



彭旭东
王成霞 编著
万 红

程序设计教程(C/C++版)上机指导和习题解析



- ◆ 教材习题解析
- ◆ 上机实验和指导
- ◆ 丰富的补充习题
- ◆ 标准模拟试卷



清华大学出版社

高等院校计算机应用技术系列教材

程序设计教程 (C/C++版)

上机指导和习题解析

彭旭东 王成霞 万红 编著

清华大学出版社

北 京

内 容 简 介

本书是《程序设计教程(C/C++版)》的配套资料,也是教程内容的补充和延续。内容主要包括 5 部分:教材习题解答、上机指导、上机实验与指导、补充习题和模拟试题。在附录部分提供了补充习题和模拟试题的参考答案。

本书习题丰富,上机实验内容深入浅出,循序渐进,便于自学,是辅导教师和学生必备的工具书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

程序设计教程(C/C++版)上机指导和习题解析/彭旭东,王成霞,万红编著. —北京:清华大学出版社,2005.6
(高等院校计算机应用技术系列教材)

ISBN 7-302-10747-5

I.程… II.①彭… ②王… ③万… III.C 语言—程序设计—高等学校—教学参考资料 IV.TP312

中国版本图书馆 CIP 数据核字(2005)第 027339 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦
http://www.tup.com.cn 邮 编:100084
社 总 机:010-62770175 客 户 服 务:010-62776969

组稿编辑:胡伟卷

文稿编辑:刘金喜

封面设计:王永

版式设计:康 博

印 刷 者:北京牛山世兴印刷厂

装 订 者:三河市李旗庄少明装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:14.5 字数:335 千字

版 次:2005 年 6 月第 1 版 2005 年 6 月第 1 次印刷

书 号:ISBN 7-302-10747-5/TP·7158

印 数:1~4000

定 价:21.00 元

前 言

本书是《程序设计教程(C/C++版)》(以下简称为《教程》)的配套书,同时,它也可以单独作为学习 C 程序设计的辅导教材。

当前流行的 C 和 C++ 的编译系统有很多。本书在“上机指导”部分介绍的是 Turbo C 2.0 和 Visual C++ 6.0 两个系统的安装步骤、编辑方法、获取帮助信息的方法和简单的程序调试技术。

“上机实验与指导”部分首先介绍了上机实验的具体步骤,然后围绕教程的进度,安排了 16 个实验,每个实验都给出了实验的目的和题目。前几个实验给出了比较详细的操作步骤,便于初学者入门。此外,作为教材内容的补充,本书在一些实验中还安排了要求学生掌握的算法。

“补充习题”部分针对教程中的多数章节给出了大量的习题,供学生平时练习和复习使用。

为了测试学生对知识的掌握情况,我们仿照目前流行的等级考试的形式给出了两套模拟试题。当然,本书并不针对任何等级考试,所以模拟试题中不含有与 C 程序设计无关的内容。

在编写本书的过程中,我们得到了天津理工大学计算机科学与工程系领导的大力支持。清华大学出版社的胡伟卷女士、天津理工大学的莫秀良老师也为本书的出版提供了大量的帮助,在此一并表示感谢。

作 者

目 录

第 I 部分 教材习题解答	1
第 1 章 概述	1
第 2 章 结构化程序设计与小规模软件设计初步	1
第 3 章 C 程序设计基础	1
第 4 章 简单的数据计算	2
第 5 章 程序控制结构基础	3
第 6 章 选择结构	3
第 7 章 循环结构	4
第 8 章 数组及其应用	6
第 9 章 字符与字符串	7
第 10 章 指针	9
第 11 章 函数	11
第 12 章 位运算	12
第 13 章 复合数据类型	12
第 14 章 文件	14
第 15 章 编译预处理	16
第 16 章 C++ 与面向对象程序设计	16
第 17 章 链表和堆栈	17
第 18 章 排序	18
第 19 章 查找	20
第 II 部分 上机指导	22
一、Turbo C 2.0	22
安装	22
启动与退出	22
编辑	23
1. 文件操作	24
2. 编辑操作	25
编译运行	26
调试初步	27
配置	29

帮助	29
工程	30
二、 Visual C++ 6.0	32
安装	32
启动与退出	33
建立工程	33
编辑	35
调试	36
帮助	37
第III部分 上机实验与指导	38
一、 上机实验的目的	38
二、 上机实验的步骤	38
【实验 1】 熟悉 C 语言的运行环境和 C 程序的运行方法	39
【实验 2】 算术运算符与表达式	40
【实验 3】 逻辑运算、关系运算和选择结构	43
【实验 4】 循环结构	43
【实验 5】 数组	45
【实验 6】 字符与字符串	49
【实验 7】 指针	53
【实验 8】 函数	55
【实验 9】 位运算	58
【实验 10】 复合数据类型	59
【实验 11】 文件	61
【实验 12】 编译预处理	62
【实验 13】 C++与面向对象	63
【实验 14】 链表和堆栈	64
【实验 15】 排序	65
【实验 16】 查找	67
第IV部分 补充习题	68
第 4 章 简单的数据计算	68
第 5 章 程序控制结构基础	74
第 6 章 选择结构	77
第 7 章 循环结构	84
第 8 章 数组及其应用	91
第 9 章 字符与字符串	96
单元测试一	101

第 10 章 指针	108
第 11 章 函数	113
第 12 章 位运算	119
第 13 章 复合数据类型	122
第 14 章 文件	130
第 15 章 编译预处理	134
单元测试二	141
第 17 章 链表和堆栈	152
第 18 章 排序	158
第 19 章 查找	162
单元测试三	164
第 V 部分 模拟试题	172
模拟试卷一	172
模拟试卷二	185
附录 A 补充习题参考答案	199
第 4 章 简单的数据计算	199
第 5 章 程序控制结构基础	200
第 6 章 选择结构	200
第 7 章 循环结构	201
第 8 章 数组及其应用	201
第 9 章 字符与字符串	202
单元测试一	202
第 10 章 指针	204
第 11 章 函数	204
第 12 章 位运算	205
第 13 章 复合数据类型	205
第 14 章 文件	206
第 15 章 编译预处理	206
单元测试二	207
第 17 章 链表和堆栈	208
第 18 章 排序	208
第 19 章 查找	209
单元测试三	209
附录 B 模拟试卷参考答案	211
模拟试卷一参考答案	211
模拟试卷二参考答案	212

附录 C 常见错误和警告信息	213
附录 D Turbo C 常用库函数	218

第 I 部分 教材习题解答

第 1 章 概 述

一、选择题

1. A 2. D

二、上机题

略

第 2 章 结构化程序设计与小规模软件设计初步

一、选择题

1. B 2. A 3. D

二、上机题

略

第 3 章 C 程序设计基础

一、选择题

1. A 2. D 3. D 4. B 5. A

二、填空题

1. /* */

2. 编译 obj

三、上机题

略

第4章 简单的数据计算

一、选择题

1. B 2. B 3. A 4. D 5. C 6. B 7. D 8. D 9. D 10. B

二、填空

1. 0~65535
2. 5 3 3.8 2
3. 1
4. 1234, 1234
01234,00001234
5. 1234.560,1234
123.456000, 123.46,123.46

三、上机题

1. 分析：变量定义时可以初始化，但不可以用一个初值连续初始化多个变量，程序第5行有错；变量a是int型，它可以存储的最大值是32767，程序第8行给变量a赋的值超过了32767，会导致数据丢失。程序应作如下修改。

第5行改为：float x=2.0,y=2.0;

第8行改为：a=8000;或者把变量a定义为长整型，用%ld格式输出。

2. 分析：用scanf输入数据时，需要把变量的地址作为参数，程序第7行有错；变量x和y是float型，只能用%f或%e格式输出，不能用%d格式输出，程序第10行输出格式错误。

第7行改为：scanf("%d%d",&a,&b);

第10行改为：printf("%f,%f\n",x,y);或 printf("%e,%e\n",x,y);

3. 分析：由华氏度转换为摄氏度的转换公式为： $C=5/9(F-32)$ 其中C代表摄氏度，F代表华氏度，由此可以推出把摄氏度转换为华氏度的转换公式： $F=9/5*C+32$ ，按问题的要求，输入以华氏度表示的温度值，带入公式计算出以华氏度表示的温度值，最后将转换后的值输出。温度值一般会有小数，所以定义两个float型变量f和c分别表示华氏度的值和摄氏度的值，输出时，指定输出宽度为5，保留一位小数。

可参考的程序如下：

```
#include <stdio.h>
main()
{
    float f,c;
    printf("please input temperature in degree Celsius:\n");
```

```
scanf("%f",&c);  
f=9.0/5.0*c+32;  
printf("the temperature in degree Fahrenheit is%5.1f\n",f);  
}
```

4. 分析: 用符号常量 PI 表示 π 值, 定义两个 float 型变量 r 和 s 分别表示圆的半径和面积。用 scanf 函数输入一个半径值给 r, 用公式 $s=PI*r*r$ 计算出面积赋值给 s, 然后输出 s 的值。

可供参考的程序如下:

```
#include<stdio.h>  
#define PI 3.1415  
main()  
{  
    float r,s;  
    printf("please input radius:");  
    scanf("%f",&r);  
    s=PI*r*r;  
    printf("area=%8.2f\n",s);  
}
```

第 5 章 程序控制结构基础

一、选择题

1. B 2. D B 3. C 4. D 5. C

二、填空题

1. 1,2,3,4,1,6,1

2. 4,1

3. 0

第 6 章 选择结构

一、选择题

1. A 2. C 3. D 4. C 5. B

二、填空题

1. 1,2,2

2. 123

3. 0

三、上机题

分析：定义变量 a、b、c 用来存放从键盘输入的方程各次项的系数，根据输入的一组系数计算方程的判别式 dt，如果 dt 的值是负的，则方程没有实根，程序结束；否则用求根公式计算方程的两个实根 x1 和 x2，并输出。

参考程序：

```
#include <math.h>
main()
{
    double a,b,c;
    double x1,x2,dt;
    scanf("%lf%lf%lf",&a,&b,&c);
    printf("%5.2f x^2 + %5.2f x + %5.2f = 0\n",a,b,c);
    dt=b*b-4*a*c;
    if(dt<0)
    {
        printf("dt<0\n");
        exit(-1);
    }
    dt=sqrt(dt);
    x1=(-b+dt)/2/a;
    x2=(-b-dt)/2/a;
    printf("x1 = %f; x2 = %f\n",x1,x2);
}
```

第7章 循环结构

一、选择题

1. 【1】C 【2】C 2. B 3. C 4. 【1】B 【2】A 5. B

二、填空题

1. -1

2. 6

3. 35

4. 9,6,3

5.

```
int s=0,d=1;
d=d*i;
s+d
```

三、上机题

1. 分析: 从题目给出的算法描述可知: 如果 y 和 y_l 异号时, 表明 x^* 在 $[x_l, x]$ 之间, 取 $x_r = x; y_r = y$; 否则如果 y 和 y_r 异号时, 表明 x^* 在 $[x, x_r]$ 之间, 取 $x_l = x; y_l = y$ 。由此可得出程序第 13 行和第 19 行中的判断关系式错误。应做如下修改:

第 13 行改为: `if(y*y_l<0)`

第 19 行改为: `else if(y*y_r<0)`

2. 分析: 定义变量 s 表示符号, 正数用 1 表示, 负数用 -1 表示, 因为多项式中正负项交替出现, 且第一项是正数, 所以每次只要使 s 取上一次的相反数就可以交替表示正负, 并且初始值为 1。用变量 i 存储分母, 第一项分母是 1, 因此, 置 i 的初值为 1, 后一项的分母比前一项的分母增加 2, 所以每求出一项值后, i 的值增加 2, 为求下一项值做好准备。变量 d 分别存储求出的各个项的绝对值, 初始值为第一项值 1, 以后用 $1.0/i$ 计算出下一项值, 直到 d 小于 10^{-6} 运算结束, 否则, 把 $d*s$ 累加到存放多项式和的变量 pi 中, 如此循环。循环结束后 pi 中得到的是 $\pi/4$, 计算 $pi*4$ 变得到 π 的近似值。

参考程序:

```
#include <math.h>
main()
{
    int s=1;
    unsigned long int i=1;
    double pi=0,d=1;
    do
    {
        pi+=s*d;
        s=-s;
        i+=2;
        d=1.0/i;
    } while(d>1e-6);
    pi=pi*4;
    printf("Result: PI = %10.7lf\n",pi);
}
```

3. 分析: 定义两个 `long int` 型变量 $f1$ 和 $f2$ 存储 Fibonacci 数列中相邻两项的值, 初始值分别存储第 1 项和第 2 项值(即 1 和 1)。用循环实现, 用 i 作为循环控制变量, 初值为 0, 终值为 19, 循环 20 次。每次输出两项值, 第一次输出 $f1$ 和 $f2$ 中的值, 输出的是第 1 项和第 2 项值, 然后用 $f1$ 和 $f2$ 求出第 3 项值存入 $f1$, 再用新的 $f1$ 和 $f2$ 求出第 4 项的值存入

f2, 此时, f1 和 f2 中分别存储的是第 3 项和第 4 项的值, 将其输出。如此反复就可以求出 Fibonacci 数列其他各项的值, 直到循环结束。另外, 为了使输出格式整齐, 除了用统一宽度输出格式控制之外, 还用 $i\%2==1$ 控制光标换行, 以确保每行输出 4 项数据。

参考程序:

```
main()
{
    long int f1=1,f2=1;
    int i=0;
    while(i<20)
    {
        printf("%15ld%15ld",f1,f2);
        f1=f1+f2;
        f2=f2+f1;
        if(i%2==1) printf("\n");
        i++;
    }
}
```

第 8 章 数组及其应用

一、选择题

1. D 2. D 3. B 4. D 5. A

二、填空题

1. 27

2. 【1】 $a[i++] = n\%10$; 【2】 $i=0; i<j; i++$

3. 【1】 $\text{int } s1=0, s2=0$; 【2】 $s1+a[i][i]$ 【3】 $s2+a[i][4-i]$

三、上机题

1. 分析: 程序中第 13~第 22 行构成的双重循环用来找出第 i 行($i=0, 1, 2, 3\cdots, M-1$)的最大值存入 max, max 的初始值是第 i 行的第 0 列元素, 然后令 $a[i][j]$ ($j=1, 2, 3\cdots, N-1$)分别与 max 比较, 从而找出该行的最大值。找任何一行的最大值时, max 的初始值都应该为本行的第 0 列元素值。每次内循环结束时, max 就是 i 的最大值, 如果 i 等于 0, 把 max 赋值给 min; 否则, 如果 min 大于 max, 则把 max 赋值给 min, 把它作为已经求出各行最大值中的最小值, 继续循环求下一行的最大值, 依次类推, 直到所有行的最大值都求完了, 外循环结束, min 中便是最终所求。基于上述分析, 应该对原程序修改如下:

第 12 行改为: $\text{max}=a[i][0]$; 并把该语句移动到第 15 行之前。

第 18 行改为: $\text{if}(i==0)$

2. 分析: 定义符号常量 N , 定义具有 N 个元素的整型数组 a , 把 N 个 20 以内的偶数存入数组 a 。为了便于验证, 把 a 中的各元素输出, 然后把要删除的数输入给变量 x , 用循环实现 x 与 a 中各元素的比较, 直到下标 p 等于 N 或比较到某个 $a[p]$ 与 x 相等为止。循环结束时, 若 p 等于 N , 则表明 x 不在 a 中, 输出 “Can not find it.”, 程序结束; 否则, 表明 $a[p]$ 与 x 相等, 应该删除 $a[p]$, 方法是把从下标 p 到 $N-1$ 之间的所有元素顺次向前移动一个元素位置。最后把删除后 a 中的各元素输出。

参考程序:

```
#define N 10
main()
{
    int a[N]={6,8,12,18,4,16,14,10,0,2},x;
    int i,p;
    for(i=0;i<N;i++)
        printf("%5d",a[i]);
    printf("\n");
    scanf("%d",&x);
    p=0;
    while(p<N&& a[p]!=x) p++;
    if(p==N)
    {
        printf("Cannot find it!\n");
        exit(0);
    }
    for(i=p;i<N-1;i++)
        a[i]=a[i+1];
    for(i=0;i<N-1;i++)
        printf("%5d",a[i]);
    printf("\n");
}
```

3. 略

第 9 章 字符与字符串

一、选择题

1. A 2. C 3. B 4. D 5. B

二、填空题

1. 6666666666

2. 【1】 $i=-1$; 【2】 $s2[i]=(s1[i]>='A' \& \& s1[i]<='Z')?s1[i]-'A'+a:s1[i]$;

三、上机题

1. 分析: 第 8~第 12 行的循环是把正数 n 的各个数位分离出来作为字符顺次存入数组 s 的各元素中, $n\%10$ 得到的是 n 中数据的个位数字, 而不是字符, 要想把它当字符处理应该加上 '0', 即应该修改第 10 行; 第一个循环结束时, 由 n 得到的字符串 s 与 n 中的各数字顺序相反, 应该把 s 中的字符串逆序, 第 17~第 20 行构成的循环实现此功能。下标 i 从 0 开始, j 从最后一个字符 ('\0' 之前的字符) 开始, 因此, 第 15 行应该实现把 $i-1$ 的值赋值给 j 。程序应做如下修改:

第 10 行改为: $s[i]=n\%10+'0'$;

第 15 行改为: $j=i-1$;

2. 分析: 加密程序可把输入字符串中的小写字母转换成其后第 3 个字符对应的大写字母。定义两个字符数组 $s1$ 和 $s2$ 分别存储输入的字符串和加密后的字符串, 把 $s1[i]$ 转换成对应的加密字符可以用公式: $(s1[i]-'a'+3)\%26+'A'$, 并将公式运算结果赋值给 $s2[i]$ 即可。

加密程序是把加了密的字符串还原。定义两个字符数组 $s2$ 和 $s3$ 分别存储输入的加密后字符串和经解密后的原字符串, 把 $s2[i]$ 还原成加密前字符可用的公式: $(s2[i]-'A'+23)\%26+'a'$, 并将公式运算结果赋值给 $s3[i]$ 即可。

参考程序(加密):

```
#include <stdio.h>
main()
{
    char s1[80],s2[80];
    int i;
    gets(s1);
    i=-1;
    do
    {
        i++;
        s2[i]=(s1[i]-'a'+3)%26+'A';
    } while(s1[i]!='\0');
    s2[i]='\0';
    printf("s1:%s\ns2:%s\n",s1,s2);
}
```

参考程序(解密):

```
#include <stdio.h>
main()
{
    char s2[80],s3[80];
    int i;
    gets(s2);
```



```

for(i=0;s2[i]!='\0';i++)
    s3[i]=(s2[i]-'A'+23)%26+'a';
s3[i]='\0';
printf("s2:%s\ns3:%s\n",s2,s3);
}

```

3. 分析: 定义一个字符数组 s 存储输入的字符串, 引入两个整型变量 i 和 j 作为下标, 分别从 s 的第 0 个字符和倒数第一个字符相向移动。初始时 i 为第 0 个字符的下标, 让 j 为 s 中最后一个字符(即 '\0' 前面一个字符)下标, 比较 $s[i]$ 和 $s[j]$, 如果二者相等, $i++$, $j--$, 再继续比较 $s[i]$ 和 $s[j]$, 直到 i 与 j 相遇或者 $s[i]$ 和 $s[j]$ 比较的结果不相等, 循环结束。如果循环结束后, i 与 j 未相遇(即 i 小于 j), 则 s 不是回文, 输出 “No.”; 否则是回文, 输出 “Yes.”。

参考程序:

```

#include <stdio.h>
main()
{
    char s[80];
    int i,j;
    gets(s);
    j=0;
    while(s[j]!='\0')j++;
    j--;
    i=0;
    while(i<j&& s[i]==s[j])
    {
        i++; j--;
    }
    if(i<j)printf("No.\n");
    else printf("Yes.\n");
}

```

第 10 章 指 针

一、选择题

1. B 2. C 3. D 4. A 5. D 6. A 7. B 8. C 9. A 10. C

二、填空题

1. 【1】 $p2=n-1$ 【2】 $p1<p2$

2. 【1】 $c[k]=a[i];i++;$ 或 $c[k]=a[i]; ++i;$ 或 $c[k]=a[i];i=i+1;$