初开始,我国的高等院校开始面向各个专业的全体大学生开展计算机教育。面向非计算机专业学生的计算机基础教育牵涉的专业面广、人数众多,影响深远,它将直接影响我国各行各业、各个领域计算机应用的发展水平。这是一项意义重大而且大有可为的工作,应该引起各方面的充分重视。

三十多年来,全国高等院校计算机基础教育研究会和全国高校从事计算机基础教育的老师始终不渝地在这片未被开垦的土地上辛勤工作,深入探索,努力开拓,积累了丰富的经验,初步形成了一套行之有效的课程体系和教学理念。高等院校计算机基础教育的发展经历了3个阶段:20世纪80年代是初创阶段,带有扫盲的性质,多数学校只开设一门入门课程;20世纪90年代是规范阶段,在全国范围内形成了按3个层次进行教学的课程体系,教学的广度和深度都有所发展;进入21世纪,开始了深化提高的第3阶段,需要在原有基础上再上一个新台阶。

在计算机基础教育的新阶段,要充分认识到计算机基础教育面临的挑战。

(1)在世界范围内信息技术以空前的速度迅猛发展,新的技术和新的方法层出不穷,要求高等院校计算机基础教育必须跟上信息技术发展的潮流,大力更新教学内容,用信息技术的新成就武装当今的大学生。

(2)我国国民经济现在处于持续快速稳定发展阶段,需要大力发展信息产业,加快经济与社会信息化的进程,这就迫切需要大批既熟悉本领域业务,又能熟练使用计算机,并能将信息技术应用于本领域的新型专门人才。因此需要大力提高高校计算机基础教育的水平,培养出数以百万计的计算机应用人才。

(3)21世纪,信息技术教育在我国中小学中全面开展,计算机教育的起点从大学下移到中小学。水涨船高,这样也为提高大学的计算机教育水平创造了十分有利的条件。

迎接21世纪的挑战,大力提高我国高等学校计算机基础教育的水平,培养出符合信息时代要求的人才,已成为广大计算机教育工作者的神圣使命和光荣职责。全国高等院校计算机基础教育研究会和清华大学出版社于2002年联合成立了“中国高等院校计算机基础教育改革课题研究组”,集中了一批长期在高校计算机基础教育领域从事教学和研究的专家、教授,经过深入调查研究,广泛征求意见,反复讨论修改,提出了高校计算机基础教育改革思路和课程方案,并于2004年7月发布了《中国高等院校计算机基础教育课程体系2004》(简称CFC 2004)。国内知名专家和从事计算机基础教育工作的广大教师一致认为CFC 2004提出了一个既体现先进性又切合实际的思路和解决方案,该研究成果具有开创性、针对性、前瞻性和可操作性,对发展我国高等院校的计算机基础教育具有重要的指导作用。根据近年来计算机基础教育的发展,课题研究

组先后于 2006、2008 和 2014 年发布了《中国高等院校计算机基础教育课程体系》的新版本,由清华大学出版社出版。

为了实现 CFC 提出的要求,必须有一批与之配套的教材。教材是实现教育思想和教学要求的重要保证,是教学改革中的一项重要的基本建设。如果没有好的教材,提高教学质量只是一句空话。要写好一本教材是不容易的,不仅需要掌握有关的科学技术知识,而且要熟悉自己工作的对象,研究读者的认识规律,善于组织教材内容,具有较好的文字功底,还需要学习一点教育学和心理学的知识等。一本好的计算机基础教材应当具备以下 5 个要素:

(1) 定位准确。要明确读者对象,要有的放矢,不要不问对象,提笔就写。

(2) 内容先进。要能反映计算机科学技术的新成果、新趋势。

(3) 取舍合理。要做到“该有的有,不该有的没有”,不要包罗万象、贪多求全,不应把教材写成手册。

(4) 体系得当。要针对非计算机专业学生的特点,精心设计教材体系,不仅使教材体现科学性和先进性,还要注意循序渐进,降低台阶,分散难点,使学生易于理解。

(5) 风格鲜明。要用通俗易懂的方法和语言叙述复杂的概念。善于运用形象思维,深入浅出,引人入胜。

为了推动各高校的教学,我们愿意与全国各地区、各学校的专家和老师共同奋斗,编写和出版一批具有中国特色的、符合非计算机专业学生特点的、受广大读者欢迎的优秀教材。为此,我们成立了“中国高等院校计算机基础教育课程体系规划教材”编审委员会,全面指导本套教材的编写工作。

本套教材具有以下几个特点:

(1) 全面体现 CFC 的思路和课程要求。可以说,本套教材是 CFC 的具体化。

(2) 教材内容体现了信息技术发展的趋势。由于信息技术发展迅速,教材需要不断更新内容,推陈出新。本套教材力求反映信息技术领域中新的发展、新的应用。

(3) 按照非计算机专业学生的特点构建课程内容和教材体系,强调面向应用,注重培养应用能力,针对多数学生的认知规律,尽量采用通俗易懂的方法说明复杂的概念,使学生易于学习。

(4) 考虑到教学对象不同,本套教材包括了各方面所需要的教材(重点课程和一般课程,必修课和选修课,理论课和实践课),供不同学校、不同专业的学生选用。

(5) 本套教材的作者都有较高的学术造诣,有丰富的计算机基础教育的经验,在教材中体现了研究所倡导的思路和风格,因而符合教学实践,便于采用。

本套教材统一规划,分批组织,陆续出版。希望能得到各位专家、老师和读者的指正,我们将根据计算机技术的发展和广大师生的宝贵意见及时修订,使之不断完善。

全国高等院校计算机基础教育研究会荣誉会长  
“中国高等院校计算机基础教育课程体系规划教材”编审委员会主任

谭浩强

C语言是国内外广泛使用的计算机语言。许多高校都开设了“C语言程序设计”课程。作者于1991年编写了《C程序设计》，由清华大学出版社出版，该书出版后，受到了广大读者的欢迎，大家普遍认为该书概念清晰、叙述详尽、例题丰富、深入浅出、通俗易懂，大多数高校选其作为教材。该书已先后修订了4次，出了5版，重印200多次，累计发行1400多万册，为国内外同类书之首。

由于全国各地、各类学校情况不尽相同，对C语言的教学要求学时数也有所差别。因此，作者除了编写出版《C程序设计》外，还针对部分学时较少的学校的情况，于2007年编写出版《C程序设计教程》一书，适当减少内容，紧扣基本要求，突出重点，同时编写出版了《C程序设计教程学习辅导》一书。2018年，作者对《C程序设计教程》一书再次进行修订，出版了《C程序设计教程（第3版）》，同时也对《C程序设计教程学习辅导》进行了修订，出版了《C程序设计教程（第3版）学习辅导》，即本书。

本书包括4个部分。

第一部分是“《C程序设计教程（第3版）》习题与参考解答”。在这一部分中包括了《C程序设计教程（第3版）》一书的全部习题，并提供了参考解答。除对其少数概念问答题，由于能在教材中直接找到答案，不另给出答案外，对所有编程题一律给出参考解答，包括程序清单和运行结果。对于相对简单的问题，只给出程序清单和运行结果，不作详细说明，以便给读者留下思考的空间。对于一些比较复杂的问题还对算法进行了详细的分析介绍，给出流程图，并在程序中加入注释以便于读者理解。对少数难度较大的题目(如链表的插入、删除等)还作了比较详细的文字说明。对有些题目，我们给出了两种参考解答，供读者参考和比较，以启发思路。全部程序都在Visual C++ 6.0环境下调试通过。

本书提供的114个不同类型、不同难度的题目，可以作为读者学习《C程序设计教程（第3版）》的补充材料。希望教师和同学们能善于利用这部分内容。教师可以根据教学要求和学生的情况，从习题中指定学生必须完成若干题目，目的是希望通过完成这些习题巩固消化教材中的内容，同时能帮助读者举一反三，深入思考。

习题中包括了不少有价值的、适用于教学的典型问题。由于篇幅和课时的限制，在教材和讲授中不可能介绍很多例子，只能介绍一些典型的例题。教师可以从习题中选一些，作为补充例题在课堂上讲授。对于一般读者，不必要求他们全部完成这些题

目,但是可以提倡他们直接阅读参考解答,这对于开阔眼界,丰富知识,理解不同的程序,了解有关算法,掌握编程思路,是大有裨益的。对 C 程序设计有兴趣、有基础的读者,最好能多选一些题目独立完成编程,对提高编程水平会有很大的帮助。有人说:如果能独立完成这些题目的编程,学习 C 语言就基本过关了。

应该说明,本书给出的程序并非是唯一正确的解答,甚至不一定是最佳的一种。对同一个题目可以编出多种程序,我们给出的只是其中的一种或两种,而且程序尽可能便于初学者容易理解,不一定直接搬用于专业的程序设计中。读者在使用本书时,千万不要照抄照搬,读者完全可以编写出更好、更专业的程序。

第二部分是“常见错误分析和程序调试”。作者根据多年教学经验,总结了学生在编写程序时常出现的问题,以提醒读者少犯类似错误。此外,介绍了调试程序的知识和方法,为上机实验打下基础。

第三部分是“C 语言上机指南”。介绍了目前常用的 Visual C++ 6.0 和 Visual Studio 2010 集成环境下的上机方法,使读者上机练习有所遵循。

第四部分是“上机实验”。在这部分中提出了上机实验的要求,介绍了程序调试和测试的初步知识,并且安排了 12 个实验,便于进行实验教学。

本书不仅可以作为《C 程序设计教程(第 3 版)》的参考书,而且可以作为任何 C 语言教材的参考书;既适用于高等学校教学,也可供报考计算机等级考试者和其他自学者参考。

南京大学金莹老师以及薛淑斌高级工程师和谭亦峰工程师参加本书的收集材料以及部分编写和调试程序的工作。

本书难免会有错误和不足之处,作者希望得到广大读者的指正。

谭亦峰

2018 年 5 月于清华园

## 第一部分 《C 程序设计教程(第 3 版)》习题与参考解答

第 1 章	程序设计和 C 语言 .....	3
第 2 章	最简单的 C 程序设计——顺序程序设计 .....	8
第 3 章	选择结构程序设计 .....	16
第 4 章	循环结构程序设计 .....	25
第 5 章	利用数组处理批量数据 .....	42
第 6 章	利用函数进行模块化程序设计 .....	64
第 7 章	善于使用指针 .....	90
第 8 章	根据需要创建数据类型 .....	114
第 9 章	利用文件保存数据 .....	147

## 第二部分 常见错误分析和程序调试

第 10 章	常见错误分析 .....	169
第 11 章	程序的调试与测试 .....	185
11.1	程序的调试 .....	185
11.2	程序错误的类型 .....	187
11.3	程序的测试 .....	189

## 第三部分 C 语言上机指南

第 12 章	Visual C++ 6.0 的上机操作 .....	197
12.1	Visual C++ 6.0 的安装和启动 .....	197
12.2	输入和编辑源程序 .....	198



12.2.1	新建一个 C 源程序的方法 .....	198
12.2.2	打开一个已有的程序 .....	200
12.2.3	通过已有的程序建立一个新程序的方法 .....	200
12.3	编译、连接和运行 .....	201
12.3.1	程序的编译 .....	201
12.3.2	程序的调试 .....	202
12.3.3	程序的连接 .....	204
12.3.4	程序的执行 .....	205
12.4	建立和运行包含多个文件的程序的方法 .....	206
12.4.1	由用户建立项目工作区和项目文件 .....	206
12.4.2	用户只建立项目文件 .....	211
<b>第 13 章</b>	<b>用 Visual Studio 2010 运行 C 程序 .....</b>	<b>213</b>
13.1	关于 Visual Studio 2010 .....	213
13.2	怎样建立新项目 .....	213
13.3	怎样建立文件 .....	217
13.4	怎样进行编译 .....	219
13.5	怎样运行程序 .....	220
13.6	怎样打开项目中已有的文件 .....	221
13.7	怎样编辑和运行一个包含多文件的程序 .....	222
13.8	关于用 Visual Studio 2010 编写和运行 C 程序的说明 .....	225
<b>第四部分 上机实验</b>		
<b>第 14 章</b>	<b>实验指导 .....</b>	<b>229</b>
14.1	上机实验的目的 .....	229
14.2	上机实验前的准备工作 .....	230
14.3	上机实验的步骤 .....	230
14.4	实验报告 .....	231
14.5	实验内容安排的原则 .....	231
<b>第 15 章</b>	<b>实验安排 .....</b>	<b>232</b>
15.1	实验 1 C 程序的运行环境和运行 C 程序的方法 .....	232
15.2	实验 2 最简单的 C 程序设计——顺序程序设计 .....	234
15.3	实验 3 选择结构程序设计 .....	236
15.4	实验 4 循环结构程序设计 .....	237
15.5	实验 5 利用数组(一) .....	238
15.6	实验 6 利用数组(二) .....	239

15.7 实验7 函数调用(一) .....	240
15.8 实验8 函数调用(二) .....	241
15.9 实验9 善用指针(一) .....	242
15.10 实验10 善用指针(二) .....	243
15.11 实验11 使用结构体 .....	244
15.12 实验12 文件操作 .....	245
参考文献 .....	246

**第一部分**

---

**《C 程序设计教程(第 3 版)》  
习题与参考解答**





# 第1章

## 程序设计和 C 语言

1.1 上机运行本章 3 个例题,熟悉所用系统的上机方法与步骤。

解:略。

1.2 请参照本章例题,编写一个 C 程序,输出以下信息:

```
*****  
Very Good!  
*****
```

解:编写程序如下:

```
#include <stdio.h>  
int main()  
{ printf("*****\n\n");  
  printf("Very Good!\n\n");  
  printf("*****\n");  
  return 0;  
}
```

运行结果:

```
*****  
Very Good!  
*****
```

1.3 编写一个 C 程序,输入 a,b,c 三个值,输出其中最大者。

解:编写程序如下:

```
#include <stdio.h>  
int main()  
{ int a,b,c,max;  
  printf("please input a,b,c: \n");  
  scanf("%d,%d,%d",&a,&b,&c);  
  max = a;  
  if (max < b)  
    max = b;  
  if (max < c)
```

```

    max = c;
    printf("The largest number is %d\n",max);
    return 0;
}

```

运行结果:

please input a,b,c:

18, -43, 34 ↵

The largest number is 34

1.4 先后输入 50 个学生的学号和成绩,要求将其中成绩在 80 分以上的学生的序号和成绩立即输出。请用传统流程图表示其算法。

解: 见图 1-1。

1.5 求  $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{99} + \frac{1}{100}$ 。请用传统流程图和结构化流程图表示其算法。

解: 传统流程图见图 1-2, 结构化流程图见图 1-3。

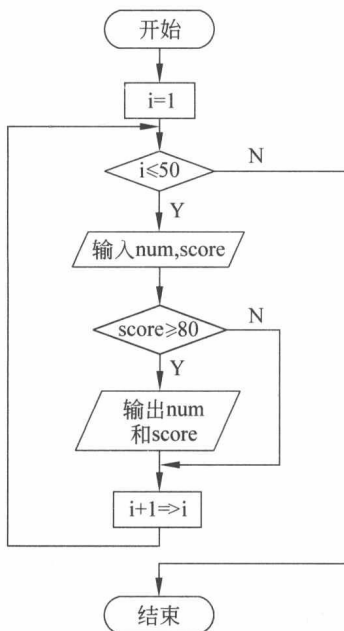


图 1-1

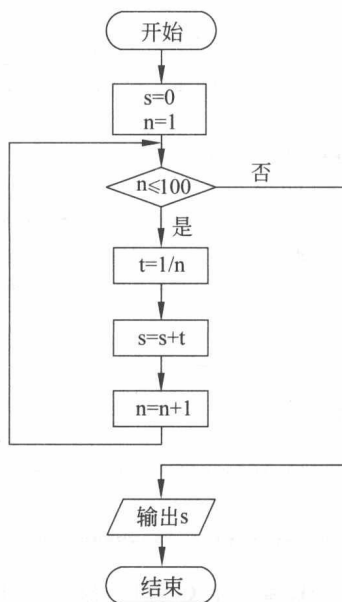


图 1-2

1.6 输入一个年份 year, 判定它是否是闰年, 并输出它是否是闰年的信息。请用结构化流程图表示其算法。

解: 闰年的条件是符合下面二者之一:

- ① 能被 4 整除, 但不能被 100 整除, 如 2016。
- ② 能被 4 整除, 又能被 400 整除, 如 2000 (注意, 能被 100 整除, 不能被 400 整除的年份不是闰年, 如

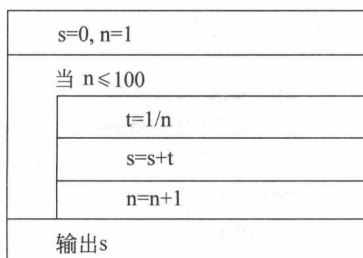


图 1-3

2100)。

画出 N-S 流程图,见图 1-4。

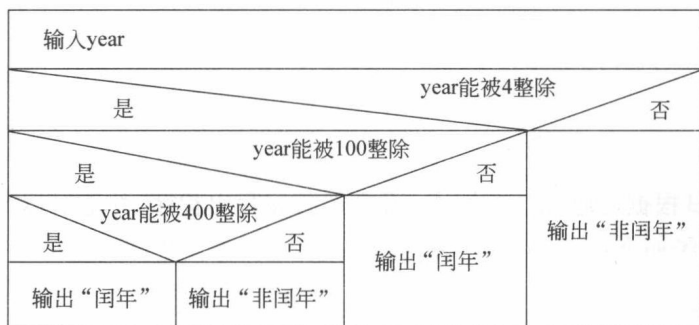


图 1-4

1.7 给出一个大于或等于3的正整数,判断它是不是一个素数。请用伪代码表示其算法。

**解:** 所谓素数(prime number)是指除了1和它本身之外,不能被其他任何整数整除的整数。例如17是一个素数,因为它不能被2~16的任何整数整除。而21不是素数,因为它能被3和7整除。要判定一个整数m是否是素数,只要把m被2~(m-1)的各整数整除,如果都除不尽,m就是素数。用伪代码表示的算法如下:

begin	(算法开始)
input m	(输入m)
i = 2	(除数从2开始)
while (i <= m - 1)	(一直进行到被m-1除)
{if (m%i is equal to 0) flag = 1	(如果m被i整除,使标志flag的值为1)
i = i + 1	(i加1,准备下一次循环)
}	
if flag is equal to 1 ,print m is not a prime number	(如果flag的值为1,输出m不是素数)
else m is a prime number	(否则m是素数)
end	(算法结束)

**说明:** 用伪代码写算法时,上面右侧括号内的说明是不需要的。由于有的读者对用伪代码表示算法不太习惯,所以在此加上必要的说明。从上面可以看到,用伪代码写算法,书写灵活,格式自由,修改方便,中英文均可,它是写给人看的(不是让计算机执行的),只要自己和别人能看懂就行。专业人员一般喜欢用伪代码,尽量写得接近计算机语言的形式,以便容易转换为源程序。

1.8 请尝试根据习题1.4的算法,用C语言编写出程序,并上机运行。

**解:** 编写程序如下:

```
#include <stdio.h>
int main()
{ int i,num,score;
  i = 1;
```

```

while (i<=50)
{scanf("%d,%d",&num,&score);
  if(score>=80) printf("%d,%d\n",num,score);
  i=i+1;
}
return 0;
}

```

1.9 请尝试根据习题 1.5 的算法,用 C 语言编写出程序,并上机运行。

解: 编写程序如下:


```

#include<stdio.h>
int main()
{ int n;
  float sum,term;
  sum=0;
  n=1;
  while (n<=100)
  { term=1.0/n; //term 代表多项式中的某一项的值
    sum=sum+t; //把各项累加到 sum 中
    n=n+1; //使 n 的值加 1,准备求下一项
  }
  printf("%f\n",sum); //输出总和
  return 0;
}

```

运行结果:

5.187378

 **说明:** 第 8 行“term=1.0/n;”,分子是 1.0,表示是实数,如果写成“term=1/n;”,在 C 语言中规定两个整数相除,结果是整数,因此当 n>1 时,1/n 的值总是等于 0,最后结果显然不正确。读者可以上机试验一下。关于这个问题,在学习第 2 章之后会进一步理解的。

1.10 请尝试根据习题 1.6 的算法,用 C 语言编写出程序,并上机运行。

解: 编写程序如下:

```

#include<stdio.h>
int main()
{ int year;
  scanf("%d",&year);
  if(year%4==0) //若 year 能被 4 整除
  {if(year%100==0) //还能被 100 整除
    if(year%400==0) //还能被 400 整除
      printf("%d is a leap year.\n",year); //是闰年
    else printf("%d is not a leap year.\n",year);
  } //不能被 400 整除的不是闰年
  else printf("%d is a leap year.\n",year); //能被 4 整除不能被 100 整除的是闰年
}

```



```

    }
    else printf("%d is not a leap year.\n",year); //不能被4整除的不是闰年
    return 0;
}

```

运行结果:

```

2100 ✓
2100 is not a leap year.

```

1.11 请尝试根据习题1.7的算法,用C语言编写出程序,并上机运行。

解:编写程序如下:

```

#include<stdio.h>
int main()
{ int m,i,flag;
  scanf("%d",&m); //输入要检测的整数
  i=2;
  while (i <=m-1)
  {if (m%i ==0) flag=1;
   i=i+1;
  }
  if (flag==1) printf("%d is not a prime number.\n",m);
  else printf("%d is a prime number.\n",m);
  return 0;
}


```

运行结果:

```

17 ✓
17 is a prime number.


```

 **程序分析:** 实际上, $m$ 不必被 $2 \sim (m-1)$ 的全部整数除,只要被 $2 \sim m$ 的平方根之间的全部整数除即可。例如,为了判别17是否素数,只要把17被2,3,4除即可。请读者思考是什么原因?程序第6行可改为

```
while (i <= sqrt(m)) //sqrt是求平方根的函数
```

如果程序中使用了C函数库中的数学函数(包括sqrt),应在程序开头写预处理指令:

```
#include<math.h>
```

 **说明:** 第1章是学习C程序设计的预备知识,并未系统介绍C语言的语法知识以及算法和编程的知识。本章的习题,目的是使读者尽早接触算法,接触程序。习题1.8~习题1.11是编程题,可能许多读者感到有些困难,希望读者能尝试一下,即使编写的程序有些问题,也没关系,可以提高对程序的兴趣,培养主动学习、善于发展知识的创造精神。如果确实编程有困难,也可以直接阅读上面的程序,能大体看懂程序就是一个收获,可以为后面的学习打下较好的基础。