工业和信息化部高等教育“十三五”规划教材

C语言程序设计实验指导

主　编　崔　健　袁　敏
副主编　逄锦聚　王兆堃　代令军
　　魏　东　于韶杰　胡凤珠

電子工業出版社
Publishing House of Electronics Industry
北京・BEIJING

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目(CIP)数据

C 语言程序设计实验指导 / 崔健, 袁敏主编. —北京: 电子工业出版社, 2017.8
ISBN 978-7-121-32551-9

Ⅰ. ①C… Ⅱ. ①崔… ②袁… 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(2017)第 203174 号

策划编辑：郝国栋
责任编辑：郝国栋
印　　刷：
装　　订：
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×1092　1/16　　印张：11　　字数：252 千字
版　　次：2017 年 8 月第 1 版
印　　次：2017 年 8 月第 1 次印刷
定　　价：22.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（0532）67772605，邮箱：majie@phei.com.cn

目 录

第一部分 实 验

第 1 章 C 语言程序设计概述 …… 1

实验目的 …… 1
实验案例 …… 1
实验内容 …… 6
实验总结与分析 …… 7

第 2 章 数据类型和表达式 …… 8

实验目的 …… 8
实验案例 …… 8
实验内容 …… 10
思考题答案 …… 12
实验总结与分析 …… 12

第 3 章 程序控制结构 …… 13

3.1 顺序结构 …… 13
实验目的…… 13
实验案例 …… 13
实验内容 …… 14
实验总结与分析 …… 15
3.2 选择结构 …… 15
实验目的 …… 15
实验案例 …… 16
实验内容 …… 19
实验总结与分析 …… 20
3.3 循环结构 …… 20
实验目的 …… 20
实验案例 …… 21
实验内容 …… 22
实验总结与分析…… 24

第 4 章　数组 ……25
4.1　一维数组 …… 25
实验目的…… 25
实验案例…… 25
实验内容…… 28
实验总结与分析…… 29
4.2　一维字符数组和字符串…… 30
实验目的…… 30
实验案例…… 30
实验内容…… 33
实验总结与分析…… 35
4.3　二维数组…… 35
实验目的…… 35
实验案例…… 35
实验内容…… 36
实验总结与分析…… 38
第 5 章　函数…… 39
5.1 函数的基本应用…… 39
实验目的…… 39
实验案例…… 39
实验内容…… 43
实验总结与分析…… 46
5.2　递归函数与数组名作函数参数…… 46
实验目的…… 46
实验案例…… 46
实验内容 …… 49
实验总结与分析 …… 53
第 6 章　指针…… 54
6.1　指针与数组…… 54
实验目的…… 54
实验案例…… 54
实验内容…… 56
实验总结与分析…… 58
6.2　指针与字符串…… 58
实验目的…… 58
实验案例…… 59
实验内容…… 60
实验总结与分析…… 63

第 7 章　构造数据类型与编译预处理 …… 64
实验目的…… 64
实验案例…… 64
实验内容…… 66
实验总结与分析…… 69
第 8 章　文件 …… 70
实验目的…… 70
实验案例…… 70
实验内容…… 71
实验总结与分析…… 74
第 9 章　等级考试二级 C 语言操作试题举例 …… 75
实验目的…… 75
实验案例…… 75
实验内容…… 78
实验总结与分析…… 81

第二部分　课堂练习及答案

第 2 章　数据类型和表达式 …… 82
第 3 章　程序控制结构…… 85
第 4 章　数组…… 99
第 5 章　函数…… 104
第 6 章　指针…… 110
第 7 章　构造数据类型与编译预处理 …… 121

第三部分　习题及参考答案

第 1 章　C 语言程序设计概述……128
第 2 章　数据类型和表达式……129
第 3 章　程序控制结构……131
第 4 章　数组……140
第 5 章　函数……145

第 6 章　指针……149
第 7 章　构造数据类型与编译预处理……156
第 8 章　文件……159

实验报告(格式参考)……166

第一部分　实　　验

第 1 章　C 语言程序设计概述

实验目的

1. 熟悉 C 语言编译环境 VC++ 6.0，掌握一个 C 语言程序从编辑、编译、连接到最后运行的全过程。

2. 了解 C 语言程序的基本框架，并掌握其规范要求，能够编写简单的 C 语言程序。

3. 理解程序调试的思想，能借助于 VC++ 6.0 编译环境修改程序中出现的简单语法错误。

实验案例

【例 1.1】在屏幕上显示一个短句“This is my first program!”。

1. 解题步骤

编写一个完整的 C 语言程序主要有以下 6 个步骤：

① 确定程序的目的或作用。

② 分析问题。

③ 绘制程序流程图。

④ 根据流程图编写程序源代码。

⑤ 利用一种 C 语言编译器编译、连接程序。

⑥ 执行程序。

2. 操作步骤

第 1 步：建立自己的文件夹。

在磁盘(最好是除系统盘分区外的其他磁盘分区)上新建一个文件夹，用于存放要编写的 C 程序，如“E:\2012021051 王某某”。

第 2 步：编辑 C 语言源程序。

可以采用各种编辑器编辑 C 语言源程序，包括最简单的“记事本”程序，还可以采用专门的程序编辑器 UltraEdit，Editplus 等，另外还有一些 C 语言的 IDE 环境中自带的编辑器。本书中采用 Visual C++ 6.0 集成环境，具体操作步骤如下：

① 启动 VC++ 6.0：执行“开始”→“程序”→“Microsoft Visual Studio 6.0”→“Microsoft

Visual C++ 6.0”，进入 VC++ 6.0 编译环境。启动后界面如图 1.1 所示。

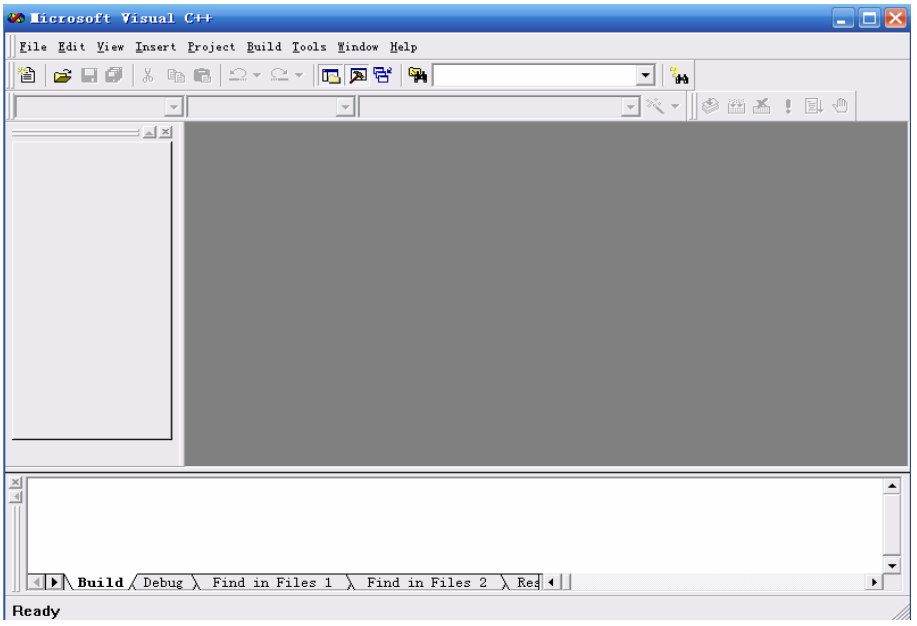


图 1.1 **Visual C++ 6.0 启动界面**

② 建立工程：工程类型应为“Win32 Console Application”，即 Win32 控制台应用程序。所有的程序必须依托在工程下才能运行，工程保存位置可以设置为第 1 步操作中建立的文件夹下的子文件夹，如图 1.2 所示。单击 OK 按钮，进入图 1.3 所示的界面。

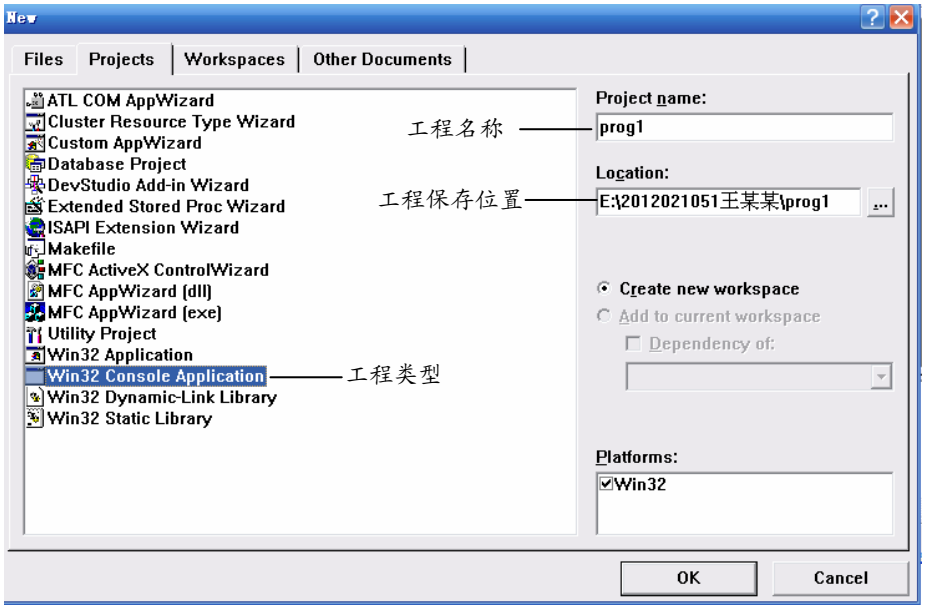


图 1.2 创建工程的界面

③ 建立 C 语言源程序文件：文件类型应选择“C++ Source File”；文件保存位置选择为上一步所示文件夹，若选择“Add to project”，文件会自动保存在建立的工程中；文件名可命名为 first.c，如图 1.3 所示。C 语言的源程序文件扩展名为.c，在命名时，如果省略文件的

扩展名，将默认为.cpp 文件。单击 OK 按钮，进入 Visual C++ 6.0 的程序编辑界面。

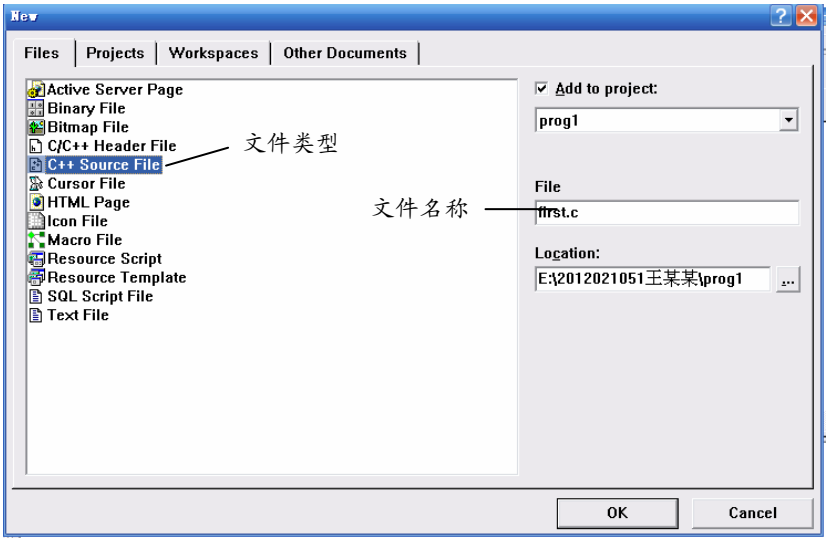


图 1.3　创建文件界面

④ 编辑源程序。在编辑窗口中输入源程序代码，并保存源程序文件。

参考代码：

```
1   /****************************************************
2   文件名：first.c
3   作者：***
4   日期：2013-7-21
5   功能：在屏幕上输出字符串"This is my first program!"
6   ****************************************************/
7   #include<stdio.h>                          /*编译预处理命令*/
8   int main(void)                             /*主函数*/
9   {
10      printf("This is my first program!");        /*在屏幕上输出字符串*/
11      return 0;
12  }
```

【说明】以上程序代码每行行首的数字在本书中代表行号，是额外添加的，用 VC++ 6.0 编写程序时，不需要加以上数字。

第 3 步：编译。

单击工具栏中的“编译”按钮，如图 1.4 所示，或执行“Build”→“Compile”菜单命令。编译后会在编辑窗口的下方显示编译信息。如果此时源程序文件有错误，则应改正，否则无法继续进行下一步操作。如果编译成功将生成.obj 文件。

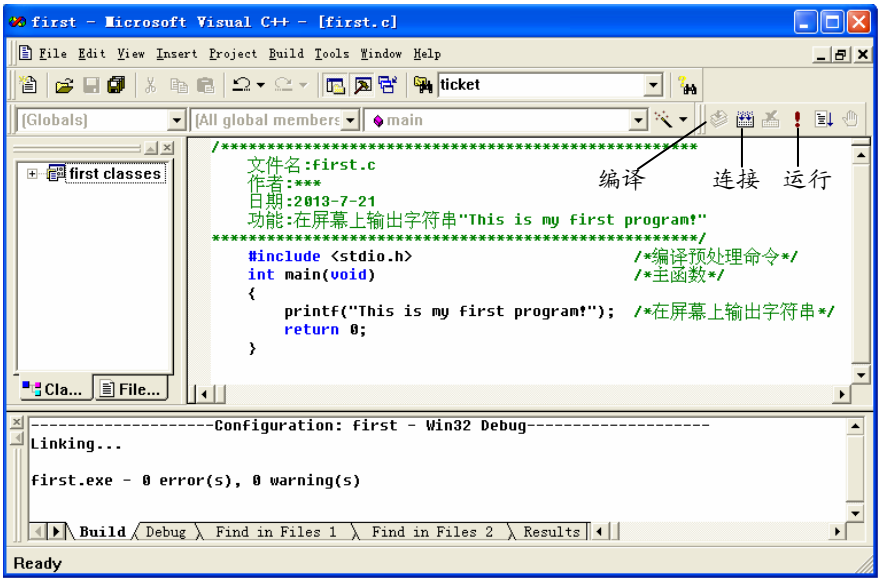


图 1.4　编辑界面

第 4 步：连接。

单击工具栏中的连接按钮，或执行“Build”→“Build”菜单命令，在环境界面的下方会显示连接信息。如果连接过程有错误，将显示错误原因。如果连接成功，将生成.exe 可执行文件。

第 5 步：运行。

单击工具栏中的运行按钮，或执行“Build”→“Execute”菜单命令，将显示运行结果窗口。若运行结果正确，则程序运行完成；若运行结果不正确，还要进行调试，直到结果正确为止。

第 6 步：调试。

C 语言程序在编译、连接过程中如果出现错误，即表示源代码程序中有 bug，为了能继续对目标代码进行编译、连接，达到正常运行程序的目的，必须找出相应的 bug，并加以修改，这个过程就称为 debugging，即调试。

【例 1.2】编写程序，输出以下两行短句。

Hello World!

This is my first program!

参考代码：

```
1  /*********************************************************************
2  文件名：second.c
3  作者：***
4  日期：2013-7-21
5  功能：在屏幕上输出两行字符串，分别为 Hello world!和 This is my first program!
6  *********************************************************************/
7
8  #include<stdio.h>                                  /*编译预处理命令*/
9  int main(void)                                     /*主函数*/
```

```
10    {
11        printf("Hello  World！ ");                    /*在屏幕上输出字符串*/
12        printf("This  is  my  first  program！ ")
13        return  0;
14    }
```

对以上程序源代码编译后，界面上给出程序中有错误的信息，如图 1.5 所示。在提示中，给出了错误所在的行，错误的类型及原因，可根据行号提示或双击该错误提示行，查找出现错误的源代码。并根据发生错误的原因对错误进行修改，直到没有错误为止。

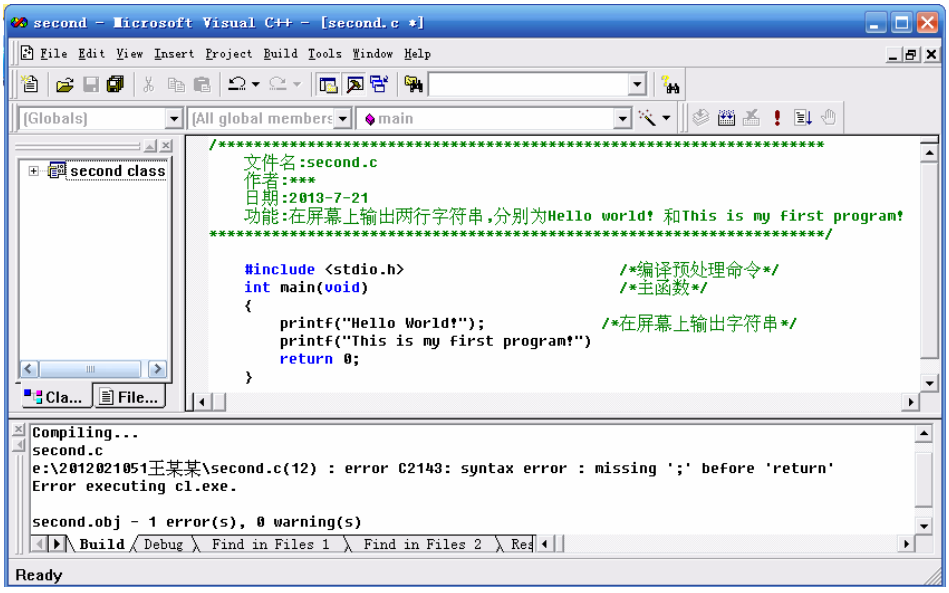


图 1.5　提示发现编译错误的界面

在信息窗口中，双击错误信息，编辑窗口就会出现一个箭头指向程序出错的位置。一般在箭头所指的当前行或上一行可以找到出错语句。本例中错误的原因指示在第 13 行，属于语法错误(syntax error)

missing ';' before 'return'

该提示指出，在第 13 行的 return 之前缺少分号，实际上是第 12 行的 printf 语句最后缺少分号，但因为 C 语言语法中规定一条语句可以书写在多行中，所以语句的分号可以写在下一行中，因此编译工具才在第 13 行中给出错误提示。

改正错误：在第 12 行的“printf”语句末尾补上一个分号。重新编译，无误，然后连接，最后运行得到输出结果窗口如图 1.6 所示。

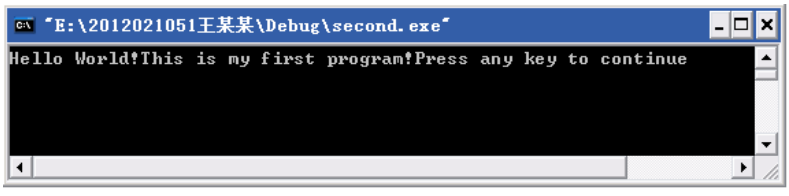


图 1.6　程序运行结果

结果中最后的“Press any key to continue”是运行程序时系统自动加上的，含义是按任

意键即可退出，返回到程序的编辑状态。

程序的运行结果与之前的要求仍然不一致，说明程序仍然存在错误，此时的错误就称为语义错误。错误原因第 11 行语句中缺少换行符'\n'，应将程序第 11 行修改为：

```
11      printf("Hello World! \n");                    /*在屏幕上输出字符串*/
```

或者将第 12 行修改为

```
12      printf("\nThis is my first program! ");
```

或者将 11 行和 12 行合并为 1 行

```
11      printf("Hello World! \n This is my first program! ");  /*在屏幕上输出字符串*/
```

为了能清楚地显示结果，最好在上一条语句的“program”之后也加一个回车换行符。

保存程序，重新编译、连接、运行后，得到程序运行结果如图 1.7 所示，与题目要求结果一致，按任意键返回。

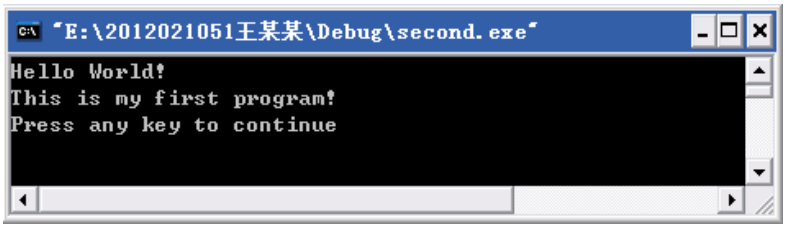


图 1.7　运行结果

【思考】根据分别在编译、连接、运行程序之前和之后工作目录中文件的变化(即程序编译后，又生成哪些文件)，了解相应文件各起什么作用？

【注意】VC++ 6.0 编译器通知的语法错误分为 error 和 warning 两种，其中 error 错误是致命的，将使程序无法继续连接，也不能继续运行；warning 是警告性错误，不影响继续进行下一步操作，但最好也对其进行修改，否则可能会导致运行结果不正确。对于初学者来说，每修改完一个错误，就应马上重新编译程序。连接中出现的错误，最大的可能是 main 函数中出现拼写错误，此外可以尝试关闭编译器，删除编译过程中新生成的文件，重新编译。

实验内容

1. 请编写程序在屏幕上显示你的名字、爱好以及你想对老师说的话。同时将你的班级以注释的方式显示出来。请注意编程规范。

2. 请编写一个程序，输出下列图形。注意编程规范。

```
**************************
*        Hello World!        *
**************************
```

3. 改正下列程序中的错误，在屏幕上显示“There are 52 weeks in a year!”。如果程序不规范，请按照编程规范进行修改。

```
/*该程序可以显示出一年有多少周
include studio.h
int mian(){
printf(There are 52 weeks in a year!); return 0}
```

4. 编写程序说明将一只大象放入冰箱中的步骤，每个步骤一行。注意编程规范。
5. 编写程序，在屏幕上输出下列图形。注意编程规范。

```
   A
 A   A
   A
```

实验总结与分析

第 2 章　数据类型和表达式

实验目的

1. 掌握不同类型变量定义的方法，体会它们之间的不同。
2. 掌握不同类型变量的输入和输出方法。
3. 掌握各种 C 语言表达式，体会它们与对应的数学表达式的不同。
4. 继续体会程序调试的思想，借助于 VC++ 6.0 编译环境修改 C 程序中的语法错误。

实验案例

【例 2.1】任意给定两个整数，例如 4 和 5，实现简单的计算器功能，即求出它们的和、差、积、商和余数。

【分析】要计算两个数的各种运算结果，首先应保存这两个数，方法是定义两个变量，同样还应定义 5 个变量来存放它们 5 种运算的运算结果。定义完变量后给两个整数变量赋初值，然后进行计算，并将计算结果分别保存在 5 个变量中，最后显示计算结果。根据以上分析，绘制如图 2.1 所示的 N-S 图。

4→lOperator
5→rOperator
lOperator+rOperator→nResult1
lOperator−rOperator→nResult2
lOperator*rOperator→nResult3
lOperator/rOperator→nResult4
lOperator%rOperator→nResult5
输出 nResult1, nResult2, nResult3, nResult4, nReult5

图 2.1　例 2.1 的 N-S 图

参考代码：

```
/*****************************************************
文件名：compute.c
作者：***
日期：2013-7-21
功能：简单计算器功能
*****************************************************/

#include<stdio.h>                          /*编译预处理命令*/
int main(void)                             /*主函数*/
{
    int  lOperator, rOperator;     /*定义 2 个整型变量，用于存放左右操作数*/
    int  nResult1, nResult2, nResult3, nResult4, nResult5;   /*定义 5 个存放结果的整型变量*/
    lOperator=4;
    rOperator=5;
```

```
15      nResult1=lOperator+rOperator;   /*lOperator 与 rOperator 相加，结果存入 nResult1 中*/
16      nResult2=lOperator-rOperator;   /*lOperator 与 rOperator 相减，结果存入 nResult2 中*/
17      nResult3=lOperator*rOperator;   /*lOperator 与 rOperator 相乘，结果存入 nResult3 中*/
18      nResult4=lOperator/rOperator;   /*lOperator 与 rOperator 相除，结果存入 nResult4 中*/
19      nResult5=lOperator%rOperator; /*lOperator 与 rOperator 相除的余数，存入 nResult5 中*/
20      printf("%d", nresult1);                 /*在屏幕上显示和*/
21      printf("%d", nresult2);                 /*在屏幕上显示差*/
22      printf("%d", nresult3);                 /*在屏幕上显示积*/
23      printf("%d", nresult4);                 /*在屏幕上显示商*/
24      printf("%d", nresult5);                 /*在屏幕上显示相除后的余数*/
25
26        return  0;
27  }
```

保存后编译，发现有错，如图 2.2 所示。

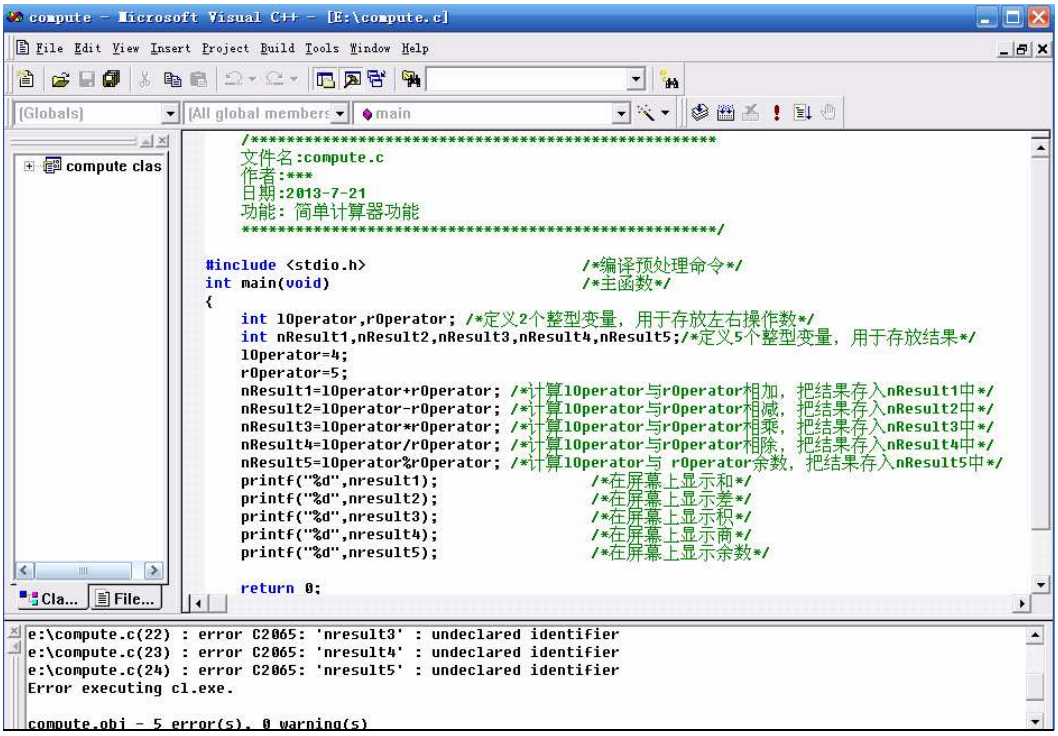


图 2.2　编译后错误提示图

第一个错误出现在第 20 行，错误原因是 nresult1 是“undeclared identifier”，即 nresult1 变量未定义。发生该类错误的原因可能有 3 种：第 1 种是该变量未定义，在 main()函数开头应添加上相应的定义语句；第 2 种可能是该变量已经定义过了，但发生了书写错误；第 3 种可能是在该变量的定义语句部分出现问题。经分析发现该错误属于第 2 种情况，变量定义部分中定义的变量是 nResult1，而第 20 行的变量为 nresult1。

修改方法：将第 20 行的 nresult1 修改为 nResult1。用类似的方法修改剩余的 4 个错误，直到编译无误。连接运行后，得到如图 2.3 所示的运行结果。

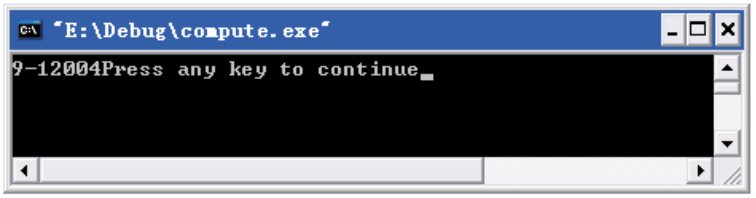


图 2.3　运行结果

运行结果为 9-12004，分析后得知结果是正确的，但这个显示结果不清楚，这时应该修改结果的显示方式，至少应该让每个结果之间有间隔(如空格)。修改 printf 语句，将第 20 行修改为：

```
20        printf("%d ", nResult1);                    /*在屏幕上显示和*/
```

同样的方法修改第 21 行到第 23 行的语句，将第 24 行修改为：

```
24        printf("%d\n", nResult5);
```

重新编译、连接后运行得到结果如图 2.4 所示。

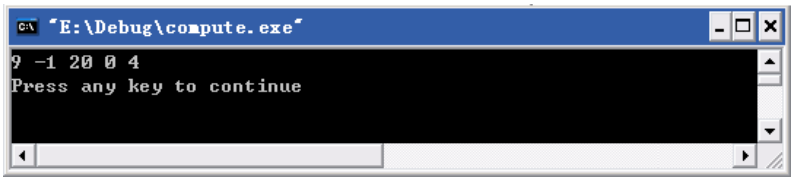


图 2.4　修改程序后的运行结果

【思考题】① 例 2.1 所示的程序只能计算 4 和 5 的和、差、积、商和相除的余数，如果要求该程序能对任意两个整数进行相关的运算，应如何修改程序？

② 如果用户不但要输入运算对象，还要输入运算符，若只进行加法运算，应该如何修改程序？如何进行键盘的输入？【提示】使用字符型变量。

实验内容

1. 输入任意两个整数，输出其中较大的一个。请补充完整程序。

```
#include<stdio.h>
void  main()
{
    int  j, k, m;
    printf("Enter  j  and  k：");
    scanf("%d,%d" ,  ____[1]____);
    m=_____[2]_____ ;
    printf("m=%d\n", m);
}
```

程序运行示例(其中带下划线者表示键盘输入的内容)：

```
Enter  j  and k：5, 8
m=8
```

2. 输入三个浮点型数据，求它们的平均值并输出。请补充完整程序。

```
#include<stdio.h>
void main()
{
    double a, b, c, sum;
    printf("Enter three double nums:\n");
    scanf(      [1]      , &a, &b, &c);
     [2]       ;
    printf("average =%lf\n",  [3]  );
}
```

程序运行示例：

```
Enter three double nums:
4.5  8.0  2.5
average =5.000000
```

3. 在例 2.1 的实验案例中，若要求输入输出如下，应如何修改程序。注意编程规范。

输入输出示例：

```
请输入两个整数，并计算它们的差：
5, 7
5-7=-2
```

4. 在例 2.1 的实验案例中，如果参加运算的两个数是浮点数，应如何修改程序，此时以上 5 种运算都可以进行吗？如果两个浮点数的值分别为 2.118 和 4.5，请计算它们的和及差。结果保留 2 位小数，请注意编程规范。

输入输出示例：

```
2.118 4.5
2.118+4.500=6.62
2.118-4.500=-2.38
```

5. 例 2.1 实验案例中给出的程序在交互性上还是有问题，修改程序，使程序的运行结果变为：

```
4+5=9，4-5=-1，4*5=20，4/5=0，4%5=4
```

注意编程规范。

6. 输入一个整数和一个浮点数(double 类型)，求出它们的倒数，并显示在屏幕上，要求程序具有较强的交互性：输入有提示，输出有说明。注意编程规范。

请按照实验案例的方式写出问题分析、画出流程图，然后再编写程序源代码，并编译、连接、运行。如果程序有错，请说明调试过程，最后写出程序的运行结果，包括输入和输出。

程序运行示例：

```
请输入一个整数和一个浮点数，并求出二者的倒数：
10, 0.5
1/10=0.10; 1/0.5=2.00
```

思考题答案

① 问题的本质就是调用 scanf()函数对变量进行初始化。可以将该程序的 13 行和 14 行改为：

```
13      scanf("%d", &lOperator);
14      scanf("%d", &rOperator);
```

如此修改之后，再运行程序，屏幕上会有光标闪烁，等待用户输入数据。如要计算 7 和 15 两个数据的相关计算结果，则输入格式为：7 15

② 要输入运算符需要定义字符型变量。对程序有关部分的修改如下所示：

```
10  {
11      int  lOperator, rOperator;          // 定义两个整型变量，用于存放左右操作数
12      int  nResult1,;                     // 定义整型变量，用于存放结果
13      char  ch;                           // 定义字符变量，用于存放运算符
14      scanf("%d%c%d",&lOperator, &ch, &rOperator);  // 输入运算式
15      nResult1=lOperator+rOperator;  // 这里假定只进行加法运算
16      printf("%d%c%d=%d",lOperator, ch, rOperator, nResult1);  /*显示运算结果*/
17      return  0;
18  }
```

程序运行示例：

```
5+6
5+6=11
```

【注意】在运行程序输入数据时，一定要注意参加运算的运算对象与运算符之间不能有任何其他字符，即 5 和“+”之间不能有任何其他字符。

实验总结与分析

第 3 章　程序控制结构

3.1　顺 序 结 构

实验目的

1. 进一步理解不同类型变量的输入和输出方法。
2. 掌握顺序结构的应用。
3. 掌握程序调试的方法。

实验案例

【例 3.1】编写程序，输入三角形的三条边长，求三角形的面积。设输入的三条边长 a、b、c 能构成三角形，已知求三角形面积的公式为：

$$area = \sqrt{s(s-a)(s-b)(s-c)}$$

$$s=\frac{a+b+c}{2}$$

流程图如图 3.1 所示。

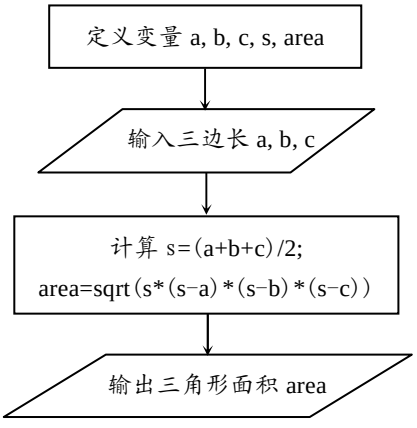


图 3.1　例 3.1 流程图

参考代码：

```
#include<stdio.h>
#include<math.h>
void  main()
{
    float  a,  b,  c,  s,  area;
    scanf("%f%f%f", &a, &b, &c);
    s=(a+b+c)/2;
    area=sqrt(s*(s-a)*(s-b)*(s-c));
    printf("a=%7.2f, b=%7.2f, c=%7.2f\n", a, b, c);
    printf("area=%8.3f\n", area);
}
```

程序运行示例：

```
3 4 5
a=    3.00, b=    4.00, c=    5.00
area=    6.000
```

实验内容

1. 阅读并运行程序，理解输入函数 scanf()的用法。

```
#include<stdio.h>
void  main()
{    int  m, n, p;
     scanf("m=%dn=%dp=%d", &m, &n, &p);
     printf("%d%d%d\n", m ,n, p);
}
```

若想从键盘上输入数据,使变量 m 中的值为 123，n 中的值为 456，p 中的值为 789，则应该怎样正确进行输入？

分别按如下输入运行程序，看到的结果是什么？

123 456 789(回车)

m=123,n=456,p=789(回车)

m=123 n=456 p=789(回车)

m=123n=456p=789(回车)

2. 运行下面的程序，使 a=3，b=7，x=8.5，y=78.82，c1='A'，c2='a'。问在键盘上应如何输入？

```
#include<stdio.h>
int  main(void)
{
    int  a, b;
    float  x, y;
    char  c1, c2;
    scanf("a=%d b=%d", &a, &b);
    scanf("%f%e", &x, &y);
    scanf("%c %c", &c1, &c2);
    printf("a=%d,b=%d,x=%f,y=%f,c1=%c,c2=%c\n", a, b, x, y, c1, c2);
    return  0;
}
```

如果键盘输入为：

a=3 b=7

8.5 78.82 A a （A 与 a 之间有一个空格）

输出是什么？

如果键盘输入为：

a=3 b=7

8.5 78.82

A a （A 与 a 之间有一个空格）

输出是什么？

3. 设圆的半径 r，求圆周长、圆面积。用 scanf()函数输入数据，输出计算结果，输出时要求有文字说明，取小数点后 2 位数字。编程并调试运行。

程序运行示例：

请输入圆半径 r：1.5
圆周长为：l=9.42
圆面积为：s=7.07

4. 编写程序将单词“China”译成密码，密码的设置规律是：把原来的字母用其在字母表中后面第 4 个字母代替。例如字母 C 后面的第 4 个字母是 G。因此“China”译为“Glmre”。用赋初值的方法使 c1，c2，c3，c4，c5 这 5 个变量的值分别为'C'，'h'，'i'，'n'，'a'，经过计算，使 c1，c2，c3，c4，c5 这 5 个变量的值分别变成 'G'，'l'，'m'，'r'，'e'，并输出。

5. 趣味编程——鸡兔同笼问题。“鸡兔同笼”是我国古代著名趣题之一，大约在 1500 年前，《孙子算经》中就记载了这个有趣的问题。书中是这样叙述的：“今有雉兔同笼，上有三十五头，下有九十四足，问雉兔各几何？”这四句话的意思是：“有若干只鸡兔在同一个笼子里，从上面数，有 35 个头；从下面数，有 94 只脚。求笼中鸡和兔各有几只？

现假设有 n 个头和 m 个脚，请编写程序计算有多少只鸡和兔。

【提示】输入头和脚的数目，输出鸡和兔子的数目。

程序运行示例：

输入：2 6
输出：rabbit=1, chicken=1

实验总结与分析

3.2　选 择 结 构

实验目的

1. 掌握分支结构的应用。
2. 掌握用多分支结构解决实际问题的方法，理解分支结构嵌套的思路。

实验案例

【例 3.2】输入一个整数，判断并输出它是奇数还是偶数。

【分析】定义一个 int 型变量 num，从键盘输入其值，通过判断 num 除以 2 的余数是 0 还是 1，来判别 num 是奇数还是偶数。流程图如图 3.2 所示。

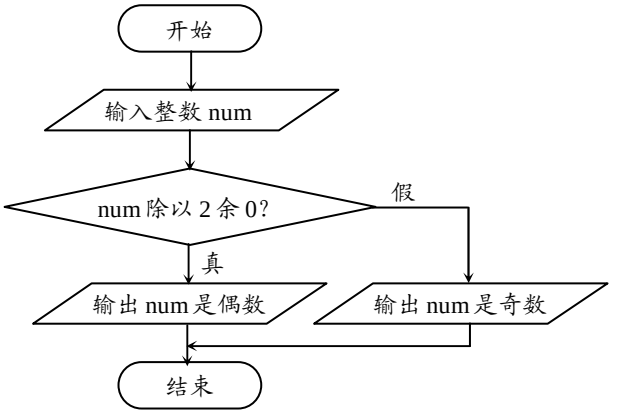


图 3.2　判断奇偶数流程

参考代码：

```
#include<stdio.h>
int  main(void)
{
    int  num;
    printf("请输入一个整数：");
    scanf("%d", &num);
    if(num % 2 == 0 )
        printf("%d 是个偶数。\n", num);
    else
        printf("%d 是个奇数。\n", num);
    return  0;
}
```

程序运行示例：

```
请输入一个整数：20
20 是个偶数。
请输入一个整数：5
5 是个奇数。
```

【思考】条件表达式 num%2==0 能够改成表达式 num%2 吗？请写出新程序。

【例 3.3】输入一个年份，根据形成闰年的条件：四年一闰，百年不闰，四百年再闰，输出该年是否为闰年。

【分析】输入的年份应该用一个整型变量存放，然后判断其是否满足闰年的条件。即若该年能被 4 整除但是不能被 100 整除，或者能被 400 整除，就是闰年，否则不是。程序 N-S 图如图 3.3 所示。

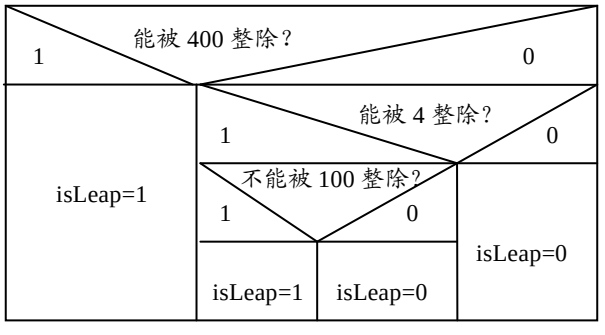


图 3.3　判断是否为闰年的 N-S 图

参考代码：

```
#include<stdio.h>
void  main(void)
{
    int  year, isLeap;   // isLeap 用来存放是否为闰年的标志，等于 1 是闰年，否则不是
```

```
        scanf("%d", &year);
        if(year % 400 == 0 )
        {
             isLeap = 1;                          // 能被 400 整除是闰年
        }
        else
        {
            if(year % 4 == 0 )
            {
                if(year % 100 != 0 )
                    isLeap = 1;                   // 能被 4 整除但不能被 100 整除是闰年
                else
                    isLeap = 0;                   // 能同时被 4 和 100 整除，不是闰年
            }
            else
            {
                isLeap = 0;             // 既不能被 400 整除，也不能被 4 整除，不是闰年
            }
        }
        //输出
        if(isLeap)
            printf("%d 是闰年。\n", year);       // isLeap 不等于 0，是闰年
        else
            printf("%d 不是闰年。\n", year);     // isLeap 等于 0，不是闰年
    }
```

程序运行示例 1：

```
2000
2000 年是闰年。
```

程序运行示例 2：

```
2013
2013 年不是闰年。
```

【思考】如何只用一个判断条件，判断 year 是否为闰年？

单步调试执行程序，如图 3.4 所示。

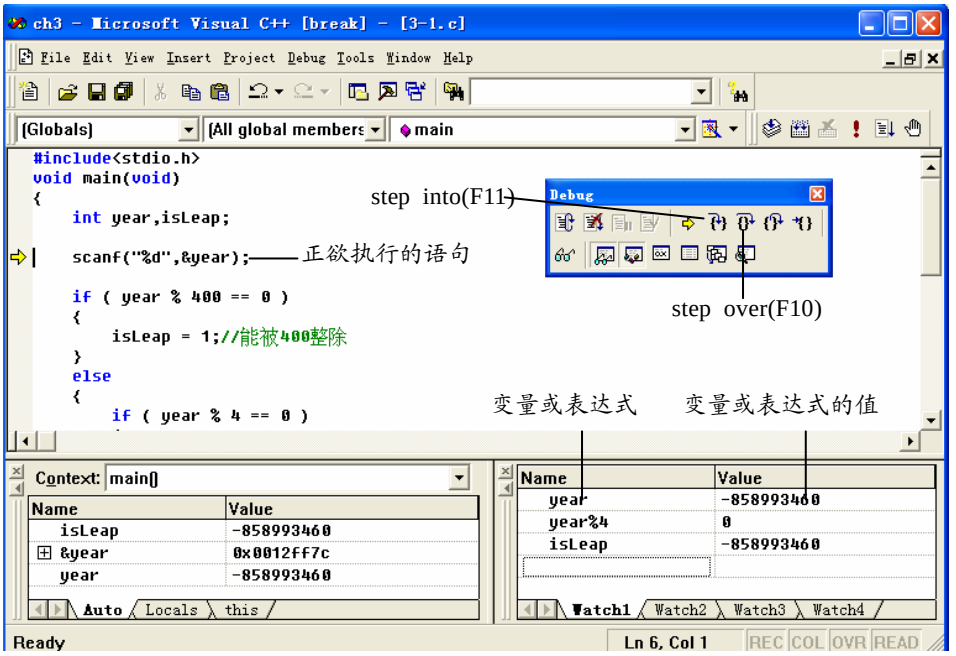


图 3.4　单步调试程序示意图

【例 3.4】根据输入的年 year，月 month 两个值，输出该年该月的天数。

【分析】year 和 month 都是整型变量，应当通过键盘分别给 year 和 month 输入两个整数值，如果 month 是 1、3、5、7、8、10 或 12 中的任何一个，则输出 31 天，如果 month 是 4、6、9 或 11 中的任何一个，则输出 30 天，如果 month 是 2，则还需要判断 year 年是否为闰年，如果是闰年则输出 29 天，否则输出 28 天。

参考代码：

```
#include<stdio.h>
void  main(void)
{
    int  year, month;
    scanf("%d-%d", &year, &month);      // 注意输入格式为  年-月
    switch(month)
    {
        case  1:
        case  3:
        case  5:
        case  7:
        case  8:
        case  10:
        case  12:
            printf("31 天。\n");
            break;
        case 4:
```

```
            case 6:
            case 9:
            case 11:
                printf("30 天。\n");
                break;
            case 2:
                if((year % 4 == 0 && year%100 != 0) || year %400 == 0 )
                    printf("29 天。\n");
                else
                    printf("28 天。\n");
                break;
            default:printf("输入错误！ ");
        }
    }
```

程序运行示例 1(判断 2010 年 1 月)：

```
2010-1
31 天
```

程序运行示例 2(判断 2010 年 2 月)：

```
2010-2
28 天
```

【注意】输入数据的必须是“年-月”这种格式。

【思考】在上述程序中的多路分支结构中，可以去掉 break 语句吗?

实验内容

1. 以下程序的功能是任意输入一个 3 位整数，倒序输出该数据，如：输入 123 输出 321，输入-123 输出-321。 请补充完整程序。

```
#include<stdio.h>
#include<math.h>
void  main()
{
    int  x, c1, c2, c3, y;
    scanf(    [1]    );
    y=abs(x);
    c1=y/100;
    c2=y/10%10;
    c3=    [2]    ;
    y=c3*100+c2*10+c1;
    if(x>=0)
        [3]    ;
    else
        printf("%d\n", -y);
}
```

2. 编写程序，输入 x 的值，计算并输出下列分段函数 $f(x)$ 的值，结果保留两位小数。

$$f(x)=\begin{cases}(x+1)^2+2x+\dfrac{1}{x} & (x<0)\\ \sqrt{x} & (x\geqslant 0)\end{cases}$$

【提示】调用数学库中的 sqrt() 函数求平方根。

程序运行示例1：

Enter x: −0.5
f(−0.50)=−2.75

程序运行示例2：

Enter x: 0
f(0.00)=0.00

程序运行示例3：

Enter x: 10
f(10.00)=3.16

3. 输入两个整数，判断两者关系并输出。

程序运行示例1：

4, 5
4<5

程序运行示例2：

7, 3
7>3

程序运行示例3：

6, 6
6=6

4. 编写一个计算器程序：num1 和 num2 都是整数，operator 为+、−、*、/和%中的一种，输入“num1 operator num2”表达式(如 3+5，5*6，7/2 等)，计算表达式的结果。

5. 输入年、月、日三个数，输出该日是这一年的第几天。

程序运行示例：

2010−4−2
第 92 天。

实验总结与分析

3.3 循 环 结 构

实验目的

1. 掌握 for，while 和 do−while 三种循环结构的应用。
2. 能够灵活选用循环结构解决实际问题。
3. 掌握 break 和 continue 两个控制语句的应用。
4. 掌握单步执行和查看变量值的调试方法。

实验案例

【例 3.5】输入三个整数，分别判断这三个数的奇偶性，并输出它们的和。

【分析】循环次数已知(3 次)，用 for 循环比较直接，每次循环所完成的任务是：输入一个数，然后判断它们的奇偶性，并进行累加。程序的 N-S 图如图 3.5 所示。

参考代码：

```
#include<stdio.h>
void  main(void)
{
    int  a, i, sum=0;
    for(i=1; i<=3; i++)
    {
        scanf("%d", &a);
        if(a % 2 == 0 )
            printf("是偶数。\n");
        else
            printf("是奇数。\n");
        sum=sum+a;
    }
    printf("sum=%d。\n",sum);
}
```

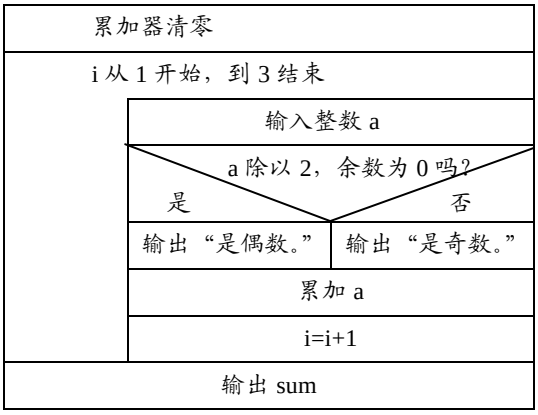


图 3.5　例 3.5 的 N-S 图

【调试】单步执行该程序，注意查看变量 i、a 以及 sum 的值。

程序运行示例：

```
1
是奇数。
2
是偶数。
3
是奇数。
sum=6。
```

【思考】① for 循环体不变的情况下，能否把 for 循环控制部分改为 for(i=1; i<6; i=i+2)。

② 程序中对 sum 的初始化能否放到 for 循环控制部分？若能应该怎么写？若不能，是为什么？

【例 3.6】使用 do-while 结构，判断多次输入的正整数 num 是不是素数，直到输入的 num 不大于 1 结束。

【分析】判断一个正整数 num 是否为素数，是一个循环问题，要是判断多

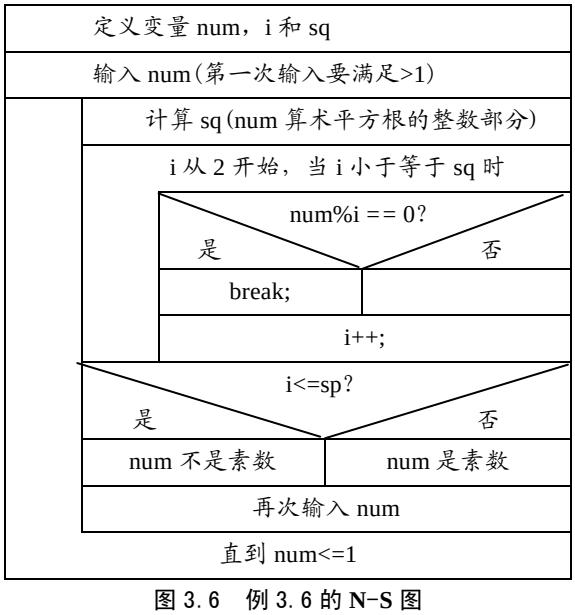


图 3.6　例 3.6 的 N-S 图

次输入的数据是否为素数，则需要在外层再有一个循环结构，其循环体即为判断一个数是否为素数的循环。需要一个循环嵌套。外层循环直到 num 为非整数结束，用 do-while 结构，内层循环还是使用 for 循环结构。N-S 图如图 3.6 所示。

参考代码：

```
#include<stdio.h>
#include<math.h>
void  main(void)
{
    int  num, i;
    int  sq;
    printf("请输入一个大于 1 的整数：");
    scanf("%d", &num);
    do{
        sq=(int)sqrt(num);
        for(i=2; i<=sq; i++)
        if(num % i == 0)
            break;
        if(i<=sq)
            printf("%d 不是素数。\n", num);
        else
            printf("%d 是素数。\n", num);
        printf("请输入一个正整数 num(输入不大于的 1 的整数退出)：");
        scanf("%d", &num);
    }while(num>1);
    printf("退出。\n");
}
```

程序运行示例：

```
请输入一个大于 1 的整数：9
9 不是素数。
请输入一个正整数 num(输入不大于 1 的整数退出)：7
7 是素数。
请输入一个正整数 num(输入不大于 1 的整数退出)：0
退出。
```

【思考】① 如何改写程序中的 for 循环为 while 循环结构。

② 如何改写程序中的 do-while 循环为 while 循环结构。

实验内容

1. 以下程序的功能是计算 s=1+12+123+1234+12345。请补充完整程序。

```
#include<stdio.h>
void  main()
```

```
{
    int  t=0, s=0, i;
    printf("s=1+12+123+1234+12345=");
    for(i=1; i<=5; i++)
      {   t=i+ [1]  ;
          s=s+t;
      }
    printf("%d\n", s);
}
```

2. 以下程序的功能是输出如下方阵，请补充完整程序。

```
 13  14  15  16
  9  10  11  12
  5   6   7   8
  1   2   3   4
```

```
#include<stdio.h>
void  main()
{
  int  i, j, x;
  for(j=4; j [1] ; j--)
    {
      for(i=1; i<=4; i++)
        {    x=(j-1)*4 + [2] ;
             printf("%4d", x);
        }
      printf("\n");
    }
}
```

3. 以下程序的功能是输出 100 以内能被 3 整除且个位数为 6 的所有整数。请补充完整程序。

```
#include<stdio.h>
void  main()
{
  int  i, j;
  for(i=0;  [1] ; i++)
      {   j=i*10+6;
          if( [2] )        continue;
          printf("%4d ", j);
      }
}
```

4. 编写程序输出1到100之间能被6或7整除的所有整数。
5. 编写程序输出n行星号。

程序运行示例：

```
请输入n：4
   *
  * * *
 * * * * *
* * * * * * *
```

6. 编写程序求 1.1+2.2+3.3+……+1.1*n的和(n是从键盘输入的整数)。

7. 输入1个多位整数，逆序输出该数及其位数。

程序运行示例1：

```
893
3位数
逆序：398
```

程序运行示例2：

```
3820
4位数
逆序：283
```

8. 编写程序，输出从n后开始的依次的m个素数。要求分别使用for、while和do-while三种循环结构。

程序运行示例1：

```
n=3
m=4
3以后依次的4个素数是：5 7 11 13
```

程序运行示例2：

```
n=7
m=3
7以后依次的3个素数是：11 13 17
```

实验总结与分析

第4章　数　　组

4.1　一 维 数 组

实验目的

1. 熟练掌握使用一维数组编程的方法。
2. 掌握与数组有关的算法(例如排序算法)。

实验案例

【例 4.1】用键盘输入 n 个整数(1<n≤10)存入数组，编制程序使该数组中的数按照从小到大的次序排列并输出。

【分析】本程序可分为输入数组、排序、输出数组三个部分，其中难点是排序，本例学习用选择法排序，每一趟在 a[i]～a[n-1]中查找最小数的下标，找到后作为有序序列中第 i 个记录。显然，在 n 个关键字中选取最小值，至少进行 n-1 次比较。程序的 N-S 图如图 4.1 所示。

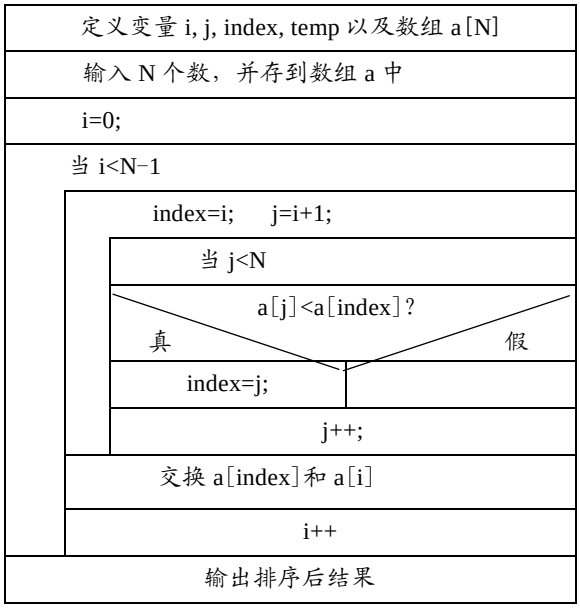


图 4.1　例 4.1 的 N-S 图

参考代码：

```
/*一维数组的定义、赋值、引用以及选择法排序*/
#include"stdio.h"
#define  N  10          //定义符号常量 N，对 N 个数排序

int  main()
{
    int  a[N], i, j, index, temp;

    /*输入 N 个数，并存到数组 a 中*/
    printf("please  input  %d  numbers\n", N);  //提示输入 N 个数
```

```
10      for(i=1; i<=N; i++)
11          scanf("%d", &a[i]);
12
13      /*对数组 a 中的 N 个数排序,每循环一次选出数组中第 i 个最小的数,i 从 0 到 N-2
        取值*/
14      for(i=0; i<N-1; i++)
15          {
16
17      /*index 用来存放最小数的下标，初始值是剩余未排序的数组元素 a[i]~a[N-1]的
        第一个元素下标 i*/
18
19              index=i;
20
21          /*在 a[i]~a[N-1]中找到最小的数，并将该数的下标放到 index 中*/
22          for(j=i+1; j<N; j++)
23              if(a[j]<a[index])
24                  index=j;
25
26          /*将找到的最小数与第一个未排序的元素 a[i]交换*/
27                  temp=a[index];
28                  a[index]=a[i];
29                  a[i]=temp;
30      }
31
32      /*输出排序后的结果*/
33      printf("the array after sort:\n");
34      for(i=0; i<N; i++)
35          printf("%5d", a[i]);
36      printf("\n");
37
38      return 0;
39  }
```

【调试程序过程】

① 编译程序，通过后执行程序，按照下面提示输入N个数字，数字间用空格(或者用回车)间隔：

```
please input 10 numbers
5 7 0 9 1 8 4 3 2 6
```

输出结果如图4.2所示，输出异常终止。分析输出结果：未输出最后输入的数6，而且输出中第一个数为系统随机值-858993460。

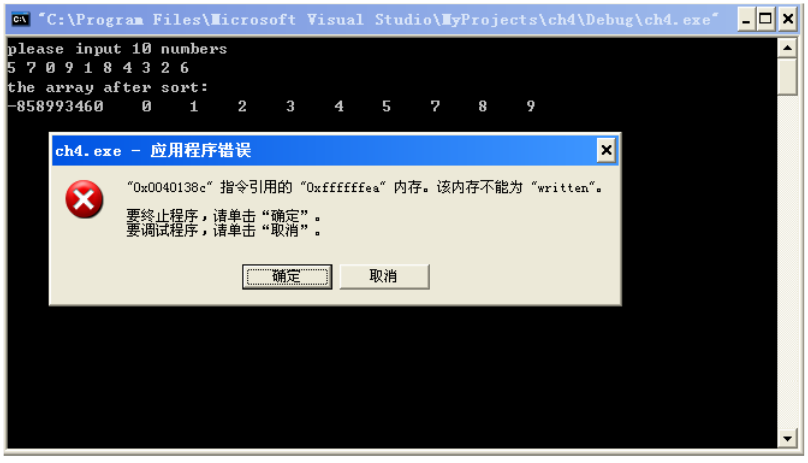


图4.2　例4.1程序运行结果图

② 为了看清楚程序中出现的问题，在程序的11行增加断点，将插入点光标移到第11行，单击(Go)按钮，开始调试程序，当程序执行到断点所在位置时，单击单步执行按钮(Step over)，输入5。

③ 再次单击单步执行按钮(Step over)两次，输入7。

④ 按照步骤③，输入0，此时变量窗口如图4.3所示。

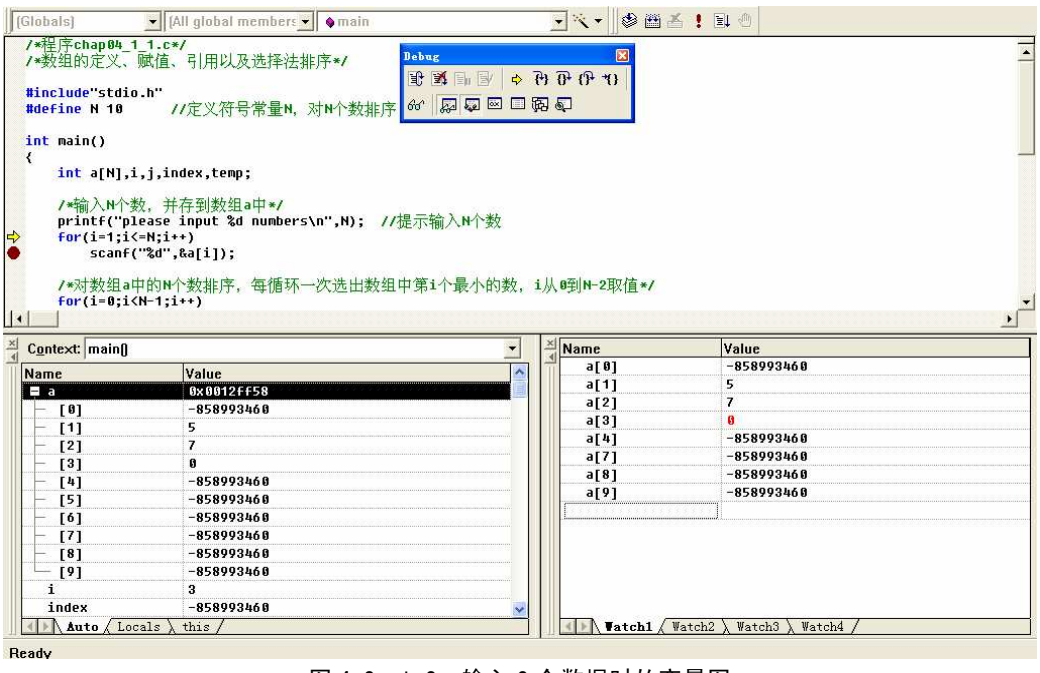


图 4.3　i=3，输入 3 个数据时的变量图

⑤ 按照步骤③，依次输入剩下数据9 1 8 4 3 2 6，在变量窗口和观察窗口中，观察存储在a[0]~a[9]的值。如图4.4所示，可以看到输入的10个数只存放了9个到数组a中，数组中元素a[0]未存放输入数据。由此可见，第10行有问题，for循环中数组下标i应该从0开始，到9结束。

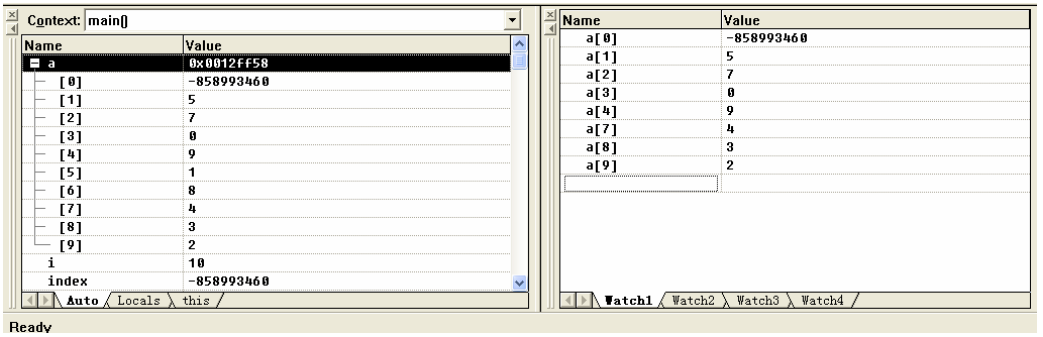


图 4.4　i=10，输入 10 个数据时的变量图

⑥ 把第10行改为“for(i=0; i<N; i++);”后运行程序，得到正确的结果。

【注意】定义数组时，数组长度是常量(包括符号常量)，引用数组时，数组下标从0开始，最大值是“数组长度-1”。

【思考】① 程序中定义了符号常量 N，是否可以不定义，定义符号常量有什么好处？

② 如果要将 N 个数从大到小排序，应该怎样修改本例的代码？

实验内容

1. 下述程序要求实现的功能是：在整型数组 a 中查找与 x 相同的元素，如果找到，输出 x 在数组 a 中的下标位置；如果没找到，输出“没有找到与 x 相同的元素!”。设 a 数组中各元素的值互异，n(n<=10)和 x 都是正整数。考查程序运行示例，可知此程序有错，请修改程序中的错误，使它能得出正确的结果。

```
#include<stdio.h>
int  main()
{    int  i, x, n, a[n];
     printf("输入数组元素的个数：");
     scanf("%d", &n);
     printf("输入%d 个数组元素的值：", n);
     for(i = 0; i < n; i++)
          scanf("%d", &a[i]);
     printf("x= " );
     scanf("%d", &x);
     for(i = 0; i < n; i++)
          if(a[i] == x)
               break;
     if(i!=n)
          printf("没有找到与%d 相等的元素!\n", n);
     else
          printf("和%d 相同的数组元素是 a[%d]=%d\n", x, i, a[i]);
     return  0;
}
```

程序运行示例：

输入数组元素的个数：6
输入 6 个数组元素的值：4 6 5 7 9 1
输入 x：5
和 5 相同的数组元素是 a[2]=5

2. 输入一个正整数 n(1<n≤10)，再输入 n 个整数存入一维数组，按逆序重新存放数组元素后再输出。

程序运行示例：

Input n：5
Input 5 integers：4 6 5 7 1
Output：1 7 5 6 4

3. 青年歌手参加歌曲大奖赛，有 10 个评委给她评分，编写程序按照从高分到低分的顺序输出评委的评分，并求出选手的最后得分(去掉一个最高分和一个最低分后的平均分)，结果保留两位小数。在程序中加注释，说明自己使用的是哪种排序方法。

程序运行示例：

请输入评委的评分：75 82 69 92 88 85 90 78 84 83
从高分到低分为：92 90 88 85 84 83 82 78 75 69
该选手最后得分为：83.13

4. 输入一个正整数 n(1<n≤10)和 n 个有序整数(从小到大的顺序)，今输入一个数 x，要求按原来排序的规律将它插入有序数组中，若原来序列中已有 x，则将原序列中的 x 删除。

程序运行示例1：

Input n：5
Input 5 integers：3 5 6 7 9
Input x：8
Output：3 5 6 7 8 9

程序运行示例2：

Input n：4
Input 4 integers：1 4 6 8
Input x：4
Output：1 6 8

实验总结与分析

4.2 一维字符数组和字符串

实验目的

1. 掌握一维字符数组的使用方法。
2. 熟练掌握字符串的存储和操作方法。
3. 学会使用常用的字符串处理函数。

实验案例

【例 4.2】阅读下面程序，分析程序实现的功能。查找问题，然后上机调试程序，查看运行结果和自己分析的结果是否一致。

```
/*字符串的存储和处理*/
#include<stdio.h>
//#include<string.h>     // string.h 是包含字符串处理函数的头文件
int  main()
{
    /*声明三个字符串 s1、s2、s3，并用不同形式初始化 s2、s3*/
    char  s1[80];
    char  s2[10]="I  am  a  boy";
    char  s3[10]={'I', '  ', 'a', 'm', '  ', 'a', '  ', 'b', 'o', 'y'};

    int i=0;

    /*输入字符串 s1*/
    printf("input  string  s1：");   //输入提示
    while((s1[i]=getchar())!='\n')
        i++;
    //s1[i]='\0';

    /*计算字符串 s1 的有效长度*/
    for(i=0; s1[i]!='\0'; i++)
        ;
    printf("字符串 s1 的长度为：%d \n", i);

    /*使用字符串处理函数 strlen()分别输出字符串 s1、s2、s3 的有效长度*/
    printf("字符串 s1 的长度为：%d \n", strlen(s1));
    printf("字符串 s2 的长度为：%d \n", strlen(s2));
    printf("字符串 s3 的长度为：%d \n", strlen(s3));

    return  0;
}
```

【思考】① 例 4.2 的程序实现什么功能？字符串的长度包含结束符'\0'吗？

② 字符串常量怎么表示？C 语言中怎样存储字符串？

③ 字符串如何初始化，程序中对字符串 s2、s3 的初始化方法正确吗？

④ 字符串 s1 由键盘输入，“/*输入字符串 s1*/” 程序段是存储 s1 字符串的代码，这段程序正确吗？

⑤ 程序中最后几行使用字符串处理函数 strlen()求字符串长度，有没有问题，思考使用字符串处理函数应该注意什么问题？

调试程序，查看程序结果与分析是否一致，再修改程序，写明修改方法，以及修改原因。

【例 4.3】输入一个以回车结束的字符串(少于 80 个字符)，计算其中空格的个数。

【分析】本题的字符串由键盘输入，首先要解决输入字符串的问题，将输入的字符串存储到字符数组 str 中，然后处理该字符数组，逐一判断字符数组的每一个元素是否是空格，是空格，统计结果增 1；无论是与不是，都继续判断数组 str 下面的元素，直到遇到字符串结束符'\0'为止。最后输出统计结果。N-S 图如图 4.5 所示。

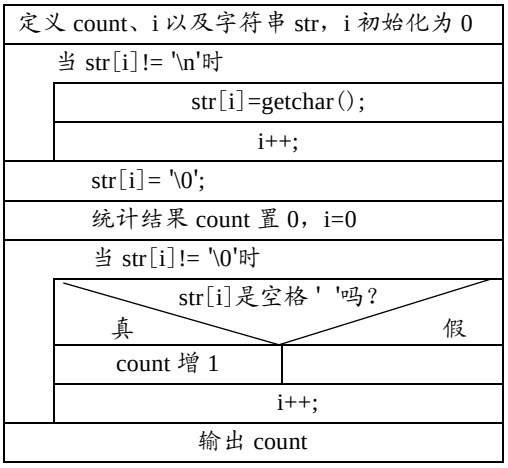


图 4.5　例 4.3 的 N-S 图

参考代码：

```
/*统计输入的字符串中的空格
个数，并输出字符串*/
#include<stdio.h>
int  main()
{
    char  str[80];
    int  i, count;

    /*输入字符串*/
    printf("input  a  string：");        //输入提示
    i=0;
    while((str[i]=getchar())!='\n')
        i++;

    /*统计字符串中空格字符的个数*/
    count=0;
    for(i=0; str[i]!='\0'; i++)
        if(str[i]==' ')
            count = count +1;
    printf("空格数为：%d \n", count);
    printf("被统计的字符串是：%s \n", str);
    return  0;
}
```

【调试程序过程】

① 编译、连接，然后执行程序结果，如图 4.6 所示，程序输出结果中，输出字符串后换行并且输出乱码。为了查找问题，对程序进行调试。

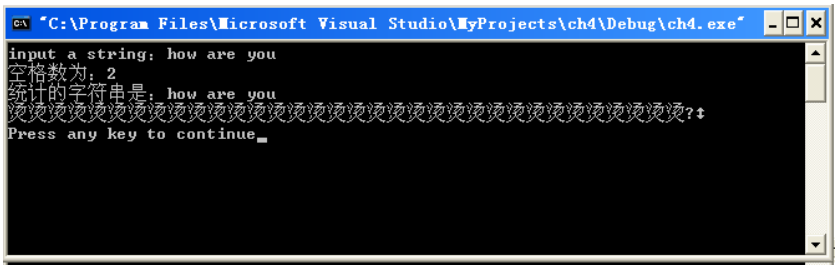


图 4.6　例 4.3 程序运行结果图

② 执行“Build”菜单下的→“start debug”→“step into”菜单命令，单击单步调试按钮，执行到“统计字符串中空格字符的个数”程序段。如图 4.7 所示。

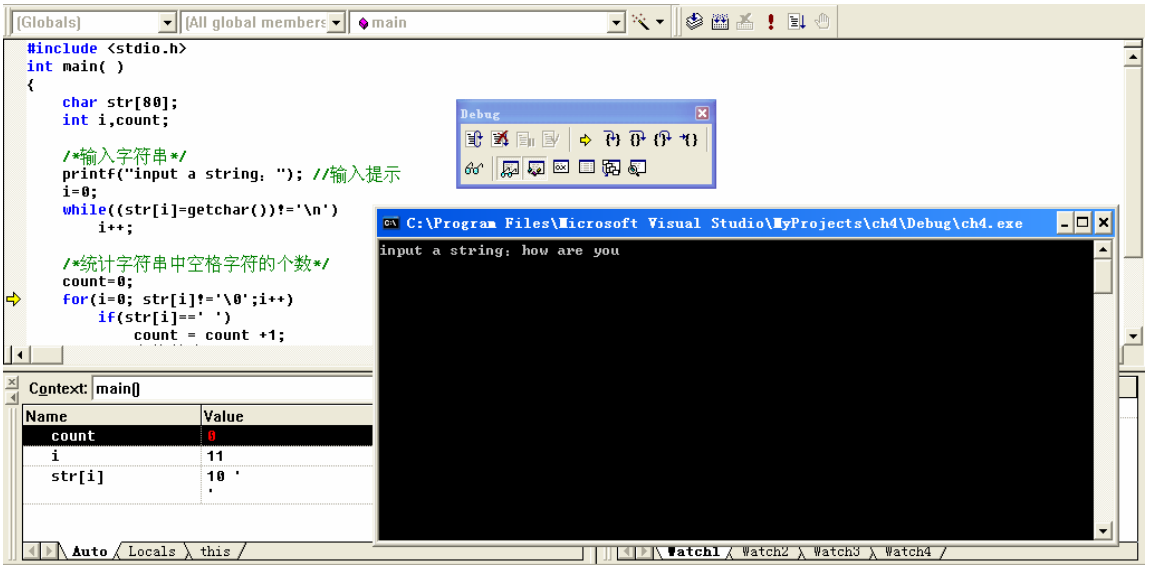


图 4.7　单步调试图

在“统计字符串中空格字符的个数”程序段中继续单步调试，会发现循环次数超过了我们输入的字符串中字符个数，如图 4.8 所示。本次调试输入的字符串是“how are you”， 应该判断字符 11 次，但是 i 此时变为 16，循环还未终止。说明循环结束条件“str[i]!='\0'”未起到作用，即使用字符数组 str 存储字符串时，没有赋值结束符'\0'。由于%s 格式控制符是遇到'\0'结束，所以使用%s 格式控制符输出时，在输出完字符串后还输出乱码。

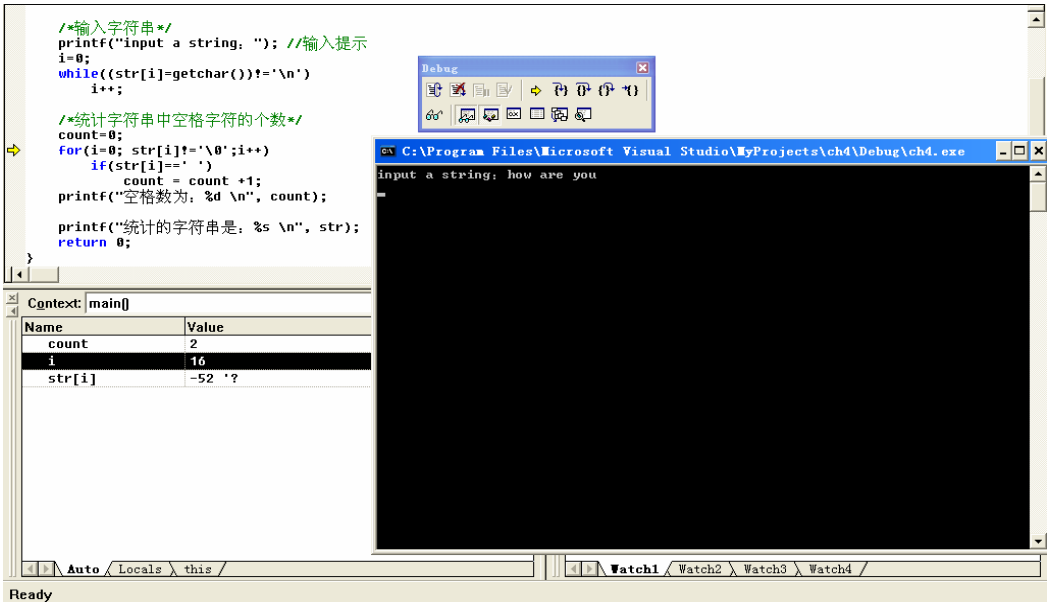


图 4.8　循环次数超过字符串有效字符个数

③ 修改程序，给字符串增加结束符'\0'，即在“输入字符串”程序段代码

```
while((str[i]=getchar())!='\n')
    i++;
```

后增加语句“str[i]= '\0';”。

④ 修改后编译、连接、执行程序，得到正确结果。

【思考】本例中使用格式控制符%s 输出字符串，还可以怎样输出字符串？输入输出字符串都有哪些常用方法？

实验内容

1. 下述程序的功能是将 tt 所指字符串中的小写字母都改为对应的大写字母，其他字符不变。请改正程序中的错误，使它能得到正确的结果。

```
void main()
{   int i;
    char tt[81];
    printf("Please input a string: ");
    gets(tt);
    for(i = 0; tt[i]; i++)
        if(('a' <= tt[i])||(tt[i] <= 'z'))
            tt[i]+= 32;
    printf("The result string is:%s\n", tt);
}
```

程序运行示例：

```
Please input a string: Ab, cD
The result string is: AB, CD
```

2. 下述程序的功能是：读入一个字符串(长度<80)，将该字符串中的所有字符按 ASCII 码升序排序后输出。请改正程序中的错误，使它能得到正确的结果。

```
#include<stdio.h>
#include<string.h>
void main()
{
    int i, j;
    char t[81], c;
    printf("Please input a string: ");
    gets(t);
    for(i = strlen(t); i; i--)
        for(j = 0; j < i; j++)
            if(t[j]<t[j + 1])
            {
                c = t[j];
                t[j] = t[ j + 1 ];
                t[j + 1 ] = c;
            }
    printf("The string after sorting:%s\n", t );
}
```

程序运行示例：

```
Please input a string: dbeac
The string after sorting: abcde
```

3. 输入一个以回车结束的字符串(长度<80)，统计字符串中英文字母的个数。

程序运行示例1：

```
Input a string：ab12&cd
Output：count=4
```

程序运行示例2：

```
Input a string：@12&34
Output：count=0
```

4. 编写程序，输入两个字符串，将输入的字符串连接成一个字符串，然后输出。

程序运行示例：

```
Input a string：aabbcc
Input another string：123ee
Output：aabbcc123ee
```

5. 统计字符串 str 中含有子串 subStr 的个数。

程序运行示例1：

```
Input a string：abcDeabchc
Input a substring：abc
Output：count=2
```

程序运行示例2：

```
Input a string：bcDebchc
Input a substring：cb
Output：count=0
```

实验总结与分析

4.3　二 维 数 组

实验目的

1. 掌握二维数组的定义、赋值和输入输出方法。
2. 熟练掌握使用二维数组编程的方法。

实验案例

【例 4.4】定义一个 2×3 的二维数组 a，数组的元素值为 a[i][j]=i+j(0≤i≤1，0≤j≤2)，按矩阵的形式输出 a。

【分析】二维数组最常见的编程结构是双重 for 循环，一般用行下标作外循环的循环变量，列下标作内循环的循环变量。处理矩阵问题主要用二维数组来实现，按矩阵的形式输出二维数组，即每输出完一行数据后加一个换行。

参考代码：

```
/*按矩阵的形式输出二维数组*/
#include<stdio.h>
int  main()
{
    int  a[2][3];
    int  i, j;
    /*给二维数组赋值*/
    for(i=0; i<2; i++)
        for(j=0; j<3; j++)
            a[i][j]=i+j;
    /*按矩阵的形式输出 a*/
    for(i=0; i<2; i++)
```

```
        {
            for(j=0; j<3; j++)
                printf("%4d", a[i][j]);
            printf("\n");
        }
        return 0;
    }
```

程序运行示例：

```
0   1   2
1   2   3
```

实验内容

1. 下述程序的功能是：输出 M 行 M 列整数方阵，然后求两条对角线上元素之和。请修改程序中的错误，使它能得出正确的结果。

```
#include<stdio.h>
#define M 5
void main()
{
    int xx[][]={{1, 2, 3, 4, 5},{4, 3, 2, 1, 0},{6, 7, 8, 9, 0},{9, 8, 7, 6, 5},{3, 4, 5, 6, 7}};
    int i, j, sum;
    printf("%d 行%d 列矩阵为:\n", M, M );
    for(i = 0; i < M; i++ )
        {  for(j = 0; j < M; j++ )
               printf("%f ", xx[i][j]);
           printf("\n");
        }
    for(i = 0; i < M; i++)
        sum+= xx[i][i]+xx[i][M-i];
    printf("两条对角线上元素之和为:%d\n", sum);
}
```

程序运行示例：

```
5 行 5 列矩阵为：
1   2   3   4   5
4   3   2   1   0
6   7   8   9   0
9   8   7   6   5
3   4   5   6   7
两条对角线上元素之和为：50
```

2. 下述程序的功能是找出方阵中每行最大元素以及所在的列号，输出方阵中每行最大元素及其所在列号和每行最大元素的和。请在程序中填空，使它能得出正确的结果。

```
#include<stdio.h>
#define  M  4
void  main()
{
    int  a[M][M]={{14, 23, 52, 3},{34, 22, 52, 41},{12, 15, 8, 9},{54, 98, 23, 21}};
    int  i, j, t, sum, ln[M];
    for(i=0; i<M; i++)
    {    t=0;
         for(j=1; j<M; j++)
             if(a[i][j]>a[i][t])
                 ___[1]___;
         ln[i]=t;
    }
    printf("\n%d 行%d 列方阵为: \n", M, M);
    for(i=0; i<M; i++)
    {
        for (j=0; j<M; j++)
            printf("%8d", a[i][j]);
        printf("\n");
    }
    ___[2]___;
    printf("\n 每行最大元素及其列号:\n");
    for(i=0; i<M; i++)
    {
         printf("%3d：%2d\n", a[i][ln[i]], ____[3]____);
         sum=sum+ a[i][ln[i]];
     }
   printf("\n 各行最大元素的和为：%d\n", sum);
}
```

程序运行示例：

```
4 行 4 列矩阵为：
14   23   52    3
34   22   52   41
12   15    8    9
54   98   23   21
每行最大元素及其列号：
52：2
52：2
15：1
98：1
各行最大元素的和为：217
```

3. 输入 2 个正整数 m 和 n(1<m≤6，1<n≤6)，然后输入 m 行 n 列二维数组中的 m×n 个元素，分别求出各行元素之和并输出。

程序运行示例：

```
Input m：2
Input n：3
Input array：
2  3  5
4  1  6
sum of row 0 is 10
sum of row 1 is 11
```

4. 输入一段英文文章存入二维字符数组中，编程输出每行英文中的空格数。

程序运行示例：

```
Input：
If you were a teardrop;In my eye,
For fear of losing you,
I would never cry
the row 0 have 6
the row 1 have 4
the row 2 have 3
```

【思考】如果要求输出每行英文中的单词数，应该如何编程？

实验总结与分析

第5章　函　　数

5.1　函数的基本应用

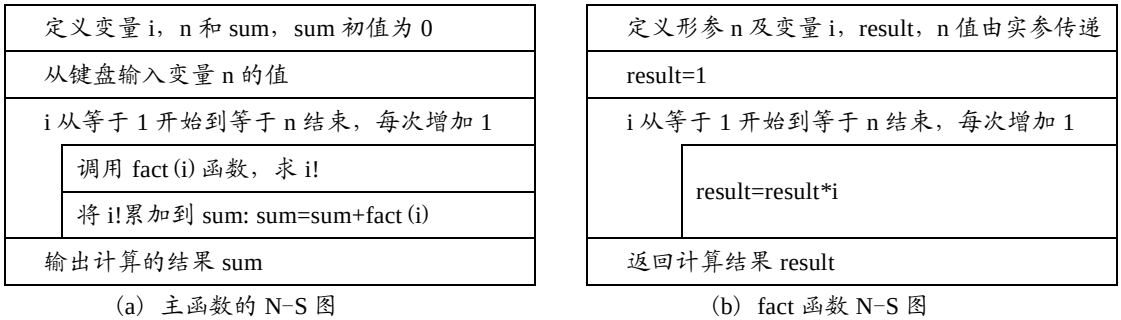
实验目的

1. 熟练掌握函数的定义和调用。
2. 掌握函数实参、形参和返回值的概念及使用方法。
3. 熟练掌握使用函数编写程序。
4. 掌握单步调试进入函数和跳出函数的调试方法。

实验案例

【例 5.1】从键盘输入一个正整数 n，计算 1!+2!+…+n!的值，要求定义并调用函数 fact(i)计算 i!，函数类型为 double。

【分析】本题是一个求累加和的问题，累加的每一项都是一个数的阶乘，用 i 控制这个数的变化，则 i 的取值由 1 到 n，所以本题的关键问题是求 i 的阶乘，按题目要求定义 fact(i)函数求 i!。主函数中用 for 循环多次调用 fact(i)函数，每调用一次求出一个 i 的阶乘，并累加到存放累加和的变量 sum 中。N-S 图如图 5.1 所示。



(a) 主函数的 N-S 图　　(b) fact 函数 N-S 图

图 5.1　例 5.1 的 N-S 图

参考代码：

```
#include<stdio.h>
int  main(void)
{
    int  i,n;
    double  sum=0;
    printf("Input  n: ");
    scanf("%d", &n);
```

```
    /*通过循环多次调用 fact 函数，每循环一次调用一次函数，求出一个 i 的阶乘并累加
      到 sum 中，i 的取值从 1 到 n*/
    for(i = 1 ; i <= n; i++)
        sum = sum + fact(i);                    //调试时设置断点
    printf("1!+2!+…+%d!= %f\n", n, sum);

    return  0;                                  //调试时设置断点
}
double  fact(int  n);
{
    int  i;
    double  result;

    for(i = 1 ; i <=n; i++)
        result=result*i;

    return  result;                             //调试时设置断点
}
```

程序运行示例：

Input n: 10

1!+2!+…+10!=4037913.000000

【调试过程】(进入函数和跳出函数的调试方法)

① 编译程序，出现 1 个 warning 和 3 个 error，如图 5.2 所示。第一条警告信息是："'fact' undefined;"，双击该信息，箭头指向"sum=sum+fact(i)"这一行。

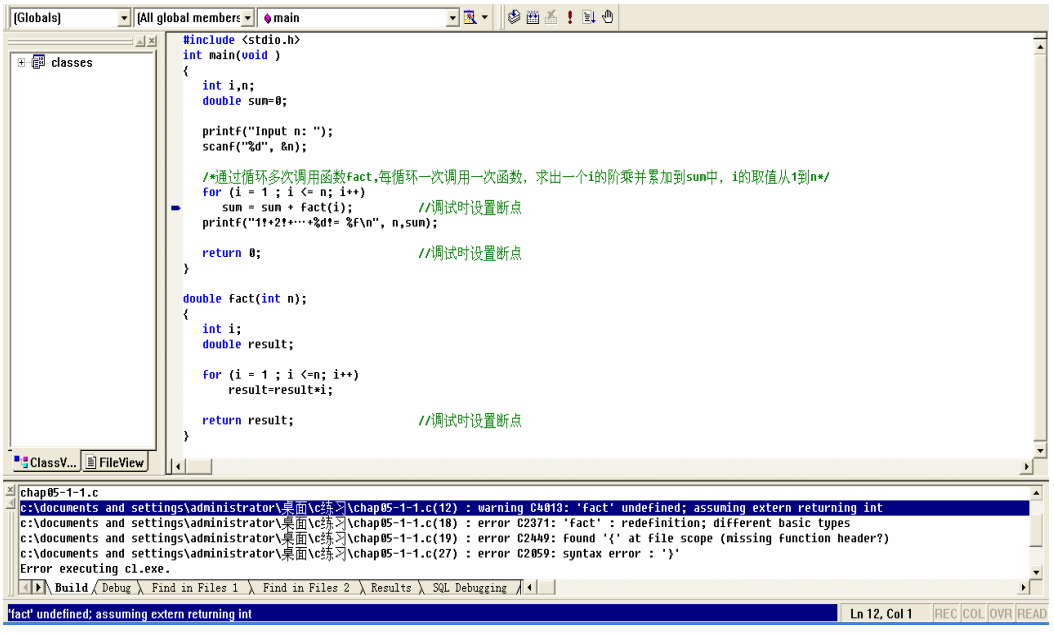


图 5.2　编译后给出错误提示的界面

警告信息指出 fact 函数没有定义，因为函数在调用前必须先定义或声明。改正程序，在

主函数的变量定义前加上函数声明“double fact(int n);”。

② 重新编译程序，出现两条错误提示信息，如图 5.3 所示。双击第一条错误信息"found '{' at file scope (missing function header?)”，箭头指向“{”所在行，错误信息指出缺少函数首部，因为函数定义时，函数首部不能加分号。改正程序，将“double fact(int n);”这一行最后的分号“;”去掉。

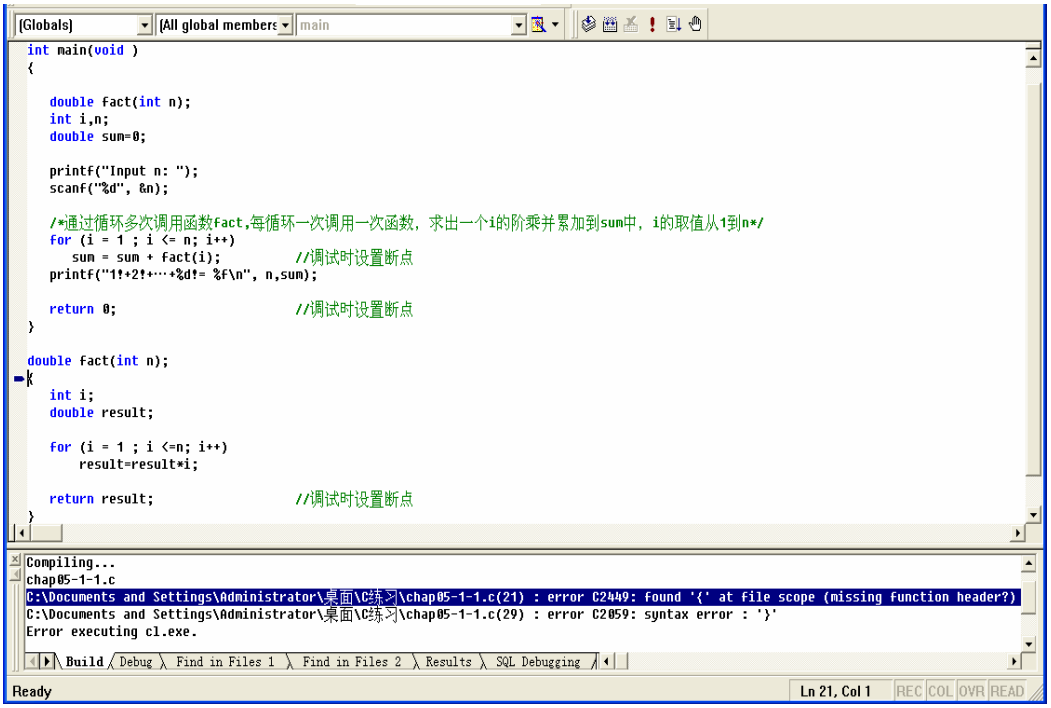


图 5.3　重新编译后给出错误提示的界面

③ 重新编译和连接程序，没有错误提示。但运行程序后，结果不正确，如图 5.4 所示。需要调试程序，

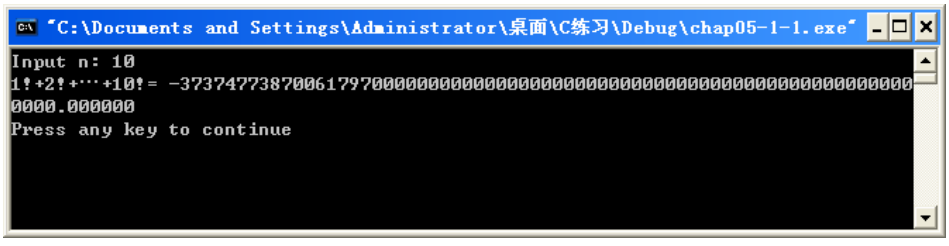


图 5.4　程序运行结果

④ 运用断点调试方法，在程序中设置 3 个断点，具体位置见上面程序中给出的注释所示。单击工具条的 (Go)按钮，输入 n 的值 10 后，程序执行到第一个断点位置暂停，如图 5.5 所示。

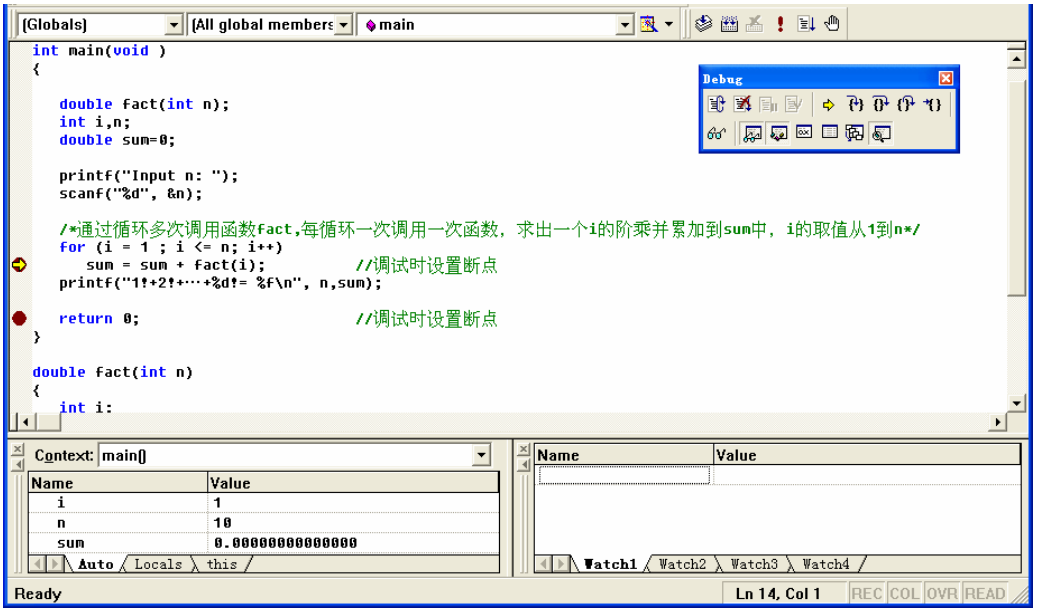


图 5.5　断点调试

⑤ 单击 (Step Into)按钮，进入函数 fact()进行调试，箭头表示程序已经执行到 fact()函数内，如图 5.6 所示。

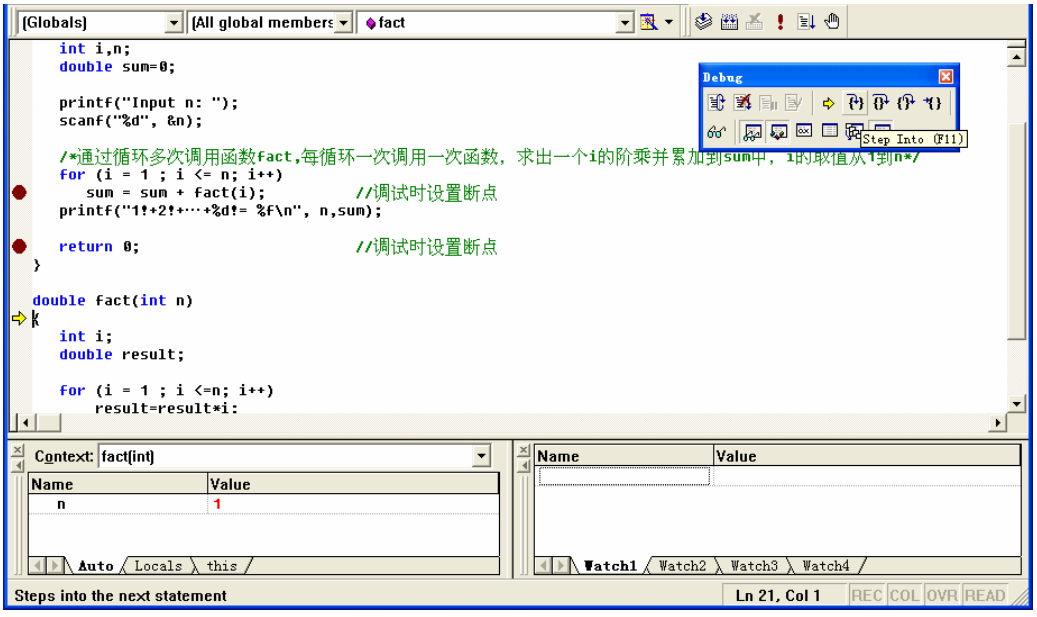


图 5.6　进入函数

⑥ 再次单击 (Go)按钮，程序执行到 fact()函数中的断点位置暂停，如图 5.7 所示。在变量窗口观察到 result 的值是-9.2559631349318e+061，显然不正确，原因是 result 未赋初值。改正错误，加上“result=1;”语句后，重新编译和连接，按以上步骤，运行到该断点位置，变量窗口中 result 的值为 1(1!)正确。

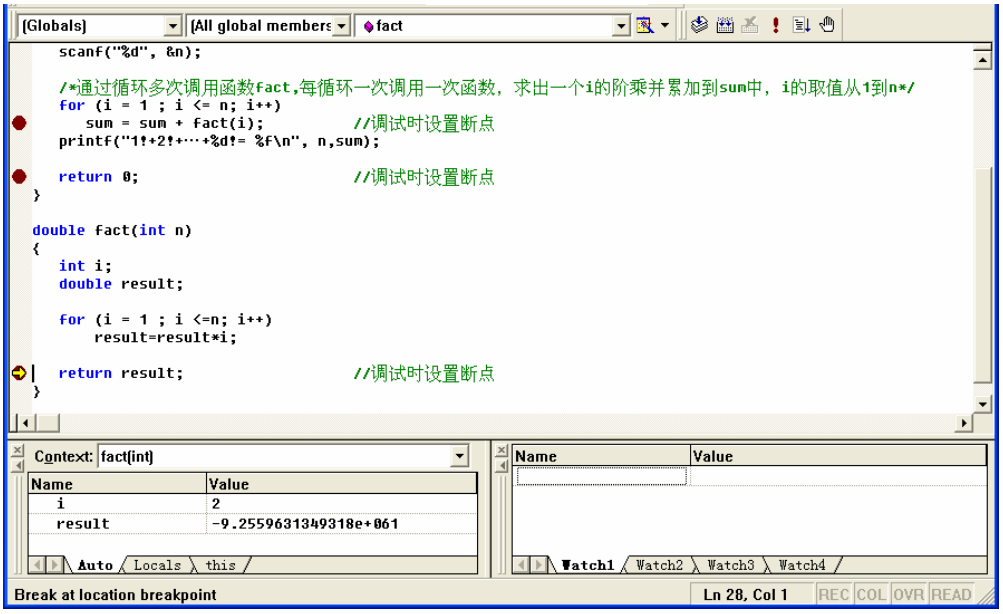


图 5.7　查看中间结果

⑦ 单击(Step Out)按钮，从被调用的 fact()函数返回主调函数 main()函数继续执行，如图 5.8 所示。

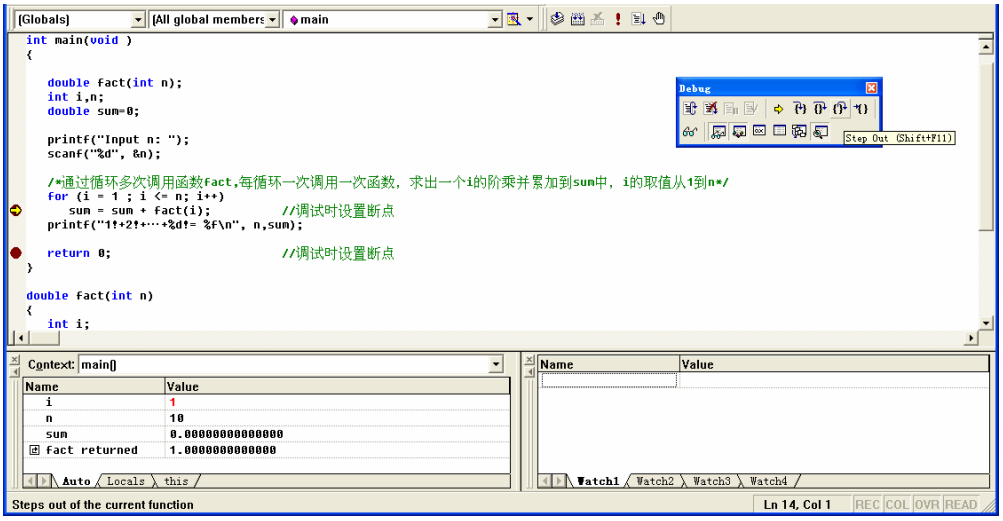


图 5.8　跳出子函数返回到主函数

⑧ 继续多次单击(Go)按钮，直到程序执行到 main()函数的最后一个断点位置，运行窗口中输出 1!+2!+…+10!=4037913.000000，结果正确。

⑨ 单击(Stop Debugging)按钮，程序调试结束。

实验内容

1. 下述程序的功能是求半个圆的面积，函数通过形参得到圆的半径，返回半个圆的面积。例如输入圆的半径值为：19.527，输出为：s=598.950017。

请在程序的下划线处填入正确的内容并把下划线删除，使程序得出正确的结果。

注意：不得增行或删行，也不得更改程序的结构！

```
#include<stdio.h>
float fun(float r)
{
    return 3.14159 * ____[1]_____ /2.0;
}
void main()
{
    ____[2]_____ x;
    printf("Enter x: ");
    scanf("%f", _____[3]_____ );
    printf ("s = %f\n ", fun(x) );
}
```

2. 下述程序中 fun()函数的功能是：计算正整数 num 的各位上的数字之积。例如，若输入 252，则输出 20；若输入：202，则输出：0。请改正程序中的错误，使它能得出正确的结果。

注意：不得增行或删行，也不得更改程序的结构！

```
#include<stdio.h>
long fun(long num)
{
    long k;
    do{
        k*=num%10 ;
        num\=10 ;
      }while(num);
    return (k) ;
}
void main()
{
    long n ;
    printf("Please enter a number:") ;  scanf("%ld",&n);
    printf("%ld\n", fun(n)) ;
}
```

3. 编写程序，求 $S=\frac{1}{1\times 2}+\frac{1}{2\times 3}+\cdots+\frac{1}{n(n+1)}$ 的值。要求定义和调用 fun(n)函数，计算 S 的值。请补充完整 fun()函数。

```
#include<stdio.h>
double fun(int n)
{




}
```

```
void main()        /* 主函数 */
{
    int n;
    printf("Input n:");
    scanf("%d", &n);
    printf("%f\n", fun(n));
}
```

程序运行示例：

```
Input n: 10
S=0.909091
```

4. 输入两个正整数 m 和 n(m>n)，编写程序求 $P=\frac{m!}{n!(m-n)!}$ 的值。要求定义和调用 fact() 函数，计算 n!。请补充完整 fun() 函数。

```
#include<stdio.h>
float fun(int n)
{




}
void main()        /* 主函数 */
{
    int m, n;
    printf("Input m and n:");
    scanf("%d%d", &m, &n);
    printf("P=%.0f\n", fact(m)/(fact(n)*fact(m-n)));
}
```

程序运行示例：

```
Input m and n: 12 8
P=495
```

5. 输入两个正整数 m 和 n(m<n 且 m≥1，n≤300)，求 m 和 n 之间所有素数的和。素数就是除了 1 之外，只能被 1 和自身整除的正整数，1 不是素数，2 是素数。要求定义和调用 prime(n) 函数判断 n 是否为素数，当 n 为素数时返回 1，否则返回 0。

程序运行示例：

```
Input m: 1
Input n: 10
The sum is: 17
```

(1 到 10 之间的素数有 4 个：2、3、5、7。)

6. 输入一个正整数 n(n≤100)，编写程序计算并输出 n 的所有因子(不包括 1 与自身)的平方和。要求定义和调用 fun(n) 函数，计算 n 的所有因子的平方和。

程序运行示例：

```
Input n: 56
sum=1113
```

实验总结与分析

5.2 递归函数与数组名作函数参数

实验目的

1. 熟练掌握使用函数编写程序。
2. 熟悉递归函数的编程方法。
3. 掌握用数组名作为函数参数的编程方法。

实验案例

【例 5.2】有 5 个人坐在一起，问第 5 个人多少岁，他说比第 4 个人大两岁；问第 4 个人岁数，他说比第 3 个人大两岁；问第 3 个人，又说比第 2 个人大两岁。问第 2 个人，说比第 1 个人大两岁。最后问第 1 个人，他说是 10 岁。请问第 5 个人多少岁？

【分析】要求第 5 个人的年龄，就必须先知道第 4 个人的年龄，而要求第 4 个人的年龄，又必须先知道第 3 个人的年龄，第 3 个人的年龄取决于第 2 个人的年龄，第 2 个人的年龄又取决于第 1 个人的年龄。而且每一个人的年龄都比其前一个人的年龄大 2，这显然是一个递归问题。即：

age(5) = age(4) +2
age(4) = age(3) +2
age(3) = age(2) +2
age(2) = age(1) +2
age(1) = 10

用数学公式表述如下：

$$age(n)=\begin{cases}10 & n=1\\ age(n-1)+1 & n>1\end{cases}$$

函数调用过程如图 5.9 所示。

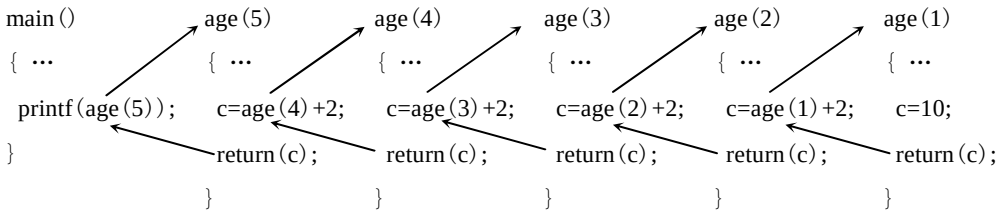


图 5.9　例 5.2 函数调用过程图

参考代码：

```
#include<stdio.h>
int  main(void)
{
     int  age(int  n);
     printf("%d\n", age(5));
}
int  age(int n)                    //定义求年龄的递归函数
{
     int  c;
     if(n==1)
          c=10;
     else
          c=age(n-1)+2;
     return(c);
}
```

【说明】递归问题可以分为“递推”和“回归”两个阶段，要经历若干步才能得到最后的值。要使递归过程不是无限制地进行下去，就必须有一个结束递归过程的条件，即递归出口。递归出口和递归式是递归函数的两大要素。

【思考】① 例 5.2 的程序中的递归结束的条件和递推公式分别是什么？

② 在某一次调用 age()函数时能否立即得到 age(n)的值？如果不能，什么时候才能得到确定的值？

【例5.3】下述程序中，fun()函数的功能是：对数组a表示的n种商品按价格分段进行统计，各段商品价格的数目存到数组b中；其中：价格为100元以下的个数存到b[0]中，价格为100元到199元的个数存到b[1]，价格为200元到299元的个数存到b[2]，价格为300元到399元的个数存到b[3]，价格为400元到499元的个数存到b[4]，价格为500元以上的个数存到b[5]中。

例如，当数组a中的数据为90、180、270、380、590、530、140、750、380时，调用该fun()函数后，b中存放的数据是1、2、1、2、0、3。

部分源程序已给出，请在fun()函数的花括号中填入相应的程序语句。

```
#include<stdio.h>
void  fun(int  a[], int  b[], int  n)
{



}
```

```
void main()
{   int i, a[100]={90, 180, 270, 380, 590, 530, 140, 750, 380}, b[6];
    fun(a, b, 9);
    printf("The result is: ");
    for(i=0; i<6; i++)
        printf("%d ", b[i]);
    printf("\n");
}
```

【分析】本题是一个计算分段函数值的问题，用两个循环来实现。第一个循环使 b 数组的所有元素值都为 0，这个循环不能少，因为若 b 中元素不初始化，则它们的值是随机值。第 2 个循环的作用是分段统计各价格段的商品数目。

参考代码:

```
void fun(int a[], int b[], int n)
{
    int i;
    for(i=0; i<6; i++)
        b[i]=0;
    for(i=0; i<n; i++)
    {   if(a[i]<100)                b[0]++;
        if(a[i]>=100&&a[i]<199)     b[1]++;
        if(a[i]>=200&&a[i]<299)     b[2]++;
        if(a[i]>=300&&a[i]<399)     b[3]++;
        if(a[i]>=400&&a[i]<499)     b[4]++;
        if(a[i]>500)                b[5]++;
    }
}
```

【例5.4】下述程序中定义了一个N×N二维数组表示一个方阵，在主函数中给数组自动赋值。请编写fun(int a[][N])函数，其功能是使方阵左下三角元素中的值全部置成0。例如：原来的数组a对应的方阵为：$\begin{pmatrix}1&9&7\\2&3&8\\4&5&6\end{pmatrix}$，则返回主程序后数组a对应的方阵为$\begin{pmatrix}0&9&7\\0&0&8\\0&0&0\end{pmatrix}$。

部分源程序已给出，请在fun()函数的花括号中填入相应的语句。

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define N 5
void fun(int a[][N])
{

}
```

```
void main()
{   int a[N][N], i, j;
    printf("***** The array *****\n");
    for(i =0; i<N; i++ )
    {   for(j =0; j<N; j++ )
        {   a[i][j] =rand()%10;          //调用库函数，以随机数给数组赋值
            printf("%4d", a[i][j]);
        }
        printf("\n");
    }
    fun(a);                              //调用函数，以二维数组名作参数
    printf("THE RESULT\n");
    for(i =0; i<N; i++ )
    {   for(j =0; j<N; j++ )
            printf( "%4d", a[i][j]);
        printf("\n");
    }
}
```

【分析】二维数组名作函数实参时，形参也用二维数组的形式，当函数调用时，实参数组与形参数组占同一段内存单元。要使方阵左下三角元素中的值全部置成 0，也就是对满足条件 i>=j 的元素值都赋为 0。

参考代码：

```
void fun(int a[][N])
{
    int i, j;
    for(i=0; i<N; i++)
        for(j=0; j<N; j++)
            if(i>=j)
                a[i][j]=0;
}
```

实验内容

1. 下述程序中 fun()函数的功能是：求 k!(k<13)，所求阶乘的值作为函数值返回。请修改程序中的错误，使它能得出正确的结果。

```
#include<stdio.h>
long fun(int k)
{
    if(k=0 )
        return 1L;
    else if k > 0
```

```
        return (k*fun(k-1));
    }
    main()
    {
        int k;
        printf("Input k:");
        scanf("%d", &k);
        printf("%d!=%ld\n", k, fun(k));
    }
```

程序运行示例：

```
Input k: 10
10!=3628800
```

2. 已知一个数列从第 1 项开始的前 3 项为 0、0、1，以后的各项都是其相邻的前 3 项之和。要求使用递归方法编写 fun(n)函数，求该数列第 n 项的值，返回值为长整型，并写出相应的主函数。

程序运行示例：

```
Input n: 8
The result is: 13
```

3. 下述程序的功能是计算数组 score 中 m 个人的平均成绩 aver，将低于 aver 的成绩放在 below 中，通过函数返回值给出低于平均成绩的人数。例如，当 score={10, 20, 30, 40, 50, 60, 70, 80, 90}，m=9 时，函数返回值应该是 4，below={10, 20, 30, 40}。

请在程序的下划线处填入正确的内容并把下划线删除，使程序得出正确的结果。

【注意】不得增行或删行，也不得更改程序的结构！

```
#include<stdio.h>
int fun(int score[], int m, int below[])
{   int i, j = 0 ;
    float aver = 0.0 ;
    for(i = 0 ; i < m ; i++)
        aver += score[i] ;
    aver /= m ;
    for(i = 0 ; i < m ; i++)
        if(score[i] < aver)
            below[j++] = _____[1]_____ ;
    return j ;
}
void main()
{   int i, n, below[9] ;
    int score[9] = {10, 20, 30, 40, 50, 60, 70, 80, 90} ;
    n = fun(score, 9, _____[2]_____) ;
    printf("\nBelow the average score are: ") ;
    for (i = 0 ; i < n ; i++)
```

```
        printf("%d ", _____[3]_____);
    }
```

4. 下述程序中 fun()函数的功能是：由形参给定 n 个实数，输出平均值，并统计在平均值以上(含平均值)的实数个数。请改正程序中的错误，使程序能输出正确的结果。

```
#include<stdio.h>
int  fun(double  x, int  n)
{
    int  j, c=0, xa=0;
    for(j=0; j<n; j++ )
        xa += x[j]/n;
    printf("平均值为%f\n", xa);
    for(j=0; j<n; j++ )
        if(x[j] => xa)
            c++;
    return  c;
}
void  main ( )
{   double  x[100] = {193.199,  195.673,  195.757,  196.051,  196.092,  196.596,  196.579,
                      196.763};
    printf("平均值以上的实数个数为：%d\n", fun(x, 8));
}
```

程序运行示例：

```
平均值为:195.8387
平均值以上的实数个数为:5
```

5. 编写程序用选择法对数组 a 中的 n 个元素按从小到大的顺序排序。要求定义和调用 fun(a, n)函数，对数组元素进行排序。请补充完整 fun 函数。

```
#include<stdio.h>
#define  N  20
void  fun(int  a[], int  n)
{



}
void  main()
{
    int a[N]={9, 6, 8, 3, -1}, i, m = 5;
    printf("排序前的数据:");
    for(i = 0; i < m; i++)
        printf("%d ", a[i]);
    printf("\n");
    fun(a, m);
```

```
    printf("排序后的数据:");
    for(i = 0; i < m; i++)
        printf("%d ", a[i]);
     printf("\n");
}
```

程序运行示例：

```
排序前的数据: 9 6 8 3 -1
排序后的数据:-1 3 6 8 9
```

6. 下述程序中定义了一个N×N二维数组表示一个方阵，在主函数中给数组自动赋值。请编写fun函数，其功能是使方阵右上三角元素中的值乘以m。例如：若m的值为2，数组a对应的方阵为：$\begin{pmatrix}1 & 9 & 7\\ 2 & 3 & 8\\ 4 & 5 & 6\end{pmatrix}$，则返回主程序后数组a对应的方阵为$\begin{pmatrix}2 & 18 & 14\\ 2 & 6 & 16\\ 4 & 5 & 12\end{pmatrix}$。

请勿改动main主函数中的任何内容，仅在fun函数的花括号中填入你编写的若干语句。

```
#include <stdio.h>
#define  N  5
void  fun int  a[][N], int  m)
{



}
void  main()
{   int  a[N][N], m, i, j;
    printf("***** The  array *****\n");
    for( i =0; i<N; i++ )
    {   for(j =0; j<N; j++ )
        {
            a[i][j] = rand()%20;
            printf("%4d", a[i][j]);
        }
        printf("\n");
    }
    do
     {
        m = rand()%10 ;
     }while(m>=3 );
    printf("m = %4d\n", m);
    fun(a, m );
    printf("THE  RESULT\n");
    for(i =0; i<N; i++ )
```

```
    {
        for(j =0; j<N; j++ )
            printf("%4d", a[i][j]);
        printf("\n");
    }
}
```

实验总结与分析

第 6 章　指　　针

6.1　指针与数组

实验目的

1. 掌握指针的概念，会定义和使用指针变量。
2. 学会使用指针操作数组元素。
3. 学会使用数组、指针变量作函数的参数。

实验案例

【例 6.1】阅读下述三个程序，先自行分析程序运行结果，然后上机调试运行。

程序 1：

```
#include<stdio.h>
void  main()
{     int  *p, a;
      a=10;
      p=&a;
      a=*p+10;
      printf("%d ", a);
}
```

程序 2：

```
#include<stdio.h>
void  main()
{     int  a[]={1, 2, 3, 4, 5};
      int  *p, x, y;
      p=&a[0];
      x=*(p+2);
      y=*(p+4);
      printf("*p=%d,x=%d,y=%d\n ",*p, x, y);
}
```

程序 3：

```
#include<stdio.h>
void  main()
{     int  a[]={1, 3, 5, 7, 9}, *p;
      p=&a[1];
```

```
    printf("%d, \n ", *p++);
    printf("%d, \n ", *p);
    printf("%d, \n ", (*p)++);
    printf("%d, \n ", *p);
    printf("%d, \n ", *p++);
    printf("%d, \n ", *p);
}
```

【例 6.2】下述程序中，用 msort 函数实现选择法排序，用指针作函数参数。

① 阅读程序，在领会程序功能的基础上，给语句加上注释。

```
#include<stdio.h>
#define  MAX  20
void  msort(int  *x, int  n);
void  main()
{
    int  *p, i, a[MAX], n;
    printf("Enter  n: ");
    scanf("%d", &n);
    printf("Enter  %d  integer：", n);
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    p=a;
    msort(p, n);
    for(i=0; i<n; i++)
        printf("%d", a[i]);
}
void  msort(int  *x, int  n)                    /*也可以写成 void  msort(int  x[], int  n)*/
{
    int  i, j, k, temp;
    for(i=0; i<n-1; i++)
    {   k= i;
        for(j=i+1; j<n; j++)
            if(*(x+j)<*(x+k))
                k=j;
        temp=*(x+i);   *(x+i)= *(x+k);   *(x+k)=temp;
    }
}
```

② 阅读改写后的 msort 函数，在领会程序功能的基础上，给语句加上注释。

```
void  msort(int  *x, int  n)              /*也可以写成 void  msort(int  x[], int  n)*/
{
    int  *i, *j, *k, temp;
    for(i=x; i<x+n-1; i++)
```

```
        {    k=i;
             for(j=i+1; j<x+n; j++)
                  if(*j<*k)
                       k=j;
             temp=*i;   *i= *k;   *k=temp;
        }
   }
```

③ 阅读再改写后的 msort 函数，在领会程序功能的基础上，给语句加上注释。

```
   void  msort(int  *x, int  n)          /*也可以写成 void  msort(int  x[], int  n)*/
   {
        int  i, j, k, temp;
        for(i=0; i<n-1; i++)
        {    k= i;
             for(j=i+1; j<n; j++)
                  if(x[j] < x[k])
                       k=j;
             temp=x[i];   x[i]= x[k];   x[k]=temp;
        }
   }
```

④ 运行上述程序，体会函数参数的形式虽然不同，但它们的实质都是传递地址。

实验内容

1. 下述程序中fun函数的功能是：将a、b两个二位正整数合并，形成一个整数c。合并的方式是：将a数的十位和个位数依次放在c数的千位和十位上，将b数的十位和个位数依次放在c数的百位和个位上。例如，若a为45，b为12，调用该函数后，c为4152。在fun()函数的花括号中填入你编写的若干语句。

```
#include<conio.h>
#include<stdio.h>
void  fun(int  a, int  b, long  *c)
{


}
void  main()
{   int  a, b; long  c;
    printf("Input  a, b:");
    scanf("%d%d", &a, &b);
    fun(a, b, &c);
    printf("The  result  is: %d\n", c);
}
```

2. 下述程序中，一维整型数组a的元素已在主函数中赋值；fun函数的功能是：找出数组a的元素中最大的值和它对应的下标，通过形参传回最大的值和它对应的下标。

主函数中x是数组名，n是x中的数据个数，max存放最大值，index存放最大值元素对应的下标。在fun函数的花括号中填入你编写的若干语句。

```
#include<stdlib.h>
#include<stdio.h>
void  fun(int  a[], int  n, int  *max, int  *d)
{


}
void  main()
{    int  i, x[20], max, index, n = 10;
     for(i=0; i < n; i++)
    {   x[i] = rand()%50;            //调用库函数，用随机数给数组赋值
        printf("%4d", x[i]);
    }
     printf("\n");
     fun(x, n , &max, &index);
     printf("Max =%5d , Index =%4d\n", max, index );
}
```

3. 下述程序中，假设数组 a 中原有 10 个元素，编写在数组中插入指定下标 j 的位置插入值为 m 的元素的 minsert(int a[], int j, int m, int n)函数。j，m 的值由键盘输入。补充完成函数的定义。

```
#include<stdio.h>
void  minsert(int  a[], int  j, int  m, int  n);
int  main()
{    int  a[11]={5, 6, 4, 8, 3, 7, 9, 2, 1, 10};
     int  i=0, j, m;
     printf("输入插入的位置和值：");
     scanf("%d%d", &j, &m);
     minsert(a, j, m, 10);
     for(i=0; i<11; i++)
         printf("%d", a[i]);
     return  0;
}
void  minsert(int  a[], int  j, int  m, int  n)
{


}
```

4. 下述程序中，fun函数的功能是：把a数组中的n个数和b数组中逆序的n个数一一相减，结果存在数组c中。

例如:当a数组中的值是：1、3、5、7、8，
　　b数组中的值是：2、3、4、5、8。
调用fun函数后，数组c中存放的数据是：−7、−2、1、4、6。
在fun函数的花括号中填入你编写的若干语句。

```
#include<conio.h>
#include<stdio.h>
void fun(int a[], int b[], int c[], int n)
{



}
void main()
{   int i, a[100]={1, 3, 5, 7, 8}, b[100]={2, 3, 4, 5, 8}, c[100];
    fun(a, b, c, 5);
    printf("The result is: ");
    for(i=0; i<5; i++)
        printf("%d ", c[i]);
    printf("\n");
}
```

实验总结与分析

6.2 指针与字符串

实验目的

1. 熟悉如何定义字符型的指针变量和将指针变量指向字符串首地址的操作。
2. 掌握通过指针判断字符串结束的方法。
3. 掌握如何将两个字符串首尾连接的基本操作。

实验案例

【例 6.3】阅读下述三个程序，先自行分析结果，然后上机调试运行。

程序 1：

```
#include<stdio.h>
void  main()
{
    int  **k, *p, a=100;
    p=&a;
    k=&p;
    printf("%d \n", **k);
}
```

程序 2：

```
#include<stdio.h>
void  main()
{
    int  a[3][4]={1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23};
    int  (*p)[4];
    int  sum=0, i, j;
    p=a;
    for(i=0; i<3; i++)
        for(j=0; j<2; j++)
            sum+=*(*(p+i)+j);
    printf("sum=%d\n ", sum);
}
```

程序 3：

```
#include<stdio.h>
void  main()
{
    char  *str="abcde";
    printf("%c, ", *str);
    printf("%c, ",   *str++);
    printf("%c, ", *++str);
    printf("%c, ", *(str)++);
    printf("%c,\n ", *str);
}
```

【例 6.4】编制程序，将 str1 和 str2 两个字符串首尾连接。阅读程序，加上注释；然后运行程序，观察结果。

【分析】定义指针变量 p 和 q，将指针 p 指向 str1 字符串串首，将 q 指针指向 str2 字符串串首；通过指针 p 找到 str1 字符串串尾；将 q 所指字符串接到 p 所指字符串后；为 p 所指字符串赋串结束标志。

【注意】存放 str1 串的数组要足够大，保证能够存放连接后的字符串。

参考代码：

```
#include<stdio.h>
int  main()
{
      char  str1[80], str2[10], *p=str1, *q=str2;
      printf("输入字符串 1(70 个字符以内)：");
      gets(str1);
      printf("输入字符串 2(10 个字符以内)：");
      gets(str2);
      while(*p!= '\0')
            p++;
      while(*q!= '\0')
      {     *p=*q;
           q++;
            p++;
      }
      *p='\0';
      printf("连接后的字符串：");
      puts(str1);
      return  0;
}
```

【思考】① 能否将程序中第 2 个 while 循环修改为：

```
while(*q)
   *p++=*q++;
```

② 改写源程序，编写实现字符串连接的 void castr(char *p, char *q)函数并调试运行。

实验内容

1. 编写程序，实现矩阵(3行3列)的转置(即行列互换)。

例如，输入下面的矩阵元素：

```
100   200   300
400   500   600
700   800   900
```

程序输出：

```
100   400   700
200   500   800
300   600   900
```

在下述程序的fun函数花括号中填入你编写的语句。

```
#include<stdio.h>
#include<conio.h>
```

```
void fun(int array[3][3])
{

}
void main()
{   int i, j;
    int array[3][3]={{100, 200, 300}, {400, 500, 600}, {700, 800, 900}};
    for(i=0; i<3; i++)
    {    for(j=0; j<3; j++)
             printf("%7d", array[i][j]);
         printf("\n");
    }
    fun(array);
    printf("Converted array:\n");
    for(i=0; i<3; i++)
    {    for(j=0; j<3; j++)
             printf("%7d", array[i][j]);
         printf("\n");
    }
}
```

2. 下述程序中fun函数的功能是：对指定字符在字符串a中出现的次数进行统计，统计的数据存到数组b中。其中：字符'a'出现的次数存放到b[0]中，字符'b'出现的次数存放到b[1]中，字符'c'出现的次数存放到b[2]中，字符'd'出现的次数存放到b[3]中，字符'e'出现的次数存放到b[4]中，其他符出现的次数存到b[5]中。

例如，当a字符串为"bacd1b+ddep"，调用该函数后，b中存放数据应是：1、2、1、3、1、3。在fun函数的花括号中填入你编写的语句。

```
#include<conio.h>
#include<stdio.h>
#include<string.h>
void fun(char *a, int b[])
{

}
void main()
{   int i, b[6]; char a[100] = "bacd1b+ddep";
    fun(a, b);
    printf("The result is: ");
    for(i=0; i<6; i++)
        printf("%d ", b[i]);
    printf("\n");
}
```

3. 下述程序中fun()函数的功能是：删除s所指字符串中下标为偶数的字符，串中剩余字符形成的新串放在t所指数组中。

例如，若s所指字符串中的内容为"ABCDEFGHIJK"，

在t所指数组中的内容应为"BDFHJ"。

在fun()函数的花括号中填入你编写的语句。

```
#include<conio.h>
#include<stdio.h>
#include<string.h>
void fun(char *s, char t[])
{

}
void main()
{
    char s[100], t[100];
    printf("\nPlease enter string S:");    scanf("%s", s);
    fun(s, t);
    printf("\nThe result is: %s\n", t);
 }
```

4. 下述程序中fun()函数的功能是：将输入的ss所指字符串中所有下标为奇数位置上的字母转换为大写(若该位置上不是字母，则不转换)。

例如，若输入为"abc4EFg"，则应输出"aBc4EFg"。

在fun函数的花括号中填入你编写的语句。

```
#include <conio.h>
#include <stdio.h>
void fun(char *ss)
{

}
void main(void)
{
    char tt[51];
    printf("\nPlease enter an character string within 50 characters:\n");
    gets(tt);
    printf("\n\nAfter changing, the string\n  \"%s\"", tt);
    fun(tt);
    printf("\nbecomes\n  \"%s\"", tt);
 }
```

5. 下述程序中fun()函数的功能是：判断字符串是否为回文？若是，函数返回1，主函数中输出：YES；否则返回0，主函数中输出NO。回文是指顺读和倒读都一样的字符串。

例如， "LEVEL"字符串是回文，而123312"字符串"就不是回文。

在fun()函数的花括号中填入你编写的语句。

```
#include<stdio.h>
#define  N  80
int  fun(char  *str)
{



}
void  main()
{
    char  s[N];
    printf("Enter  a  string: ");
    gets(s);
    printf("\n\n");
    puts(s);
    if(fun(s))
        printf("YES\n");
    else
        printf("NO\n");
 }
```

实验总结与分析

第 7 章　构造数据类型与编译预处理

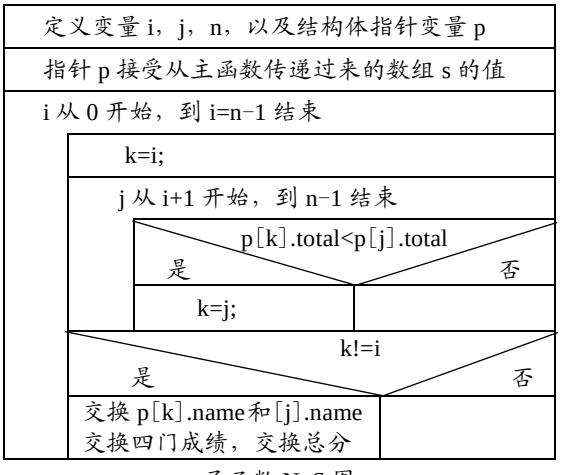
实验目的

1. 掌握结构变量和结构数组的基本使用方法。
2. 掌握结构的简单嵌套应用。
3. 掌握结构指针的概念，以及用结构指针作为函数参数的编程方法。

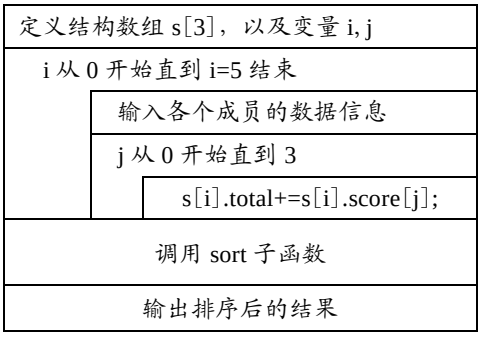
实验案例

【例 7.1】用结构数组存放一个数据库，库中包含 5 个学生的有关信息，包括姓名和数学、计算机、英语、体育 4 门课程的成绩及总分，其中，总分由程序自动计算，主程序能输出排序后的数组，按总分从高到低排序由 sort()子程序完成。

【分析】在主程序中定义结构数组，输入各成员数据，进而计算出各人的总分，调用子程序对结构数组排序，最后在主程序中输出排序后的结果。



子函数 N-S 图



主函数 N-S 图

图 7.1　例 7.1 的 N-S 图

由于存在函数调用，且在函数间要传递结构数组，因此，最好先定义一个外部结构类型，以保证各函数中的结构类型一致。当结构数组在函数间传递时，实参为结构数组名或结构指针，形参可以是结构指针或结构数组名。程序的 N-S 图如图 7.1 所示。

参考代码：

```
#include<stdio.h>
#include<string.h>
struct  stu
{
    char  name[30];
```

```
    int  score[4];
    int  total;
};
void  sort(struct  stu  *p, int  n)                    /*对结构数组用选择法排序*/
{
    int  i, j, k, m, temp;
    char  t[30];
    for(i=0; i<n-1; i++)
    {
        k=i;
        for(j=i+1; j<n; j++)
          if(p[k].total<p[j].total)
              k=j;
        if(k!=i)                                       /*交换元素*/
        {
            strcpy(t, p[k].name);                      /*交换姓名*/
            strcpy(p[k].name, p[i].name);
            strcpy(p[i].name, t);
            for(m=0; m<4; m++)                         /*交换四门课程成绩*/
            {
                temp=p[k].score[m];
                p[k].score[m]=p[i].score[m];
                p[i].score[m]=temp;
            }
            temp=p[k].total;                           /*交换总分*/
            p[k].total=p[i].total;
            p[i].total=temp;
        }
    }
}
int  main(void)
{
    struct  stu s[5];
    int  i, j;
    printf("\nEnter  name  math  computer  English  Physical:\n");
    for(i=0; i<5; i++)                  /*输入数据*/
    {
        scanf("%s%d%d%d%d", s[i].name, &s[i].score[0], &s[i].score[1],
              &s[i].score[2], &s[i].score[3]);
        s[i].total=0;
        for(j=0; j<4; j++)              /*计算总分*/
```

```
            s[i].total+=s[i].score[j];
        }
        sort(s, 5);                    /*调用排序子程序*/
        for(i=0; i<5; i++)             /*输出排序后的结果*/
        printf("%s, %d, %d, %d, %d, %d\n", s[i].name, s[i].score[0], s[i].score[1],
            s[i].score[2], s[i].score[3], s[i].total);
        return  0;
    }
```

程序运行示例：

```
Li  89  98  43  69
Wang  90  80  70  80
Wu  80  100  70  80
Zhang  90  89  90  99
Zhao  68  90  90  89
Zhang, 90, 89, 90, 99, 368
Zhao, 68, 90, 90, 89, 337
Wu, 80, 100, 70, 80, 330
Wang, 90, 80, 70, 80, 320
Li, 89, 98, 43, 69, 229
```

【思考】① 能够直接交换两个结构体变量的值吗？如何修改程序实现这一点？

② 如何把输出的数据格式化？（姓名列左对齐，数字列右对齐。）

实验内容

1. 已知学生的记录由学号和学习成绩构成，N名学生的数据已经存入a结构体数组中。下述给定程序的功能是找出成绩最低的学生记录，通过形参返回主函数。请在程序的下划线处填入正确的内容并把下划线删除，使程序得出正确的结果。

```
#include<stdio.h>
#include<string.h>
#define  N  10
typedef  struct  ss
{   char  num[10]; int  s;}STU;
fun(STU  a[], STU  *s)
{   _____[1]_____  h; int  i ;
    h = a[0];
    for(i = 1; i < N; i++ )
        if(a[i].s < h.s )
            _____[2]_____ = a[i];
    *s = _____[3]_____ ;
}
```

```
void  main()
{   STU  a[N]={{"A01", 81}, {"A02", 89}, {"A03", 66}, {"A04", 87}, {"A05", 77},
                {"A06", 90}, {"A07", 79}, {"A08", 61}, {"A09", 80}, {"A10", 71}}, m;
    int  i;
    printf("***** The original data *****\n");
    for(i=0; i< N; i++ )
        printf("No = %s    Mark = %d\n", a[i].num, a[i].s);
    fun(a, &m );
    printf ("***** THE    RESULT *****\n");
    printf("The lowest:    %s , %d\n", m.num, m.s);
}
```

2. 下述程序中已建立了一个带有头结点的单向链表，链表中的各结点按结点数据域中的数据从小到大顺序链接。fun函数的功能是：把形参x的值放入一个新结点并插入到链表中，插入后各结点仍保持从小到大顺序排列。请在程序的下划线处填入正确的内容并把下划线删除，使程序能得到正确的结果。

注意：不得增行或者删行，也不得修改程序的结构。

```
#include<stdio.h>
#include<stdlib.h>
#define  N  8
typedef  struct  list
{   int  data;
    struct  list  *next;
}SLIST;
void  fun(SLIST  *h, int  x)
{   SLIST  *p, *q, *s;
    s=(SLIST  *)malloc(sizeof(SLIST));
    s->data=___[1]___;
    q=h;
    p=h->next;
    while(p!=NULL && x>p->data)
    {
        q=___[2]___;
        p=p->next;
    }
    s->next=p;
    q->next=___[3]___;
}
SLIST  *creatlist(int  *a)
{   SLIST  *h, *p, *q; int  i;
    h=p=(SLIST *)malloc(sizeof(SLIST));
    for(i=0; i<N; i++)
```

```
    {   q=(SLIST *)malloc(sizeof(SLIST));
        q->data=a[i];   p->next=q;   p=q;
    }
    p->next=0;
    return h;
}
void outlist(SLIST *h)
{   SLIST *p;
    p=h->next;
    if(p==NULL)
        printf("\nThe list is NULL!\n");
    else
    {   printf("\nHead");
        do { printf("->%d", p->data);     p=p->next; } while(p!=NULL);
        printf("->End\n");
    }
}
void main()
{   SLIST *head; int x;
    int a[N]={11, 12, 15, 18, 19, 22, 25, 29};
    head=creatlist(a);
    printf("\nThe list before inserting:\n");   outlist(head);
    printf("\nEnter a number:  ");   scanf("%d", &x);
    fun(head, x);
    printf("\nThe list after inserting:\n");   outlist(head);
}
```

3. 下述程序通过定义学生结构体变量存储学生的学号、姓名和3门课的成绩。fun函数的功能是将形参a所指结构体变量中的数据赋给函数中的结构体变量b，并修改b中的学号和姓名，最后输出修改后的数据。例如：a所指变量中的学号、姓名和3门课的成绩依次是：

10001、"ZhangSan"、95、80、88

则修改后输出b中的数据应为：

10002、"LiSi"、95、80、88

请在程序的下划线处填入正确的内容并把下划线删除，使程序能得出正确的结果。

注意：不得增行或删行，也不得更改程序的结构。

```
#include<stdio.h>
#include<string.h>
struct student{
    long sno;
    char name[10];
    float score[3];
};
```

```
void fun(struct student a)
{   struct student b; int i;
    b = __[1]__;
    b.sno = 10002;
    strcpy(__[2]__, "LiSi");
    printf("\nThe data after modified :\n");
    printf("\nNo: %ld  Name: %s\nScores:  ", b.sno, b.name);
    for(i=0; i<3; i++)
        printf("%6.2f ", b.__[3]__);
    printf("\n");
}
void main()
{   struct student s={10001, "ZhangSan", 95, 80, 88};
    int i;
    printf("\n\nThe original data :\n");
    printf("\nNo: %ld  Name: %s\nScores:  ", s.sno, s.name);
    for(i=0; i<3; i++)
        printf("%6.2f ", s.score[i]);
    printf("\n");
    fun(s);
}
```

4. 编写程序，定义手机通讯录的结构体类型，要求包含联系人的姓名、年龄、联系电话、通信地址(城市、街道、门牌号、邮编)。输入一个联系人信息，并从屏幕上显示出来。

5. 编写程序，利用结构数组输入三个人的姓名和年龄，并输出三个人中最年长者的姓名和年龄。

6. 编写程序，用结构数组建立含 5 个人的通讯录，包括姓名、地址和电话号码。能根据键盘输入的姓名，输出该姓名及对应的电话号码。

实验总结与分析

第8章 文　　件

实验目的

1. 熟练掌握文件的打开、关闭、读、写等函数的使用方法。
2. 掌握文本文件及二进制文件的建立和使用。

实验案例

【例 8.1】先建立一个 wb.txt 文本文件，将下列字符串写入该文件：London，Paris，Bon，Tokyo，Detroit，Jerusalim，Bomgey，Beijing，Washington，要求每个字符串占 11 个字节。然后，读取该文件的各个字符串，并在屏幕上显示出来。

【分析】按如下步骤完成此题：

① 用"w+"或"r+"方式打开 wb.txt 文本文件。

② 用 fputs()每次写入一个字符串，共写 10 次。

③ 用 rewind()函数将文件指针移到文件开头，以便从头开始读入字符串。

④ 用 fgets()从文件中读取字符串，并输出到屏幕上，共输出 10 次。

⑤ 关闭文件。

注意：在用 fputs()函数向文件写入一个字符串时，字符串末尾的'\0'不会写入文件。例如：

```
fputs("London", fp);　　fputs("Paris", fp);
```

写入文件的一串字符为："LondonParis"。

再读入文件时，将无法分辨出这两个字符串。解决的方法是在写入一个字符串后，再写入一个换行符'\n'。例如：

```
fputs("London", fp);　　fputs("\n", fp);　　fputs("Paris", fp);　　fputs("\n", fp);
```

这样，用 fgets()函数读入文件时，只要将指定的字符个数 n 设定大一些，就能在读入 n-1 个字符之前遇到换行符而使读入终止。

用 fgets()函数从文件中读入一串字符后，会自动在末尾增加一个'\0'字符，形成字符串。例如，可以用下面的方法读入前面的两个字符串：

```
char s[2][11];
fgets(s[0], 11, fp);
fgets(s[1], 11, fp);
```

这时，s[0]中的内容为 London\n\0，s[1]中的内容 Paris\n\0。

参考代码：

```
#include<stdio.h>
#include<stdlib.h>
int main(void)
{
```

```
    FILE  *fp;
    char  st[][11]={"London", "Paris", "Bon", "Tokyo", "Detroit", "Moscow", "Jerusalim",
                    "Bomgey", "Beijing", "Washington"};
    char  s[10][11];
    int  i;
    if((fp=fopen("wb.txt", "w+"))==NULL)
    {    printf("Cannot  open  the  file!\n");
         exit(0);
    }
    for(i=0; i<10; i++)
    {    fputs(st[i], fp);
         fputs("\n", fp);
    }
    rewind(fp);
    printf("results\n");
    for(i=0; i<10; i++)
    {    fgets(s[i], 11, fp);
         puts(s[i]);
    }
    fclose(fp);
    return  0;
}
```

实验内容

1. 下述程序中，fun的函数功能是：把自然数1，2，3，…，10以及它们的平方根写到名为myfile3.txt的文本文件中，然后再顺序读出显示在屏幕上。请在程序中下划线处填入正确的内容并把下划线删除，使程序得出正确的结果。注意，不得增行或删行，也不得更改程序的结构。

```
#include<math.h>
#include<stdio.h>
int  fun(char  *fname )
{   FILE  *fp; int  i, n; float  x;
    if((fp=fopen(fname, "w"))==NULL)
        return  0;
    for(i=1; i<=10; i++)
        fprintf(____[1]____, "%d %f\n", i, sqrt((double)i));
    printf("\nSucceed!\n");
    ____[2]____;
    printf("\nThe data in file :\n");
    if((fp=fopen(____[3]____, "r"))==NULL)
```

```
        return 0;
    fscanf(fp,"%d%f", &n, &x);
    while(!feof(fp))
        {  printf("%d %f\n", n, x);
           fscanf(fp, "%d%f", &n, &x);
        }
    fclose(fp);
    return 1;
}
void main()
{  char fname[]="myfile3.txt";
   fun(fname);
}
```

2. 下述程序的功能是：从键盘输入若干行文本(每行不超过 80 个字符)，写到 myfile4.txt 文件中，用-1 作为字符串输入结束的标志。然后将文件的内容读出显示在屏幕上。读写文件分别由自定义的 ReadText 和 WriteText 函数实现。请在程序中下划线处填入正确的内容并把下划线删除，使程序得出正确的结果。注意，不得增行或删行，也不得更改程序的结构。

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
void WriteText(FILE *);
void ReadText(FILE *);
void WriteText(FILE ____[1]____)
{  char str[81];
   printf("\nEnter string with -1 to end :\n");
   gets(str);
   while(strcmp(str,"-1")!=0)
   {  fputs(____[2]____, fw);  fputs("\n", fw);
      gets(str);
   }
}
void ReadText(FILE *fr)
{  char str[81];
   printf("\nRead file and output to screen :\n");
   fgets(str, 81, fr);
   while(!feof(fr))
   {  printf("%s", ____[2]____);
      fgets(str, 81, fr);
   }
}
```

```
void main()
{   FILE *fp;
    if((fp=fopen("myfile4.txt", "w"))==NULL)
      {   printf("open fail!!\n");      exit(0);}
    WriteText(fp);
    fclose(fp);
    if((fp=fopen("myfile4.txt", "r"))==NULL)
      {   printf("open fail!!\n");      exit(0);}
    ReadText(fp);
    fclose(fp);
}
```

3. 下述程序的功能是：调用fun函数建立班级通讯录。通讯录中记录每个学生的编号、姓名和电话号码。班级的人数和学生的信息从键盘输入，每个人的信息作为一个数据块写到myfile5.dbf二进制文件中。请在程序中下划线处填入正确的内容并把下划线删除，使程序得出正确的结果。注意，不得增行或删行，也不得更改程序的结构。

```
#include<stdio.h>
#include<stdlib.h>
#define N 5
typedef struct
{   int num;
    char name[10];
    char tel[10];
}STYPE;
void check();
int fun(____[1]____ *std)
{   ____[2]____ *fp; int i;
    if((fp=fopen("myfile5.dat", "wb"))==NULL)
        return(0);
    printf("\nOutput data to file !\n");
    for(i=0; i<N; i++)
        fwrite(&std[i], sizeof(STYPE), 1, ____[3]____);
    fclose(fp);
    return (1);
}
void main()
{   STYPE s[10]={{1, "aaaaa", "111111"},{2, "bbbbb", "222222"},{3, "ccccc", "333333"},
                 {4, "ddddd", "444444"},{5, "eeeee", "555555"}};
    int k;
    k=fun(s);
    if(k==1)
      {   printf("Succeed!");      check();}
```

```
    else
        printf("Fail!");
}
void check()
{   FILE *fp; int i;
    STYPE s[10];
    if((fp=fopen("myfile5.dat", "rb"))==NULL)
        {   printf("Fail !!\n");      exit(0);}
    printf("\nRead file and output to screen:\n");
    printf("\n    num      name        tel\n");
    for(i=0; i<N; i++)
        {   fread(&s[i], sizeof(STYPE), 1, fp);
            printf("%6d      %s      %s\n", s[i].num, s[i].name, s[i].tel);
        }
    fclose(fp);
}
```

4. 编写程序从键盘输入一行字符，写到a.txt文件中。

5. Int_Data.dat 文件中存放了若干整数，将文件中所有数据相加，并把累加和写入到该文件的最后。

① 文件 Int_Data.dat 中的初始数据为：

10 15 20 50 100 200 220 280 300

② 运行程序后，文件 Int_Data.dat 中的数据：

10 15 20 50 100 200 220 280 300 1195

【提示】① 在运行程序前，读者应该首先建立 Int_Data.dat 文件。

② 运行程序时，不需要从键盘输入数据，也没有屏幕输出。

③ 运行程序后，读者可打开 Int_Data.dat 文件，检查数据是否正确。

④ 每次运行程序，都会将文件 Int_Data.dat 中所有数据的累加和写入该文件的最后，由于读写操作针对同一个 Int_Data.dat 文件，故每次运行后文件的内容都不同。

实验总结与分析

第 9 章　等级考试二级 C 语言操作试题举例

实验目的

1. 熟悉计算机等级考试二级 C 语言操作题型及特点。
2. 了解计算机等级考试二级 C 语言上机考试解题方法。

实验案例

【例 9.1】程序填空题。

下列给定程序中 fun 函数的功能是：统计 3～n(n>2)之间所有素数的个数，素数的个数作为函数值返回。

请在程序的下划线处填入正确的内容并把下划线删除，使程序得出正确的结果。

注意：不得增行或删行，也不得更改程序的结构。

```
#include<stdio.h>
int  fun(int  n)
{   int  i, j, count=0;
    printf("\nThe  prime  number  between  3  to  %d\n", n);
    for(i=3; i<=n; i++){
    /**********found**********/
        for(____1____; j<i; j++)
    /**********found**********/
            if (___2___%j == 0)
                break;
    /**********found**********/
        if (___3___>=i)
        {   count++;     printf(count%15?  "%5d" : "\n%5d", i);   }
    }
    return  count;
}
void  main()
{   int  n=20, r;
    r = fun(n);
    printf("\nThe  number  of  prime  is : %d\n", r);
}
```

【主要考点】

① 素数的判断方法。

② 自增运算符“++”的使用。

【参考答案】

1. j=2　　2. i　　3. j

【答案解析】

填空 1：这里考查判断素数时除数的变化范围，素数是只能被 1 和它本身整除的 1 以上的正整数。判断素数时，变量 j 应从 2 开始，直至 i-1，所以此处应填 j=2。

填空 2：这里考查判断素数时循环提前结束的条件。若 2 到 i-1 中，有任意一个数能整除 i，即满足条件 i%j==0，就能断定 i 不是素数，因此就没有继续循环的必要了，可以用 break 语句提前终止循环，所以此处应填 i。

填空 3：循环结束后，可以根据变量 i 和 j 的大小关系，得出结论。若是在循环中途通过 break 跳出循环，那么 j 必定小于 i，此时 i 不是素数；若 j 大于等于 i，那么 i 就是素数，使用变量 count 统计其个数。所以此处应填 j。

【例 9.2】程序修改题。

下列给定程序中，fun 函数的功能是：删除 s 所指字符串中所有的小写字母 c。

请改正程序中的错误，使程序能得出正确的结果。

注意：不得增行或删行，也不得更改程序的结构。

```
#include<stdio.h>
void fun(char *s)
{   int i, j;
    for(i=j=0; s[i]!='\0'; i++)
      if(s[i]!='c')
    /************found************/
        s[j]=s[i];
    /************found************/
    s[i]= '0';
}
void main()
{  char s[80];
   printf("Enter a string:      "); gets(s);
   printf("The original string:  "); puts(s);
   fun(s);
   printf("The string after deleted:  "); puts(s); printf("\n\n");
}
```

【主要考点】

① 数组元素下标的引用。

② 字符串结束标志'\0'。

【参考答案】

① 错误 1：s[j]=s[i];　　正确：s[j++]=s[i]; 或 s[j]=s[i]; j++;

② 错误 2：s[i]= '0';　　正确：s[j]= '\0';

【答案解析】

错误 1：在 fun 函数中，使用循环遍历字符串的每一个字符，直至遇到字符串结束符'\0'。由条件 if(s[i]!='c')可知，如果字符串中的字符不是'c'，则将保留该字符，并放到 s[j]的位置，然后下标 j 进行加 1 操作，移向下一个元素，以便保存下一个不是'c'的字符。因此，此处应在赋值之后，执行 j++，后移一个元素。

错误 2：循环结束后要在新字符串的结尾加上字符串结束符'\0'，它和数组字符'0'是不相同的，数组字符'0'的 ASCII 码是 48，因此，此处应将'0'改为'\0'。另外，新字符串的下标用 j 控制，所以应将下标 i 改为 j。

【例 9.3】程序设计题。

请编写 fun 函数，其功能是：求出 1 到 1000 之内能被 5 或 13 整除、但不能同时被 5 和 13 整除的所有整数，并将它们放在 a 所指的数组中，通过 n 返回这些数的个数。

注意：请勿改动主函数 main 和其他函数中的任何内容，仅在 fun 函数的花括号中填入所编写的若干语句。

试题程序：

```
#include<conio.h>
#include<stdio.h>
void  fun(int  *a, int  *n)
{



}
void  main()
{
    int  aa[1000], n, k;
    fun(aa, &n);
    for(k=0; k<n; k++)
        if((k+1)%10==0)
        {
            printf("%5d ", aa[k]);
            printf("\n ");
        }
        else
            printf("%5d ", aa[k]);
    printf("\n");
        printf("n=%d\n", n);
}
```

【主要考点】

求余运算符%的使用。

① 指针和数组的引用。

② 指针变量作函数形参，通过形参获得返回值。

【参考答案】

```
void  fun(int  *a, int  *n)
{
    int  i, j=0;
    for(i=1; i<=1000; i++)
        if((i%5==0||i%13==0)&&i%65!=0)
    a[j++]=i;
    *n=j;
}
```

【答案解析】

本题要求找出 1 到 1000 之内能被 5 或 13 整除但不能同时被 5 和 13 整除的所有整数。能同时被 5 和 13 整除的整数一定能被 65 整除，而不能被 65 整除的数不一定是能被 5 或 13 整除的数。所以可得出程序中的 if((i%5==0||i%13==0)&&i%65!=0)。(i%5==0||i%13==0)要作为一个整体被判断，按运算优先级可知，此时两边必须要有小括号。

fun()函数的形参 a 是指向实参数组 aa 的指针，形参 n 是指向实参 n 的指针。使用循环遍历指针 a 指向的数组 aa，可以使用指针加下标的形式访问数组元素，如果循环变量满足上面的条件，就把 i 存放在数组 a 中，最后使用*n 传回数组 a 中元素的个数。

实验内容

1. 程序填空题。

下列给定程序中，fun()函数的功能是：把形参 a 所指数组中的偶数按原顺序依次存放到 a[0]、a[1]、a[2]……中，把奇数从数组中删除，偶数的个数通过函数值返回。

例如，若 a 所指数组中数据的最初排 列为：9，1，4，2，3，6，5，8，7，删除奇数后，a 所指数组中的数据为：4，2，6，8，返回值为 4。

请在程序的下划线处填入正确的内容并把下划线删除，使程序得出正确的结果。

注意：不得增行或删行，也不得更改程序的结构。

```
#include<stdio.h>
#define  N  9
int  fun(int  a[], int  n)
{   int  i, j;
    j = 0;
    for(i=0; i<n; i++)
    /**********found**********/
      if(___①___== 0)
      {
      /**********found**********/
          ___②___ = a[i];      j++;
       }
    /**********found**********/
    return  ___③___;
```

```
}
void main()
{   int b[N]={9, 1, 4, 2, 3, 6, 5, 8, 7}, i, n;
    printf("\nThe original data   :\n");
    for(i=0; i<N; i++)
        printf("%4d ", b[i]);
    printf("\n");
    n = fun(b, N);
    printf("\nThe number of even   ：%d\n", n);
    printf("\nThe even   :\n");
    for(i=0; i<n; i++)
         printf("%4d ", b[i]);
    printf("\n");
}
```

2. 程序修改题。

在主函数中从键盘输入若干个数放入数组中，用 0 结束输入并把它放在最后一个元素中。下列给定程序中，fun 函数的功能是计算数组元素中值为负数的平均值(不包括 0)。

如：数组中元素的值依次为 43，-47，-21，53，-8，12，0，则程序的运行结果为-25.333333。

请改正程序中的错误，使它能得到正确结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构。

```
#include<stdio.h>
double fun(int x[])
{
    double sum=0.0;
    int c=0, i=0;
    /**********************found**********************/
    while(x[i]==0)
        {
          if(x[i]<0)
          {
              sum=sum+x[i];
              c++;
          }
        i++;
    }
    /**********************found**********************/
    sum=sum\c;
    return sum;
}
void main()
{
```

```
    int  x[1000];
    int  i=0;
    printf("\nPlease enter some data(end with 0) :");
    do
    {    scanf("%d", &x[i]);
    }while(x[i++]!=0);
    printf("%f\n", fun(x));
}
```

3. 程序设计题。

请编写 fun 函数，其功能是计算并输出下列多项式：

$$S=(1-1/2)+(1/3-1/4)+\cdots+(1/(2n-1)-1/2n)$$

的值。

例如，若主函数从键盘输入 n 的值为 8，则输出为：S=0.662872。

注意：要求 n 的值大于 1 但不大于 100。请勿改动主函数中的任何内容，仅在 fun 函数的花括号中填入所编写的若干语句。

```
#include<stdio.h>
double  fun(int  n)
{


}
void  main()
{   int  n; double  s;
    printf("\nInput  n:   ");   scanf("%d", &n);
    s=fun(n);
    printf("\ns=%f\n", s);
}
```

【参考答案】

1. 程序填空题

① a[i]%2　　② a[j]　　③ j

2. 程序修改题

① 错误 1：while(x[i]==0)　　正确：while(x[i]!=0)

② 错误 2：sum=sum\c;　　正确：sum=sum/c;

3. 程序设计题

```
double  fun(int  n)
{
    int  i;
    double  s=0;
    for(i=1; i<=n; i++)
        s=s+(1.0/(2*i-1)-1.0/(2*i));
    return  s;
}
```

实验总结与分析

第二部分　课堂练习及答案

第 2 章　数据类型和表达式

【练习2.1】已知半径，求一个圆的面积，应如何书写相关变量的定义语句？

【答案】float radius, area; 或者 double radius, area; 或者 float radius; double area;

【练习2.2】分析以下程序段是否有错误，可编译后查看编译结果，若有错误，查看对应的错误类型和原因。

```
#include<stdio.h>
int main()
{
    printf("请输出 n 的值：");
    int n;
    n=5;
    printf("n=%d", n);
    return 0;
}
```

【答案】错误在于在VC ++ 6.0编译环境下，函数中变量的定义必须放在函数体最开头的几行中，不能放在其他语句之后。所以，“int n;”应该是函数体内的第一条语句。

【练习2.3】运行以下程序段，验证程序的输出结果是否与a，b的初值相同。

```
#include<stdio.h>
int main()
{
    float a;
    double b;
    a=1234567.89;
    b=1234567890123.45678;
    printf("%f,%f\n", a, b);
    return 0;
}
```

【答案】1234567.875000,1234567890123.456800

结果与初值不同，这是因为浮点数在系统内部不能精确表示。

【练习 2.4】① 20、020、0x20 是否代表同一数值，如果不是，它们的数值分别为多少？

② 8、010、0x8 是否代表同一数值，如果不是，它们的数值分别为多少？

③ C 语言中是否有二进制常量？

【答案】① 它们不是同一数值，分别代表十进制下的 20、16、32。

② 它们代表同一数值，均是十进制下的 8。

③ C 语言中没有二进制常量，只有十进制、八进制和十六进制常量。

【练习 2.5】① 判断下述字符常量"a"、'ch'、"\n"、"\0x23"、'\025'、'\223'、'x41'的合法性。

② 如下程序段的输出结果为_____。

```
#include<stdio.h>
int  main()
{
    int  i;
    char  c='2';
    i=c+2;
    printf("%d %d", c, i);
    return  0;
}
```

【答案】① '\025'、'\223'为合法字符，其余为非法字符。

② 50 52

【练习 2.6】分析以下程序段输出结果。两次调用 putchar()函数的输出结果相同吗？

```
char  c='d';
putchar(c);
putchar('c');
```

【答案】两次输出结果不同。第一次输出变量c的值：d，第二次输出的是字符常量：c。

【练习 2.7】已知英文字母 'a'的 ASCII 码为 97(十进制)，则以下程序段的输出结果为：_____。

```
char  c='0', b='j', a='a';
printf("%d, %c, %x, %X, %o", c, c, b, b, a);
```

【答案】48, 0, 6a, 6A, 141

【练习2.8】有如下程序：

```
#include<stdio.h>
int  main()
{
    int  m, n, p;
    scanf("m=%d,n=%d,p=%d", &m, &n, &p);
    printf("%d%d%d\n", m, n, p);
    return  0;
}
```

若想从键盘上输入数据，使变量 m 中的值为 123，n 中的值为 456，p 中的值为 789，则

正确的输入是_____。

A. m=123n=456p=789　　B. m=123　n=456　p=789

C. m=123,n=456,p=789　　D. 123　456　789

【答案】C

【练习 2.9】① 执行程序段“int i=3;　printf("%d", i--);”输出结果为_____。

A. 3　　B. 4

② 以下不能正确表达 $\frac{2ab}{cd}$ 的C语言表达式的是_____。

A. 2*a*b/c/d　　B. a*b/c/d*2　　C. a/c/d*b*2　　D. 2*a*b/c*d

【答案】① A　　② D

【练习2.10】① 已有定义“char c=' ';　int a=1, b;”(此处c的初值为空格字符)，执行语句“b=!c&&a;”后b的值为_____。

② 执行以下程序段后，w的值为_____，x的值为_____。

```
int w='A', x=10, y=15;
x=((x&&y)||(w='B'));
```

【答案】① 0　　② w的值为'A'，x的值为1D。

【练习2.11】① 若变量均已正确定义并赋值，以下语句中，合法的C语言赋值语句是_____。

A. x=y= =5;　　B. x=n%2.5;　　C. x+n=i;　　D. x=5=4+1;

② 设变量a和b已正确定义并赋初值。与a-=a+b等价的赋值表达式为_____。

【答案】① A　　② a= a-(a+b)或 a=-b。

【练习2.12】运行以下程序后的输出结果是_____。

```
#include <stdio.h>
int main()
{
    int x, a=1, b=2, c=3, d=4;
    x=(a<b)? a:b;    x=(x<c)? x:c;    x=(d>x)?x:d;
    printf("%d\n", x);
    return 0;
}
```

【答案】1

【练习2.13】以下程序段的输出结果为_____。

```
#include<stdio.h>
int main()
{
    int x;
    printf("%d", (x=5, 2*x, 3*x));
    return 0;
}
```

【答案】15

【练习 2.14】设有定义“double x=123.4567;”则执行以下语句后的输出结果是_____。

```
printf("%f\n", (int)(x*100+0.5)/100.0);
```

【答案】123.460000

第 3 章　程序控制结构

【练习 3.1】从键盘输入两个整数，然后计算这两个数的和，最后输出结果。运行结果如下：

```
请输入两个整数：5 9
5+9=14
```

【答案】参考代码：

```
#include<stdio.h>
int main()
{
    //1. 定义变量
    int a, b, sum;

    //2. 输入两个整数
    printf("请输入两个整数：");
    scanf("%d%d", &a, &b);

    //3. 求和
    sum=a+b;

    //4. 输出和 sum
    printf("%d+%d=%d\n", a, b, sum);
    return 0;
}
```

【练习 3.2】输入一个整数，用单分支结构求其绝对值，并输出。

【答案】参考代码：

```
#include<stdio.h>
int main()
{
    //1. 定义变量
    int a;

    //2. 输入 a
    printf("请输入一个整数：");
    scanf("%d", &a);
```

```
    //3. 求 a 的绝对值，并赋给 a
    if(a<0)
        a=-a;

    //4. 输出绝对值
    printf("a 的绝对值为%d\n", a);
    return 0;
}
```

【练习 3.3】输入一个整数，判断并输出它是偶数还是奇数。

【答案】参考代码：

```
#include<stdio.h>
int main()
{
    //1. 定义变量
    int a;

    //2. 输入
    printf("请输入一个整数：");
    scanf("%d", &a);

    //3. 判断并输出
    if(a%2==0)                    //判断是否满足偶数的条件
        printf("%d 是偶数。\n", a);
    else
        printf("%d 是奇数。\n", a);
    return 0;
}
```

【练习 3.4】一个少年乘坐公交车，如果身高超过了 1.4m，则应按成人投票 1 元，否则投 0.5 元。现在输入身高，输出应投票的金额。

【答案】参考代码：

```
#include<stdio.h>
int main()
{
    //1. 定义变量
    double tall, ticket;

    //2. 输入身高
    printf("请输入你的身高：");
    scanf("%lf", &tall);

    //3. 计算票价
    if(tall>1.4)
        ticket=1;
    else
```

```
        ticket=0.5;
    //4. 输出票价
    printf("你应付票价%.2f", ticket);
    return 0;
}
```

【练习 3.5】执行下面的程序后，变量 y 的值是_____。

A. 1000　　B. 1200　　C. 1400　　D. 3400

```
#include<stdio.h>
void main()
{
    int x, y;
    x=7;
    if(x<6)
        y=1000;
    else
        if(x<=18)
            y=(x-6)*200+1000;
        else
            y=(x-18)*300+3400;
    printf("y=%d.\n", y);
}
```

【答案】B

【练习 3.6】某公司销售人员的工资算法如下表所示。销售的产品单价为 1000 元/件，若每月销售 5 件以下(不含 5)，没有提成；5 件以上 10 件以下(含 5，不含 10)，该部分每件提成 4%；若每月销售 10 件以上 15 件以下(含 10，不含 15)，该部分每件提成 6%；每月销售 15 件以上(含 15)，该部分每件提成 10%。编制程序完成以下要求：输入销售件数，输出销售人员的工资。

销售提成计算表

销售件数 x	底薪工资(元)	提成比例
[1,5)	2000	0%
[5,10)	2000	4%
[10,15)	2200	6%
15 以上(含 15)	2500	10%

【答案】参考代码：

```
#include<stdio.h>
int main()
{
    int num, salary;
    //1. 输入
    printf("请输入销售数量：");
```

```
    scanf("%d", &num);

    //2. 计算提成
    if(num>=15)
        salary=2500+1000*5*(0.04+0.06)+(num-14)*1000*0.1;
    else if(num>=10)
            salary=2200+5*1000*0.04+(num-9)*1000*0.06;
    else if(num>=5)
            salary=2000+(num-4)*1000*0.04;
    else
            salary=2000;

    //3. 输出
    printf("salary=%d\n", salary);
    return 0;
}
```

【练习 3.7】某市出租车的收费情况只有两种。一种是起步价为 7 元，车程 3 公里；3 公里以上每公里 1.2 元；6 公里以上要加收 50%的回空费，即每公里 1.8 元。另一种是排气量在 1.8 升以上的豪华型轿车，起步价 10 元，车程也是 3 公里；3 公里以上每公里 1.8；6 公里以上加收 50%的回空费，即每公里 2.7 元。低速行驶及等待，每 5 分钟按照 1 公里计费。编制程序完成以下要求，输入乘坐的车型、行驶里程数和低速行驶时长，计算并输出应付的出租车费(四舍五入到元)。

【答案】参考代码：

```
#include<stdio.h>
int main()
{
    //1. 定义变量
    char p;
    int l, t;
    double m;

    //2. 输入车型 p，里程 l 和低速行驶时间 t
    printf("请输入车型 P(A 或 a 代表低级，B 或 b 代表高级)");
    scanf("%c", &p);
    printf("请输入行驶里程和低速行驶时间(都为整数)");
    scanf("%d%d", &l, &t);

    //3. 计算费用
    if(p=='A' || p=='a')
    {
        if((l+t/5)<=3)
            m=7;
        else
```

```
            if((l+t/5)<=6)
                m=(l+t/5-3)*1.2+7;
            else
                m=(l+t/5-6)*1.8+10.6;
        }
        else
            if(p=='B' || p=='b')
                if((l+t/5)<=3)
                    m=10;
                else
                    if((l+t/5)<=6)
                        m=(l+t/5-3)*1.8+10;
                    else
                        m=(l+t/5-6)*2.7+15.4;
        m=(int)(m*10+5)/10;                    //四舍五入到元

        //4. 输出费用 m
        printf("您乘坐%c 型车,行驶里程%d 公里，其中低速行驶时间%d 分钟，合计费用为
            %.2f 元。\n", p, l, t, m);
        return 0;
    }
```

【练习 3.8】用 switch 结构编制程序，输入 1 个学生的 C 语言课程考试成绩(百分制整型数)，输出成绩相应等级。

A 级：90～100　B 级：80～89　C 级：70～79　D 级：60～69　E 级：<60

【答案】参考代码：

```
#include<stdio.h>
int main()
{
    int grade;
    char level;

    printf("请输入一个整数(0-100):");
    scanf("%d", &grade);

    if(grade>=0 && grade<=100)
    {
        switch(grade/10)          //两整数相除结果为整数
        {
          case 10:
          case 9: level='A';   break;
          case 8: level='B';   break;
          case 7: level='C';   break;
          case 6: level='D';   break;
```

```
            case  5:
            case  4:
            case  3:
            case  2:
            case  1:
            case  0:   level='E';   break;
            default:   printf("判断错误。\n");
        }
        printf("Level %c.\n", level);
    }
    else
        printf("输入错误。\n");
    return  0;
}
```

【练习 3.9】输入年月，输出该月有几天。

运行实例：

```
请输入年月：2010-3
2010 年 3 月有 31 天。
```

【答案】参考代码：

```
#include<stdio.h>
int  main()
{
    int  year, month;
    int  days=0;        //天数

    printf("请输入年-月：");
    scanf("%d-%d", &year, &month);
    switch(month)
    {
      case  1:
      case  3:
      case  5:
      case  7:
      case  8:
      case  10:
      case  12:
            days=31;
            break;
      case  4:
      case  6:
      case  9:
```

```
        case  11:
              days=30;
              break;
        case  2:
              if(year%4==0 && year%100!=0 || year%400==0)
                 days=29;
              else
                 days=28;
              break;
        default: printf("Input  Month  Error!\n");
    }
    if(days)
        printf("%d 年%d 月有%d 天。\n", year, month, days);
    return  0;
}
```

【练习 3.10】用 switch 语句编制程序，输入 10 个学生的 C 语言成绩(百分制整型数)，统计各等级的人数。

A 级：90～100　B 级：80～89　C 级：70～79　D 级：60～69　E 级：<60

【答案】参考代码：

```
#include<stdio.h>
int  main()
{
    int  grade;
    int  A, B, C, D, E;
    A=B=C=D=E=0;

    printf("请输入第 1 个学生成绩(0-100):");
    scanf("%d", &grade);

    if(grade>=0 && grade<=100)
    {
        switch(grade/10)
        {
          case  10:
          case  9:   A++;   break;
          case  8:   B++;   break;
          case  7:   C++;   break;
          case  6:   D++;   break;
          case  5:
          case  4:
          case  3:
          case  2:
```

```
            case  1:
            case  0:    E++;    break;
            default:    printf("判断错误。\n");
        }
}
else
        printf("输入错误。\n");

printf("请输入第 2 个学生成绩(0-100):");
scanf("%d", &grade);

if(grade>=0 && grade<=100)
{
        switch(grade/10)
        {
            case  10:
            case  9:    A++;    break;
            case  8:    B++;    break;
            case  7:    C++;    break;
            case  6:    D++;    break;
            case  5:
            case  4:
            case  3:
            case  2:
            case  1:
            case  0:    E++;    break;
            default:    printf("判断错误。\n");
        }
}
else
        printf("输入错误。\n");
//……该处省略第 3 到第 9 个学生的成绩输入与等级判断……

printf("请输入第 10 个学生成绩(0-100):");
scanf("%d", &grade);

if(grade>=0 && grade<=100)
{
        switch(grade/10)
        {
            case  10:
            case  9:    A++;    break;
            case  8:    B++;    break;
```

```
            case 7:   C++;   break;
            case 6:   D++;   break;
            case 5:
            case 4:
            case 3:
            case 2:
            case 1:
            case 0:   E++;   break;
            default:  printf("判断错误。\n");
        }
    }
    else
        printf("输入错误。\n");

    printf("A 等级有%d 个。\n", A);
    printf("B 等级有%d 个。\n", B);
    printf("C 等级有%d 个。\n", C);
    printf("D 等级有%d 个。\n", D);
    printf("E 等级有%d 个。\n", E);

    return 0;
}
```

【练习 3.11】用 for 循环结构编程求 1+3+5+…+97+99 的值(可以采取两种方法完成)。

【答案】参考代码:

```
//方法 1
#include<stdio.h>
int main()
{
    int i, sum=0;

    for(i=1; i<100; i+=2)
        sum+=i;
    printf("%d", sum);
    return 0;
}

//方法 2
#include<stdio.h>
int main()
{
    int i, sum=0;

    for(i=1; i<100; i++)
        if(i%2==1 )
```

```
            sum+=i;
        printf("%d", sum);
        return 0;
    }
```

【练习 3.12】用 while 循环结构编程求 1+3+5+…+97+99 的值。

【答案】参考代码：

```
#include<stdio.h>
int main()
{
    int i, sum=0;
    i=1;
    while(i<=99)
    {
        sum=sum+i;
        i+=2;
    }
    printf("%d.\n", sum);
    return 0;
}
```

【练习 3.13】用 while 循环结构编程求$1+\frac{1}{3}+\frac{1}{5}+\cdots+\frac{1}{97}+\frac{1}{99}$的值。

【答案】参考代码：

```
#include<stdio.h>
int main()
{
    int i;
    double sum=0.0;
    i=1;
    while(i<=99)
    {
        sum=sum+1.0/i;
        i+=2;
    }
    printf("%f.\n", sum);
    return 0;
}
```

【练习 3.14】根据公式$\frac{\pi}{4}=1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\frac{1}{11}+\cdots$，求 π 的近似值，要求最后一项的绝对值小于 1e−7。

【答案】参考代码：

```
#include<stdio.h>
```

```
int  main()
{
    int  i, k;
    double  pi, sum=0.0;
    i=1;    k=1;
    while(1.0/i>1e-7)
    {
        sum=sum+k*1.0/i;
        i+=2;
        k=-1*k;
    }
    pi=4*sum;
    printf("pi=%f.\n", pi);
    return  0;
}
```

【练习 3.15】用 do-while 结构，计算表达式 1+3+5+7+9+…+97+99 的值。

【答案】参考代码：

```
#include<stdio.h>
int  main()
{
    int  i, sum=0;
    i=1;
    do
    {
        sum=sum+i;
        i+=2;
    }while(i<=99);
    printf("%d.\n", sum);
    return  0;
}
```

【练习 3.16】根据公式 $\frac{\pi}{4}=1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\frac{1}{11}+\cdots$，用 do-while 结构求 π 的近似值，要求直到最后一项的绝对值不小于 1e-7。

【答案】参考代码：

```
#include<stdio.h>
int  main()
{
    int  k, i;
    double  pi, sum=0.0;
    i=1;   k=1;
    do
```

```
        {
            sum=sum+k*1.0/i;
            i+=2;
            k=-1*k;
        }while(1.0/i>1e-7);
        pi=4*sum;
        printf("pi=%f.\n", pi);
        return 0;
    }
```

【练习 3.17】根据 s=1+2+4+7+11+…+x，(x<100)，计算 s 的值。

【答案】参考代码：

```
#include<stdio.h>
int main()
{
    int i, j;
    long s;
    for(i=1, j=0, s=0; i<100; i=i+j)          //注意步长 j 的变化
    {
        s=s+i;
        printf("%d+", i);
        j++;
    }
    printf("\b");                              //退格去掉最后的“+”
    printf("=%ld", s);
    return 0;
}
```

【练习 3.18】输入整数 n，输出 n 行星，第 1 行 1 个，第 2 行 3 个，第 3 行 5 个……例如输入正整数 3。

输出如下星型图：

```
    *
   * * *
  * * * * *
```

【答案】参考代码：

```
#include<stdio.h>
int main()
{   //1. 定义变量
    int i, j, n;

    //2. 输入 n
    printf("请输入一个正整数 n:");
    scanf("%d", &n);
```

```
    //3. 输出由星号组成的三角形
    for(i=1; i<=n; i++)
    {    //3.1  输出第 i 行的前导空格
        for(j=1; j<=2*(n-i); j++)
            printf(" ");
        //3.2  输出第 i 行的星号
        for(j=1; j<=2*i-1; j++)
            printf("* ");
        //3.3  输出换行符号
        printf("\n");
    }
    return 0;
}
```

【练习 3.19】编制程序，求 1!+2!+3!+…+n!的值，n 为键盘输入的正整数。

【答案】参考代码：

```
#include<stdio.h>
void main()
{
    int i, j, n;
    long fac=1, sum=0;
    printf("请输入一个正整数 n：");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
    {
        fac=1;
        for(j=1; j<=i; j++)
        {
            fac=fac*j;
        }
        sum=sum+fac;
    }
    printf("1!+2!+…+%d!=%d", n, sum);
}
```

【练习 3.20】输入一个正整数 n，输出紧靠 n 前的能够被 3 或 5 整除的整数。

【答案】参考代码：

```
#include<stdio.h>
int main()
{
    int i, n;
    printf("请输入一个正整数 n:");
```

```
        scanf("%d", &n);
        for(i=n-1; i>=3; i--)
            if(i%3==0 || i%5==0)
            {
                printf("%d 能够被 3 或 5 整除.\n", i);
                break;
            }
        if(i<3)
            printf("%d 前，没有能够被 3 或 5 整除的整数.\n", n);
        return  0;
    }
```

【练习 3.21】下面程序执行后，i 的值是_____。

```
#include <stdio.h>
void  main()
{
    int  i;
    for(i=0; i<10; i++)
        if(i%3!=0)
            printf("%d ", i);
        else
            break;
    printf("\ni=%d", i);
}
```

【答案】0

【练习 3.22】执行下列程序后 i 的值为_____。

```
#include<stdio.h>
void  main()
{
    int  i;
    for(i=0; i<10; i++)
    {   if(i%3==0)
            continue;
        i=i+3;
    }
    printf("%d\n", i);
}
```

【答案】10

【练习 3.23】执行下列程序后得到的情况是_____。

```
#include<stdio.h>
void  main(void)
{
    int  i=0;
    while(i<10)
    {
        if(i%3==0)
            continue;
        i+=3;
    }
    printf("%d\n", i);
}
```

【答案】死循环。

第4章　数　　组

【练习 4.1】输入一个正整数 n(1<n≤10)，再输入 n 个整数存入一维数组，先输出最大值及其下标(设最大值唯一)，再将最大值与最后一个数交换，并输出交换后的 n 个数。

【答案】参考代码：

```
#include<stdio.h>
#define  N  10          //定义符号常量 N
int  main()
{
    int  a[N], i, index, temp, n;
    printf("input  n(1-10):");
    scanf("%d", &n);

    //1. 输入 n 个数，并存到数组 a 中
    printf("please  input  %d  numbers\n", n);
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    index=0; //index 用来存放最大值的下标，初值是第一个元素下标 0

    //2. 在 a[0]~a[n-1]中找最大值的下标放到 index 中
    for(i=1; i<n; i++)
        if(a[i]>a[index])
            index=i;
```

```
        //3. 将找到的最大值与最后一个数 a[n-1]交换
        temp=a[index];
        a[index]=a[n-1];
        a[n-1]=temp;

        //4. 输出交换后的结果
        printf("the array after chang:\n");
        for(i=0; i<n; i++)
            printf("%3d", a[i]);
        printf("\n");

        return 0;
    }
```

【练习4.2】若要定义一个具有5个元素的整型数组，以下定义语句中错误的是_____。

A. int a[5]={0};　　B. int b[]={0, 0, 0, 0, 0};

C. int c[2+3];　　D. int i=5, d[i];

【答案】D

【练习4.3】若有定义语句“int a[]={5, 4, 3, 2, 1}, i=4;”，则下面对a数组元素的引用中错误的是_____。

A. a[--i]　　B. a[2*2]　　C. a[a[0]]　　D. a[a[i]]

【答案】C

【练习4.4】有以下程序：

```
#include<stdio.h>
void main()
{
    int a[5]={1, 2, 3, 4, 5}, b[5]={0,2,1,3,0},i,s=0;
    for(i=1; i<5; i++)
        s=s+a[b[i]];
    printf("%d\n", s);
}
```

运行程序后的输出结果是_____。

A. 6　　B. 10　　C. 11　　D. 15

【答案】B

【练习4.5】有以下程序：

```
#include<stdio.h>
int main()
{   int s[12]={1, 2, 3, 4, 4, 3, 2, 1, 1, 1, 2, 3}, c[5]={0}, i;
    for(i=0; i<12; i++)
        c[s[i]]++;
    for(i=1; i<5; i++)
        printf("%d", c[i]);
    printf("\n");
```

```
    return 0;
}
```

运行程序后的输出结果是_____。

A. 4332　　B. 4321　　C. 1234　　D. 2334

【答案】A

【练习 4.6】编程解决：输入一个正整数 n(1<n≤10)，再输入 n 个浮点数存入一维数组，然后输出该数组中的最小值及其下标。

【答案】参考代码：

```
#include<stdio.h>
void main()
{
    int i, n, index;
    double a[10];
    printf("输入 n:");
    scanf("%d", &n);
        printf("输入%d 个浮点数:", n);
    for(i=0; i<n; i++)
        scanf("%lf", &a[i]);
    index=0;
    for(i=1; i<n; i++)
        if(a[i]<a[index])
            index=i;
    printf("最小值：%f，下标：%d\n", a[index], index);
}
```

【练习4.7】以下各项中不能正确将字符串存入一维字符数组的是____。

A. char s[10]="abcdefg";　　B. char t[]="abcdefg";

C. char s[10]; s="abcdefg";　　D. char s[10]={'a', 'b', 'c' ,'d', 'e', 'f', 'g'};

【答案】C

【练习 4.8】有以下程序：

```
#include<stdio.h>
int main()
{
    char s[]="\n12ab\\";
    printf("%d, %d\n", strlen(s), sizeof(s));
    return 0;
}
```

运行程序后的结果是_____。

A. 赋初值的字符串有错　　B. 输出 6, 7

C. 输出 5, 6　　D. 输出 6, 6

【答案】B

【练习 4.9】输入一个以回车结束的字符串(少于 80 个字符)，统计字符串中数字字符的个数。

【答案】参考代码：

```
#include<stdio.h>
int  main(void)
{
    int  count, i;
    char  str[80];
    printf("Enter  a  string: ");
    /*输入字符串*/
    i = 0;
    while((str[i] = getchar()) != '\n')
        i++;
    str[i] = '\0';     /*  输入结束符转换为字符串结束符  */
    count = 0;
    for(i = 0; str[i] != '\0'; i++)
        if(str[i] <= '9' && str[i] >= '0')
            count++;
    printf("count = %d\n", count);
    /*  输出字符串  */
    for(i = 0; str[i] != '\0'; i++)
        putchar(str[i]);
    return  0;
}
```

【练习4.10】以下语句中，能正确定义数组并正确赋初值的语句是_____。

A. int a[2][]={{1,2}, {3,4}};　　B. int b[1][2]={{1}, {3}};

C. int N=5, c[N][N];　　D. int d[3][2]={{1,2}, {3,4}};

【答案】D

【练习4.11】有以下程序：

```
#include<stdio.h>
void  main()
{
    int  m[][3]={1, 4, 7, 2, 5, 8, 3, 6, 9};
    int  i, j, k=2;
    for(i=0; i<3; i++)
        printf("%d ", m[k][i]);
}
```

运行程序后的输出结果是_____。

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

【答案】C

【练习4.12】输入一个正整数n(1<n≤6)，根据下式生成一个n*n的方阵，然后将该方阵转置(行号列号互换)后输出。a[i][j] = i * n + j + 1(0≤i≤n-1，0≤j≤n-1)。

【答案】参考代码：

```
#include<stdio.h>
int  main(void)
{
    int  i, j, n, temp;
    int  a[6][6];
    printf("Enter n(1<n≤6): ");     scanf("%d", &n);
    /* 1. 给二维数组赋值*/
    for(i = 0; i < n; i++)
        for(j = 0; j < n; j++)
            a[i][j] = i * n + j + 1;
    printf("原矩阵:\n");
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < n; j++)
            printf("%d ", a[i][j]);
        printf("\n");
    }
    /* 2. 行列互换*/
    for(i = 0; i < n; i++)
        for(j = 0; j < n; j++)
            if(i <= j)          /* 只遍历上三角阵 */
            {   temp = a[i][j];
                a[i][j] = a[j][i];
                a[j][i] = temp;
            }
    /* 3. 按矩阵的形式输出 a */
    printf("转置后矩阵:\n");
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < n; j++)
            printf("%d ", a[i][j]);
        printf("\n");
    }
    return 0;
}
```

第5章 函 数

【练习5.1】有如下函数定义：

```
void fun(int n, double x) {……}
```

若以下选项中的变量都已正确定义并赋值，则正确调用fun函数的语句是_____。

A. fun(int y, double m);　　B. k=fun(10, 12.5);

C. fun(x, n);　　D. void fun(n, x);

【答案】C

【练习5.2】有以下程序：

```
#include<stdio.h>
fun(int a, int b)
{
    if(a>b)
        return(a);
    else
        return(b);
}
void main()
{
    int x=3, y=8, z=6, r;
    r=fun(fun(x, y), 2*z);
    printf("%d", r);
}
```

运行程序后的输出结果是_____。

A. 3　　B. 6　　C. 8　　D. 12

【答案】D

【练习 5.3】有以下程序：

```
#include<stdio.h>
void f(int x, int y)
{
    int t;
    if(x<y){ t=x;    x=y;    y=t; }
}
void main()
{
    int a=4, b=3, c=5;
    f(a, b);  f(a, c);  f(b, c);
    printf("%d, %d, %d\n", a, b, c);
```

```
}
```

运行程序后的输出结果是_____。

A. 3, 4, 5　　B. 5, 3, 4　　C. 5, 4, 3　　D. 4, 3, 5

【答案】D

【练习 5.4】输入两个正整数，输出其中较大的数。要求通过定义和调用 max(x, y)函数求 x 和 y 中较大的数。

【答案】参考代码：

```
#include<stdio.h>
void  main()
{
    int  max(int  x, int  y);
    int  x, y, z;
    printf("input  two  numbers:\n");
    scanf("%d%d", &x, &y);
    z=max(x, y);
    printf("max=%d\n", z);
}
int  max(int  x, int  y)
{
    if(x>y)
        return  x;
    else
        return  y;
}
```

【练习 5.5】输入两个正整数 m 和 n(m≥1，n≤300，m<n)，求 m 和 n 之间所有素数的和。要求通过定义和调用 prime(p)函数判断 p 是否为素数，当 p 为素数时返回 1，否则返回 0。

【答案】参考代码：

```
#include<stdio.h>
#include<math.h>                    // 需要调用平方根函数
int  main(void)
{
    int  m, n, i, sum=0;
    int  prime(int  p);             // 函数声明
    printf("input  m:");
    scanf("%d", &m);
    printf("input  n:");
    scanf("%d", &n);
    for(i = m; i <= n; i++)
      {
        if(prime(i))                // 调用 prime(i)判断 i 是否为素数
            sum=sum+i;
```

```
        }
    printf("sum=%d\n", sum);
}
// 定义判断素数的函数，如果 p 是素数则返回 1(真)；否则返回 0(假)
int prime(int p)
{
    int i, n;
    if(p == 1)
        return 0;                    // 1 不是素数，返回 0
    n = sqrt(p);
    for( i = 2; i <= n; i++)
        if (p % i == 0){             // 如果 p 不是素数
            return 0;                // 返回 0
        }
    return 1;                        // p 是素数，返回 1
}
```

【练习5.6】有以下程序：

```
#include<stdio.h>
int m=13;
int fun2(int x, int y)
{
    int m=3;
    return(x*y-m);
}
void main( )
{
    int a=7, b=5;
    printf("%d\n", fun2(a, b)/m);
}
```

运行程序后的输出结果是_____。

A. 1　　B. 2　　C. 7　　D. 10

【答案】B

【练习5.7】有以下程序：

```
#include<stdio.h>
int fun()
{
    static int x=1;
    x*=2;
    return x;
}
```

```
void main()
{
    int i, s=1;
    for(i=1; i<=2; i++)
        s=fun();
    printf("%d", s);
}
```

运行程序后的输出结果是_____。

A. 0　　B. 1　　C. 4　　D. 8

【答案】C

【练习 5.8】下列程序的输出结果是_____。

```
#include<stdio.h>
long fib(int g)
{
    switch(g)
    {
        case 0: return(0);
        case 1:
        case 2: return(2);
    }
    printf("g=%d, ", g);
    return(fib(g-1) + fib(g-2));
}
void main()
{
    long k;
    k = fib(4);
    printf("k=%ld\n", k);
}
```

【答案】g=4,g=3,k=6

【练习 5.9】已知一个数列从第 1 项开始的前 3 项为 0、0、1，以后的各项都是其相邻的前 3 项之和。要求使用递归方法编写函数 fun(n)，求该数列第 n 项的值，返回值为长整型，并写出相应的主函数。

【答案】参考代码：

```
#include<stdio.h>
int main(void)
{
    int n;
    long fun(int n);
    printf("Input n:");
    scanf("%d", &n);
```

```
    printf("The result is %ld\n", fun(n));
    return 0;
}
long fun(int n)
{
    if(n==1||n==2)
        return 0;
    else if(n==3)
        return 1;
    else
        return fun(n-3)+fun(n-2)+fun(n-1);
}
```

【练习 5.10】输入一个正整数 n(n≤10)，编写 fun(a, n)函数，其功能是计算 n 门课程的平均分，计算结果作为函数的返回值。正整数 n 和 n 门课程的成绩在主函数中输入。

【答案】参考代码：

```
#include<stdio.h>
int main(void)
{
    int i, n;
    double a[10];
    double fun(double a[], int n);                    //函数声明
    printf("Input n:");
    scanf("%d", &n);
    printf("Input %d scores:",n);
    for(i=0; i<n; i++)
        scanf("%lf", &a[i]);
    printf("Average score is:%.2f\n", fun(a, n));     //输出函数调用的结果
    return 0;
}
double fun(double a[], int n)                         //定义求平均值的函数
{
    int i;
    double sum;
    sum=0;
    for(i=0; i<n; i++)
        sum=sum+a[i];
    return sum/n;
}
```

【练习5.11】有以下程序：

```
#include<stdio.h>
int f(int a)
```

```
{
    return a%2;
}
void main()
{
    int s[8]={1, 3, 5, 2, 4, 6}, i, d=0;
    for(i=0; f(s[i]); i++)    d+=s[i];
    printf("%d", d);
}
```

运行程序后的输出结果是_____。

A. 9　　B. 11　　C. 19　　D. 21

【答案】C

【练习5.12】有以下程序：

```
#include<stdio.h>
void fun(int a[], int n)
{
    int i, t;
    for(i=0; i<n/2; i++)
    {t=a[i];  a[i]=a[n-1-i];  a[n-1-i]=t;}
}
void main()
{
    int k[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, i;
    fun(k, 5);
    for(i=2; i<8; i++)
        printf("%d", k[i]);
    printf("\n");
}
```

运行程序后的输出结果是_____。

A. 345678　　B. 876543　　C. 1098765　　D. 321678

【答案】D

【练习5.13】有以下程序：

```
#include<stdio.h>
int f(int b[][4])
{
    int i, j, s=0;
    for(j=0; j<4; j++)
    {  i=j;
       if(i>2)  i=3-j;
       s+=b[i][j];
    }
```

```
    return s;
}
void main()
{
    int a[4][4]={{1, 2, 3, 4},{0, 2, 4, 6},{3, 6, 9, 12},{3, 2, 1, 0}};
    printf("%d", f(a));
}
```

运行程序后的输出结果是_____。

A. 12　　B. 11　　C. 18　　D. 16

【答案】D

第6章　指　　针

【练习 6.1】① 有以下程序：

```
void main()
{
    int a=7, b=8, *p, *q, *r;
    p=&a;  q=&b;
    r=q;   q=p;  p=r;
    printf("%d, %d, %d, %d", *p, *q, a, b);
}
```

运行程序后的输出结果是_____。

A. 8, 7, 8, 7　　B. 7, 8, 7, 8　　C. 8, 7, 7, 8　　D. 7, 8, 8, 7

② 若有“int n=7, *p=&n, *q=p;”，以下语句中非法的赋值语句是_____。

A. p=q;　　B. *p=*q;　　C. n=*q;　　D. p=n;

③ 若有“int i=0, j=1, *p=&i, *q=&j;” 以下语句中非法的赋值语句是_____。

A. i=*&j;　　B. p=&*&i;　　C. j=*p;　　D. i=*&q;

④ 有以下程序：

```
void fun(char *a, char *b)
{
    a=b;
    (*a)++;
}
void main()
{
    char c1='A', c2='a', *p1, *p2;
    p1=&c1;
    p2=&c2;
```

```
    fun(p1, p2);
    printf("%c%c", c1, c2);
}
```

运行程序后的输出结果是_____。

A. Ab　　B. aa　　C. Aa　　D. Bb

【答案】① C　② D　③ D　④ A

【练习6.2】有以下程序，运行程序后的输出结果是_____。

```
void swap(int x, int y)
{
    int t;
    t=x;    x=y;    y=t;
    printf("%d %d", x, y);
}
int main()
{
    int a=3, b=4;
    swap(a, b);
    printf("%d %d", a, b);
    return 0;
}
```

【答案】4 3 3 4

【练习 6.3】以下程序运行后的输出结果是_____。

```
void swap(int *x, int *y)
{
    int t;
    t= *x;    *x = *y;    *y =t;
    printf("%d %d ", *x, *y);
}
int main()
{
    int a=3, b=4;
    swap(&a, &b);
    printf("%d %d", a, b);
    return 0;
}
```

【答案】4 3 4 3

【练习 6.4】以下程序运行后的输出结果是_____。

```
void byvalue(int x, int y, int z)
{   x=0;
    y=0;
    z=0;
```

```
}
void byref(int *x, int *y, int *z)
{
    *x=0;
    *y=0;
    *z=0;
}

int main()
{
    int a=3, b=4, c=5;
    printf("\n Befor calling byvalue(), ");
    printf("a=%d, b= %d, c=%d", a, b, c);

    byvalue(a, b, c);                    //函数调用
    printf("\n After calling byvalue(), ");
    printf("a=%d, b= %d, c=%d ", a, b, c);

    byref(&a, &b, &c);                   //函数调用
    printf("\n After calling byref(), ");
    printf("a=%d, b= %d, c=%d ", a, b, c);
    return 0;
}
```

【答案】

Befor calling byvalue(), a=3, b=4, c=5
After calling byvalue(), a=3, b=4, c=5
After calling byref(), a=0, b=0, c=0

【练习 6.5】① 若有定义语句：

```
double x[5]={1.0, 2.0, 3.0, 4.0, 5.0}, *p=x;
```

则下述各项中，错误引用 x 数组元素的是_____。

A. *p　　B. x[5]　　C. *(p+1)　　D. *x

② 有以下程序：

```
void main()
{   int a[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, *p=&a[3], *q=p+2;
    printf("%d\n", *p+*q);
}
```

运行程序后的输出结果是_____。

A. 16　　B. 10　　C. 8　　D. 6

③ 若有以下定义及语句：

```
int a[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, *p=&a[3], b;
b=p[5];
```

b 中的值是_____。

A. 5　　B. 6　　C. 8　　D. 9

④ 已有定义“int i, a[10], *p;”则下述语句中合法的赋值语句是_____。

A. p=100;　　B. p=a[5];　　C. p=a[2]+2;　　D. p=a+2;

【答案】① B　　② B　　③ D　　④ D

【练习 6.6】① 有以下程序：

```
void prt(int *m, int n)
{
    int i;
    for(i=0; i<n; i++)
        m[i]++;
}
void main()
{
    int a[]={1, 2, 3, 4, 5},i;
    prt(a, 5);
    for(i=0; i<5; i++)
        printf("%d, ", a[i]);
}
```

运行程序后的输出结果是_____。

A. 1, 2, 3, 4, 5,　　B. 2, 3, 4, 5, 6,　　C. 3, 4, 5, 6, 7,　　D. 2, 3, 4, 5, 1,

② 有以下程序：

```
void sum(int *a)
{
    a[0]=a[1];
}
main()
{   int aa[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, i;
    for(i=2; i>=0; i--)
        sum(&aa[i]);
    printf("%d", aa[0]);
}
```

运行程序后的输出结果是_____。

A. 4　　B. 3　　C. 2　　D. 1

③ 有以下程序：

```
#include<stdio.h>
void main()
{
    int a[]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,};
    int *p=a+5,*q=NULL;
    *q=*(p+5);
    printf("%d  %d \n", *p, *q);
}
```

运行程序后的结果是_____。

A. 报错　　B. 输出 6　6　　C. 输出 6　12　　D. 输出 5　5

【答案】① B　　② A　　③ B

【练习 6.7】下述程序的运行结果是_____。

```
#include<stdio.h>
void f(int b[], int n)
{
    int i, r;
    r=1;
    for(i=0; i<=n; i++)
        r=r*b[i];
    return r;
}
void main()
{
    int x, a[]={2, 3, 4, 5, 6, 7, 8, 9};
    x=f(a, 3);
    printf("%d\n", x);
}
```

【答案】120

【练习 6.8】① 有以下程序：

```
#include<stdio.h>
#include<stdlib.h>
int fun(int n)
{
    int *p;
    p=(int*)malloc(sizeof(int));
    *p=n;
    return *p;
}
void main()
{
    int a;
    a=fun(10);
    printf("%d\n", a+fun(10));
}
```

运行程序的结果是_____。

A. 输出 0　　B. 输出 10　　C. 输出 20　　D. 出错

② 有以下程序段：

```
int *p;
p= ____malloc(sizeof(int));
```

若要使指针变量 p 指向一个存储整型变量的动态存储单元，则应填入____。

A. int　　B. int *　　C. (*int)　　D. (int*)

③ 若指针 p 已正确定义，int 型数据占 2 个字节，则下述使指针 p 指向两个连续整型动态存储单元的语句中，不正确的语句是____。

A. p=2*(int*)malloc(sizeof(int));　　B. p=(int*)malloc(2*sizeof(int));

C. p=(int*)malloc(2*2);　　D. p=(int*)calloc(2, sizeof(int));

【答案】① C　　② D　　③ A

【练习 6.9】① 若有定义语句“char s[10]="1234567\0\0";”，则 strlen(s)的值是_____。

A. 7　　B. 8　　C. 9　　D. 10

② 有以下程序：

```
#include<stdio.h>
#include<string.h>
void main()
{
    char a1[]="abcd", a2[]="ABCD", *p1=a1, *p2=a2, str[50]="xyz";
    strcpy(str+2, strcat(p1+2, p2+1));
    printf("%s", str);
}
```

运行程序后的输出结果是_____。

A. xyabcAB　　B. abcABz　　C. ABabcz　　D. xycdBCD

③ 以下语句或语句组中，能正确进行字符串赋值的语句是_____。

A. char *sp; *sp="right!";　　B. char s[10]; s="right!";

C. char s[10]; *s="right!";　　D. char *sp="right!";

④ 有以下程序：

```
void ss(char *s, char t)
{
    while(*s)
    {
        if(*s==t)
            *s=t-'a'+'A';
        s++;
    }
}
void main()
{
    char str1[100]="abcddfefdbd", c='d';
    ss(str1, c);
    printf("%s\n", str1);
}
```

运行程序后的输出结果是_____。

A. ABCDDFEFDBD　　B. abcDDfefDbD　　C. abcAAfefAbA　　D. Abcddfefdbd

⑤ 有以下程序：

```
#include<stdio.h>
#include<string.h>
void  main()
{
    char  *s1="AbCdEf", *s2="aB";
    s1++;   s2++;
    printf("%d\n", strcmp(s1, s2));
}
```

运行程序后的输出结果是_____。

A. 正数　　B. 负数　　C. 零　　D. 不确定的值

⑥ 有以下程序：

```
#include<stdio.h>
f(char  *s)
{   char  *p=s;
    while(*p!='\0')
        p++;
    return(p-s);
}
void  main()
{
    printf("%d\n", f("ABCDEF"));
}
```

运行程序后的输出结果是_____。

A. 3　　B. 6　　C. 8　　D. 0

⑦ 已定义如下函数：

```
fun(char  *p2,char  *p1)
{   while((*p2=*p1)!='\0')
      {p1++;     p2++;}
}
```

函数的功能是____。

A. 将p1所指的字符串复制到p2所指的内存空间

B. 将p1所指的字符串的地址赋给指针p2

C. 对p1和p2两个指针所指的字符串进行比较

D. 检查p1和p2两个指针所指的字符串中是否有'\0'

⑧ 有以下程序：

```
#include<stdio.h>
#include<string.h>
main()
{
     char  ss[10]="12345";
```

```
    strcat(ss, "6789");
    gets(ss);
    printf("%s\n", ss);
}
```

当执行上述程序且输入ABC时，输出的结果是_____。

A. ABC　　B. ABC9　　C. 123456ABC　　D. ABC456789

【答案】① A　② D　③ D　④ B

⑤ A　⑥ B　⑦ A　⑧ A

【练习 6.10】运行以下程序后的输出结果是____。

```
#include<string.h>
void fun(char *s, int p, int k)
{   int i;
    for(i=p; i<k-1; i++)
        s[i]=s[i+2];
}
void main()
{   char s[]="abcdefg";
    fun(s, 3, strlen(s));
    puts(s);
}
```

【答案】abcfg

【练习 6.11】运行以下程序后的输出结果是____。

```
void main()
{
    char a[]="language";
    char *ptr=a;
    while(*ptr)
    {
        printf("%c", *ptr-32);
        ptr++;
    }
}
```

【答案】LANGUAGE

【练习 6.12】① 设有定义：

```
int n=0, *p=&n, **q=&p;
```

以下赋值语句中，正确的语句是_____。

A. p=1;　　B. *q=2;　　C. q=p;　　D. *p=5;

② 设有语句“int (*ptr)[M];”，其中的标识符ptr是_____。

A. M个指向整型变量的指针

B. 指向M个整型变量的函数指针

C. 一个指向M个整型元素的一维数组的指针

D. 具有M个指针元素的一维指针数组，每个元素都只能指向整型变量

③ 设有语句“static int a[2][3]={2, 4, 6, 8, 10, 12};”，下述各项中，正确表示数组元素地址的是_____。

A. *(a+1)　　B. *(a[1]+2)　　C. a[1]+3　　D. a[0][0]

④ 有以下程序：

```
#include<stdio.h>
void main()
{
    char *alpha[6]={"ABCD", "EFGH", "IJKL", "MNOP", "QRST", "UVWX"};
    char **p;
    int i;
    p=alpha;
    for(i=0; i<4; i++)
        printf("%s", p[i]);
    printf("\n");
}
```

运行程序后的输出结果为_____。

A. ABCDEFGHIJKL　　B. ABCD

C. ABCDEFGHIJKLMNOP　　D. AEIM

⑤ 有以下程序：

```
void main()
{   char *s[]={"one", "two", "three"}, *p;
    p=s[1];
    printf("%c, %s", *(p+1), s[0]);
}
```

运行程序后的输出结果为_____。

A. n, two　　B. t, one　　C. w, one　　D. o, two

【答案】① D　② C　③ C　④ C　⑤ C

【练习 6.13】运行以下程序后的输出结果是_____。

```
void main()
{   int a[5]={2, 4, 6, 8, 10}, *p, **k;
    p=a;
    k=&p;
    printf("%d, ", *p++);
    printf("%d\n", **k);
}
```

【答案】2, 4

【练习 6.14】运行以下程序后的输出结果是_____。

```
#include<string.h>
void main()
{   char ch[]="abc", x[3][4];
```

```
    int  i;
    for(i=0; i<3; i++)
        strcpy(x[i], ch);
    for(i=0; i<3; i++)
        printf("%s", &x[i][i]);
}
```

【答案】abcbcc

【练习 6.15】① 有以下程序：

```
#include<string.h>
main(int  argc, char  *argv[])
{   int  i, len=0;
    for(i=1; i<argc; i+=2)
        len+=strlen(argv[i]);
    printf("%d", len);
}
```

经编译链接后生成的可执行文件是ex.exe，若运行时输入以下带参数的命令行

```
ex  abcd  efg  h3  k44
```

运行程序后的输出结果是____。

A. 14　　B. 12　　C. 8　　D. 6

② 假定下列程序的可执行文件名为prg.exe：

```
main(int  argc, char  *argv[])
{   int  i;
    if(argc<=0)
        return;
    for(i=1; i<argc; i++)
        printf("%c", *argv[i]);
}
```

则在该程序所在的子目录下输入命令行：

```
prg  hello  good
```

按回车后，输出结果是____。

A. hello good　　B. hg　　C. hel　　D. hellogood

【答案】① D　　② B

【练习 6.16】① 若有说明语句“double *p, a;”则下述各程序段中，能通过语句正确给输入项读入数据的程序段是_____。

A. *p=&a;　scanf("%1f", p);　　B. *p=&a;　scanf("%f", p);

C. *p=&a;　scanf("%1f", *p);　　D. p=&a;　scanf("%1f", p);

② 有以下程序：

```
void  swap(char  *x, char  *y)
{
    char  t;
    t=*x;    *x=*y;    *y=t;
```

```
}
void main()
{
    char s1[]="abc", s2[]="123";
    swap(s1, s2);  printf("%s,%s\n", s1, s2);
}
```

运行程序后的输出结果是_____。

A. 123, abc　　B. abc, 123　　C. 1bc, a23　　D. 321, cba

③ 有以下程序：

```
void main()
{
    char s[]={"aeiou"}, *ps;
    ps=s;     printf("%c\n", *ps+4);
}
```

运行程序后的输出结果是_____。

A. a　　B. e　　C. u　　D. 元素s[4]的地址

④ 有以下程序：

```
int *f(int *x, int *y)
{  if(*x<*y)
       return x;
   else
       return y;
}
void main()
{
   int a=7, b=8, *p, *q, *r ;
   p=&a;    q=&b;
   r= f(p, q);
   printf("%d, %d, %d", *p, *q, *r);
}
```

运行程序后的输出结果是_____。

A. 7, 8, 8　　B. 7, 8, 7　　C. 8, 7, 7　　D. 8, 7, 8

⑤ 定义语句“int (*ptr)();”的含义是_____。

A. ptr 是指向一维数组的指针变量

B. ptr 是指向 int 型数据的指针变量

C. ptr 是指向函数的指针，该函数返回一个 int 型数据

D. ptr 是一个函数名，该函数的返回值是指向 int 型数据的指针

⑥ 已有定义“int (*p)();”指针 p_____。

A. 代表函数的返回值　　B. 指向函数的入口地址

C. 表示函数的类型　　D. 表示函数的返回值的类型

⑦ 已有函数 max(a, b)，为了让函数指针变量 p 指向函数 max，正确的赋值方法是_____。

A. p=max;　　B. *p=max;　　C. p=max(a,b);　　D. *p=max(a,b);

⑧ 已有函数 max(a, b)，函数指针变量 p 已经指向函数 max，正确调用该函数的方法是_____。

A. (*p)=max(a,b);　　B. *pmax(a,b);　　C. (*p)(a, b);　　D. *p(a, b);

【答案】① D　② C　③ B　④ B

⑤ C　⑥ B　⑦ A　⑧ C

第 7 章　构造数据类型与编译预处理

【练习 7.1】以下结构体类型变量 student1 占用多少个字节内存空间？

```
struct  Student
{
    char  no[10], name[20], sex;        //学号，姓名，性别
    int  age, classno;                  //年龄，班级
    float  grade;                       //成绩
}student1;
```

【答案】sizeof(struct student1)=10+20+1+4+4+4=43

【练习 7.2】① 定义一个有关时间的结构体变量(其中包括小时、分钟、秒)，从键盘为其输入数据，并显示。

② 定义一个日期结构体变量(包括年、月、日)。计算该日在本年中是第几天？注意闰年问题。

【答案】参考代码：

①
```
#include<stdio.h>
int  main(void)
{
    struct  time
    {
        int  hour, minute, second;
    }t;

    printf("请输入时间：\n");
    scanf("%d%d%d", &t.hour, &t.minute, &t.second);
    printf("输出显示时间：\n");
    printf("%d:%d:%d", t.hour, t.minute, t.second);
    return  0;
}
```

②
```
#include<stdio.h>
int  main(void)
```

```
{
    struct
    {
        int  year, month, day;
    }d;
    int  a[2][13]={{0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},
                   {0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}};
    int  i, k;
    printf("请输入一个日期：\n");
    scanf("%d%d%d", &d.year, &d.month, &d.day);
    k=(d.year%4==0&&d.year%100!=0)||d.year%400==0;
    for(i=0; i<d.month; i++)
        d.day=d.day+a[k][i];
    printf("该日是本年中的第%d 天\n", d.day);
    return  0;
}
```

【练习 7.3】定义一个结构体数组用以保存 10 个学生的以下信息：学号、姓名、性别、家庭住址、6 门课程成绩。要求：

① 从键盘输入 10 个学生的数据。

② 显示每个学生 6 门课程中的最低分和最高分。

③ 显示 10 个学生中第一门课程的最低分和最高分。

④ 显示 10 个学生中有一门不及格和所有课程均不及格的人数。

⑤ 检索学号为 NUM 的学生的 6 门课程成绩，NUM 由键盘输入。

【答案】参考代码：

```
#include<stdio.h>
#define  N  10
int  main(void)
{
    struct  student
    {
        int  no;
        char  name[20];
        char  sex;
        char  addr[20];
        float  score[6];
    }stu[10];
    float  min, max;
    int  i, j, count, c1=0, c2=0, NUM;
    //1. 从键盘输入 10 个学生的数据
    printf("请输入 10 个学生的信息数据：\n");
    for(i=0; i<N; i++)
```

```
{
    scanf("%d,%s,%c,%s", &stu[i].no, stu[i].name, &stu[i].sex, stu[i].addr);
    for(j=0; j<6; j++)
        scanf("%f", &stu[i].score[j]);
}
//2. 显示每个学生 6 门课程中的最低分和最高分
printf("显示每个学生 6 门课程中的最低分和最高分:\n");
for(i=0; i<N; i++)
{
    max=min=stu[i].score[0];
    for(j=1; j<6; j++)
    {
        if(stu[i].score[j]<min)
            min=stu[i].score[j];
        if(stu[i].score[j]>max)
            max=stu[i].score[j];
    }
    printf("第%d 个学生的最高分是：%.2f　最低分是%.2f\n", i+1, max, min);
}
//3. 显示 10 个学生中第一门课程的最低分和最高分
printf("显示 10 个学生中第一门课程的最低分和最高分\n");
for(i=1; i<N; i++)
{
    max=min=stu[0].score[0];
    if(stu[i].score[0]>max)
        max=stu[i].score[0];
    if(stu[i].score[0]<min)
        min=stu[i].score[0];
}
printf("10 个学生中第一门课的最低分：%f，最高分是：%f\n", min, max);
//4. 显示 10 个学生中有一门课程不及格和所有课程均不及格的人数
printf("显示 10 个学生中有一门课程不及格和所有课程均不及格的人数\n");
for(i=0; i<N; i++)
{
    count=0;
    for(j=0; j<6; j++)
      if(stu[i].score[j]<60)
        count++;
    if(count==1)
      c1++;
    else  if(count==6)
```

```
            c2++;
        }
        printf("有一门课程不及格的人数为%d\n, 所有课程不及格的人数%d\n", c1, c2);
        //5. 检索学号为 NUM 的学生的 6 门课程成绩
        printf("输出学号为 NUM 的学生的 6 门课程成绩\n");
        printf("请输入学号 NUM:");
        scanf("%d", &NUM);
        for(i=0; i<N; i++)
        {
            if(stu[i].no==NUM)
            {
                printf("学号为%d 的学生的 6 门课程成绩分别是:", NUM);
                for(j=0; j<6; j++)
                    printf("%6.2f", stu[i].score[j]);
            }
        }
        return 0;
    }
```

【练习 7.4】有以下程序：

```
struct STU
{   char name[10];
    int num;
};
void f1(struct STU c)
{   struct STU b={"LiSiGuo", 2042};
    c=b;
}
void f2(struct STU *c)
{   struct STU b={"SunDan", 2044};
    *c=b;
}
main()
{   struct STU a={"YangSan", 2041}, b={"WangYin", 2043};
    f1(a);   f2(&b);
    printf("%d %d", a.num, b.num);
}
```

运行程序后的输出结果是____。

A. 2041 2044　　B. 2041 2043　　C. 2042 2044　　D. 2042 2043

【答案】A

【练习 7.5】有如下定义：

```
struct person
```

```
{    char name[9];
     int age;
};
struct person class[10] = {"John", 17, "Paul", 19, "Mary", 18, "Adam", 16,};
```

根据上述定义，下述语句中能输出字母 M 的语句是_____。

A. printf("%c\n", class[3].name[0]);　　B. printf("%c\n", class[3].name[1]);

C. printf("%c\n", class[2].name[1]);　　D. printf("%c\n", class[2].name[0]);

【答案】D

【练习 7.6】有以下结构体说明和变量的定义，且指针 p 指向变量 a，指针 q 指向变量 b：

```
struct node
{    char data;
     struct node *next;
}a, b, *p=&a, *q=&b;
```

下述语句中，不能把结点 b 连接到结点 a 之后的语句是_____。

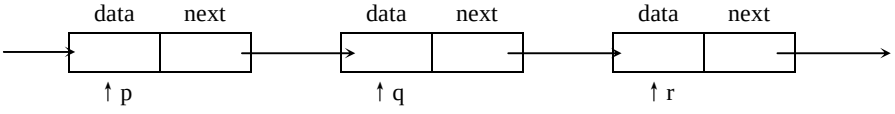
A. p=&a;　　B. p->next=&b;　　C. (*p).next=q;　　D. p->next=q;

【答案】A

【练习7.7】有以下结构体说明和变量定义：

```
struct node
   {int data;    struct node *next;}*p, *q, *r;
```

如下图所示，指针p，q，r分别指向链表中的三个连续结点。



练习7.7的链表示意图

现要将q所指结点从链表中删除，同时要保持链表的连续，以下语句中不能完成这项指定操作的语句是____。

A. p->next=q->next;　　B. p->next=p->next->next;

C. p->next=r;　　D. p=q->next;

【答案】D

【练习 7.8】有以下程序：

```
union myun
{   struct
    {
        int x, y, z;
    }u;
    int k;
}a;
void main()
{   a.u.x=4;
    a.u.y=5;
```

```
    a.u.z=6;
    a.k=0;
    printf("%d\n", a.u.x);
}
```

运行程序后的输出结果是_____。

A. 4　　B. 5　　C. 6　　D. 0

【答案】D

【练习7.9】若有以下定义和语句：

```
union data
{  int i;
   char c;
   float f;
}x;
int y;
```

以下语句中正确的语句是_____。

A. x=10.5;　　B. x.c=101;　　C. y=x;　　D. printf("%d\n", x);

【答案】B

【练习 7.10】已知枚举类型定义为“enum color{yellow, red, blue, green, white, black};”从键盘输入一个整数(0～5)，显示与其相对应的枚举常量的英文名称。

【答案】参考代码：

```
#include<stdio.h>
int main(void)
{
    enum color{yellow, red, blue, green, white, black}c;
    int k;
    printf("input a number(0～5)");
    scanf("%d", &k);
    c=(enum color)k;
    switch(c)
    {
      case yellow: printf("yellow\n");  break;
      case red:    printf("red\n");  break;
      case blue:   printf("blue\n");  break;
      case green:  printf("green\n");  break;
      case white:  printf("white\n");  break;
      case black:  printf("black\n");  break;
      default:     printf("input error\n");  break;
    }
    return 0;
}
```

【练习 7.11】以下各选项中，不能把 c1 定义成结构体变量的是_____。

A.
```
typedef struct
{   int red;
    int green;
    int blue;
}COLOR;
COLOR c1;
```

B.
```
struct color c1
{   int red;
    int green;
    int blue;
};
```

C.
```
struct color
{   int red;
    int green;
    int blue;
}c1;
```

D.
```
struct
{   int red;
    int green;
    int blue;
}c1;
```

【答案】B

【练习7.12】有以下程序：

```
#include<stdio.h>
#define SUB(a) (a)-(a)
main()
{   int a=2, b=3, c=5, d;
    d=SUB(a+b)*c;
    printf("%d\n", d);
}
```

运行程序后的输出结果是_____。

A. 0　　B. −12　　C. −20　　D. 10

【答案】C

【练习 7.13】有以下程序：

```
#define f(x) x*x
main()
{   int i;
    i=f(4+4)/f(2+2);
    printf("%d", i);
}
```

运行程序后的输出结果是_____。

A. 28　　B. 22　　C. 16　　D. 4

【答案】A

第三部分　习题及参考答案

第 1 章　C 语言程序设计概述

1. 以下不能定义为用户标识符的是_____。

A. scanf　　B. Void　　C. _3com_　　D. int

【答案】D

2. 计算机能直接执行的程序是_____。

A. 源程序　　B. 目标程序　　C. 汇编程序　　D. 可执行程序

【答案】D

3. C语言源程序名的后缀是_____。

A. .exe　　B. .c　　C. .obj　　D. .cp

【答案】B

4. 对于一个正常运行的C程序，以下叙述中正确的是_____。

A. 程序的执行总是从main函数开始，在main函数结束

B. 程序的执行总是从程序的第一个函数开始，在main函数结束

C. 程序的执行总是从main函数开始，在程序的最后一个函数中结束

D. 程序的执行总是从程序中的第一个函数开始，在程序的最后一个函数中结束

【答案】A

5. 以下叙述中正确的是_____。

A. C程序的注释部分可以出现在程序中任意合适的地方

B. 大括号“{”和“}”只能作为函数体的定界符

C. 构成C程序的基本单位是函数，所有函数名都可以由用户命名

D. 分号是C语言之间的分隔符，不是语句的一部分

【答案】A

6. 在一个C程序中_____。

A. main函数必须出现在所有函数之前　　B. main函数可以在任何地方出现

C. main函数必须出现在所有函数之后　　D. main函数必须出现在固定位置

【答案】B

7. 以下4组用户定义的标识符中，全部合法的一组是____。

A. _main
enclude
sin

B. If
−max
turbo

C. txt
REAL
3COM

D. int
k_2
_001

【答案】A

8. 一个算法应该具有“确定性”等5个特性，下面对另外4个特性的描述中错误的是_____。

A. 有零个或多个输入　　B. 有零个或多个输出

C. 有穷性　　D. 可行性

【答案】B

9. 以下叙述中正确的是_____。

A. C语言的源程序不必通过编译就可以直接运行

B. C语言中的每条可执行语句最终都将被转换成二进制的机器指令

C. C语言源程序经编译形成的二进制代码可以直接运行

D. C语言中的函数不可以单独进行编译

【答案】B

第 2 章　数据类型和表达式

1. 已定义a为字符型变量，则下列语句中正确的是_____。

A. a='97'　　B. a="97";　　C. a=97;　　D. a="a";

【答案】C

2. 有定义语句“int x, y;”。若要通过“scanf("%d,%d", &x, &y);”语句使变量x得到数值11，变量y得到数值12，下面四组输入形式中，错误的是_____。

A. 11 12↙　　B. 11,12↙　　C. 11, 12↙　　D. 11,↙12↙

【答案】A

3. 以下选项中可作为C语言合法整数的是_____。

A. 10110B　　B. 0386　　C. 0Xffa　　D. x2a2

【答案】C

4. 有以下程序，程序运行后的输出结果是_____。

A. 6, 1　　B. 2, 1　　C. 6, 0　　D. 2, 0

```
void main()
{
    int a, b, d=25;
    a=d/10%9;
```

```
    b=a&&(-1);
    printf("%d, %d\n", a, b);
}
```

【答案】B

5. 以下语句中，非法的赋值语句是_____。

A. n=(i=2, ++i);　　B. j++;　　C. ++(i+1);　　D. x=j>0;

【答案】C

6. 若函数中有以下语句“int k;”，则_____。

A. 系统将自动给 k 赋初值 0　　B. 这时 k 中的值无定义

C. 系统将自动给 k 赋初值-1　　D. k 的值为随机值

【答案】D

7. 以下各项中，能用作数据常量的是_____。

A. o123　　B. 028　　C. 1.3e1.2　　D. 115L

【答案】D

8. 有以下程序：

```
void main()
{   int k=11;
    printf("k=%d, k=%o, k=%x\n", k, k, k);
}
```

运行程序后的输出是_____。

A. k=11, k=12, k=11　　B. k=11, k=13, k=13

C. k=11, k=013, k=0xb　　D. k=11, k=13, k=b

【答案】D

9. 若整型变量 a，b 的值分别为 7 和 9，要求按以下格式输出 a，b 的值:

```
a=7
b=9
```

请完成输出语句：“printf("_____", a, b);”。

【答案】a=%d\nb=%d

10. 以下各项中，不能作为 C 语言合法常量的是_____。

A. 'ab'　　B. 0.1e+6　　C. "\a"　　D. '\011'

【答案】A

11. 以下各项中，正确的定义语句是_____。

A. double a; b;　　B. double c=d=1;　　C. double, c, d;　　D. double c=7, d=7;

【答案】D

12. 若有表达式(w)?(x++):(y--)，则下述选项中可以替换式中 w 的是_____。

A. w==0　　B. w!=0　　C. w==1　　D. w!=1

【答案】B

13. 以下能正确定义变量且赋初值的语句是_____。

A. int n1=n2=10;　　B. char c=32;

C. float f=f+1.1;　　D. double x=12.3E2.5;

【答案】B

14. 以下程序的功能是给 r 输入数据后计算半径为 r 的圆面积 s，程序在编译时出错。

```
void main()
/* Beginning */
{
    int  r; float  s;
    scanf("%d", &r);
    s=*∏*r*r;
    printf("s=%f\n", s);
}
```

出错的原因是_____。

A. 注释语句书写位置错误

B. 存放圆半径的变量 r 不应该定义为整型

C. 输出语句中格式描述符非法

D. 计算圆面积的赋值语句中使用了非法变量

【答案】D

15. 已知字符 A 的 ASCII 码值为 65，以下程序运行后的输出结果是_____。

```
void  main()
{   char  a, b;
    a='A'+'5'-'3';     b=a+'6'-'2';
    printf("%d%c\n", a, b);
}
```

【答案】67G

16. 以下各项中，值为 1 的表达式是_____。

A. 1-'0'　　B. 1-'\0'　　C. '1'-0　　D. '\0'-'0'

【答案】B

第 3 章　程序控制结构

1. 有以下程序：

```
void  main()
{   int  i;
    for(i=0; i<3; i++)
        switch(i)
        {   case  0:  printf("%d", i);
            case  2:  printf("%d", i);
            default:  printf("%d", i);
        }
}
```

运行程序后的输出结果是_____。

A. 022111　　B. 021021　　C. 000122　　D. 012

【答案】C

2. 有以下程序：

```
void main()
{  int a=3, b=4, c=5, d=2;
   if(a>b)
      if(b>c)
         printf("%d", d++ +1);
      else
         printf("%d", ++d +1);
   printf("%d", d);
}
```

运行程序后的输出结果是_____。

A. 2　　B. 3　　C. 43　　D. 44

【答案】A

3. 若变量已正确定义，要求程序段完成求5!的计算，不能完成此操作的程序段是_____。

A.
```
for(i=1, p=1; i<=5; i++)
    p*=i;
```

B.
```
for(i=1; i<=5; i++)
{  p=1;    p*=i;}
```

C.
```
i=1; p=1;
while(i<=5)
{  p*=i;   i++;}
```

D.
```
i=1;      p=1;
do
{  p*=i;      i++;}while(i<=5);
```

【答案】B

4. 有以下程序：

```
void main()
{  int i, s=0;
   for(i=1; i<10; i+=2)
      s+=i+1;
   printf("%d", s);
}
```

运行程序后的输出结果是_____。

A. 自然数1～9的累加和　　B. 自然数1～10的累加和

C. 自然数1～9中奇数之和　　D. 自然数1～10中偶数之和

【答案】D

5. 以下程序段中与语句“k=a>b?(b>c?1:0):0;”功能等价的是_____。

A.
```
if((a>b)&&(b>c)) k=1;
else k=0;
```

B.
```
if((a>b)||(b>c)) k=1;
else k=0;
```

C.
```
if(a<=b) k=0;
else if(b<=c) k=1;
```

D.
```
if(a>b) k=1;
else if(b>c) k=1;
else k=0;
```

【答案】A

6. 有以下程序：

```
void main( )
{  int i;
   for(i=1; i<6; i++)
   {   if(i%2)
       {
           printf("#");
           continue;
       }
       printf("*");
   }
   printf("\n");
}
```

运行程序后的输出结果是_____。

A. #*#*#　　B. #####　　C. *****　　D. *#*#*

【答案】A

7. 有以下程序：

```
void main()
{  int k=4, n=0;
   for( ; n<k;)
   {  n++;
      if(n%3!=0)
         continue;
      k--;}
   printf("%d,%d", k, n);
}
```

运行程序后的输出结果是_____。

A. 1, 1　　B. 2, 2　　C. 3, 3　　D. 4, 4

【答案】C

8. 有如下程序：

```
void main()
{  int i, sum;
   for(i=1; i<=3; sum++)
      sum+=i;
   printf("%d\n", sum);
}
```

运行程序后的结果是_____。

A. 输出6　　B. 输出3　　C. 输出0　　D. 陷入死循环

【答案】D

9. 有以下程序：

```
#include<stdio.h>
void  main()
{   int  x=1, y=2, z=3;
    if(x>y)
        if(y<z)
            printf("%d", ++z);
        else
            printf("%d", ++y);
    printf("%d\n", x++);
}
```

运行程序后的输出结果是_____。

A. 331　　B. 41　　C. 2　　D. 1

【答案】D

10. 有以下程序段：

```
int  a=3, b=5, c=7;
if(a>b)
     a=b;
c=a;
if(c!=a)
     c=b;
printf("%d, %d, %d", a, b, c);
```

运行程序后的结果是____。

A. 程序段有语法错误　　B. 输出3, 5, 3　　C. 输出3, 5, 5　　D. 输出3, 5, 7

【答案】B

编制程序，完成 11～18 题的要求。

11. 从键盘上输入一个英文字母，如果输入的英文字母为 y 或者 Y，则输出 Yes；如果输入的英文字母为 n 或者 N，则输出 No。

【答案】参考代码：

```
#include<stdio.h>
int  main()
{
     char  c;

     //1. 输入
     printf("请输入一个字符：");
     scanf("%c", &c);

     //2. 判断
     if((c=='y') || (c=='Y'))
          printf("Yes.\n");
     else  if((c=='n') || (c=='N'))
```

```
            printf("No.\n");
        else
            printf("Input  error.\n");
        return  0;
    }
```

12. 输出 100～999 之间的水仙花数。(三位水仙花数是指一个三位数，各位的立方和等于该三位数，如 153，370 等。)

【答案】参考代码：

```
#include<stdio.h>
int  main()
{
    int  i, bai, shi, ge;
    for(i=100; i<=999; i++)
    {
        //1. 求每一位数
        ge=i%10;            //个位
        shi=i%100/10;       //十位
        bai=i/100;          //百位

        //2. 判断是否水仙花数
        if(i==bai*bai*bai+shi*shi*shi+ge*ge*ge)
            printf("%d 是水仙花数。\n", i);
    }
    return  0;
}
```

13. 从键盘输入一个正整数，如果该数为素数，则输出该素数，否则输出该数的所有因子(除去 1 与自身)。

【答案】参考代码：

```
#include<stdio.h>
int  main()
{
    int  i, n;

    //1. 输入
    printf("请输入一个正整数：");
    scanf("%d", &n);

    //2. 判断 n 是否为素数
    for(i=2; i<n; i++)
        if(n%i==0)
            break;

    //3. 若 n 是素数，则显示其为素数；否则，输出其因子
```

```
    if(i==n)
        printf("%d 为素数。\n", n);
    else
    {
        printf("%d 不是素数，因子有：", n);
        for(i=2; i<n; i++)
            if(n%i==0)
                printf("%d ", i);
    }
    return 0;
}
```

14. 显示输出能写成两个数平方和的所有三位数。

【答案】参考代码：

```
#include<stdio.h>
int main()
{
    //穷举法
    int i, j, k;
    for(i=100; i<=999; i++)
    {
        //判断 i 是否为两个数的平方和
        for(j=1; j<33; j++)              //依据题意，减少循环次数
        {
            for(k=j; k<33; k++)
                if(i==j*j+k*k)       //i 是两个数的平方和
                {
                    printf("%d*%d+%d*%d=%d\n ", j, j, k, k, i);
                    break;
                }
            if(k<33)
                break;
        }
    }
    return 0;
}
```

15. 从键盘输入 20 个实数，分别计算并输出所有正数之和，所有负数之和，所有数的绝对值之和，正数的个数和负数的个数。

【答案】参考代码：

```
#include<stdio.h>
int main()
{
```

```
    int  i, num, cnt1, cnt2, sum1, sum2, sum3;
    cnt1=0;                 //正数个数
    cnt2=0;                 //负数个数
    sum1=0;                 //正数之和
    sum2=0;                 //负数之和
    sum3=0;                 //所有数绝对值之和
    for(i=1; i<=20; i++)
    {
        scanf("%d", &num);
        if(num>0)
        {
            sum1+=num;
            sum3+=num;
            cnt1++;
        }
        else
            if(num<0)
            {
                sum2+=num;
                sum3+=-num;
                cnt2++;
            }
    }
    printf("所有正数之和%d，所有负数之和%d，所有数的绝对值之和%d，正数的
        个数%d 和负数的个数%d", sum1, sum2, sum3, cnt1, cnt2);
    return  0;
}
```

16. 求 2 到 500 之间的所有亲密数对。亲密数对的定义为：如果 m 的因子(不含 1 和 m)之和为 n，且 n 的因子(不含 1 和 n)之和为 m，则称 m 与 n 为一对亲密数。

【答案】参考代码：

```
#include<stdio.h>
int  main()
{
    int  m, j, cnt1, cnt2;
    for(m=2; m<=500; m++)
    {
        cnt1=cnt2=0;
        //求 m 的亲密对数
        for(j=2; j<m; j++)
            if(m%j==0)
                cnt1+=j;
```

```
            if(cnt1<=500 && cnt1!=0)
                for(j=2; j<cnt1; j++)
                    if(cnt1%j==0)
                        cnt2+=j;
            if(cnt2==m && cnt2!=0)
                printf("亲密对数：%d — %d\n", m, cnt1);
        }
        return 0;
    }
```

17. 某参观团按以下条件限制从 A、B、C、D 和 E 五个地方中选定若干参观点。

① 如果去 A 地，则必须去 B 地；

② D 和 E 两地中只能去一地；

③ B 和 C 两地中只能去一地；

④ C 和 D 两地要么都去，要么都不去；

⑤ 如果去 E 地，则必须去 A 和 D 地。

问该参观团能去哪些地方？

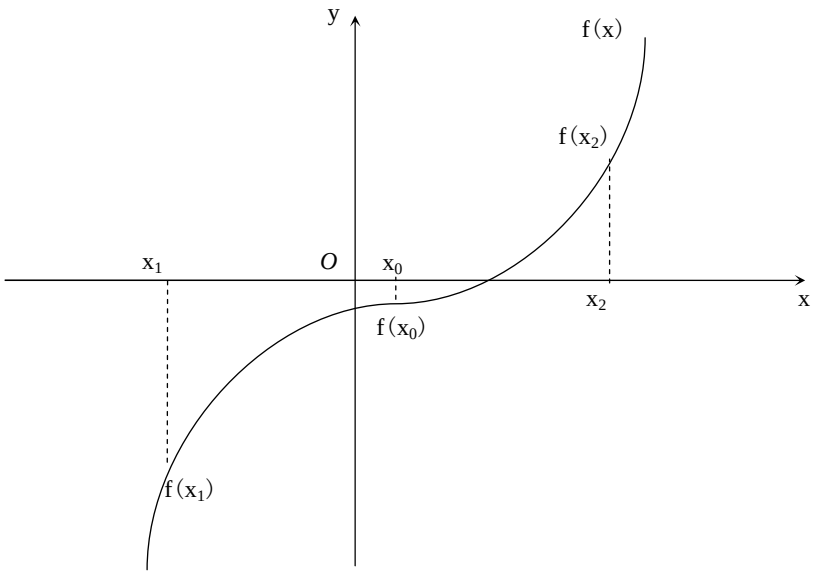
【答案】参考代码：

```
#include<stdio.h>
int main()
{ int A, B, C, D, E;
  for(A=0; A<=1; A++)
    for(B=0; B<=1; B++)
      for(C=0; C<=1; C++)
        for(D=0; D<=1; D++)
          for(E=0; E<=1; E++)
            if(((A==1 && B==1)||A==0) &&(D+E!=2) &&(B+C!=2) && C==D
                &&(E==0 ||(E==1 && A==1 && D==1)))
              printf("A=%d, B=%d, C=%d, D=%d, E=%d\n", A, B, C, D, E);
  return 0;
}
```

18. 用对分法求方程 $2x^3-4x^2+6x-1=0$ 在区间[−10，10]上的实根，精度要求 $\varepsilon=10^{-6}$。

【答案】

二分法的思路如下：设方程左边的含 x 的式子为函数 f(x)。先指定一个区间$[x_1, x_2]$，如果函数 f(x)在此区间内为单调函数，且 $f(x_1)$和 $f(x_2)$的值异号，则 f(x)=0 在$[x_1, x_2]$区间必有一个(且只有一个)实根；如果 $f(x_1)$和 $f(x_2)$的值同号，则在$[x_1, x_2]$区间无实根，要重新改变 x_1和 x_2的值；如果 $f(x_1)$[或 $f(x_2)$]的值为 0，则方程的根为 x1(或 x2)。如果确定$[x_1, x_2]$中(非端点)有一个实根后，采取二分法将$[x_1, x_2]$区间一分为二，再判断在哪一个小区间中有实根。如此不断进行下去，直到小区间足够小为止，如下图所示。可以验证，本题中的 $f(x)=2x^3-4x^2+6x-1$ 为单调上升函数，f(−10)<0，f(10)>0。



用二分法求方程实根的示意图

参考代码：

```
#include<stdio.h>
#include<math.h>
void  main()
{
    double  x0, x1, x2, fx0, fx1, fx2;
    x1=-10.0;   x2=10.0;
    fx1=2*x1*x1*x1-4*x1*x1+6*x1-1;
    fx2=2*x2*x2*x2-4*x2*x2+6*x2-1;
    do{
        x0=(x1+x2)/2;                    //取 x1，x2 的中间值，得两个区间[x1, x0]和[x0, x2]
        fx0=2*x0*x0*x0-4*x0*x0+6*x0-1;
        if(fx0 == 0)
            break;
        else
            if(fx0*fx1<0)                //根在区间[x1, x0]内
            {
                x2=x0;
                fx2=fx0;
            }
            else                         //根在区间[x0, x2]内
            {
                x1=x0;
                fx1=fx0;
```

```
        }
    }while(fabs(x2-x1)>=1e-6);        //当区间长度大于 1e-6 时继续循环
    printf("x=%6.2f\n", x0);
    printf("f(x)=%6.2f\n", fx0);
}
```

第 4 章　数　　组

1. 编写程序，输入 5 个互异的整数存入一维数组 a 中，再输入一个整数 x，然后在数组中查找 x，如果找到输出相应的下标，否则输出“not found”。

【答案】参考代码：

```
#include<stdio.h>
int main(void)
{
    int i, flag, x;
    int a[5];
    printf("Enter 5 integers: ");
    for(i = 0; i < 5; i++)
        scanf("%d", &a[i]);
    printf("Enter x: ");
    scanf("%d", &x);
    flag = 0;
    for(i = 0; i < 5; i++)
        if(a[i] == x)
        {
            printf("Index is %d\n", i);
            flag = 1;
            break;
        }
        if(flag == 0)
            printf("Not Found\n");
    return 0;
}
```

2. 编写程序，输入一个正整数 n(1<n≤10)，再输入 n 个整数存入一维数组，把奇数从数组中删除，并统计出偶数个数。

【答案】参考代码：

```
#include"stdio.h"
int main()
```

```
{
    int  a[10], n, i;
    int  count=0;    //偶数的个数，同时作为数组下标
    printf("请输入一个不大于 10 的正整数：");
    scanf("%d", &n);

    //输入 n 个数，并存到数组 a 中
    printf("请输入 %d 个数\n", n);
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    for(i=0; i<n; i++)
        if(a[i]%2==0)    //是偶数，则保留并且 count 增 1
        {
            a[count]=a[i];
            count++;
        }
    //输出统计结果
    printf("删除奇数后的数组为:\n");
    for(i=0; i<count; i++)
        printf("%3d", a[i]);
    printf("\n 偶数的个数为%d\n", count);

    return 0;
}
```

3. 编写程序，输入一个正整数 n(1<n≤10)和 n 个有序整数(从小到大的顺序)，然后输入一个数 x，要求按原来排序的规律将它插入有序数据中，若原来序列中已有 x，则将原序列中的 x 删除。

【答案】参考代码：

```
#include<stdio.h>
int  main()
{
    int  n, a[11], x;
    int  i, isInsert=1, index;
    printf("请输入一个不大于 10 的正整数\n");
    scanf("%d", &n);
    printf("请输入%d 个有序的整数\n", n);
    for(i=0; i<n; i++)
        scanf("%d",&a[i]);

    printf("请输入一个整数，如果它包含在上面的有序数据中，则删除它，否则按原来
        排序规律插入\n");
    scanf("%d", &x);
```

```
        index=n;      /* 若 x 大于原来所有的数，则不执行下面的循环体，x 应插入到 a[n]
                         的位置 */
        for(i=0; i<n; i++)
            if(a[i]>=x)
            {
                if(a[i]==x)
                    isInsert=0;
                index=i;
                break;
            }
        if(isInsert)
        {
            printf("插入%d 后，序列变为：\n", x);
            for(i=n-1; i>=index; i--)
                a[i+1]=a[i];
            a[index]=x;
            for(i=0; i<n+1; i++)
                printf("%d ", a[i]);
        }
        else
        {
            printf("删除%d 后，序列变为：\n", x);
            for(i=index; i<n; i++)
                a[i]=a[i+1];
            for(i=0; i<n-1; i++)
                printf("%d ", a[i]);
        }
        return 0;
    }
```

4. 编写程序，输入一个以回车结束的字符串(少于80个字符)，再输入一个字符，统计并输出该字符在字符串中出现的次数，然后再输出该字符串。

【答案】参考代码：

```
#include<stdio.h>
int  main(void)
{
    int  count, i;
    char  str[80], ch;
    printf("请输入一个字符串: ");
    //输入字符串
    i = 0;
```

```
    while((str[i] = getchar())!= '\n')
        i++;
    str[i] = '\0';    //把回车符替换为字符串结束符
    printf("请再输入一个字符: ");
    ch=getchar();
    count = 0;
    for(i = 0; str[i] != '\0'; i++)
        if(str[i] ==ch)
            count++;
    printf("count = %d\n", count);
    //输出字符串
    for(i = 0; str[i] != '\0'; i++)
        putchar(str[i]);
    return  0;
}
```

5. 编写程序，输入一个以回车结束的字符串(少于 80 个字符)，将字符串中相同的字符存放在一起，并按 ASCII 码升序存放。

【答案】参考代码：

```
#include<stdio.h>
int  main()
{
    char  str[80];
    int  i, j, index;
    char  t;
    //输入字符串，输入字符串的方法有三种，思考一下其他方法
    printf("input  a  string：");
    i=0;
    while((str[i]=getchar())!='\n')
        i++;
    str[i]='\0';
    //使用选择法排序
    for(i=0; str[i]!='\0'; i++)
    {
        index=i;
        for(j=i+1; str[j]!='\0'; j++)
            if(str[index]>str[j])
                index=j;
        t=str[index];
        str[index]=str[i];
        str[i]=t;
```

```
    }
    printf("处理后的字符串是：%s \n", str);
    return 0;
}
```

6. 编写程序，输入一个正整数n(1≤n≤10)，再输入n阶方阵a，计算该矩阵除副对角线、最后一列和最后一行以外的所有元素之和(副对角线为从矩阵的右上角到左下角的连线)。

【答案】参考代码：

```
#include<stdio.h>
void main(void)
{   int n, i, j, sum=0;
    int a[10][10];
    printf("Enter an integers(1~10):\n");
    scanf("%d", &n);
    printf("Enter %d*%d integers:\n", n, n);
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            scanf("%d", &a[i][j]);
    for(i=0; i<n; i++)
     {
        for(j=0; j<n; j++)
            /* 副对角线上的元素下标满足关系 i+j=n-1，
               最后一列的列号 j=n-1，最后一行的行号 i=n-1 */
            if((i+j!=n-1)&&(j!=n-1)&&(i!=n-1))
                sum+=a[i][j];
     }
    printf("输入的方阵为:\n");
    for(i=0; i<n; i++)
    {
        for(j=0; j<n; j++)
            printf("%4d", a[i][j]);
        printf("\n");
    }
    printf("sum = %d\n", sum);
}
```

7. 编写程序，输出一个九九乘法表。

【提示】用乘数、被乘数做下标，乘积放在一个二维数组中，再输出该二维数组。

【答案】参考代码：

```
#include<stdio.h>
int main()
{
    int i, j;
```

```
    int  a[10][10];
    //给 a[i][j]赋值
    for(i=1; i<=9; i++)
        for(j=1; j<=i; j++)
            a[i][j]=i*j;
    // 输出结果
    for(i=1; i<=9; i++)
    {
        for(j=1; j<=i; j++)
            printf("%d*%d=%2d    ", j, i, a[i][j]);
        printf("\n");
    }
    return  0;
}
```

第5章　函　　数

1. 输入两个正整数m和n(m>n)，编写程序求 $C_m^n=\frac{m!}{n!(m-n)!}$ 的值。要求定义和调用fact(n)函数计算n!。

【答案】参考代码：

```
#include<stdio.h>
double  fact(int  n);
int  main(void)
{
    int  m, n;
    double  c;
    printf("Input  n:");
    scanf("%d", &n);
    printf("Input  m:");
    scanf("%d", &m);
    c=fact(m)/(fact(n)*fact(m-n));
    printf("C = %.0f\n", c);
    return 0;
}
double  fact(int  n)
{   int  i;
```

```
    double  result = 1;
    for(i = 1; i <= n; i++)
        result = result * i ;
    return  result ;
}
```

2. 输入精度 e，使用格里高利公式求 π 的近似值，精确到最后一项的绝对值小于 e。要求定义和调用 funpi(e)函数求 π 的近似值。格里高利公式为：$\frac{\pi}{4}=1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\cdots$

【答案】参考代码：

```
#include<stdio.h>
#include<math.h>
int  main(void)
{
    double  e, pi;                          // e 代表精度
    double  funpi(double  e);               // 函数声明
    printf("Enter  e:");                    // 提示用户输入精度
    scanf("%lf", &e);
    pi = funpi(e);                          // 调用函数，把返回值赋值给 pi
    printf ("pi = %.4f\n", pi);
    return  0;
}
double  funpi(double  e)                    // 定义计算 pi 的函数
{
    int  denominator, flag;
    double  item, sum;
    flag = 1;                               // flag 表示第 i 项的符号，初始为正
    denominator = 1;                        // denominator 表示第 i 项的分母，初始为 1
    item = 1.0;                             // item 中存放第 i 项的值，初值取 1
    sum = 0;                                // 置累加和的初值为 0
    while(fabs(item) >= e)                  // 当|item| ≥ e 时，执行循环
    {
        item = flag * 1.0 / denominator;    // 计算第 i 项的值
        sum = sum + item;                   // 累加第 i 项的值
        flag = -flag;                       // 改变符号，为下一次循环做准备
        denominator = denominator + 2;      // 分母递增 2，为下一次循环做准备
    }
    return  sum * 4;
}
```

3. 输入一个正整数 n，编写程序用递归方法按逆序输出整数 n。

【答案】参考代码：

```
#include<stdio.h>
```

```
int  main(void)
{
    int  num;
    void  fun(int  n);              //函数声明
    printf("Input  a  number:");
    scanf("%d", &num);
    printf("Output:");
    fun(num);                       //函数调用
    return  0;
}
void  fun(int  n)
{
    printf("%d", n%10);
    n=n/10;
    if(n>0)                         //递归出口，n<=0 时函数返回
        fun(n);                     //递归调用
}
```

4. 输入一个正整数 n(n≤10)，再输入 n 个整数，定义并调用 fun(a, n)函数，将数组中的整数按逆序重新存放。

【答案】参考代码：

```
#include<stdio.h>
int  main(void)
{
    int  a[10], n, i;
    void  fun(int  a[], int  n);          //函数声明
    printf("Input  n:");
    scanf("%d", &n);
    printf("Input  %d  integers:", n);
    for(i=0; i<n; i++)
        scanf("%d,", &a[i]);
    fun(a, n);                            //函数调用
    printf("Output:\n");
    for(i=0; i<n; i++)
        printf("%d", a[i]);
    return  0;
}
void  fun(int  a[], int  n)               //函数定义
{
    int  i, t;
    for(i=0; i<n/2; i++)
      {
```

```
            t=a[i];  a[i]=a[n-1-i];  a[n-1-i]=t;
        }
    }
```

5. 输入一个正整数 n(n≤10)，再输入 n 个整数，定义并调用 sort(a, n)函数，用冒泡法对 n 个数按从大到小排序。

【答案】参考代码：

```
#include<stdio.h>
int main()
{   int a[10];
    int i, n;
    void sort(int a[], int n);          //函数声明

    printf("Input n(1～10): ");
    scanf("%d", &n);
    //从键盘输入 n 个整数给数组赋值
    printf("Input %d numbers: ", n);
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    sort(a, n);                         //函数调用
    //输出排序后的结果
    printf("The sorted numbers:");
    for(i=0; i<n; i++)
        printf("%d ", a[i]);
    printf("\n");

    return 0;
}
void sort(int a[], int n)               //定义函数，冒泡法排序
{   int i, j, t;
    for(i=1; i<n; i++)
        for(j=0; j<n-i; j++)
            if(a[j]<a[j+1])
            {
                t=a[j];
                a[j]=a[j+1];
                a[j+1]=t;
            }
}
```

6. 输入一个 3 行 4 列的矩阵，定义并调用函数，求矩阵元素中的最大值。

【答案】参考代码：

```
#include<stdio.h>
int main()
```

```
{    int a[3][4];
     int i, j;
     int max(int a[][4]);                    //函数声明
     //从键盘输入 12 个整数给二维数组赋值
     printf("Enter 12 integers:\n");
     for(i=0; i<3; i++)
          for(j=0; j<4; j++)
               scanf("%d", &a[i][j]);
     //按矩阵的形式输出二维数组
     for(i=0; i<3; i++)
     {    for(j=0; j<4; j++)
               printf("%4d", a[i][j]);
          printf("\n");
     }
     printf("max=%d\n", max(a));      //函数调用，输出数组中最大值
     return 0;
}
int max(int a[][4])                   //函数定义
{    int i, j, m;
     m=a[0][0];                       //m 存放最大值，初值为 a[0][0]
     for(i=0; i<3; i++)
          for(j=0; j<4; j++)
               if(a[i][j]>m)
                    m=a[i][j];
     return m;                        //返回最大值
}
```

第 6 章　指　　针

1. 从键盘输入 5 个整数，按照从小到大的顺序排序，然后输出。排序用子函数实现。

【答案】参考代码：

```
#include<stdio.h>
int main(void)
{
     int i;
     int a[5];
     void fsort(int a[], int n);          //函数声明
```

```
    printf("请输入 5 个整数：");
    for (i=0; i<5; i++)
      scanf("%d", &a[i]);
    fsort(a, 5 );                          //函数调用
    for(i = 0; i < 5; i++)
      printf("%d ", a[i]);
    return  0;
}
void  fsort(int  a[], int  n)              //函数定义
{
    int  k, j, temp;
    for(k = 1; k < n; k++)                 //用冒泡法排序
    for(j = 0; j < n-k; j++)
        if(a[j] > a[j+1])
        {    temp = a[j];
             a[j] = a[j+1];
             a[j+1] = temp;
        }
}
```

2. 从键盘输入 5 个字符串，按照从小到大的顺序排序，然后输出。排序用子函数实现。

【答案】参考代码：

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int  main(void)
{
    int  i, n = 0;
    char  *pcolor[20], str[10];
    void  fsort(char  *color[], int  n);        //函数声明
    printf("请输入 5 个字符串，每行一个，#结束输入：\n");
    scanf("%s", str);
    while(str[0] != '#')
    {
       pcolor[n] = (char *)malloc(sizeof(char) * (strlen(str) + 1));
       //调用动态分配内存函数获得内存
       strcpy(pcolor[n], str);
       n++;
       scanf("%s", str);
     }
    fsort(pcolor, 5 );                          //函数调用
```

```
    for(i = 0; i < 5; i++)
        printf("%s ", pcolor[i]);
    return  0;
}
void  fsort(char  *color[], int  n)                    //函数定义
{
    int  k, j;
    char  *temp;
    for(k = 1; k < n; k++)                             //冒泡法排序
        for(j = 0; j < n-k; j++)
            if(strcmp(color[j], color[j+1])>0)         //比较两个字符串大小
              {
                    temp = color[j];
                    color[j] = color[j+1];
                    color[j+1] = temp;
              }
}
```

3. 输入一个字符串，再输入一个字符 ch，将字符串中包含的所有 ch 字符删除后输出该字符串。定义和调用 delchar(s, c)函数来实现，delchar(s, c)函数的功能是删除字符串 s 中出现的所有字符 c。

【答案】参考代码：

```
#include<stdio.h>
#include<string.h>
int  main(void)
{    char  str1[80], c;
     void  delchar(char  *s, char  c);              //函数声明
     printf("请输入一行字符：");
     scanf("%s", str1);
     printf("请输入一个要删除的字符：");
     getchar();                                     //吸收回车符
     c=getchar();                                   // c 存放要删除的字符
     delchar(str1, c);                              //函数调用
     printf("输出删除字符后的字符串：%s\n", str1);
}
void  delchar(char  *s,char  c)                     //函数定义
{
     int  i=0, j=0;
     while(s[i]!='\0')
     {
        if(s[i]==c)                 //找到相同的字符，后面的字符依次前移，即删除
            {    j=i;
```

```
                do{   s[j]=s[j+1];
                      j++;
                  }while(s[j]!='\0');
            }
        if(s[i]!=c)     i++;          //判断下一个是否仍然是所要删除的字符
    }
}
```

4. 判断输入的一串字符是否是回文。回文就是顺读和倒读都一样的字符串。

【答案】参考代码：

```
#include<stdio.h>
#include<string.h>
int  main(void)
{
    int  n;
    char  str[80], *p1, *p2;

    printf("请输入一行字符：");
    gets(str);
    n=strlen(str);
    p1=str;                          //p1 指向首字符
    p2=str+n-1;                      //p2 指向末字符

    while(p1<p2)
    {
        if(*p1!=*p2)                 //相应位置有不同字符即可断定不是回文
            break;
        else
            {   p1++;
                p2--;                // p1，p2 两个指针相向移动
            }
    }

    if(p1<p2)                        //若由 break 退出循环说明不是回文，否则说明是回文
        printf("No!\n");
    else
        printf("Yes!\n");
}
```

5. 输入一行文字，统计其中的大写字母、小写字母、空格、数字以及其他字符各有多少。

【答案】参考代码：

```
#include<stdio.h>
#include<string.h>
int  main(void)
```

```
{
    int  caplet=0, lowlet=0, digit=0, blank=0, other=0;
    char  *p, str[20];

    printf("请输入一行字符(20 个以内)：");
    gets(str);

    p=str;              //指针 p 指向第一个字符
    while(*p!='\0')
     {
        if((*p>='A')&&(*p<='Z'))
            caplet++;
        else  if((*p>='a')&&(*p<='z'))
            lowlet++;
        else  if((*p>='0')&&(*p<='9'))
            digit++;
        else  if(*p==' ')
            blank++;
        else
            other++;
        p++;
     }

    printf("有%d 个大写字母\n 有%d 个小写字母\n 有%d 个数字\n", caplet, lowlet, digit);
    printf("有%d 个空格\n 有%d 个其他字符\n", blank, other);
    return  0;
}
```

6. 将字符串 programming 赋给一个字符数组，然后从第一个字母开始间隔地输出该字符串，用指针编程实现。

【答案】参考代码(下述代码中采用更一般的输入方法给字符数组赋值)：

```
#include<stdio.h>
#include<string.h>
int  main(void)
{   char str1[20], str2[20];
    void  putstring(char  *p1, char  *p2);          //函数声明
    printf("请输入一行字符(20 个以内)：");
    scanf("%s", str1);
    putstring(str1, str2);                          //函数调用
    printf("输出字符串：%s\n", str2);
}
void  putstring(char  *p1, char  *p2)               //函数定义
{   int  i=0;
```

```
    while(*p1!='\0')
    {
        *p2=*p1;
        p1++;
        p2++;
        if(*p1=='\0')                          //判断第一个字符串是否结束
            break;
        else
            p1++;
    }
    *p2='\0';
}
```

7. 编写函数，将字符串中的第 m 个字符开始的全部字符复制成另一个字符串。要求在主函数中输入字符串及 m 的值并输出复制结果，在被调函数中完成复制。

【答案】参考代码：

```
#include<stdio.h>
#include<string.h>
int  main(void)
{   int  m;
    char  str1[20], str2[20];
    void  copystring(char  *p1, char  *p2, int  n);
    printf("请输入一行字符(20 个以内)：");
    scanf("%s", str1);
    printf("请输入要复制的开始位置 m：");
    scanf("%d", &m);
    if((int)strlen(str1) < m)
        printf("输入错误！");
    else
    {   copystring(str1, str2, m);
        printf("复制后字符串：%s\n", str2);
    }
    return  0;
}
void  copystring(char  *p1, char  *p2, int  n)
{
    int  i=0;
    while(i<n-1)              //p1 定位到第一个字符串的第 n 个字符
    {
        i++;
        p1++;
    }
```

```
    while(*p1!='\0')          //从第一个字符串第 n 个字符开始复制
    {
        *p2=*p1;
        p1++;
        p2++;
    }
    *p2='\0';
}
```

8. 编程实现将字符串 str1 的所有字符传送到字符串 str2 中，要求每传送 3 个字符后再存放一个空格，例如字符串 str1 为“abcdefg”，则字符串 str2 为“abc def g”。

【答案】参考代码：

```
#include<stdio.h>
#include<string.h>
int main(void)
{   char str1[20], str2[20];
    void sendstring(char *p1, char *p2);   //函数声明
    printf("请输入一行字符(20 个以内)：");
    scanf("%s", str1);
    sendstring(str1, str2);                //函数调用
    printf("传送后字符串：%s", str2);
    return 0;
}
void sendstring(char *p1, char *p2)
{
    int i=0;
    while(*p1!='\0')                       //对字符串 1 进行处理
    {
        *p2=*p1;
        p1++;
        p2++;
        i++;                               //对字符串 1 处理的字符计数
        if(i%3==0)                         //如果是 3 的倍数，字符串 2 加一个空格
        {
            *p2=' ';
            p2++;
        }
     }
    *p2='\0';              //对字符串 2 加字符串结束符
}
```

第 7 章　构造数据类型与编译预处理

1. 以下程序运行的结果是_____。

```
#define ADD(x) x+x
void main()
{
    int m=1, n=2, k=3;
    int sum=ADD(m+n)*k;
    printf("sum=%d", sum);
}
```

A. sum=9　　B. sum=10　　C. sum=12　　D. sum=18

【答案】B

2. 以下程序运行的结果是_____。

```
#define X 5
#define Y X+1
#define Z Y*X/2
void main()
{
    int a=Y;
    printf("%d, %d", Z, --a);
}
```

A. 7, 6　　B. 12, 6　　C. 12, 5　　D. 7, 5

【答案】D

3. 程序中头文件 type1.h 的内容是：

```
#define N 5
#define M1 N*3
```

程序如下：

```
#include"type1.h"
#define M2 N*2
void main()
{   int i;
    i=M1+M2;
    printf("%d\n", i);
}
```

编译后运行程序，输出结果是_____。

A. 10　　B. 20　　C. 25　　D. 30

【答案】C

4. 下列程序中定义学生结构体变量，存储学生的学号、姓名和 3 门课的成绩。fun()函数的功能是：对形参结构体变量 a 中的数据进行修改，把修改后的数据作为函数值返回主函数进行输出。例如，若传给形参 a 的数据中，学号、姓名和三门课的成绩依次是：10001，"ZhangSan"，95，80，88，修改后的数据应为：10002，"LiSi"，96，81，89。请在下划线处填入正确的内容，使程序得出正确结果。

```
#include<stdio.h>
#include<string.h>
struct student{
    long sno;
    char name[10];
    float score[3];
};
/**********found**********/
___1___ fun(struct student a)
{
    int i;
    a.sno = 10002;
    /**********found**********/
    strcpy(___2___, "LiSi");
    /**********found**********/
    for(i=0; i<3; i++)
        ___3___+= 1;
    return a;
}
void main()
{
    struct student s={10001, "ZhangSan", 95, 80, 88}, t;
    int i;
    printf("\n\nThe original data :\n");
    printf("\nNo: %ld Name: %s\nScores:  ", s.sno, s.name);
    for(i=0; i<3; i++)
        printf("%6.2f ", s.score[i]);
    printf("\n");
    t = fun(s);
    printf("\nThe data after modified :\n");
    printf("\nNo: %ld Name: %s\nScores:  ", t.sno, t.name);
    for(i=0; i<3; i++)
        printf("%6.2f ", t.score[i]);
    printf("\n");
}
```

【答案】1. struct student　　2. a.name　　3. a.score[i]

5. 学生的记录由学号和成绩组成，N 名学生的数据放入主函数中的结构体数组 s 中，请编写函数 fun()，其功能是：按分数降序排列学生的记录，高分在前，低分在后。

【答案】参考代码：

```
#include<stdio.h>
#define N 8
typedef struct
{   char num[10];
    int s;
}STREC;
void fun(STREC a[])
{

}
void main()
{
    STREC s[N]={{"GA003", 76}, {"GA002", 69}, {"GA007", 72}, {"GA008", 64},
                {"GA006", 87}, {"GA015", 85}, {"GA013", 91}, {"GA011", 66}};
    int i;
    fun(s);
    printf("The data after sorted :\n");
    for(i=0; i<N; i++)
        printf("%s  %4d\n", s[i].num, s[i].s);
}
```

6. 学生记录由学号、8 门课程成绩和平均分组成，学号和 8 门课程成绩在主函数中给出，请编写 fun()函数，其功能是：求出该学生的平均分，并放入记录的 ave 成员中。例如，学生的成绩是 85.5，76，69.5，85，91，72，64.5，87.5，则他的平均分应为 78.875。

参考代码：

```
#include<stdio.h>
#define N 8
struct student
{   char num[10];
    double s[N];
    double ave;
};
void fun(struct student *a)
{

}
void main()
{
    struct student s={"GA005", 85.5, 76, 69.5, 85, 91, 72, 64.5, 87.5};
```

```
    int  i;
    fun(&s);
    printf("The  %s's  student  data:\n", s.num);
    for(i=0; i<N; i++)
        printf("%4.1f\n", s.s[i]);
    printf("\nave=%7.3f\n", s.ave);
}
```

第8章　文　　件

1. 从键盘输入一个字符串，将其中小写字母全部转换成大写字母，然后输出到 test.txt 磁盘文件中保存。输入的字符串以“!”结束。

【答案】参考代码：

```
#include<stdio.h>
int  main(void)
{
    char  ch;
    FILE  *fp =fopen("text.txt", "w+");
    if(fp == NULL)
    {
        printf("打开文件失败！");
        return  -1;
    }
    while((ch = getchar()) != '!')
    {
        if(ch >= 'a' && ch <= 'z')
            ch -= 32;
        fputc(ch, fp);
    }
    fclose(fp);
    return 0;
}
```

2. 有 A.txt 和 B.txt 两个磁盘文件，各存放一行字母，要求把这两个文件的信息合并(按字母顺序排列)，输出到一个新的 C.txt 文件中。

【答案】参考代码：

```
#include<stdio.h>
#include<stdlib.h>
int  main(void)
```

```
{
    FILE *fp;
    int i, j, n, ni;
    char c[160], t, ch;
    if((fp=fopen("A.txt", "r"))==NULL)          //以读方式打开 A.txt 文件
    {
        printf("file A.txt cannot be opened\n");
        exit(0);
    }
    printf("\n A.txt contents are :\n");
    for(i=0; (ch=fgetc(fp))!=EOF; i++)          //把 A.txt 文件的内容读到数组 c 中
    {
        c[i]=ch;
        putchar(c[i]);
    }
    fclose(fp);
    ni=i;
    if((fp=fopen("B.txt", "r"))==NULL)          //以读方式打开 B.txt 文件
    {
        printf("file B.txt cannot be opened\n");
        exit(0);
    }
    printf("\n B.txt contents are :\n");
    //把 B.txt 文件的内容读到数组 c 中，接在 A.txt 文件的内容之后
    for(i=ni; (ch=fgetc(fp))!=EOF; i++)
    {
        c[i]=ch;
        putchar(c[i]);
    }
    fclose(fp);
    n=i;
    for(i=1; i<n-1; i++)                        //用冒泡法对数组 c 排序
        for(j=0; j<n-i; j++)
            if(c[j]>c[j+1])
            {
                t=c[j+1];
                c[j+1]=c[j];
                c[j]=t;
            }
    printf("\n C.txt file is:\n");
    fp=fopen("C.txt", "w");                     //以写方式打开 C.txt 文件
```

```
    for(i=0; i<n; i++)
        {
            fputc(c[i], fp);                    //把数组 c 内容写到 C.txt 文件中
            putchar(c[i]);
        }
    fclose(fp);
    return 0;
}
```

3. 有 5 个学生，每个学生有 3 门课程成绩，从键盘输入学生数据(包括学生号、姓名、3 门课程的成绩)，计算出每个学生的平均成绩，将原有数据和计算出的平均分数存放在 stu.txt 磁盘文件中。

【答案】参考代码：

```
#include<stdio.h>
#include<stdlib.h>
#define M 5
#define stu struct student
stu
{
    int num;
    char name[20];
    float s1, s2, s3, avg;
};
main()
{
    stu st[M];
    FILE *fp;
    int i;
    printf("输入 5 个学生的信息数据(学号 姓名 成绩 1 成绩 2 成绩 3)：\n");
    for(i=0; i<M; i++)
    {
        scanf("%d%s%f%f%f",&st[i].num, st[i].name, &st[i].s1, &st[i].s2, &st[i].s3);
        st[i].avg=(st[i].s1+st[i].s2+st[i].s3)/3;
    }
    if((fp=fopen("stu.txt", "w"))==NULL)        //以写方式打开 stu.txt 文件
    {
        printf("cannot open file\n");
    }
    //把 5 个学生的信息写入 stu.txt 文件
    for(i=0; i<M; i++)
    {
        fprintf(fp, "%d\t%s\t%f\t%f\t%f\t%f\n", st[i].num, st[i].name, st[i].s1, st[i].s2,
```

```
            st[i].s3, st[i].avg);
        }
        fclose(fp);                    //关闭文件
        fp=fopen("stu.txt", "r");      //再次以读方式打开 stu.txt 文件
        printf("输出每个学生的平均分：\n");
        printf("num\t  name\t  score1\t  score2\t  score3\t  average\n");
        for(i=0; i<M; i++)             //从 stu.txt 文件中读出学生信息，并输出到屏幕上
        {
            fscanf(fp, "%d\t%s\t%f\t%f\t%f\t%f\n", &st[i].num, st[i].name, &st[i].s1,
                &st[i].s2, &st[i].s3, &st[i].avg);
            printf("%d\t%s\t%f\t%f\t%f\t%f\n", st[i].num, st[i].name, st[i].s1,
                st[i].s2, st[i].s3, st[i].avg);
        }
        return 0;
    }
```

4. 函数调用语句“fseek(fp, -20L, 2);”的含义是_____。

A. 把文件位置指针移到距离文件头 20 个字节处

B. 把文件位置指针从当前位置向后移动 20 个字节

C. 把文件位置指针从文件末尾处向文件头方向移动 20 个字节

D. 把文件位置指针移到离当前位置 20 个字节处

【答案】C

5. 有以下程序：

```
#include<stdio.h>
main()
{
    FILE *fp1;
    fp1=fopen("f1.txt", "w");
    fprintf(fp1, "abc");
    fclose(fp1);
}
```

若f1.txt文本文件中原有内容为good，则运行程序后，f1.txt文件中的内容为_____。

A. goodabc　　B. abcd　　C. abc　　D. abcgood

【答案】C

6. 有以下程序：

```
#include<stdio.h>
main()
{
    FILE  *fp;
    int  k, n, a[6]={1, 2, 3, 4, 5, 6};
    fp=fopen("d2.dat", "w");
    fprintf(fp, "%d%d%d\n", a[0], a[1], a[2]);
```

```
        fprintf(fp,"%d%d%d\n", a[3], a[4], a[5]);
        fclose(fp);
        fp=fopen("d2.dat", "r");
        fscanf(fp, "%d%d", &k, &n);
        printf("%d %d\n", k, n);
        close(fp);
    }
```

运行程序后的输出结果是_____。

A. 1 2　　B. 1 4　　C. 123 4　　D. 123 456

【答案】D

7. 有以下程序：

```
#include<stdio.h>
main()
{
    FILE *fp;
    int i, k, n;
    fp=fopen("data.dat", "w+");
    for(i=1; i<6; i++)
      {
        fprintf(fp, "%d    ", i);
        if(i%3==0)
            fprintf(fp, "\n");
      }
    fclose(fp);
    fp=fopen("data.dat", "r");
    fscanf(fp,"%d%d", &k, &n);
    printf("%d %d\n", k, n);
    close(fp);
}
```

运行程序后的输出结果是_____。

A. 0 0　　B. 123 45　　C. 1 4　　D. 1 2

【答案】D

8. fp 是指向某文件的指针，且已读到此文件的末尾，则 feof(fp)函数的返回值是_____。

A. 0　　B. NULL　　C. EOF　　D. 非零值

【答案】D

9. 有以下程序：

```
#include<stdio.h>
main()
{
    FILE *fp;    int i;
    char ch[]="abcd", t;
```

```
    fp=fopen("abc.dat", "wb+");
    for(i=0; i<4; i++)
        fwrite(&ch[i], 1, 1, fp);
    fseek(fp, -2L, SEEK_END);
    fread(&t, 1, 1, fp);
    fclose(fp);
    printf("%c\n", t);
}
```

运行程序后的输出结果是_____。

A. d　　B. c　　C. b　　D. a

【答案】B

10. 设有定义“FILE *fw;”，请将以下打开文件的语句补充完整，以便可以向 readme.txt 文本文件的最后续写内容。

```
fw=fopen("readme.txt", _____);
```

【答案】"a"

11. 下面程序的功能是以二进制“写”方式打开 d1.dat 文件，写入 1～100 这 100 个整数后关闭文件；再以二进制“读”方式打开 d1.dat 文件，将这 100 个整数读入到另一个数组 b 中，并显示输出。请填空。

```
#include<stdio.h>
main()
{
    FILE  *fp;
    int  i, a[100], b[100];
    fp= fopen("d1.dat", "wb");
    for(i=0; i<100; i++)
        a[i]=i+1;
    fwrite(a, sizeof(int), 100, fp);
    fclose(fp);
    fp=fopen("d1.dat", _________);
    fread(b, sizeof(int), 100, fp);
    fclose(fp);
    for(i=0; i<100; i++)
        printf ("%d\n", b[i]);
}
```

【答案】"rb"

12. 已有 test.txt 文本文件，其内容为

Hello，everyone!

以下程序中，test.txt 文件已正确为“读”而打开，因此文件指针 fr 指向文件，则程序的输出结果是_____。

```
#include<stdio.h>
```

```
main()
{
    FILE  *fr;     char  str[40];
    ……
    fgets(str, 5, fr);
    printf("%s\n", str);
    fclose(fr);
}
```

【答案】Hell

13. 以下各项中，与函数 fseek(fp, 0L, SEEK_SET)有相同作用的是_____。

A. feof(fp)　　B. ftell(fp)　　C. fgetc(fp)　　D. rewind(fp)

【答案】D

实 验 报 告 (格式参考)

实验室：软件基础实验室　　2013 年 5 月 8 日　第 3-4 节

<table>
<tr><td>实验题目</td><td colspan="2">3.1 顺序结构实验</td><td>班级</td><td>2011 计算机 1 班</td></tr>
<tr><td rowspan="2">实验目的</td><td rowspan="2">1. 理解不同类型数据的输入输出方法
2. 掌握顺序结构的应用
3. 了解 VC6.0 环境下 C 语言的调试技巧</td><td>学号</td><td>20110102011105</td></tr>
<tr><td>姓名</td><td>张程员</td></tr>
<tr><td>实验基础与案例</td><td colspan="2">(1. 思考题；2. 知识点总结；3. 实践技能总结)
1. 一般的 C 程序包含三部分：输入数据、数据处理和输出。
2. 输入数据的格式要实用、友好、方便，应按设置好的格式进行输入，否则变量无法得到预计的值。
3. 输出的格式是按照程序使用方的需求设置的。数据处理按照功能要求和一般的公式进行计算处理
4. 输出中间变量(或结果)。当程序输出与预想的结果不同时，可以使用这个方法。</td><td>备注
%7.2f 格式输出 2 位小数。</td></tr>
<tr><td>实验内容</td><td colspan="2">(从 6 个方面进行记录　1. 题目；2. 题意分析与解题思路；3. 流程图；　4. 参考程序；5. 测试数据与实验结果；6. 调试过程遇到的问题与解决方法)
第一题：输入输出练习
int m,n,p;
scanf("m=%dn=%dp=%d",&m,&n,&p);
printf("%d%d%d\n",m,n,p);
要使得 m 为 123，n 为 456，p 为 789，正确输入为以下两种：
m=123 n=456 p=789 (回车)
m=123n=456p=789 (回车)
第二题(本报告示例省略)
第三题：顺序结构练习
1. 题目：设圆的半径为 r，求圆周长、圆面积。用 scanf 函数输入数据，输出计算结果，输出时要求有文字说明，取小数点后 2 位数字。编程并调试运行。
2. 分析与解题思路
1) 定义变量
2) 输入半径 r
3) 数据处理
(可续表)</td><td>心得与总结
1. 整数输入的结束一般用空格或回车，后一个输入方式也可以。</td></tr>
<tr><td>考核评价</td><td></td><td>得分</td><td></td><td>指导教师</td><td></td></tr>
</table>

（续表）

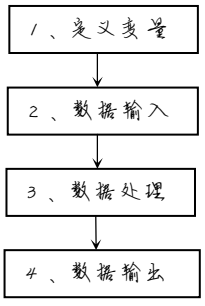
实验内容	（从 6 个方面进行记录　1. 题目；2. 题意分析与解题思路；3. 流程图；　4. 参考程序；5. 测试数据与实验结果；6. 调试过程遇到的问题与解决方法）	心得与总结

① 计算圆的周长

② 计算圆的面积

4）数据输出

3. 流程图



4. 参考代码

```
#include <stdio.h>
#define PI 3.14159
void main()
{     // 1. 定义变量
      double r,l,s;
      // 2. 数据输入r
      printf("请输入半径r（小数）");
        scanf("%lf",&r);

      // 3. 数据处理
      l = PI * 2 *r;
      s = 2 * PI * PI;

      // 4. 数据输出
      printf("圆周长为：l=%.2f\n",l);
      printf("圆面积为：s=%.2f\n",s);
}
```

5. 运行测试

第一组数据：
请输入圆半径r：1.5
圆周长为：l=9.42
圆面积为：s=7.07

第二组数据：
请输入圆半径r：2.6
圆周长为：l=16.34
圆面积为：s=21.24

第三组数据：
请输入圆半径r：3
圆周长为：l=18.85
圆面积为：s=28.27

6. 调试过程中遇到的问题

1. 严格按照顺序结构的思想把 1-4 这四步程序按顺序写好，颠倒顺序会出现错误，导致不能执行程序或输出的结果不正确。

2. 输入 double 型变量数据时，应使用%lf 格式，最初使用%f 编译时会有警告。忽略后，程序运行无论输入r值为多少，周长和面积输出都为 0.00。

第四题（本报告示例省略）

总结： 通过本次实验，我理解了一般程序都有变量、数据输入、数据处理和数据输出几个部分，要按顺序执行。语句顺序如果颠倒了，会引起程序本身语法错误，也有出现计算（算法）错误的情况。

流程图
矩形：过程
方向箭头：流程
菱形：判断（本次实验没用上）

在程序的注释中，我喜欢加上序号，对应每一个流程图中的矩形过程。

在输入 double 类型数据时，其格式应设置为%lf。如果设置格式为%f，程序有警告。不能录入半径r，如果r没有值，数据处理是无意义的。输出的结果也是错误的。

鸣　　谢

本书在编写过程中，以下同志参与了资料收集、文字录入及校对、图片收集与处理等方面的工作，在此一并表示感谢。

潘崇黎、崔荣博、刘雪锋、赵静、仇利克、刘伟、陈晓君、王艳丽、张永正、何娜、徐伶伶、郭玉芝、张德民、张欣、袁鹏、王朋朋、朱爱春、王学玲、葛苏慧、陈淑芳、王克进、吕志芳、吴伟、张蕾妮、王晓妍、孙月江、王莹莹