

上机实验与指导

第 1 章 上机实验的目的和要求

C 语言程序设计是一门理论与实践紧密结合的课程，在学习过程中既要注重 C 语言理论知识学习，又要突出上机实验环节，学生只有经过严格、规范的上机实验训练，才能将书本知识与应用实践结合在一起，掌握程序设计的基本方法，培养良好的分析与解决问题的逻辑思维能力，提高算法设计与独立编程的能力，达到“实践出真知”的目的。

1.1 上机实验的目的

学习 C 语言程序设计课程不能止步于“看懂了”，满足于能看懂书本上的语句、结构与程序，而应当熟练地掌握程序设计的全过程，要做到能独立分析问题，设计出合适的数据结构与算法，编写出源程序、上机调试程序、运行程序和分析结果。因此上机实验学习环节绝不仅仅是为了验证书本和教师讲课的内容，或者验证自己所编的程序正确与否。学习 C 语言程序设计，上机实验的目的包括以下几个方面。

(1) 加深对教师课堂教学内容的理解，C 语言中的语句、语法规则、程序结构等知识，光靠教师的课堂教学，当时可能一听就懂，但过后又难以记住，但它们是很重要的，初学者的程序出错往往错在语法上。只有通过反复的上机实验，才能自然而又熟练地掌握和运用这些知识。因此通过上机来掌握语法规则是行之有效的方法。

(2) 熟悉所用的计算机系统的操作方法，一是要了解计算机系统的软硬件配置情况，一个程序必须在一定的外部环境下才能运行，所谓“环境”，是指所用的计算机系统的硬件和软件条件，或者说工作平台。使用者应该了解为了运行一个 C 程序需要哪些必要的外部条件(例如硬件配置、软件配置)，可以利用哪些系统的功能来帮助自己开发程序。二是了解和熟悉 C 语言程序的集成开发环境(简称 IDE)，在 Windows 操作系统下的 C 语言 IDE 有多种，比较常用的有以下几种：一是微软开发的 Visual Studio(简称 VS)，它是 Windows 下的标准 IDE，实际开发中大家也都在使用；二是 Dev C++，这是一款 C/C++IDE，内嵌 GCC 的编译器(Linux GCC 编译器的 Windows 移植版)；三是 Visual C++ 6.0

(简称 VC 6.0)是微软开发的一款经典的 IDE,很多高校都以 VC 6.0 为教学工具来讲解 C 和 C++。每一种计算机系统的功能和操作方法不完全相同,但只要熟练掌握一两种计算机系统的使用,再遇到其他系统时便会触类旁通,很快就能学会。

(3) 学会上机调试程序,也就是要善于发现程序中的错误,并且能尽快排除这些错误,使程序能正确运行,得到所需的结果。经验丰富的人,在编译连接过程中出现“出错信息”时,一般能很快地判断出错误所在,并做出正确的修改。而初学者即使在明确的“出错提示”下也往往不知错在哪里,只好求助于别人。调试程序固然可以借鉴他人的现成经验,但更重要的是要通过自己的直接实践来累积经验,而且有些经验是只能“意会”难以“言传”的。别人的经验不能代替自己的经验,因此调试程序的能力是每个程序设计人员应当掌握的一项基本功。

(4) 培养良好的逻辑思维能力,程序的核心由数据和算法两部分组成,只有具备良好的逻辑思维能力,才能针对不同的问题与需求设计出恰当的数据结构和算法。上机实验是检验编程者设计的数据结构与算法正确与否、质量好坏的唯一途径。通过严格的上机实验与训练,才能达到培养学生科学思维,提高逻辑思维能力的目的。

1.2 上机实验前的准备工作

在高校的实验教学中,上机实验课大多以 2 个课时为一次上机时间,为了提高上机实验的效率,在上机实验前应做好如下准备工作。

(1) 了解学校机房的计算机设备型号、安装的操作系统、C 语言程序的开发环境,掌握系统的性能和使用方法。

(2) 做好实验所需知识的准备:复习相应章节的教学内容,掌握与实验内容有关的知识点、需要采用的程序结构及相关算法。

(3) 分析好实验题目,设计出相应算法,编写出上机所需的程序。上机实验教学配有指导教师,能及时解决学生实验过程中出现的问题,但由于每次上机课时有限,因此要珍惜实验时间,在上机前按实验题目要求编写好程序,并经人工检查无误后才能上机,以提高上机效率。初学者切忌不编程序或只抄书本的程序去上机实验,应从一开始就养成严谨的科学作风。

(4) 对运行中可能出现的问题事先做出估计,对程序中自己有疑问的地方,应做出记号,以便在上机时给予注意。

(5) 准备好调试和运行时所需的测试数据。

1.3 上机实验的步骤

(1) 开机运行 C 语言集成开发环境,进入 C 程序开发环境(例如 VC 6.0、Visual Studio 2015)。

(2) 输入自己编好的程序,保存到指定文件夹中。



(3) 检查确认已输入的程序是否有错(包括输入时打错的和编程中的错误), 如发现有错, 则及时改正。

(4) 进行编译。如果在编译和连接过程中发现错误, 输出窗口会出现“出错信息”, 根据提示找到出错位置和原因, 加以改正, 再进行编译, 如此反复, 直到顺利通过编译和连接为止。

(5) 运行程序, 并分析运行结果是否合理和正确, 在运行时应该输入多组不同的测试数据, 以检查程序是否完全正确, 若存在问题, 还需要重新修改程序, 直至结果完全正确为止。

(6) 输出程序清单和运行结果。

1.4 实验报告格式

1.4.1 实验报告格式

上机实验完成后, 应及时撰写实验报告, 实验报告的格式与内容应包括以下几方面。

(1) 实验名称、类型、实验时间

写出本次实验的名称, 如实验一: C 语言程序开发环境的使用。实验类型一般分为基础性实验(以给定数据验证给定程序是否正确为主)、设计性实验(以独立设计程序为主)、综合性设计实验(综合应用所学知识设计功能较多的程序)。

(2) 实验目的

根据要求写出上机实验所需要达到的目的。

(3) 实验内容与题目

简要写出本次实验的题目与要求。

(4) 实验过程与分析

针对每一道题目, 写出基本算法、程序代码、输入的测试数据和运行结果, 并判断程序与结果是否正确。

(5) 实验小结

根据实验过程与结果写出本次实验的小结和体会。

1.4.2 实验报告范例

一、实验名称

实验四: 选择结构程序设计

类型: 设计性实验

实验时间: 2018-03-08

二、实验目的

(1) 了解条件与程序流程的关系。

(2) 学会正确使用逻辑运算符和逻辑表达式。



(3) 熟练掌握 if 语句和 switch 语句，能正确编写选择结构的程序。

三、实验内容与题目

(1) 案例分析题：输入三个整数 a, b, c, 并找出最大值放于 max 中输出，分析给定的算法与代码是否正确。

(2) 编写一个程序，由键盘输入三个整数作为三角形的三条边，判断是否能构成一个三角形，如能并判断是等边三角形、等腰三角形、直角三角形，还是其他一般的三角形。

四、实验过程与分析

题目 1：求三个整数的最大值，给定的算法如下。

(1) 由题目可知，需定义 4 个整型变量 a, b, c, max。

(2) 通过键盘任意输入 3 个数值存入 a, b, c 中。

(3) 通过比较找出最大值存入 max 中。先比较 a, b, 把大数存入 max 中，可以用 if...else 形式语句；再比较 max 和 c, 把大数存入 max 中，用 if 形式语句。

(4) max 中存放的是最大值，输出 max。

通过分析，算法正确，能够求出三个整数的最大值。

程序代码如下。

```
#include<stdio.h>
void main( )
{ int a, b, c, max;          /* 定义四个整型变量 */
scanf("a=%d, b=%d, c=%d", &a, &b, &c);
if (a>=b)
max=a;                      /* a>=b */
else
    max=b;                  /* a<b */
if (c>max)
    max=c;                  /* c 是最大值 */
printf("\n 最大值 max=%d\n", max);
}
```

通过调试与运行，程序代码正确。

输入 a=5, b=4, c=9, 得到最大值 9。

输入 a=35, b=10, c=15, 得到最大值 35。

题目 2：给定三个整数，需要判定是否能构成一个三角形，若能构成，还需要判定三角形的形状。

通过对题目的分析，主要算法如下。

(1) 由题目可知，需定义 3 个整型变量 a, b, c 代表三条边，还需定义三个变量表示最大边、最小边及 a+b+c-h-g 的值，分别用 h, g, i 表示。

(2) 通过键盘任意输入 3 个数值存入 a, b, c 中。



(3) 通过比较找出最大值存入 h 中, 最小值存入 g 中, 第三条边值存入 i 中。

(4) 利用 `if` 语句进行三角形的判断, 若 `if((i+g>h)&&(h-i<g))` 的值为真, 则构成三角形, 再比较三条边的关系, 确定三角形的形状(等边、直角、普通)。

程序代码如下。

```
#include<stdio>
#include<cmath>
int main( )
{
    int a, b, c, g, h, i;
    printf("请输入三个整数 a, b, c:\n");
    scanf("%d%d%d", &a, &b, &c);
    g = (a<b?a:b);
    g = (g<c?g:c);           //找出最短边;
    h = a>b?a:b;
    h = h>c?h:c;             //找出最长边
    i = a+b+c - g - h;
    if((i+g>h)&&(h-i<g))
    {
        printf("这三个数能构成一个三角形,且");
        if(a==b&&b==c)
            printf("这是等边三角形\n");
        else if(a==b || b==c || c==a)
            printf("这是等腰三角形.\n");
        else if(h==sqrt(g*g+i*i))
            printf("这是直角三角形.\n");
        else
            printf("这是一般三角形.\n");
    }
    else
        printf("不能构成三角形.\n");
    return 0;
}
```

通过调试运行, 当输入 3, 4, 5 时, 运行结果如下: 这三个数构成三角形, 且这是直角三角形。当输入 2, 4, 8 时, 运行结果如下: 这三个数不能构成三角形。

五、实验小结

1. 通过实验, 分析了实现选择结构时所使用的的方法, 总结 `if...else` 语句中判断条件的构成方法和语句体的组成, 掌握了选择语句使用方法。

2. 通过实验, 较熟练掌握选择结构程序设计方法, 独立编写选择结构的程序。

第 2 章 实验一：C 语言程序开发环境的使用

2.1 实验目的

- (1) 熟悉使用 C 语言运行环境(VS 2015)，学会独立使用该系统。
- (2) 掌握 C 语言程序的书写格式和 C 语言程序的结构，初步了解 C 语言源程序的特点。
- (3) 掌握 C 语言上机步骤，了解在 VS 2015 上如何编辑、编译、连接和运行一个 C 语言程序方法。

2.2 实验内容与步骤

2.2.1 VS 2015 的基本操作

第一步：启动计算机，打开 VS 2015，进入主界面，如图 2-1 所示，然后点击“新建项目”，如图 2-2 所示。



图 2-1 启动 Visual Studio 2015 软件界面



图 2-2 新建项目

第二步：依次点击，模板—Visual C++ —Win 32，选择“Win 32 控制台应用程序”，项目名称可任意命名，完成后点击“确定”，如图 2-3 所示。

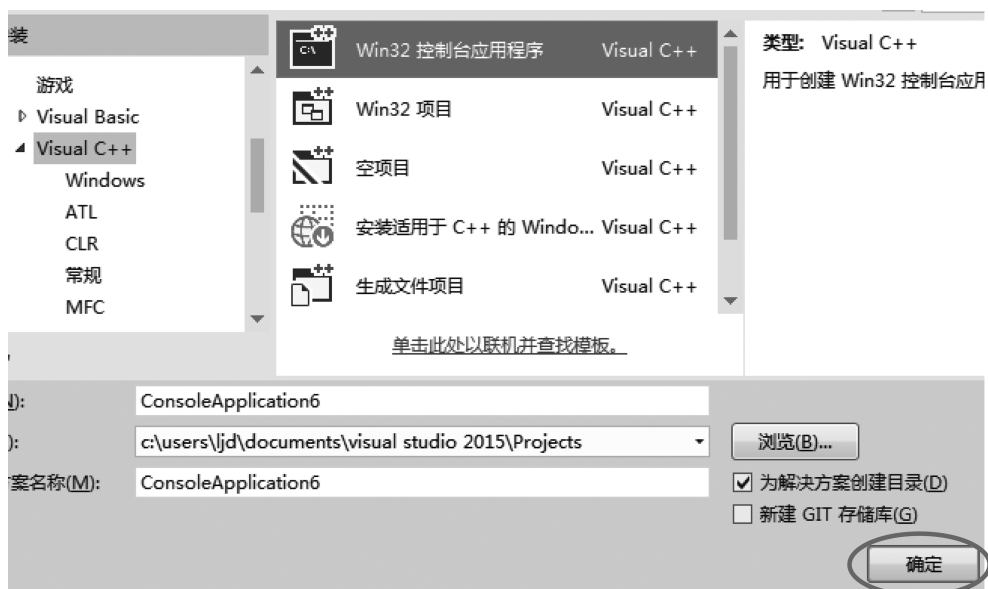


图 2-3 在 Visual Studio 2015 软件中新建工程

第三步：在出现的向导页面中，点击“下一步”，如图 2-4 所示。



欢迎使用 Win32 应用程序向导

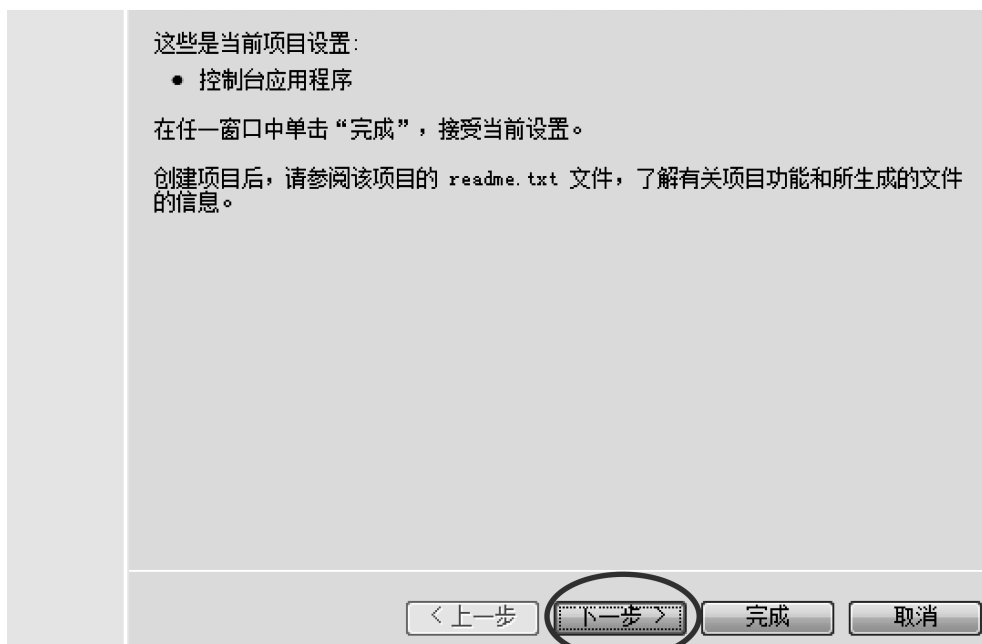


图 2-4 使用 Win 32 应用程序向导

第四步：在新弹出的页面中，选定“空项目”，然后点击“完成”，如图 2-5 所示。

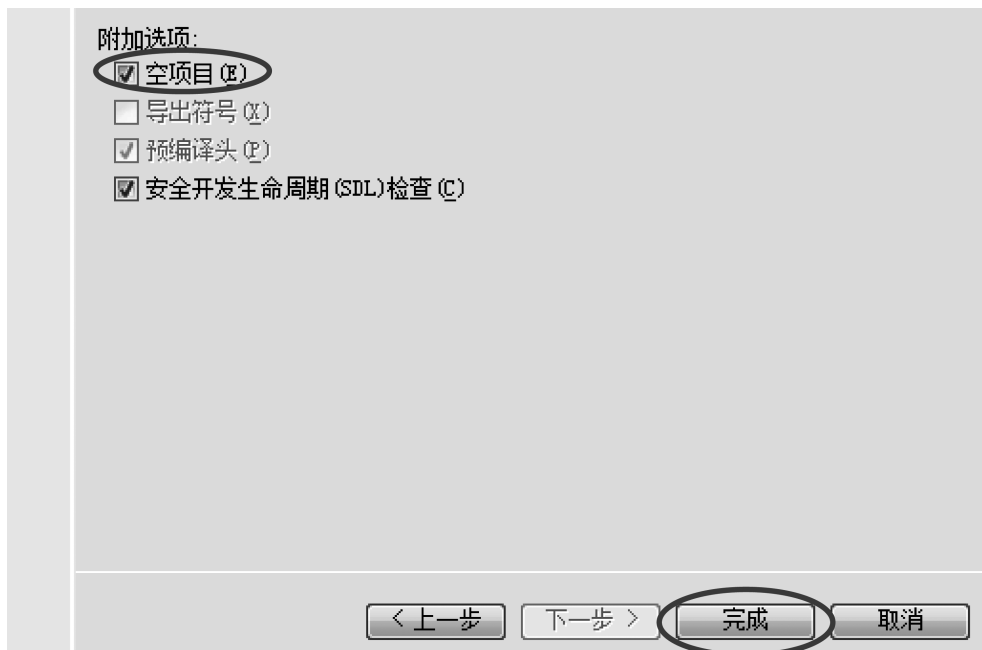


图 2-5 在 Visual Studio 2015 软件中勾选空项目



第五步：在出现的新建项目的页面中，点击“解决方案资源管理器”，找到“源文件”并右键点击，依次选择“添加—新建项”，如图 2-6 所示。

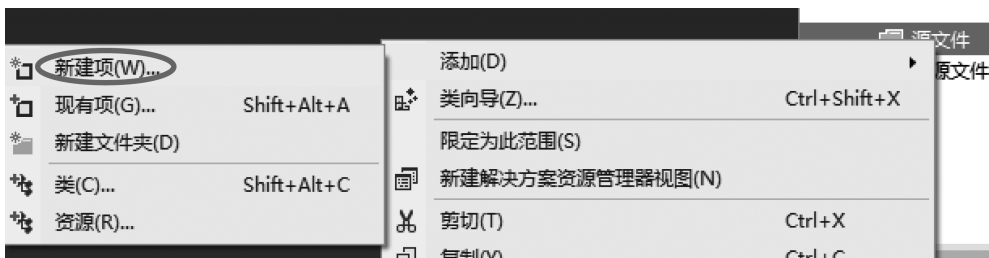


图 2-6 在 Visual Studio 2015 软件中添加新建项

第六步：在弹出的页面中，命名加上后缀“.c”，如：“源.c”。接着点击“添加”，C 文件建立完成，如图 2-7 所示。



图 2-7 在 Visual Studio 2015 软件中命名 C 文件

2.2.2 程序的运行和调试

第一步：在编辑窗口中编写好程序。

第二步：在菜单栏中找到“调试”按钮，点击“调试”之后，会跳出一个插条，点击“进行调试”将源程序编译成目标程序，编译通过后，最后再次点击“调试”里面的“执行调试”，当点击“执行调试”之后，VS 2015 就会自动运行你所输入的程序，如果程序运行正常，就可得到程序运行结果。



2.3 案例分析

计算由键盘输入的任意两个整数 num1 和 num2 的乘积。运行如下程序，找出错误，并进行调试。

```
#include<stdio. h>
void main( )           //主函数
{ int  num1, num2, p ;  //定义变量
scanf("%d, %", &x, &y) ; //键盘输入两个整数,注意格式
p=prodct(num1, num2);  //调用函数 prodct 计算两个数 num1 与 num2 的
乘积
printf(" The product is %d", p); //输出结果
}
int prodct(int a, int b ) //定义求两个数乘积的函数
{ int c;
  c=a*b;
return c;
}
```

图 2-8 为编译时出现错误的截图。

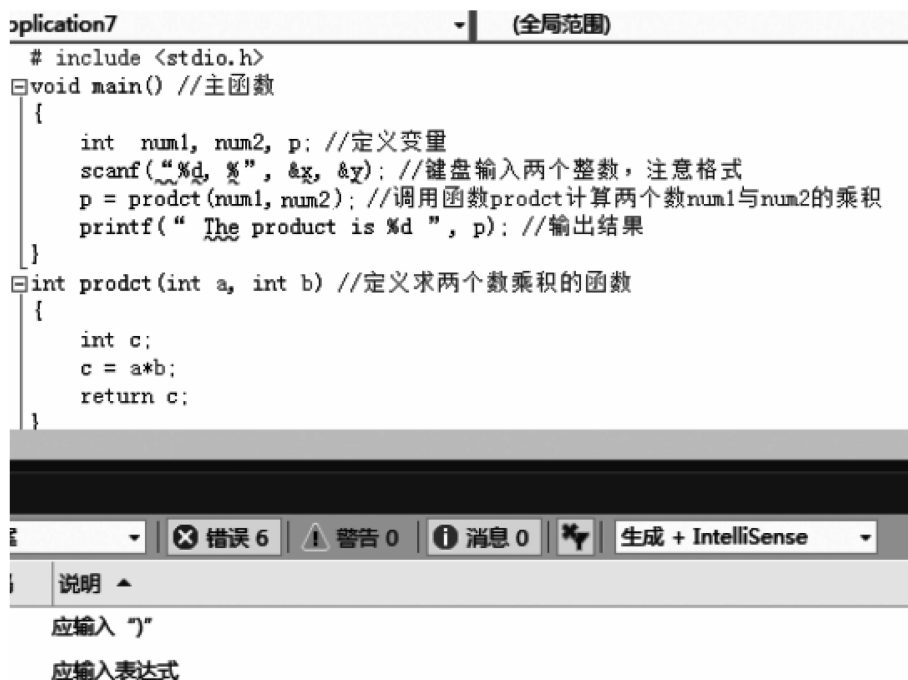


图 2-8 编译时出现错误的截图



- 分析错误：
1. 各符号必须在英文状态下输入，如：双引号、逗号、分号等；
 2. 键盘输入两个整数，应注意格式，应为%d；
 3. 应先定义变量 x , y ，然后再使用；
 4. 大括号是成对出现的，最后少}；
 5. 在使用 VS 2015 版时，需要将 scanf 改为 scanf_s；
 6. 在使用变量时，部分变量应该进行初始化。

经调试后，编译与运行图如图 2-9 所示。

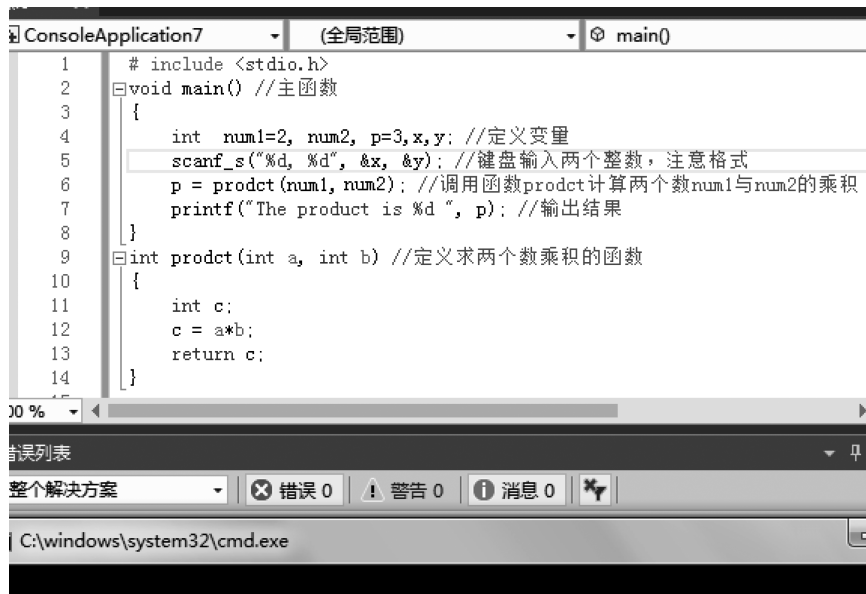


图 2-9 经调试后编译与运行截图

2.4 程序调试

1. 求 $1+2+3+\dots+100$ 的和。

```
# include<stdio. h>
min( )           //主函数
{ int  i, sum=0;
    for(i=1; i<= 100; i++)  sum=sum+i;
    printf("sum = %d \n", sum);
}
```

2. 输出一个数字三角形。

```
# include<stdio. h>
void main ( )
{ int  i, j ;
    for ( i=1; i<=5; i++)
```



```
{ for ( j=0; j<=10-2* i; j++)
    printf(" ");          /* 输出一个空格 */
    for ( j=0; j<=i; j++)
        printf("%4d", i);    // 输出一行数字
    printf("\n");          // 换行
} }
```

3. 程序调试成功的参考程序代码。

C 程序一修改为

```
# include<stdio. h>
void main( ) //主函数
{
    int i, sum=0;
    for (i=1; i<= 100; i++) sum =sum+i;
    printf("sum = %d\n", sum);
}
```

C 程序二修改为

```
# include<stdio. h>
void main( )
{
    int i, j;
    for (i=1; i<= 5; i++)
    {
        for (j=0; j<= 10-2*i; j++)
            printf(" ");          /* 输出一个空格 */
        for (j=0; j<=i; j++)
            printf("%4d", i);    // 输出一行数字
        printf("\n");          // 换行
    }
}
```

2.5 分析与讨论

(1) 记下在调试过程中所发现的错误、系统给出的出错信息和对策, 分析讨论对策成功或失败的原因。

(2) 总结 C 语言程序结构和书写规则。

第3章 实验二：数据类型、运算符和表达式的应用

3.1 实验目的

- (1) 了解 C 语言中数据类型的意义，掌握变量与常量的定义方法。
- (2) 掌握不同类型数据之间的赋值规律。
- (3) 掌握 C 语言中各种运算符的作用、优先级和结合性，能正确书写各种表达式。
- (4) 掌握不同类型数据运算时数据类型的转换规则。
- (5) 进一步熟悉 C 语言程序的编辑环境。

3.2 案例分析

1. 下面的程序试图计算由键盘输入的任意两个整数的平均值。

```
#include<stdio. h>
void main( )
{ int x, y, a;
  scanf("%x, %y", &x, &y);
  a=(x+y)/2;
  printf(" The average is%d \n", a);
}
```

调试无语法错误后，分别使用下列几组数据对上述程序进行测试，分析程序运行结果。

- (1) 2, 6 (2) 3, 8 (3) -4, -6
(4) 32800, 33000 (5) -32800, 33000

答：程序修改如下，如图 3-1 所示。

```
#include<stdio. h>
void main( )
{
  int x, y, a;
  printf("please input x, y\n ");
  scanf_s("%d, %d", &x, &y);
  a=(x+y) / 2;
  printf("The average is%d \n ", a);
}
```

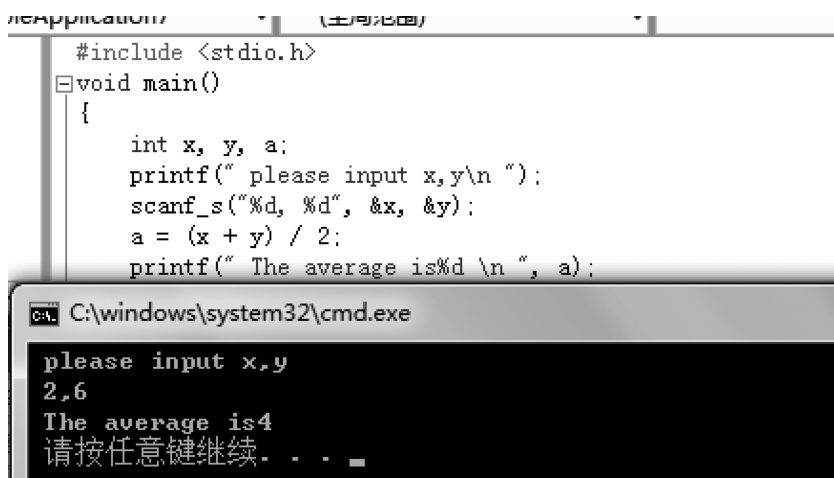


图 3-1 程序修改后运行结果

2. 输入以下程序。

```

#include<stdio. h>
void main( )
{
    int j, k, m, n;
    k=8;
    j=10;
    m=k++;
    n=++j;
    printf("%d, %d, %d, %d", k, j, m, n);
}
    
```

(1) 运行程序，观察结果。

(2) 分析++在变量之前和在变量之后的不同点？

答：(1)经运行程序，可看到如下结果，如图 3-2 所示；

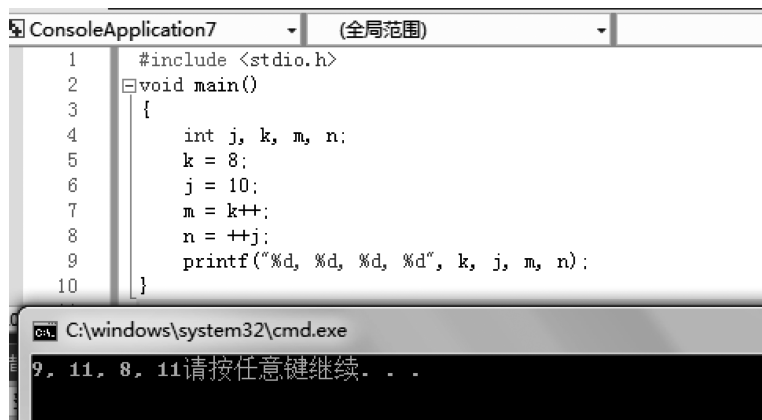


图 3-2 程序运行结果



(2) 由上述运行结果可知, ++在变量之前是先进行加 1 后运行, ++在变量之后是先运行后进行加 1。

3.3 自主编程

1. 将“China”译成密码。密码规律: 用原来的字母后面第 4 个字母代替原来的字母。例如, 字母“A”后面第 4 个字母是“E”, 用“E”代替“A”。因此, “China”应译为“Glmre”。请编一程序, 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”。经过运算, 使 c1、c2、c3、c4、c5 分别变为“G”、“l”、“m”、“r”、“e”, 并输出。

2. 编写程序, 使其能实现求 3 个数中的最大数。

3. 输入一个华氏温度, 要求输出摄氏温度。公式为 $C = (F - 32) * 5/9$, 结果保留两位小数。

3.4 自主编程参考程序

(1) 程序如下。

```
#include<stdio. h>
void main( )
{
    char c1='C', c2='h', c3='i', c4='n', c5='a';
    c1+=4;
    c2+=4;
    c3+=4;
    c4+=4;
    c5+=4;
    printf("password is %c%c%c%c%c\n", c1, c2, c3, c4, c5);
}
```

(2) 程序如下。

```
#include<stdio. h>
void main( )
{
    int a, b, c, max;
    printf("please input three numbers : \n ");
    scanf_s("%d, %d, %d\n", &a, &b, &c);
    max=a>b? a:b;
    max=max>c? max:c;
    printf("max=%d\n", max);
}
```



(3) 程序如下。

```
#include<stdio. h>
void main( )
{
    float c, f;
    printf("请输入一个华氏温度:");
    scanf_s("%f", &f);
    c = (5. 0/9. 0)*(f- 32);
    printf("摄氏温度为:%5. 2f\n", c);
}
```

3.5 分析与讨论

如何正确地选用数据类型，总结运算符的优先级？

第 4 章 实验三：顺序结构程序设计

4.1 实验目的

- (1) 掌握 C 语言中赋值语句的使用方法。
- (2) 掌握各种类型数据的输入、输出的方法，能正确使用各种格式转换符。
- (3) 学会编写顺序结构的 C 语言程序。
- (4) 进一步掌握编写程序和调试程序的方法。

4.2 实例案例分析

4.2.1 案例一

由终端输入两个整数给变量 x 和 y；然后输出 x 和 y；在交换 x 和 y 中的值后，再输出 x 和 y。验证两个变量中的数是否正确地进行了交换。

1. 算法分析

(1) 变量是存放数据的容器，定义两个整型数据变量 x 和 y，用于存放由终端输入的数据。定义一个整型变量 t，用于临时存放数据，实现 x 和 y 的数据交换。

(2) 用 scanf 函数实现终端输入数据。

(3) 用 printf 函数实现数据输出。

2. 程序代码

```
#include<stdio. h>
void main( )
{
    int x, y, t;
    printf("please input x, y: \n");
    scanf("%d%d", &x, &y);           //由终端输入数据给变量 x, y
    printf("(1) x=%d, y=%d \n", x, y); //输出 x, y
    //以下语句实现交换 x 和 y 数据
    t=x;
    x=y;
    y=t;
    printf("(2) x=%d, y=%d \n", x, y); //输出 x, y
}
```



3. 程序调试和运行

- (1) 编译(Ctrl+F7)。
- (2) 连接(F7)。
- (3) 运行(Ctrl+F5)，运行结果如图 4-1 所示。

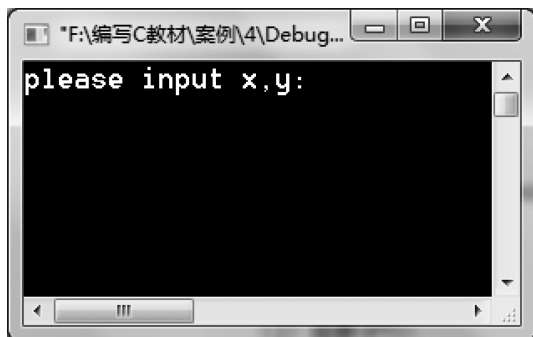


图 4-1 运行窗口

- (4) 在图 4-1 中输入 3 5，运行结果如图 4-2 所示。

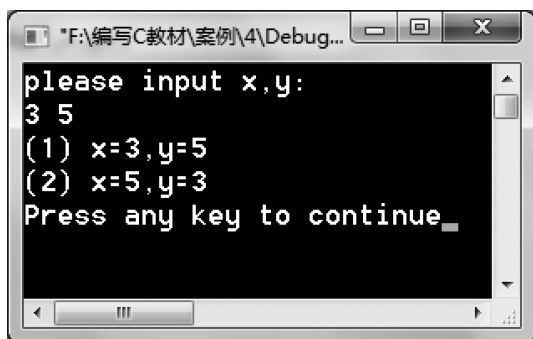


图 4-2 运行结果窗口

4.2.2 案例二

设圆半径 $r=1.5$ ，圆柱高 $h=3$ ，求圆周长、圆面积、圆柱表面积、圆柱体积，输出计算结果，取小数点后 2 位数字。

1. 算法分析

- (1) 定义变量数据类型及名称：圆半径 r ，圆柱高 h ，圆周长 c ，圆面积 s ，圆柱表面积 sb ，圆柱体积 v 。
- (2) 练习赋值语句。给圆半径 r ，圆柱高 h 赋值。
- (3) 计算所需求的各值。
- (4) 运用 `printf` 函数中的输出格式控制方法输出所求出的各个结果。
- (5) 调试运行。

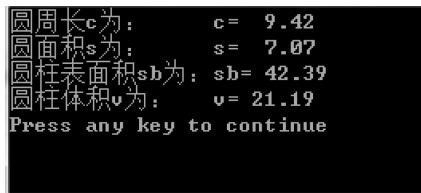


2. 程序代码

```
#include<stdio. h>
int main( )
{
    int h=3;
    float r, c, s, sb, v, pi;
    r=1. 5;
    pi=3. 14;
    c=2*pi*r;
    s=pi*r*r;
    sb=2*s+c*h;
    v=s*h;
    printf("圆周长 c 为:      c=%6. 2f\n", c);
    printf("圆面积 s 为:      s=%6. 2f\n", s);
    printf("圆柱表面积 sb 为: sb=%6. 2f\n", sb);
    printf("圆柱体积 v 为:    v=%6. 2f\n", v);
    return 0;
}
```

3. 程序调试和运行

- (1) 编译(Ctrl+F7)。
- (2) 连接(F7)。
- (3) 运行(Ctrl+F5)，运行结果如图 4-3 所示。



```
圆周长c为:      c=  9.42
圆面积s为:      s=  7.07
圆柱表面积sb为: sb= 42.39
圆柱体积v为:    v= 21.19
Press any key to continue
```

图 4-3 运行结果

4. 分析与讨论

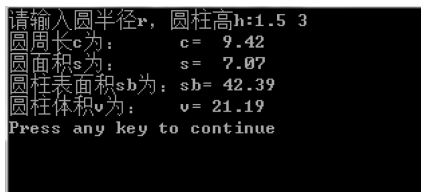
改用 scanf 输入 r, h 等数据, 并将圆周率定义为符号常量, 重新调试运行程序。参考程序代码如下。

```
#include<stdio. h>
#define PI 3. 14
int main( )
{
    float r, h, c, s, sb, v;
    printf("请输入圆半径 r, 圆柱高 h:");
```



```
scanf("%f%f", &r, &h);  
c=2*PI*r;  
s=PI*r*r;  
sb=2*s+c*h;  
v=s*h;  
printf("圆周长 c 为:    c=%6.2f\n", c);  
printf("圆面积 s 为:    s=%6.2f\n", s);  
printf("圆柱表面积 sb 为: sb=%6.2f\n", sb);  
printf("圆柱体积 v 为:   v=%6.2f\n", v);  
return 0;  
}
```

程序运行结果如图 4-4 所示。



```
请输入圆半径r, 圆柱高h:1.5 3  
圆周长c为:    c=  9.42  
圆面积s为:    s=  7.07  
圆柱表面积sb为: sb= 42.39  
圆柱体积v为:   v= 21.19  
Press any key to continue
```

图 4-4 运行结果

4.2.3 案例三

输入一个三角形的三条边长，求出三角形的面积。

1. 算法分析

按照输入数据(三条边长)、计算公式、输出结果的顺序进行。其中三角形的面积公式为

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

其中， $s = (a+b+c)/2$

2. 程序代码

```
#include<stdio.h>  
#include<math.h>  
void main( )  
{  
    int a, b, c;  
    float s, area;  
    printf("请输入三角形的三条边:");  
    scanf("%d%d%d", &a, &b, &c);  
    s=1.0/2*(a+b+c);
```



```

    area=sqrt(s*(s-a)*(s-b)*(s-c));
    printf("三角形的面积=%8.3f\n", area);
}

```

3. 程序运行和调试

经编译(Ctrl+F7)、连接(F7)、运行(Ctrl+F5)后,通过键盘输入三角形的三条边长 4, 5, 6 后,运行结果如图 4-5 所示。

```

请输入三角形的三条边: 4 5 6
三角形的面积= 9.922
Press any key to continue

```

图 4-5 运行结果

4.3 自主编程题

1. 用 getchar 函数读入两个字符给 c1、c2,然后分别用 putchar 函数和 printf 函数输出这两个字符,并思考以下问题。

- (1) 变量 c1、c2 应定义为字符型还是整型,或者二者皆可?
- (2) 要求输出 c1 和 c2 值的 ASCII 码,应如何处理? 用 putchar 函数还是 printf 函数?
- (3) 整型变量与字符变量是否在任何情况下都可以互相代替? 如:

char c1, c2; 与 int c1, c2;

是否无条件地等价?

(1) 参考程序如下。

```

#include<stdio.h>
int main( )
{
    char c1, c2;
    printf("请输入两个字符 c1, c2:");
    c1 = getchar( );
    c2 = getchar( );
    printf("用 putchar 语句输出结果为:");
    putchar(c1);
    putchar(c2);
    printf("\n 用 printf 语句输出结果为:");
    printf("%c %c\n", c1, c2);
    return 0;
}

```

运行结果如图 4-6 所示。



```
请输入两个字符c1,c2:xy
用 putchar 语句输出结果为: xy
用 printf 语句输出结果为: x y
Press any key to continue
```

图 4-6 运行结果

(2) 分析与讨论。

程序 1

```
#include<stdio. h>
int main( )
{
    int c1, c2;
    printf("请输入两个整数 c1, c2:");
    scanf("%d, %d", &c1, &c2);
    printf("按字符输出结果:");
    printf("%c, %c\n", c1, c2);
    printf("按 ASCII 码输出结果为:");
    printf("%d, %d\n", c1, c2);
    return 0;
}
```

运行结果如图 4-7 所示。

```
请输入两个整数c1,c2:97,98
按字符输出结果: a,b
按ASCII码输出结果为: 97,98
Press any key to continue
```

图 4-7 运行结果

程序 2

```
#include<stdio. h>
int main( )
{
    char c1, c2;
    int i1, i2;
    printf("请输入两个字符 c1, c2:");
    scanf("%c, %c", &c1, &c2);
    i1 = c1;
    i2 = c2;
    printf("按字符输出结果:");
    printf("%c, %c\n", i1, i2);
}
```



```
printf("按整数输出结果:");
printf("%d, %d\n", c1, c2);
return 0;
}
```

运行结果如图 4-8 所示。

```
请输入两个字符c1,c2:a,b
按字符输出结果: a,b
按整数输出结果: 97,98
Press any key to continue
```

图 4-8 运行结果

程序 3

```
#include<stdio. h>
int main( )
{
    char c1, c2;
    int i1, i2;
    printf("请输入两个整数 i1, i2:");
    scanf("%d, %d", &i1, &i2);
    c1 = i1;
    c2 = i2;
    printf("按字符输出结果:");
    printf("%c, %c\n", c1, c2);
    printf("按整数输出结果:");
    printf("%d, %d\n", c1, c2);
    return 0;
}
```

若输入 97, 98, 则运行结果如图 4-9 所示。

```
请输入两个整数i1,i2:97,98
按字符输出结果: a,b
按整数输出结果: 97,98
Press any key to continue
```

图 4-9 运行结果

同样的程序 3, 若输入 289, 330, 则运行结果如图 4-10 所示。

```
请输入两个整数i1,i2:289,330
按字符输出结果: ?,J
按整数输出结果: 33,74
Press any key to continue
```

图 4-10 运行结果

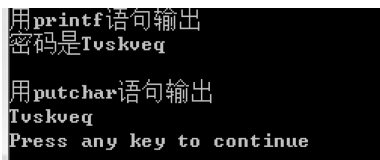
请同学们认真分析原因。注意：字符变量在计算机内存中占 1 个字节，而整型变量占 2 个或 4 个字节。因此整型变量在可输出字符的范围内是可以与字符数据互相转换的。如果整数在此范围外，不能代替。

2. 请编写程序将“Program”译成密码，密码规律是：用原来的字母后面第 4 个字母代替原来的字母。例如，字母“A”后面第 4 个字母是“E”，用“E”代替“A”。因此，“Program”应译为“Tvskveq”。请编一程序，用赋初值的方法使 c1=‘P’，c2=‘r’，c3=‘o’，c4=‘g’，c5=‘r’，c6=‘a’，c7=‘m’，经过运算，使 c1，c2，c3，c4，c5，c6，c7 分别变为‘T’，‘v’，‘s’，‘k’，‘v’，‘e’，‘q’。分别用 putchar 函数和 printf 函数输出这 7 个字符。

参考程序代码如下。

```
#include<stdio. h>
int main( )
{
    char c1=' P' , c2=' r' , c3=' o' , c4=' g' , c5=' r' , c6=' a' , c7=' m' ;
    c1=c1+4;
    c2=c2+4;
    c3=c3+4;
    c4=c4+4;
    c5=c5+4;
    c6=c6+4;
    c7=c7+4;
    printf("用 printf 语句输出 \n");
    printf("密码是%c%c%c%c%c%c%c\n", c1, c2, c3, c4, c5, c6, c7);
    printf("\n 用 putchar 语句输出 \n");
    putchar(c1);
    putchar(c2);
    putchar(c3);
    putchar(c4);
    putchar(c5);
    putchar(c6);
    putchar(c7);
    putchar(' \n' );
}
```

运行结果如图 4-11 所示。



```
用printf语句输出
密码是Tvskveq

用putchar语句输出
Tvskveq
Press any key to continue
```

图 4-11 运行结果



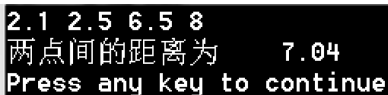
3. 输入平面上两个点的坐标(double 类型), 计算两个点之间的距离。请将下面的程序填写完整。

```
#include<stdio. h>
#include<math. h>
void main( )
{ float a, b, c, d, e, f, g;
  scanf(_____);
  e=(a- c);
  f=(b- d);
  g=sqrt(e*e+f*f);
  printf(_____);
}
```

参考程序代码如下。

```
#include<stdio. h>
#include<math. h>
void main( )
{ float a, b, c, d, e, f, g;
  scanf("%f%f%f%f", &a, &b, &c, &d);
  e=(a- c);
  f=(b- d);
  g=sqrt(e*e+f*f);
  printf("两点间的距离为%8. 2f\n", g);
}
```

运行结果如图 4-12 所示。



```
2.1 2.5 6.5 8
两点间的距离为      7.04
Press any key to continue
```

图 4-12 运行结果

4.4 实验总结

1. 通过实验调试运行程序, 观察程序运行结果, 掌握变量定义、赋值语句、输出数据的格式控制等知识。
2. 总结在 scanf 、 printf 函数中可以使用的各种格式指定符, 并给出样例。
3. 总结在 printf 函数中可以使用的各种转义字符及其功能。

第 5 章 实验四：选择结构程序设计

5.1 实验目的

- (1) 学会使用流程图，了解条件与程序流程的关系。
- (2) 学会正确使用逻辑运算符和逻辑表达式。
- (3) 熟练掌握 if 语句和 switch 语句，能正确编写选择结构的程序。

5.2 实例案例分析

5.2.1 案例一

输入三个整数 a, b, c, 并找出最大值放于 max 中输出。

1. 算法分析

- (1) 由题目可知，需定义 4 个整型变量 a, b, c, max。
- (2) 通过键盘任意输入 3 个数值存入 a, b, c 中。
- (3) 通过比较找出最大值存入 max 中。先比较 a, b, 把大数存入 max 中，可以用 if...else 形式语句；再比较 max 和 c, 把大数存入 max 中，用 if 形式语句。
- (4) max 中存放的是最大值，输出 max。

2. 程序代码

```
#include<stdio. h>
main( )
{
int a, b, c, max;          /*定义四个整型变量 */
scanf("a = %d, b = %d, c = %d", &a, &b, &c);
if (a>=b)
    max = a;                /* a>= b */
else
    max = b;                /* a<b */
if (c>max)
    max = c;                /* c 是最大值 */
printf("\n 最大值 max = %d \n", max);
}
```



3. 程序运行和调试

调试运行程序，输入 $a=5, b=4, c=9$ <回车>，运行结果如图 5-1 所示。

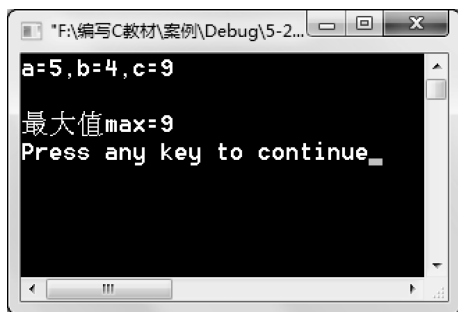


图 5-1 运行结果

5.2.2 案例二

有 4 个圆塔，圆心分别为 $(2, 2)$ ， $(-2, 2)$ ， $(-2, -2)$ ， $(2, -2)$ ，圆半径为 1，如图 5-2 所示。这 4 个塔的高度为 10 m，塔以外无建筑物。今输入任一点的坐标，求该点的建筑高度(塔外的高度为零)。

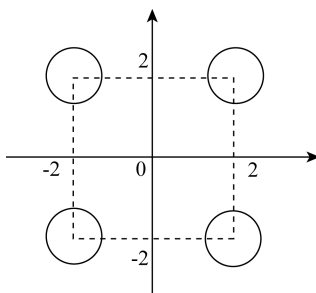


图 5-2 圆塔位置图

1. 算法分析

根据题意，流程图如图 5-3 所示。

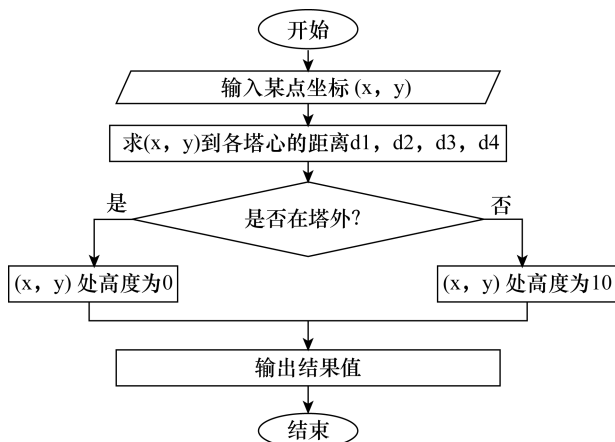


图 5-3 求某点的建筑高度流程图



2. 程序代码

```
#include<stdio. h>
int main( )
{
    int h=10;
    float x1=2, y1=2, x2=-2, y2=2, x3=-2, y3=-2, x4=2, y4=-2, x, y, d1, d2, d3, d4;
    printf("请输入一个点(x, y):");
    scanf("%f, %f", &x, &y);
    d1=(x-x1)*(x-x1)+(y-y1)*(y-y1);
    d2=(x-x2)*(x-x2)+(y-y2)*(y-y2);
    d3=(x-x3)*(x-x3)+(y-y3)*(y-y3);
    d4=(x-x4)*(x-x4)+(y-y4)*(y-y4);
    if(d1>1 && d2>1 && d3>1 && d4>1) h=0;
    printf("该点高度为%d\n", h);
    return 0;
}
```

3. 程序运行和调试

(1) 经编译、连接、运行后，屏幕提示“请输入一个点(x, y):”，当输入 1.1, 0.5 之后，得运行结果如图 5-4 所示。

```
请输入一个点 (x,y):1.1,0.5
该点高度为0
Press any key to continue
```

图 5-4 运行结果

(2) 再次编译、连接、运行后，屏幕提示“请输入一个点(x, y):”，当输入 2.5, 2.8 之后，得运行结果如图 5-5 所示。

```
请输入一个点 (x,y):2.5,2.8
该点高度为10
Press any key to continue
```

图 5-5 运行结果

5.2.3 案例三

编写程序，判断某一年是否为闰年。

1. 算法分析

(1) 首先要知道判别某一年为闰年的条件。闰年的条件是符合下面二者之一：①能被 4 整除，但不能被 100 整除，如 2008；②能被 400 整除，如 2000。因此，可得判别闰年的



逻辑表达式。

$(year \% 4 == 0 \ \&\& \ year \% 100 != 0) \ || \ year \% 400 == 0$

year 为整数年份, 如果上述表达式的值为真, 则 year 为闰年; 否则 year 不是闰年。

(2) 根据闰年判别方法, 可以画出判别闰年算法的流程图, 如图 5-6 所示。

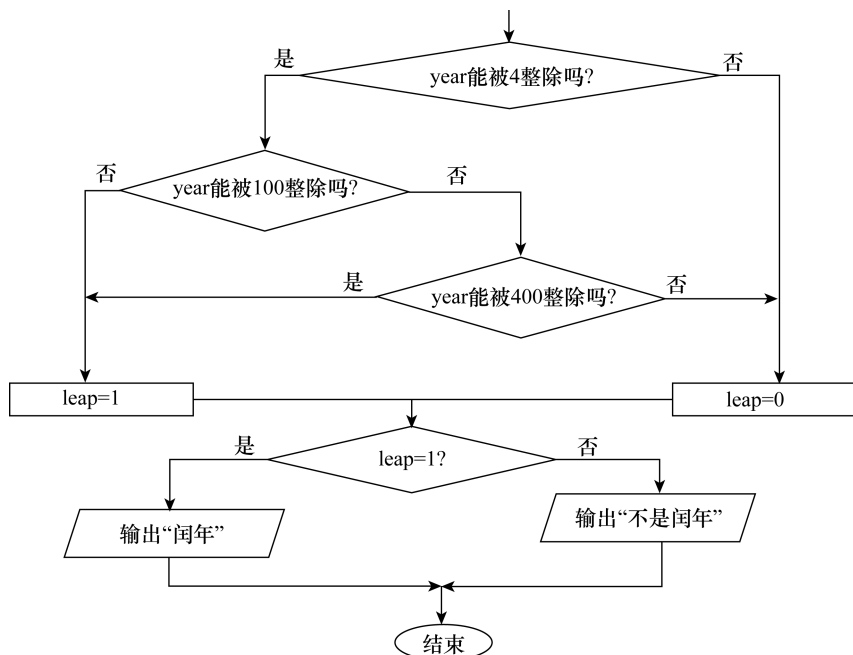


图 5-6 判断闰年的流程图

(3) 定义变量 year 表示年份, leap 是一个“标志变量”, 用来表示相应的年份是否为闰年。如果是闰年, 就使 leap 等于 1, 如果不是闰年, 就使 leap 等于 0。最后检查 leap 的值, 如果是 0, 就不是闰年, 输出“不是闰年”的信息, 如果是 1, 就输出“是闰年”的信息。

2. 程序代码

```

#include<stdio. h>
int main( )
{
    int year, leap;
    printf("enter year:");
    scanf("%d", &year);
    if(year%4==0)
    {
        if(year%100==0)
        {
            if(year%400==0)
                leap=1;
        }
    }
}
  
```

```

        else
            leap = 0;
    }
    else
        leap = 1;
    }
    else
        leap = 0;
    if(leap)
        printf("%d 是闰年\n", year);
    else
        printf("%d 不是闰年\n", year);
    return 0;
}

```

3. 程序运行和调试

通过编译、连接、运行后，屏幕显示“enter year:”，通过键盘输入 2018<回车>，运行结果如图 5-7 所示。

```

enter year:2018
2018 不是闰年
Press any key to continue

```

图 5-7 运行结果

再次编译、连接、运行后，屏幕显示“enter year:”，通过键盘输入 2020<回车>，运行结果如图 5-8 所示。

```

enter year:2020
2020 是闰年
Press any key to continue

```

图 5-8 运行结果

4. 分析与讨论

(1) 请仔细分析程序中各层 if 与 else 的配对关系。写程序时采取锯齿形式，可以清楚地看明白嵌套关系。建议今后写程序时一定要采用锯齿形式。

(2) 也可以将 7~20 行改成以下 if 语句。

```

if(year%4!=0)
    leap=0;
else if(year%100!=0)
    leap=1;

```



```
else if(year % 400 != 0)
    leap = 0;
else
    leap = 1;
```

5.2.4 案例四

运输公司对用户计算运输费用。路程(s/km)越远,每吨·千米运费越低。标准如下。

$s < 250$	没有折扣
$250 \leq s < 500$	2%折扣
$500 \leq s < 1000$	5%折扣
$1000 \leq s < 2000$	8%折扣
$2000 \leq s < 3000$	10%折扣
$3000 \leq s$	15%折扣

设每吨每千米货物的基本运费为 P , 货物重为 W , 距离为 s , 折扣为 D , 则总运费 F 的计算公式为 $F = P \times W \times S \times (1 - D)$ 。

1. 算法分析

(1) 仔细观察, 可知折扣的“变化点”都是 250 的倍数(250, 500, 1000, 2000, 3000)。

(2) 由这一特点, 找出折扣与距离的关系。定义变量 c 代表 250 的倍数, c 与距离 s 是整型变量, 即 $c = s/250$ 。当 $c < 1$ 时, 表示 $s < 250$, 无折扣; 当 $1 \leq c < 2$ 时, 表示 $250 \leq s < 500$, 折扣 $D = 2\%$; $2 \leq c < 4$ 时, $D = 5\%$; $4 \leq c < 8$ 时, $D = 8\%$; $8 \leq c < 12$ 时, $D = 10\%$; $c \geq 12$ 时, $D = 15\%$ 。

(3) 运用 `switch...case` 语句处理。

2. 程序代码

```
#include<stdio.h>
int main( )
{
    int c, s;
    float P, W, D, F;
    printf("请输入基本运费,货物重量,距离:");
    scanf("%f, %f, %d", &P, &W, &s);
    if(s >= 3000) c = 12;
    else c = s/250;
    switch(c)
    {
        case 0:D=0; break;
        case 1:D=2; break;
```

```

        case 2:
        case 3:D=5; break;
        case 4:
        case 5:
        case 6:
        case 7:D=8; break;
        case 8:
        case 9:
        case 10:
        case 11:D=10; break;
        case 12:D=15; break;
    }
    F=P*W*s*(1D/100);
    printf("总运费为:%10.2f\n", F);
    return 0;

```

3. 程序运行和调试

经编译、连接、运行后，屏幕显示“请输入基本运费，货物重量，距离：”，通过键盘输入“105，20，300”<回车>。运行结果如图 5-9 所示。



```

请输入基本运费，货物重量，距离：105,20,300
总运费为： 617400.00

```

图 5-9 运行结果

5.2.5 案例五

给一个不多于 5 位的正整数，要求：①求出它是几位数；②分别输出每一位数字；③按逆序输出各位数字，例如原数为 98765，应输出 56789。

1. 算法分析

(1) 根据题目，定义代表个位(indiv)、十位(ten)、百位(hundred)、千位(thousand)、万位(ten_thousand)和位数(place)的变量。

(2) 通过键盘输入一个数 num，运用多分支 if 语句形式，判断输入数据 num 的位数。

(3) 分别求每一位上的数字，关系如下。

求万位上的数：ten_thousand=num/10000；

求千位上的数：thousand= (num%10000)/1000；

求百位上的数：hundred= (num%1000)/100；

求十位上的数：ten= (num%100)/10；

求个位上的数：indiv=num%10。

(4) 运用 switch...case 多分支语句，根据位数 place，逆序输出各位数字。



2. 程序代码

```
#include<stdio. h>
int main( )
{
    int num, indiv, ten, hundred, thousand, ten_thousand, place;
        //分别代表个位,十位,百位,千位,万位和位数。
    printf("请输入一个整数(0- 99999):");
    scanf("%d", &num);
    if(num>0&&num<10)  place=1;
        else  if(num>=10 && num<100) place=2;
        else  if(num>=100 && num<1000) place=3;
        else  if(num>=1000 && num<10000) place=4;
        else  if(num>=10000 && num<100000) place=5;
        else
        {
            printf("输入错误\n");
            place=0;
        }
    printf("这是一个%d 位数。 \n", place);
    ten_thousand = num/10000;           //求万位上的数
    thousand = (num% 10000)/1000;       //求千位上的数
    hundred = (num% 1000)/100;          //求百位上的数
    ten = (num% 100)/10;                //求十位上的数
    indiv = num% 10;                   //求个位上的数
    switch(place)
    {
        case 5:printf("%d, %d, %d, %d, %d", ten_thousand, thousand, hundred, ten, indiv);
            printf (" \n 反序数字为:%d%d%d%d%d\n", indiv, ten, hundred, thousand,
                ten_thousand);
            break;
        case 4:printf("%d, %d, %d, %d", thousand, hundred, ten, indiv);
            printf(" \n 反序数字为:%d%d%d%d\n", indiv, ten, hundred, thousand);
            break;
        case 3:printf("%d, %d, %d", hundred, ten, indiv);
            printf(" \n 反序数字为:%d%d%d\n", indiv, ten, hundred);
            break;
        case 2:printf("%d, %d", ten, indiv);
```

```

        printf("\n 反序数字为:%d%d\n", indiv, ten);
        break;
    case 1:printf("%d", indiv);
        printf("\n 反序数字为:%d\n", indiv);
        break;
    }
    return 0;
}

```

3. 程序运行和调试

经编译、连接、运行后，屏幕显示“请输入一个整数(0-99999):”，通过键盘输入“98765”<回车>。运行结果如图 5-10 所示。

```

请输入一个整数(0-99999):98765
这是一个5位数。
9,8,7,6,5
反序数字为: 56789
Press any key to continue

```

图 5-10 运行结果

若通过键盘输入“-123”<回车>，运行结果如图 5-11 所示。

```

请输入一个整数(0-99999):-123
输入错误
这是一个0位数。
Press any key to continue

```

图 5-11 运行结果

5.3 自主编程题

1. 有一个函数

$$y = \begin{cases} -1 & (x < 0) \\ 0 & (x = 0) \\ 1 & (x > 0) \end{cases}$$

编一程序，输入一个 x 值，要求输出相应的 y 值。

(1) 采用嵌套的 if 语句处理，参考程序代码 1。

```

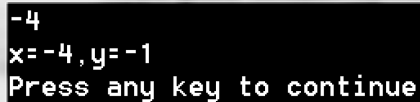
#include<stdio.h>
int main( )
{

```



```
int x, y;
scanf("%d", &x);
if(x<0)
    y = - 1;
else
    if(x==0) y=0;
    else y=1;
printf("x=%d, y=%d\n", x, y);
return 0;
}
```

当输入-4时，运行结果如图5-12所示。



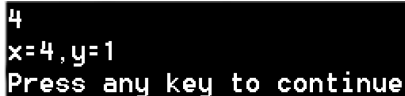
```
-4
x=-4,y=-1
Press any key to continue
```

图 5-12 运行结果

(2) 参考程序代码2。

```
#include<stdio. h>
int main( )
{
    int x, y;
    scanf("%d", &x);
    if(x>=0)
        if(x>0) y=1;
        else y=0;
    else
        y = - 1;
    printf("x=%d, y=%d\n", x, y);
    return 0;
}
```

当输入4时，运行结果如图5-13所示。



```
4
x=4,y=1
Press any key to continue
```

图 5-13 运行结果

(3) 分析与讨论。

① 弄清楚嵌套 if 中各个 if 的配对关系以及在程序中对嵌套 if 的书写格式。为了使逻辑关系清晰，避免出错，一般把内嵌的 if 语句放在外层的 else 子句中(如程序代码1)。

② 有人分别编写了以下两个程序，请分析它们是否能实现题目要求。不要急于上机运行程序，先分析程序的逻辑，画出它们的流程图，分析它们的运行情况。然后上机运行程序，观察和分析结果。

程序 a:

```
#include<stdio. h>
int main( )
{
    int x, y;
    printf("enter x:");
    scanf("%d", &x);
    y = - 1;
    if(x!=0)
        if(x>0)
            y = 1;
        else
            y = 0;
    printf("x = %d, y = %d \n", x, y);
    return 0;
}
```

程序 b:

```
#include<stdio. h>
int main( )
{
    int x, y;
    printf("enter x:");
    scanf("%d", &x);
    y = 0;
    if(x>=0)
        if(x>0)    y = 1;
        else
            y = - 1;
    printf("x = %d, y = %d \n", x, y);
    return 0;
}
```

2. 输入百分制学生的成绩，要求输出成绩等级 ‘A’，‘B’，‘C’，‘D’，‘E’。等级分类如下。

90 分以上(包括 90)： A

80 至 90 分(包括 80)： B



70 至 80 分(包括 70): C

60 至 70 分(包括 60): D

60 分以下: E

(1) 方法一: (用 if 嵌套)参考程序代码。

```
#include"stdio. h"
void main( )
{
    int  score;
    char  grade;
    printf("\nplease input a student score:");
    scanf("%d", &score);
    if(score>100 || score<0)
        printf("\n input error!");
    else
    { if(score>=90)
        grade=' A' ;
      else
      { if(score>=80)
          grade=' B' ;
        else
        {if(score>=70)
            grade=' C' ;
          else
          { if(score>=60)
              grade=' D' ;
            else grade=' E' ;
          }
        }
      }
    }
    printf("\nthe student grade:%c\n", grade);
}
```

当输入 90 时, 运行结果如图 5-14 所示。

```
please input a student score:90
the student grade:A
Press any key to continue
```

图 5-14 运行结果

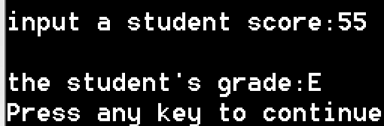


(2) 方法二: (用 switch 语句) 参考程序代码。

```
#include "stdio. h"
main( )
{
    int score, s;
    char grade;
    printf("\ninput a student score:");
    scanf("%d", &score);

    if(score>100 || score<0)
    {
        printf("\ninput error!\n");
        scanf("%d", &score);
    }
    else
    {
        switch (score/10)
        { case 10:
          case 9:  grade=' A' ;  break;
          case 8:  grade=' B' ;  break;
          case 7:  grade=' C' ;  break;
          case 6:  grade=' D' ;  break;
          default: grade=' E' ;
        }
        printf("\nthe student' s grade:%c\n", grade);
    }
}
```

当输入 55 时, 运行结果如图 5-15 所示。



```
input a student score:55
the student's grade:E
Press any key to continue
```

图 5-15 运行结果

(3) 分析与讨论

① 输入测试数据, 调试程序。测试数据要覆盖所有路径, 注意临界值, 例如此题中得 100 分, 60 分, 0 分以及小于 0 和大于 100 的数据。



② switch 语句是用于处理多分支的语句。注意, case 后的表达式必须是一个常量表达式, 所以在用 switch 语句之前, 必须把 0~100 之间的成绩分别化成相关的常量。

3. 编写一个程序, 由键盘输入三个整数作为三角形的三条边, 判断是否能构成一个三角形, 如能并判断是等边三角形、等腰三角形、直角三角形, 还是其他一般的三角形。

参考程序代码。

```
#include<stdio.h>
#include<math.h>
int main( )
{
    int a, b, c, g, h, i;
    printf("请输入三个整数 a, b, c:\n");
    scanf("%d%d%d", &a, &b, &c);
    g = (a < b ? a : b);
    g = (g < c ? g : c);          // 找出最短边;
    h = a > b ? a : b;
    h = h > c ? h : c;          // 找出最长边
    i = a + b + c - g - h;
    if((i + g > h) && (h - i < g))
    {
        printf("这三条边能构成一个三角形, 且");
        if(a == b && b == c)
            printf("这是等边三角形\n");
        else if(a == b || b == c || c == a)
            printf("这是等腰三角形.\n");
        else if(h == sqrt(g * g + i * i))
            printf("这是直角三角形.\n");
        else
            printf("这是一般三角形.\n");
    }
    else
        printf("不能构成三角形.\n");
    return 0;
}
```

通过调试运行, 当输入 3, 4, 5 时, 运行结果如图 5-16 所示。

```
请输入三个整数a,b,c:
3 4 5
这三条边能构成一个三角形, 且这是直角三角形.
Press any key to continue
```

图 5-16 运行结果

通过调试运行，当输入 5, 6, 6 时，运行结果如图 5-17 所示。

```
请输入三个整数a,b,c:
5 6 6
这三条边能构成一个三角形，且这是等腰三角形。
Press any key to continue
```

图 5-17 运行结果

通过调试运行，当输入 6, 6, 6 时，运行结果如图 5-18 所示。

```
请输入三个整数a,b,c:
6 6 6
这三条边能构成一个三角形，且这是等边三角形
Press any key to continue
```

图 5-18 运行结果

通过调试运行，当输入 7, 8, 9 时，运行结果如图 5-19 所示。

```
请输入三个整数a,b,c:
7 8 9
这三条边能构成一个三角形，且这是一般三角形。
Press any key to continue
```

图 5-19 运行结果

通过调试运行，当输入 5, 5, 10 时，运行结果如图 5-20 所示。

```
请输入三个整数a,b,c:
5 5 10
不能构成三角形。
Press any key to continue
```

图 5-20 运行结果

5.4 实验总结

1. 通过实验，分析各题实现选择结构时所使用的方法，总结 if...else 语句中判断条件的构成方法和语句体的组成。
2. 通过实验，掌握 switch 语句中表达式的构成，语句体中常量的表示方法。
3. 通过实验，熟练掌握选择结构程序设计方法，独立编写选择结构的程序。

第 6 章 实验五：循环结构程序设计

6.1 实验目的

- (1) 掌握在程序设计条件型循环结构时，如何正确地设定循环条件，以及如何控制循环的次数。
- (2) 了解条件型循环结构的基本测试方法。
- (3) 掌握如何正确地控制计数型循环结构的次数。
- (4) 了解对计数型循环结构进行测试的基本方法。
- (5) 掌握循环嵌套及复合结构，提高程序效率的方法。

6.2 实例案例分析

1. 求 1, 4, 7, 10, 13, …, 且末项不大于 100 的各项之和。

问题分析：这是一个数列求和问题，通过观察数列的数字排列，可知数字的排列规律为：初值为 1，公差为 3，末项 a_n 为不大于 100 的数。

利用数列累加求和的方法，画出对应的算法流程图如图 6-1 所示。

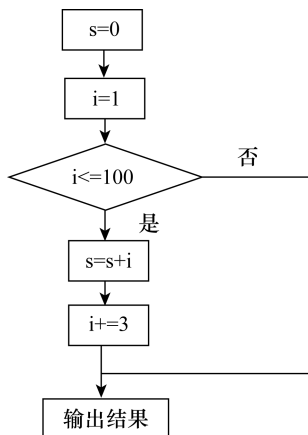


图 6-1 算法流程图

程序代码如下。

```
#include<stdio. h>
void main( )
{ int s, i;    //为什么数据类型为整数?
  s=0;        //给变量 s 赋初值为 0,为什么要赋初值,且初值为 0?
  i=1;        //给变量 i 赋初值为 1
```



```
while(i<=100)
{
    s=s+i;
    i+=3;
}
printf("s=%d\n",s);
}
```

运行结果如图 6-2 所示。

```
s=1717
Press any key to continue
```

图 6-2 运行结果

2. 求 1~100 之间的奇数之和与偶数之和。

问题分析：这也是一个求和问题，涉及如何判断某个整数是奇数或者偶数。常用的方法就是用该整数除 2 取余数，若余数为 0 则为偶数，否则为奇数。其对应的算法流程图如图 6-3 所示下。

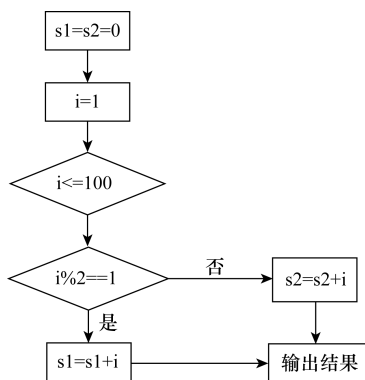


图 6-3 算法流程图

解法 1:

```
#include<stdio.h>
void main( )
{
    int s1, s2, i;
    s1=s2=0;
    for(i=1; i<=100; i++)
    {
        if(i%2==1)
            s1=s1+i; /* 奇数之和 */
        else
            s2=s2+i; /* 偶数之和 */
    }
    printf("s1=%d, s2=%d\n", s1, s2);
}
```



运行结果如图 6-4 所示。

```
s1=2500,s2=2550
Press any key to continue
```

图 6-4 运行结果

解法 2:

```
#include<stdio. h>
void main( )
{ int s1, s2, i;
    s1=s2=0;
    for(i=1; i<=99; i+=2)
        s1=s1+i; /* 奇数之和 */
    for(i=2; i<=100; i+=2)
        s2=s2+i; /* 偶数之和 */
}
printf("s1=%d, s2=%d\n", s1, s2);
}
```

运行结果如图 6-5 所示。

```
s1=2500,s2=2550
Press any key to continue
```

图 6-5 运行结果

解法 3:

```
#include<stdio. h>
void main( )
{ int s1, s2, i;
    s1=s2=0;
    i=1;
    while(i<=99)
    { s1=s1+i; /* 奇数之和 */
      i++;
      s2=s2+i; /* 偶数之和 */
      i++;
    }
    printf("s1=%d, s2=%d\n", s1, s2);
}
```

运行结果如图 6-6 所示。

```
s1=2500,s2=2550
Press any key to continue
```

图 6-6 运行结果

6.3 自主编程

1. 输入 n 个整数，求这 n 个数中的偶数平均值，并输出。
2. 输入一串字符，直到输入一个星号($*$)为止，统计(输出)其中的字母个数和数字字符个数。
3. 要求输出 100~200 之间的不能被 3 整除的数。
4. 输出以下 $4 * 5$ 的矩阵。

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5. 有一个分数序列 $2/1+3/2+5/3+8/5+13/8+\cdots$ ，求出这个数列前 20 项的和。
6. 打印如下菱形。

```

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

7. 猴子第一天摘了若干个桃子，当即吃了一半，还不解馋，又多吃了一个；第二天，吃剩下的桃子的一半，还不过瘾，又多吃了一个；以后每天都吃前一天剩下的一半多一个，到第十天想再吃时，只剩下一个桃子了。问第一天共摘了多少个桃子？
8. 设某县 2000 年工业总产值为 200 亿元，如果该县预计平均年工业总产值增长率为 4.5%，那么多少年后该县年工业总产值将超过 500 亿元。
9. 求爱因斯坦数学题。有一条长阶，若每步跨 2 阶，则最后剩 1 阶；若每步跨 3 阶，则最后剩 2 阶；若每步跨 5 阶，则最后剩 4 阶；若每步跨 6 阶，则最后剩 5 阶；若每步跨



7 阶，最后正好跨完。

10. 用二分法求方程 $2x^3 - 4x^2 + 3x = 0$ 在 $(-10, 10)$ 附近的根。

二分法：先找到 a 、 b ，使 $f(a)$ 、 $f(b)$ 异号，说明在区间 (a, b) 内一定有零点；然后求 $f[(a+b)/2]$ ，现在假设 $f(a) < 0$ ， $f(b) > 0$ ， $a < b$ ，如果 $f[(a+b)/2] = 0$ ，该点就是零点，如果 $f[(a+b)/2] < 0$ ，则在区间 $[(a+b)/2, b]$ 内有零点；按上述方法再求该区间中点的函数值，通过每次把 $f(x)$ 的零点所有小区间收缩一半的方法，使区间的两个端点逐步逼近函数的零点，以求得零点的近似值，该值即方程的根。

6.4 参考程序

1. 输入 n 个整数，求这 n 个数中的偶数平均值，并输出。

编程思路：首先输入 n 的值，然后从键盘上依次输入 n 个正整数，逐个判断这些正整数的奇偶性，对偶数进行累加，计数器加 1，最后用求得和除以偶数的个数得到平均值。

```
#include<stdio.h>
void main( )
{ int n, i, x, sum=0, count=0;
    printf("Input n:");          /* 请输入整数的个数 */
    scanf("%d", &n);             /* 输入整数的个数 */
    for(i=0; i<n; i++)
    { scanf("%d", &x);            /* 输入整数 */
        if(x%2 == 0)              /* 判断输入正整数是奇数还是偶数 */
        { sum+=x;                 /* 若是偶数,进行累加 */
            count++;              /* 偶数计数器加 1 */
        }
    }
    printf("Average is:%f\n", (float)sum/count);
    getch( );
}
```

2. 输入一串字符，直到输入一个星号 (*) 为止，统计(输出)其中的字母个数和数字字符个数。

编程思路：通过 while 循环判断输入字符是否为结束字符 '*'。然后在循环体内进行字符的分类判断：若字符在 'a' 与 'z' 或者在 'A' 与 'Z' 之间，则该字符是字母，字母个数加 1；若字符在 '0' 与 '9' 之间，则该字符是数字，数字个数加 1。

```
#include<stdio.h>
void main( )
{ char ch;
    int charcnt=0, numcnt=0;      /* 定义字母计数器 charcnt 和数字计数器 numcnt */
    while((ch=getchar())!='*')    /* 输入字符并判断该字符是否为结束字符 '*' */
```



```
{ if(ch>=' a' &&ch<=' z' || ch>=' A' &&ch<=' Z' ) /* 判断字符是否是字母 */  
    charcnt++; /* 字母计数器 charcnt+1 */  
if(ch>=' 0' &&ch<=' 9' ) /* 判断字符是否是字母 */  
    numcnt++; /* 数字计数器 numcnt+1 */  
}  
printf("Characters number:%d\nDigits number:%d\n", charcnt, numcnt);  
getch( );  
}
```

3. 要求输出 100~200 之间的不能被 3 整除的数。

编程思路：对 100~200 之间的每一个整数进行检查，如果不能被 3 整除，输出，否则不输出，无论是否输出此数，都要接着检查下一个数。

```
#include<stdio. h>  
void main( )  
{ int n;  
  for(n=100;n<=200;n++)  
    { if(n%3==0) continue;  
      printf("%5d",n);  
    }  
}
```

4. 输出以下 4 * 5 的矩阵。

```
1   2   3   4   5  
2   4   6   8   10  
3   6   9   12  15  
4   8   12  16  20
```

解题思路：可以用循环的嵌套来处理此问题，用外循环来输出一行数据，用内循环来输出一列数据，按矩阵的格式(每行 5 个数据)输出，不难发现，若行数和列数均从 1 开始计算，矩阵的元素值为行数与列数的积。

```
#include<stdio. h>  
int main( )  
{ int i, j, n=0;  
  for(i=1; i<=4; i++) /* 外循环控制行数 */  
    {for (j=1; j<=5; j++, n++) /* 内循环控制列数 */  
      { printf ("%5d", i*j); /* 输出矩阵各元素值,域宽控制为 5 */  
        if(j%5==0) printf ("\n"); /* 每输出 5 个元素,执行换行 */  
      } }  
  return 0;  
}
```



5. 有一个分数序列 $2/1+3/2+5/3+8/5+13/8+\dots$ ，求出这个数列前 20 项的和。

参考程序：

```
#include<stdio. h>
int main( )
{
    float i = 1;
    float j = 2;
    float sum = 0;
    int k;
    for(k = 1; k <= 20; k++)
    {
        int x;
        sum = sum + j/i;
        x = i;
        i = j;
        j = x + j; }
    printf("结果为%f\n", sum);
    return 0;
}
```

6. 打印一个菱形图案，程序代码如下。

```
#include<stdio. h>
void main( )
{
    int n = 6;
    int i, a, b;
    for(i = 1; i <= n; i++)
    {
        for(a = n - 1; a >= i; a--)
            printf(" ");
        for(b = 1; b <= 2 * i - 1; b++)
            printf("* ");
        printf("\n");
    }
    for(i = n - 1; i >= 1; i--)
    {
        for(a = i; a <= n - 1; a++)
            printf(" ");
        for(b = 1; b <= 2 * i - 1; b++)
            printf("* ");
        printf("\n");
    }
}
```



7. 猴子第一天摘了若干个桃子，当即吃了一半，还不解馋，又多吃了一个；第二天，吃剩下的桃子的一半，还不过瘾，又多吃了一个；以后每天都吃前一天剩下的一半多一个，到第十天想再吃时，只剩下一个桃子了。问第一天共摘了多少个桃子？

思路解析：采用逆向思维，从后往前推断，发现其中有相同的地方，即出现递推公式，可以采用递归方法。

方法一：

```
#include<stdio. h>
int main( )
{   int day=9,x1=0,x2=1;
    while (day>0)
    {   x1=(x2+1)*2;
        x2=x1;
        day-;
    }
    printf("The first day of picking peaches number is:%d\n", x1);
}
```

方法二：

```
#include<stdio. h>
int main( )
{   int day=9, x1=0, x2=1;
    for(; day>0; day- )
    {   x1=(x2+1)*2;
        x2=x1;
    }
    printf("The first day of picking peaches number is:%d\n", x1);
}
```

方法三：

```
#include "stdio. h"
int sumPeach(int day)
{   if (day==10)
    return 1;
    else
    return 2* sumPeach(day+1)+2;
}
int main( )
{   int sum;
    sum = sumPeach(1);
}
```




```
printf("The first day of picking peaches number is:%d\n", sum);  
}
```

8. 设某县 2000 年工业总产值为 200 亿元, 如果该县预计平均年工业总产值增长率为 4.5%, 那么多少年后该县年工业总产值将超 500 亿元。

参考程序:

```
#include<stdio. h>  
void main( )  
{ float x=200;  
  int n=0;  
  do  
  { x* =(1+0. 045);  
    n++;  
  }  
  while(x<500);  
  printf("n=%d", n);  
  getch( );  
}
```

9. 求爱因斯坦数学题。有一条长阶, 若每步跨 2 阶, 则最后剩 1 阶; 若每步跨 3 阶, 则最后剩 2 阶; 若每步跨 5 阶, 则最后剩 4 阶; 若每步跨 6 阶, 则最后剩 5 阶; 若每步跨 7 阶, 最后正好到顶, 求长阶段的阶数。

参考程序:

```
#include<stdio. h>  
void main( )  
{ int x=0,flag=0;  
  while(1)  
  { x++;  
    if(x%2==1)  
    if(x%3==2)  
    if(x%5==4)  
    if(x%6==5)  
    if(x%7==0)  
    { printf("%d", x);  
      break;  
    }  
    getch( );  
  }
```

10. 用二分法求方程 $2x^3 - 4x^2 + 3x = 0$ 在 $(-10, 10)$ 附近的根。



二分法：先找到 a 、 b ，使 $f(a)$ 、 $f(b)$ 异号，说明在区间 (a, b) 内一定有零点；然后求 $f[(a+b)/2]$ ，现在假设 $f(a) < 0$ ， $f(b) > 0$ ， $a < b$ ，如果 $f[(a+b)/2] = 0$ ，该点就是零点，如果 $f[(a+b)/2] < 0$ ，则在区间 $((a+b)/2, b)$ 内有零点；按上述方法再求该区间中点的函数值，通过每次把 $f(x)$ 的零点所在小区间收缩一半的方法，使区间的两个端点逐步逼近函数的零点，以求得零点的近似值，该值即方程的根。

参考程序：

```
#include <math.h>
#include <stdio.h>
void main( )
{ double x, fx, fa, fb, a = -10, b = 10, z = 0.0001;
  fa = 2*a*a*a - 4*a*a + 3*a;
  fb = 2*b*b*b - 4*b*b + 3*b;
  if(fa*fb < 0)
  { do
    { x = (a+b)/2;
      fx = 2*x*x*x - 4*x*x + 3*x;
      if(fx < 0)
      { a = x;
        fa = 2*a*a*a - 4*a*a + 3*a;
      }
      else
      { b = x;
        fb = 2*b*b*b - 4*b*b + 3*b;
      }
    } while(fabs(fa - fb) > z);
    printf("The root is:%lf\n", x);
  }
  getch( );
}
```

第 7 章 实验六：函数的应用

7.1 实验目的

- (1) 掌握 C 语言函数定义及调用的规则。
- (2) 理解函数间数据的传递、库函数的使用。
- (3) 掌握变量的作用域与存储类型。
- (4) 掌握函数的定义和函数的声明有何区别。
- (5) 了解什么是模块化程序设计，为什么要采用模块化程序设计。

7.2 实例案例分析

1. 想输出以下的结果，用函数调用实现。

```
* * * * *
```

```
How are you !
```

```
* * * * *
```

分析：在输出的文字上下分别有一行“*”号，显然不必要重复写这段代码，用一个函数 print_star 来实现输出一行“*”号的功能。然后再写一个 Print_Message 函数来输出中间一行英文语句，用主函数分别来调用这两个自定义函数。

```
#include<stdio. h>
int main( )
{ void print_star( );      /* 声明调用 print_star( )函数 */
  void print_message( );  /* 声明调用 print_message( )函数 */
  print_star( );          /* 调用执行 print_star( )函数 */
  print_message( );       /* 调用执行 print_message( )函数 */
  print_star( );          /* 调用执行 print_star( )函数 */
  return 0;
}
void print_star( )
{ printf("*****\n"); }
void print_message( )
{ printf("How are you!\n"); }
```

运行结果如图 7-1 所示。



图 7-1 运行结果

2. 输入 4 个整数，找出其中最大的数，用函数嵌套调用来处理。

分析：根据题意，可以用函数嵌套调用来实现，在 main 函数中调用 max4 函数，max4 的作用是找到 4 个数中的最大值，在 max4 中再调用另外一个函数 max2，max2 的作用是找出两个数字中的最大值，在 max4 中经过多次调用 max2 函数，就可以找出 4 个数字中的最大值，然后将最大值返回到主函数 main，输出结果。

```
#include<stdio. h>
int main( )
{ int max4(int a, int b, int c, int d);          /* 声明调用函数 max4( ) */
  int a, b, c, d;
  int max;
  printf("Please enter 4 interger number:"); /* 提示输入四个数字 */
  scanf("%d %d %d %d", &a, &b, &c, &d); /* 输入四个数字 */
  max=max4(a, b, c, d);    /* 调用函数 max4,得到四个数字中最大数字 */
  printf("max=%d\n", max);
  return 0;
}
int max4(int a, int b, int c, int d)/* 定义函数 max4( ) */
{ int max2(int a, int b);          /* 声明调用函数 max2( ) */
  int m;
  m=max2(a, b);                  /* 调用函数 max2 得到 a, b 中最大数,赋值给 m */
  m=max2(m, c);                  /* 调用函数 max2 得到 a, b, c 中最大数,赋值给 m */
  m=max2(m, d);                  /* 调用函数 max2 得到 a, b, c, d 中最大数,赋值给 m */
  return m;                      /* 返回 m 的值给 main 函数 */
}
int max2(int a, int b)            /* 定义函数 max2( ) */
{ if(a >= b)
  return a;                      /* 返回 a 的值 */
  else
  return b;                      /* 返回 b 的值 */
}
```

运行结果如图 7-2 所示。



```

"C:\Users\Administrator\Desktop\Debug\5.exe"
Please enter 4 interger number:12 11 13 14
max = 14
Press any key to continue

```

图 7-2 运行结果

分析：在主函数中需要调用函数 max4，所以在主函数的开头需要对 max4 函数进行声明。而在 max4 函数中 3 次调用了 max2 函数，所以在 max4 的开头需要对 max2 函数做声明。由于在主函数中没有直接调用 max2 函数，因此主函数中不必对 max2 函数做声明，只需在 max4 函数中做声明即可。

程序改进如下。

(1) 可以将 max2 函数的函数体修改为只需要一个 return 返回函数值。

```

int max2(int a, int b)          /* 定义函数 max2( ) */
{ return(a>=b? a: b); }

```

(2) 在 max4 函数中，3 个调用 max2 的语句(如 `m = max2(a, b);`)可以用以下一行代替。

```

m = max2(max2(max2(a, b), c), d);

```

3. 求 $1! + 2! + 3! + \dots + 10!$ 程序。

分析：该程序的算法为

```

i=1; s=0;
当 i<= 10
s=s+i!
所以此时需定义求 i! 的函数
#include<stdio. h>
void main( )
{   long mm(int x);          /* 自定义求阶乘函数应先声明 */
    int i;   long s=0;
    for (i=1; i<= 10; i++)
        s+=mm(i );          /* 调用求阶乘函数, 求 i 的阶乘 */
    printf("\n%ld\n", s);
}
long mm( int  n)
{   long  t=1;   int i;
    for (i=1; i<=n; i++)
        t*=i;
    return(t );
}

```



运行结果如图 7-3 所示。

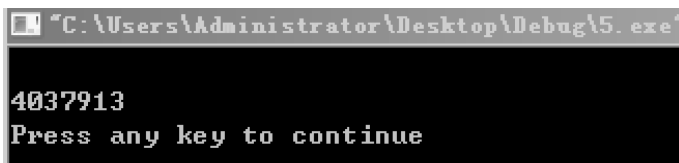


图 7-3 运行结果

4. 写一函数，用“起泡法”对输入的 10 个数字按由小到大的顺序排序。
源程序如下。

```
/* 起泡法排序 */
#include<stdio.h>
void main( )
void stor( ) /* 声明调用自定义函数 stor */
    int a[10], k; /* 定义数字 a, 有 10 个元素 */
    printf("输入十个数字: \n");
    for (k=0; k<10; k++)
        scanf("%d", &a[k]); /* 给予数组 a[10]每个元素分别赋值 */
    stor(a); /* 调用自定义函数 stor */
    printf("the sorted numbers: \n");
    for (k=0; k<10; k++)
        printf("%d ", a[k]);
    printf("\n");
}
void stor(int* b) { ... } /* 自定义函数 stor */
{ int t, i, j;
    for (i=0; i<10; i++)
        printf("%d ", b[i]);
        printf("\n");
    for (j=0; j<9; j++)
        for (i=0; i<9-j; i++)
            if (b[i]>b[i+1])
            { t=b[i];
              b[i]=b[i+1];
              b[i+1]=t; /* 冒泡法得到较大值 */
            }
}
```

运行结果如图 7-4 所示。



```

"C:\Users\Administrator\Desktop\Debug\6.exe"
输入十个数字:
1
2
4
5
3
6
7
9
8
10
1 2 4 5 3 6 7 9 8 10
the sorted numbers:
1 2 3 4 5 6 7 8 9 10
Press any key to continue

```

图 7-4 运行结果

7.3 自主编程

1. 输入两个实数，用一个函数求出它们之和。
2. 定义一个宏，实现将两个数互换，并写出程序，输入两个数作为使用参数，并显示结果。
3. 输入一个 3 位数，判断是不是一个“水仙花数”。水仙花数是指 3 位数的各位数字的立方和等于这个 3 位数本身。例如， $153 = 1 * 1 * 1 + 5 * 5 * 5 + 3 * 3 * 3$ 。
4. 写两个函数，分别求两个整数的最大公约数和最小公倍数，用主函数调用这两个函数，并输出结果。两个整数由键盘输入。
5. 输入 10 个学生 5 门课成绩，分别用函数实现下列功能。
 - (1) 计算每个学生的平均分。
 - (2) 计算每科的平均分。
 - (3) 找出 50 个分数中最高分和对应的学生课程。
6. 求出平均分方差： $\sigma = 1/n \sum x_i^2 - \left(\frac{\sum x_i}{n} \right)^2$ 中， x_i 为某一学生的平均分。

7.4 参考程序

1. 输入两个实数，用一个函数求出它们之和。

参考程序：

```

#include<stdio.h>
int main( )
{ int a, b, c;
  printf("please enter a and b:");

```



```
scanf("%d %d", &a, &b);
c = add(a, b);
printf("sum is %f. \n", c);
return 0;}

int add(int x, int y)
{ int z;
  z = x + y;
  return (z);
}
```

2. 定义一个宏, 实现将两个数互换, 并写出程序, 输入两个数作为使用参数, 并显示结果。

参考程序:

```
#define f(a, b) (a = a + b, b = a - b, a = a - b)

void main( )
{ int x, y;
  printf("Input x, y:");
  scanf("%d %d", &x, &y);
  printf("%d %d", x, y);
  f(x, y);
  printf("<=>%d %d \n", x, y);
  getch( );
}
```

3. 输入一个 3 位数, 判断是不是一个“水仙花数”。水仙花数是指 3 位数的各位数字的立方和等于这个 3 位数本身。例如, $153 = 1 * 1 * 1 + 5 * 5 * 5 + 3 * 3 * 3$ 。

参考程序:

```
#include <stdio.h>

int main( )
{ int x, i, f;
  printf("Input number:");
  scanf("%d", &x);
  直接调用 fun(x) 并判断。
  if(f == 1) printf("%d \t", i);
}

getch( );
}
```




```
int fun(int x)
{ int a, b, c;
  a=x/100;
  b=x/10%10;
  c=x%10;
  if(a*a+a+b*b+b+c*c*c==x) return 1;
  else return 0;
}
```

4. 写两个函数，分别求两个整数的最大公约数和最小公倍数，用主函数调用这两个函数，并输出结果。两个整数由键盘输入。

参考程序：

```
#include<stdio. h>
int main( )
{ unsigned int m, n, p, q;
  for(;;)
  { printf("m, n = "); scanf("%d, %d", &m, &n);
    if(m<0 || n<0) { printf("The end. \n"); break;}
    p=funp(m, n);
    q=funq(m, n);
    printf("p(%d, %d)=%d \nq(%d, %d)=%d \n", m, n, p, m, n, q);
  } }
int funp(int m, int n)
{ int i;
  if(m>n){i=m; m=n; n=i;}
  for(i=m; i>0; i- )if(m%i==0&& n%i==0)
    return(i);
  }
int funq(int m, int n)
{ int k;
  k=funp(m, n);
  return(m*n/k);
}
```

5. 输入 10 个学生 5 门课成绩，分别用函数实现下列功能。

(1) 计算每个学生的平均分。

(2) 计算每科的平均分。

(3) 找出 50 个分数中的最高分和对应的学生、课程。



参考程序：

```
#include<stdio. h>
int main( )
{ void max(int [ ][5]);//注意二位数组传递参数时,行数信息可以省略,但列数一定不能
  void averagek(int [ ][5]);
  void averages(int [ ][5]);
  int a[10][5], i;
  for(i=0; i<=9; i++)
  { printf("请输入第%d 名学生五科成绩(中间用空格隔开):", i+1);
    scanf("%d %d %d %d %d", &a[i][0], &a[i][1], &a[i][2], &a[i][3], &a[i][4]);
  }
  averages(a);
  averagek(a);
  max(a);
  return 0;
}

void averages(int a[ ][5])
{ int m, i;
  for(i=0; i<=9; i++)
  { m = (a[i][0]+a[i][1]+a[i][2]+a[i][3]+a[i][4])/5;
    printf("第%d 名学生平均分为:%d\n", i+1, m);
  }
}

void averagek(int a[ ][5])
{ int m,i;
  for(i=0;i<=4;i++)
  { m = (a[0][i]+a[1][i]+a[2][i]+a[3][i]+a[4][i]+a[5][i]+a[6][i]+a[7][i]+a[8][i]+a[9][i])/10;
    printf("第%d 科成绩平均分为:%d\n", i+1, m);
  }
}

void max(int a[ ][5])
{ int i, j, max, k, g;
  max=a[0][0];
  for(i=0; i<=9; i++)
  for( j=0; j<=4; j++)
  { if(a[i][j]>max)
    { max=a[i][j];
```



```

        k=i+1; g=j+1;
    }
    printf("第%d 名学生的第%d 科成绩取得最高分:%d\n", k, g, max);
}

```

6. 求出平均分方差: $\sigma = 1/n \sum x_i^2 - \left(\frac{\sum x_i}{n} \right)^2$ 中, x_i 为某一学生的平均分。

参考程序:

```

#include<stdio. h>
#define N 10
float a_stu[N];
int average1(int str1[ ][5], int m, int n);
int average2(int str1[ ][5], int m, int n);
int max(int str1[ ][5], int m, int n);
int average4(int str1[ ][5], int m, int n);
int main( )
{ int str[10][5]={{87, 88, 92, 67, 78},{88, 86, 87, 98, 90}, {76, 75, 65, 65, 78}, {67, 87,
60, 90, 67}, {77, 78, 85, 64, 56}, {76, 89, 94, 65, 76}, {78, 75, 64, 67, 77}, {77, 76, 56,
87, 85}, {84, 67, 78, 76, 89}, {86, 75, 64, 69, 90}};
int i, j;
printf("每个学生的平均成绩\n");
average1(str, 10, 5);
printf("每门课的平均成绩\n");
average2(str, 10, 5);
printf("最高分学生: \n");
max(str, 10, 5);
printf("某个学生的平均成绩的偏差\n");
average4(str, 10, 5);
return 0;
}
int average1(int str1[ ][5], int m, int n)//单个学生的平均成绩
{ int i, j;
float average1;
for(i=0;i<m;i++)
{ float sum=0;
for(j=0;j<n;j++)
{ sum+=str1[i][j];
}
}
}

```



```
a_stu[j] = sum/n;
printf("%.2f\n", a_stu[i]);
}}

int average2(int str1[ ][5], int m, int n)//单门课的平均成绩
{ int i, j;
  float average2;
  for(j=0; j<n;j++)
  { float sum=0;
    for(i=0; i<m;i++)
    { sum+=str1[i][j];
    }
    average2 = sum/m;
    printf("%.2f\n", average2);
  } }

int max(int str1[ ][5], int m, int n) //求最高分学生
{ int max=str1[0][0];
  int p, k, i, j;
  for(i=0; i<10; i++)
  { for(j=0; j<5; j++)
    { if(str1[i][j]>max)
      { max=str1[i][j];
        k=i+1;
        p=j+1;
      }
    }
  }
  printf("最高分对应的学生和课程\n");
  printf("max=%d, k=%d, p=%d\n", max, k, p);
}

int average4(int str1[ ][5], int m, int n) //某个学生的平均偏差
{ int i;
  float sum1=0, sum2=0;
  float average4;
  for(i=0; i<m; i++)
  { sum1+=a_stu[i];
    sum2+=a_stu[i]*a_stu[i];
    average4 = sum2/m - (sum1/m)*(sum1/m);
  }
  printf("%.2f\n", average4);
}
```

第 8 章 实验七：数组的构造与应用

8.1 实验目的

- (1) 掌握一维数组的定义、初始化方法。
- (2) 掌握一维数组的输入、输出方法。
- (3) 掌握二维数组的定义、初始化和输入输出方法。
- (4) 掌握二维数组有关的算法，如查找、矩阵转置、数据修改等。
- (5) 掌握字符数组的定义、初始化和引用。
- (6) 掌握字符数组的常用处理算法。
- (7) 掌握字符串处理函数的使用方法。

8.2 案例分析

【案例 1】 在键盘上输入 N 个整数，试编程序使该数组中的数按照从小到大的次序排列。

分析：C 语言中数组长度必须是确定大小，即指定 N 的值。排序的方法有多种，我们取出其中两种作为参考。

方法一：起泡排序

从第一个数开始依次对相邻两数进行比较，如次序对则不做任何操作；如次序不对则使这两个数交换位置。第一遍的(N-1)次比较后，最大的数已放在最后，第二遍只需考虑(N-1)个数，以此类推直到第(N-1)遍比较后就可以完成排序。

源程序如下：

```
#include<stdio. h>
#define N 10
int main( )
{ int a[N], i, j, temp;
  printf("please input %d numbers\n", N);
  for(i=0; i<N; i++)
    scanf("%d", &a[i]);
  for(i=0; i<N- 1; i++)
    for(j=0; j<N- 1- i; j++)
      { if(a[j]>a[j+1])
        { temp=a[j]; a[j]=a[j+1]; a[j+1]=temp;}
```



```
    }  
    printf("the array after sort:\n");  
    for(i=0; i<N; i++)  
        printf("%5d", a[i]);  
    printf("\n");  
    return 0;  
}
```

运行程序，从键盘输入 10 个整数，将会输出排序好的结果。程序运行输出如图 8-1 所示。

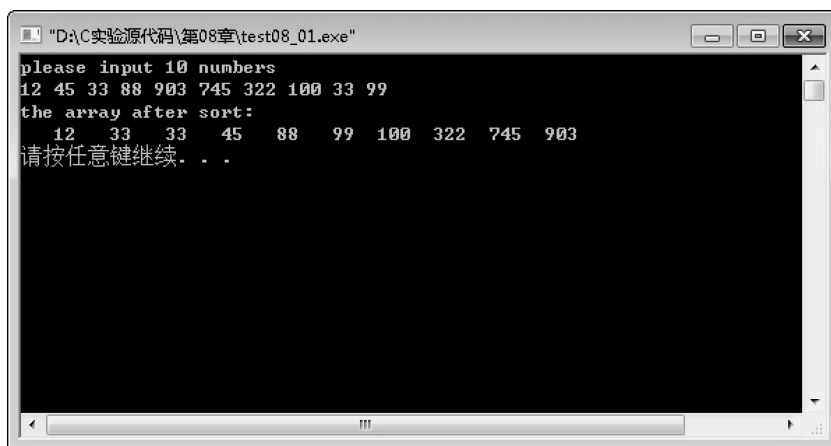


图 8-1 起泡排序法程序运行结果图

方法二：选择排序

首先找出值最小的数，然后把这个数与第一个数交换，这样值最小的数就放到了第一个位置；然后，再从剩下的数中找值最小的，把它和第二个数互换，使得第二小的数放在第二个位置上。以此类推，直到所有的值从小到大的顺序排列完为止。

```
#include<stdio. h>  
#define N 10  
int main( )  
{  
    int a[N], i, j, r, temp;  
    printf("please input %d numbers\n", N);  
    for(i=0; i<N; i++)  
        scanf("%d", &a[i]);  
    for(i=0; i<N- 1; i++)  
    {  
        r=i;  
        for(j=i+1; j<N; j++)
```



```

        if(a[j]<a[r]) r=j;
    if(r!=i)
    {
        temp=a[r]; a[r]=a[i]; a[i]=temp;
    }
}
printf("the array after sort:\n");
for(i=0; i<N; i++)
    printf("%5d", a[i]);
printf("\n");
}

```

选择排序法程序运行结果如图 8-2 所示。

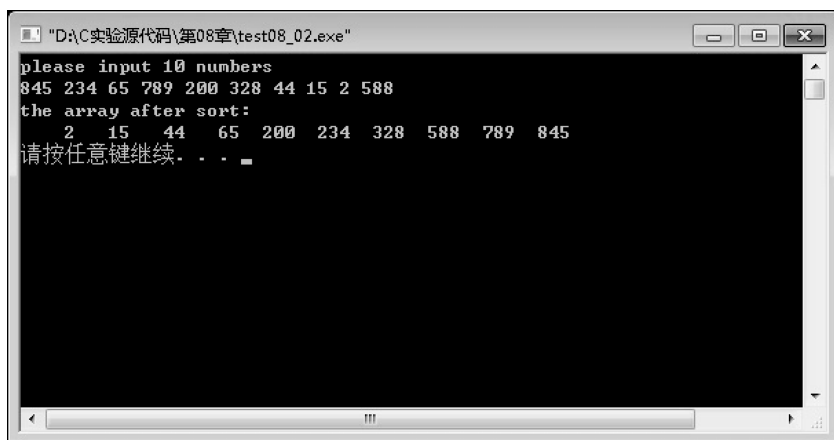


图 8-2 选择排序法程序运行结果图

【案例 2】有一个 3×4 的矩阵，要求输出其中值最大的元素的值，以及它的行号和列号。

分析：从第一个元素开始对二维数组的所有元素进行遍历，找出最大值对应的行号、列号。首先，定义两个变量 row 和 col 分别用于记录最大值元素所在的行号和列号，将第一个元素记为最大值 (row=0, col=0)；接着从第二个元素开始与所记录的最大值元素进行比较，如果出现更大值，则更新所记录的行号、列号 (即更新 row=i, col=j)，直到遍历完整个二维数组，这时 row 和 col 的值就是最大值元素所在的行号、列号，程序源代码如下所示。

```

#include<stdio. h>
#define M 3
#define N 4
int main( )
{

```

```

int max, i, j, row, col;
int a[M][N]={{123, 94, - 10, 218}, {3, 9, 10, - 83}, {45, 16, 44, - 99}};
row=0;
col=0;
max=a[0][0];
for(i=0; i<M; i++)
    for(j=0; j<N; j++)
        if ( a[i][j]>max)
        {
            max=a[i][j];
            row=i;
            col=j;
        }
printf("max=%d, row=%d, colum=%d \n", max, row, col);
return 0;
}

```

程序运行结果如图 8-3 所示。

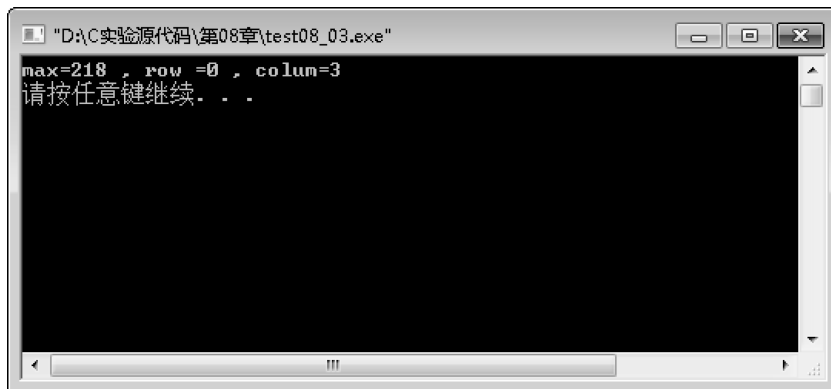


图 8-3 找矩阵最大元素程序运行图

8.3 自主编程

请根据实验题目要求，设计出算法或画出流程图，编写实验程序，上机运行验证结果，并对所编写程序进行代码优化。

1. 青年歌手参加歌曲大奖赛，有 10 个评委对歌手进行打分，试编程求这位选手的平均得分(去掉一个最高分和一个最低分)。
2. 输入一串字符，计算其中空格的个数。
3. 求 M 行 M 列整数方阵两条对角线上各元素之和。
4. 输入任一字符串存放在字符数组 a 中，将该字符串中与字符‘c’相等的所有元素



的下标值分别存放在整型数组 b 中，并输出数组 b 的内容。

5. 定义一个有 10 个元素的 int 型一维数组，从键盘输入 10 个整数存放到数组中，用选择法对这 10 个整数排序并输出排序后的数据。

6. 有一个已经排序好的数组，要求输入一个数后，按原来的排序规律将它插入数组中。

7. 输入一字符串，包含数字及其他字符，将连在一起的数字转换为对应的十进制数，并保存到一个整型数组中，并输出结果。例如输入的字符串为：“ab123cd456xbc3mDWf,45ldx”，则提取到的十进制数有 4 个：123 456 345。

8.4 参考程序

1. 先对输入的 10 个数进行排序，统计时将第一个数和最后一个数去掉。参考程序如下。

```
#include<stdio. h>
int main( )
{
    int i, j, min;
    float temp, ave=0;
    float a[11];
    printf("\n 输入评委所打的分数:\n");
    for(i=1; i<=10; i++)
        scanf("%f", &a[i]);
    for(i=1; i<=9; i++)
    {
        min=i;
        for(j=i+1; j<=10; j++)
            if(a[min]>a[j])
                min=j;
        temp=a[i];
        a[i]=a[min];
        a[min]=temp;
    }
    for(i=2; i<=9; i++) /* 注意 i 值的变化范围 */
        ave=ave+a[i];
    printf("选手所得最后分数:%6. 1f", ave/8);
    printf("\n");
    return 0;
}
```



2. 参考程序如下。

```
#include<stdio. h>
int main( )
{
    char c[30];
    int i, sum=0;
    gets(c);
    for(i=0; i<strlen(c); i++)
        if(c[i]==")
            sum = sum+1;
    printf("空格数为:%d \n", sum);
    return 0;
}
```

3. 参考程序如下。

```
#include<stdio. h>
#define M 4
int main( )
{
    int aa[M][M]={1, 2, 3, 4}, {4, 3, 2, 1}, {6, 7, 8, 9}, {9, 8, 7, 6}};
    int i, j, sum;
    printf("\nThe %d x %d matrix:\n", M, M);
    for(i=0; i<M; i++)
    {
        for(j=0; j<M; j++)
            printf( "%4d", aa[i][j]);
        printf("\n");
    }
    for(i=0; i<M; i++)
        sum += aa[i][i]+aa[i][M- i- 1];
    printf("\nThe sum is %d\n", sum);
    return 0;
}
```

4. 参考程序如下。

```
#include<stdio. h>
int main( )
{
    char a[80];
    int i, b[80], k=0;
```



```
    gets(a);
    for(i=0; a[i]!=' \0' ; i++)
        if(a[i]==' c' )
        {
            b[k]=i;
            k++;
        }
    for(i=0;i<k;i++)
        printf("%3d", b[i]);
    printf("\n");
    return 0;
}
```

5. 参考程序如下。

```
#define N 10
#include<stdio. h>
int main( )
{
    int a[N];
    int i, j, temp;
    printf("请输入%d 个整数: \n", N );
    for(i=0; i<N; i++)
        scanf( "%d", &a[i] );
    for(i=0; i<N; i++)
    {
        for(j=i+1; j<N; j++)
            if(a[i]<a[j])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
    }
    printf("排序好的数据为: \n" );
    for(i=0; i<N; i++)
        printf("%3d", a[i]);
    printf("\n");
    return 0;
}
```



6. 参考程序如下。

```
#define N 10
#include<stdio.h>
int main( )
{
    int a[N+1]={12, 23, 35, 50, 201, 255, 390, 456, 803, 899};
    int i, j, x, temp;
    printf("请输入要插入的整数: \n");
    scanf( "%d", &x );
    printf("原来的数据为: \n");
    for(i=0; i<N; i++)
        printf( "%d ", a[i] );
    for(i=0; i<N; i++)
    {
        if(a[i]>x)
        {
            for(j=N; j>i; j- )
            {
                temp = a[j];
                a[j] = a[j- 1];
                a[j- 1] = temp;
            }
            a[i] = x;
            break;
        }
    }
    if(i==N) //插入的位置在最后
        a[N] = x;
    printf("\n 插入一个整数后数组为: \n");
    for(i=0; i<N+1; i++)
        printf("%d ", a[i]);
    printf("\n");
    return 0;
}
```

7. 参考程序如下。

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



```
{
    char str[128];
    int i, j, k, x, a[50] = {0};
    printf("请输入一串字符串: \n");
    gets(str);
    i = k = 0;
    while( str[i] != '\0' )
    {
        if( str[i] >= '0' && str[i] <= '9' )
        {
            x = 0;
            while( str[i] >= '0' && str[i] <= '9' )
            {
                x = x * 10 + str[i] - '0';
                i++;
            }
            a[k++] = x;
        }
        else
            i++;
    }
    printf("提取到的整数为: \n");
    for(i = 0; i < k; i++)
        printf("%d ", a[i]);
    printf("\n");
    return 0;
}
```

第9章 实验八：指针的应用

9.1 实验目的

- (1) 掌握指针的基本概念，学习指针变量的定义和使用方法。
- (2) 掌握指针作为函数参数的使用方法。
- (3) 掌握指针在数组中的应用，正确使用数组的指针和指向数组的指针变量；区别指向数组的指针与指针数组。
- (4) 掌握利用指针处理字符串的使用方法。

9.2 案例分析

【案例1】从键盘上输入一行文字，统计其中的大写字母、小写字母、空格、数字字符及其他字符的个数，要求用指针对字符串进行访问。

分析：定义一个字符指针 p，让 p 指向字符串的首部，通过移动指针访问字符串中的每一个字符。在此过程中，如果指针指向的字符是大写字母，则让统计大写字母的变量加 1；如果是小写字母，则让统计小写字母的变量加 1；如果是空格，则让统计空格的变量加 1；如果是数字字符，则让统计数字字符的变量加 1；否则，就让统计其他字符的变量加 1。

参考程序如下。

```
#include<stdio. h>
#include<string. h>
int main( )
{
    int upper=0, lower=0, space=0, digital=0, others=0;
    char str[128], *p;
    printf("please input a string:\n");
    gets( str );
    p=str;
    while( *p)
    {
        if(*p>= ' A'  && *p<= ' Z' )        //大写字母
            upper++;
        else if(*p>= ' a'  && *p<= ' z' )    //小写字母
            lower++;
```



```

        else if(*p>='0' && *p<='9' )           //数字字符
            digital++;
        else if(*p=="")                       //空格字符
            space++;
        else
            others++;
        p++;
    }
    printf("upper=%d, lower=%d, digital=%d, space=%d, others=%d\n", upper,
        lower, digital, space, others);
    return 0;
}

```

【案例 2】从键盘上输入两串字符串，将字符串 2 复制到字符串 1 的后面连接起来，要求复制时在字符串 1 中出现过的字符不复制，用指针实现。

分析：从字符串 2 中分别取出每一个字符，检查取出的字符是否在字符串 1 中出现过，如果不是，则复制到字符串 2 的后面，否则不复制。

参考程序如下。

```

#include<stdio. h>
#include<string. h>
int main( )
{
    char str1[128], str2[128], *p, *p1, *p2, *p_end;
    printf("please input tow string:\n");
    gets( str1 );
    gets( str2 );
    p=p_end=str1;
    while( *p!='\0' )
    {
        p++;
        p_end++;
    }
    p2=str2;
    while( *p2!='\0' )
    {
        p1=str1;
        while( p1!=p_end )
        {
            if(*p1==*p2)

```

```
        break;
        p1++;
    }
    if( p1 == p_end )
    {
        *p=*p2;
        p++;
    }
    p2++;
}
puts( str1 );
return 0;
}
```

9.3 自主编程

请根据实验题目要求，设计出算法或画出流程图，编写实验程序，上机运行验证结果，并对所编写程序进行代码优化。

1. 编写实现两个字符串连接的函数，要求不可使用 `strcat` 库函数。
2. 将一个整型一维数组的最大元素与第一个元素交换，最小元素与最后一个元素交换，其他元素保持不变，要求编写一个函数用指针实现。
3. 利用指针作函数参数，设计一个函数实现将 n 个元素进行排序。
4. 利用指针作函数参数，设计一个函数实现求字符串的长度。
5. 有 n 个整数按从小到大的顺序排列，将其首尾相接围成一个圈，任意输入一个整数 x ，查找这 n 个数，以第一个比 x 大的数作为队首，输出这圆圈中的 n 个数。要求编写一个函数，通过指针实现该功能。
6. 输入一个字符串保存在字符数组中，编写一函数 `left(char *str, char c)`，从字符串右边开始往左查找字符 c ，出现该字符的右边截掉(包括该字符)，保留左边的子串。

9.4 参考程序

1. 参考程序如下。

```
#include<stdio.h>
#include<string.h>
void str_cat(char *str1, char *str2)
{
    while( *str1!='\0' )
        str1++;
```




```
while( *str2!=' \0' )
{
    *str1=*str2;
    str1++;
    str2++;
}
*str1=' \0' ;
}
int main( )
{
    char s1[100], s2[50];
    gets(s1);
    gets(s2);
    str_cat( s1, s2 );
    puts(s1);
    return 0;
}
```

2. 参考程序如下。

```
#include<stdio. h>
void exchage( int *array, int n)
{
    int *p, *pmax, *pmin, i;
    p = pmax = pmin = array;
    for(i=0; i<n; i++)
    {
        if(*pmax<*p)
            pmax = p;
        p++;
    }
    i = *pmax;
    *pmax = *array;
    *array = i;
    p = array;
    for(i=0; i<n; i++)
    {
        if(*pmin>*p)
            pmin = p;
        p++;
    }
}
```



```
    }  
    i=*pmin;  
    *pmin=*(array+n-1);  
    *(array+n-1)=i;  
}  
int main( )  
{  
    int a[10]={22, 34, 9, 354, 390, 805, 94, 33, 284, 49};  
    int i,*p;  
    p=a;  
    exchage(a, 10);  
    for(i=0; i<10; i++)  
        printf("%d ", *p++);  
    printf("\n");  
    return 0;  
}
```

3. 参考程序如下。

```
#include<stdio. h>  
int sort(int*array, int n)  
{  
    int i=0, k=0, j=0;  
    int t=0;  
    for(i=0; i< n- 1; i++)  
    {  
        k=i;  
        for(j=i+1; j< n; j++)  
        {  
            if (*(array+k) >*(array+j))  
                k=j;  
            if (k !=i)  
            {  
                t=*(array+i);  
                *(array+i)=*(array+k);  
                *(array+k)=t;  
            }  
        }  
    }  
}
```



```
int main( )
{
    int array[10];
    int i=0;
    for(i=0; i< 10; i++)
        scanf("%d", &array[i]);
    sort(array, 10);
    for(i=0; i< 10; i++)
        printf("%d ", array[i]);
    printf("\n");
    return 0;
}
```

4. 参考程序如下。

```
#include<stdio. h>
#include<string. h>
int longs(char *s)
{
    int i=0, n=0;
    for(i=0; *(s+i)!=' \0' ; i++)
        n++;
    return (n- 1);
}
int main( )
{
    char *s;
    char c[100];
    int num;
    s = c;
    gets(c);
    num = longs(s);    //传递一个指针
    printf("%d \n", num);
    return 0;
}
```

5. 参考程序如下。

```
#include<stdio. h>
#define N 10
void output(int *a, int b)
{
```



```
int *p1,*p2, i;
p1 = a;
for(i=0; i<N; i++)
{
    if(*p1>b)
        break;
    p1++;
}
if( p1>=a+N )
    printf("找不到比%d 大的数! \n", b);
else
{
    for(i=0; i<N; i++)
    {
        printf("%d ", *p1);
        p1++;
        if(p1>=a+N)
            p1 = a;
    }
    printf("\n");
}
}

int main( )
{
    int a[N]={15, 22, 30, 38, 45, 59, 77, 85, 93, 209};
    int x;
    scanf("%d", &x );
    output( a, x );
    return 0;
}
```

6. 参考程序如下。

```
#include<stdio. h>
#include<string. h>
void left(char *str, char c)
{
    char *p;
    p = str;
    while(*p!=' \0' )
```



```
        p++;
    while( p>str )
    {
        p- ;
        if(* p==c)
            break;
    }
    if(p<= str)
        str= ' \0' ;
    else
        *p= ' \0' ;
}
int main( )
{
    char s[100]="kldsfasd dsfjke89xdc534xdr33", ch;
    ch= ' x' ;
    puts(s);
    left( s, ch );
    puts(s);
    return 0;
}
```

第 10 章 实验九：结构体数据的应用

10.1 实验目的

- (1) 掌握结构体类型与结构体变量的定义。
- (2) 掌握结构体变量的引用与初始化。
- (3) 掌握结构体数组的应用。
- (4) 掌握使用指针操作结构体的基本应用。
- (5) 理解链表的基本操作。

10.2 实验案例分析

1. 编程实现：编写函数 input 实现从键盘输入学生信息(学号、姓名、成绩)，编写函数 output 实现输出学生信息，编写函数 sort 实现将学生信息按成绩进行排序。

分析：题目要求编写三个函数，那么函数的调用就必须在主函数中实现，因此函数需要传递一个数组作为参数才能将数据返回到主函数中。至于每个函数的写法，可以参照数组一章的输入、输出、排序的算法实现，只是数据使用的是结构体，具体语句做些修改就可以了。

程序参考代码如下。

```
#include<stdio. h>

typedef struct{           //根据题目定义学生结构体
    int no;               //学号
    char name[10];        //姓名
    float score;          //成绩
}Student;                //起个别名 Student
//输入 n 个学生数据,参数 s[ ]存放具体数据,n 代表要输入的学生个数
void input(Student s[ ], int n)
{
    int i;
    for(i=0; i<n; i++)
    {
        //输入单个学生数据,并加上提示
        printf("请输入第%d 个学生数据: \n", i+1);
        printf("学号:");
        fflush(stdin);
```



```
        scanf("%d", &s[i]. no);
        printf("姓名:");
        fflush(stdin);
        gets(s[i]. name);
        fflush(stdin);
        printf("请输入成绩:");
        scanf("%f", &s[i]. score);
    }
}
//以表格的方式输出 n 个学生数据
void output(Student s[ ], int n)
{
    int i;
    //输出表格头
    printf("\n%6s %8s %8s", "学号", "姓名", "成绩");
    for(i=0; i<n; i++){
        //输出一个学生数据,宽度和表格头对齐
        printf("\n%6d %8s %8. 2f", s[i]. no, s[i]. name,s[i]. score);
    }
    printf("\n");
}
//对 n 学生数据进行排序,这里使用选择排序
void sort(Student s[ ], int n)
{
    int i, j, t;
    Student max;
    //排序进行 n-1 趟,每趟先找出范围中的最大值,和范围内的最后一个数据进行交换
    for(i=0; i<n- 1; i++)
    { //找最大值
        t=0;
        for(j=0; j<n- i; j++)
        {
            if(s[t]. score<s[j]. score)
                t=j;
        }
        //交换数据
        if(t!= n- i){
            max = s[t];
```



```
        s[t]=s[n- i- 1];
        s[n- i- 1]=max;
    }
}
}
void main( )
{ Student s[10];
  int n;
  printf("请输入学生数据个数:");
  scanf("%d",&n);
  //调用函数
  input(s,n);
  sort(s,n);
  //输出结果
  output(s,n);
}
```

程序运行结果如图 10-1 所示。

学号	姓名	成绩
1002	张翠山	76.00
1001	张无忌	80.00
1006	杨过	82.00
1005	郭靖	85.00
1003	张三丰	90.00

请按任意键继续. . .

图 10-1 运行结果截图

2. 编程实现：用带头节点的链表保存整形数组 a 中的数据，并按从小到大的顺序输出。

分析：因为程序要求使用链表来实现，所以定义结构体时注意必须有一个指针指向下一个节点的地址，链表的排序可以在建立链表的同时进行，即每加入一个节点时先找到插入位置(第一个比它大的数前面)，然后插入节点即可。

程序参考代码如下。

```
#include<stdio. h>
#include<malloc. h>
typedef struct node
{ int data;
  struct node * next;
} Node;
void insert(Node * L, int a[ ], int n)
{
  int i;
```




```
Node *p, *q;
//将数据以插入的方式加到链表中
for(i = 0; i < n; i++)
{
    p = L;
    //找到链表中第一个比现在要插入的数据 a[i]大的数,然后在这个数前面插入 a[i]
    while(p->next != NULL)
    {
        if(p->next->data > a[i])
            break;
        p = p->next;
    }
    //插入数据
    q = (Node *)malloc(sizeof(Node));
    q->data = a[i];
    q->next = p->next;
    p->next = q;
}
}

void display(Node *L)
{
    Node *p;
    p = L->next;
    printf("链表的数据为: \n");
    //输出链表数据
    while(p != NULL){
        printf("%d ", p->data);
        p = p->next;
    }
    printf("\n");
}

void main( )
{
    int n = 10, a[ ] = {12, 3, 56, 7, 8, 15, 34, 90, 1, 10};
    Node *L;
    //建立头节点
    L = (Node *)malloc(sizeof(Node));
    L->next = NULL;
```

```

//调用函数将数据插入到 L 中并有序
insert(L, a, n);
//输出结果
display(L);
}

```

程序运行结果如图 10-2 所示。

```

链表的数据为：
1 3 7 8 10 12 15 34 56 90
请按任意键继续. . .

```

图 10-2 运行结果截图

10.3 自主编程题

1. 编程实现：从键盘输入一个职工结构体数据(包括身份证号、姓名、性别、籍贯、联系电话)，并输出。
2. 编程实现：从键盘输入 5 个学生信息(包括学号、姓名、年龄、籍贯)，并以表格方式输出结果。
3. 编程实现：现有一个学生结构体(包括学号、姓名、年龄、籍贯)数组 A[10]，编写函数 find 实现按籍贯查找并显示查找结果。
4. 编程实现：编写函数 days 实现计算给定日期是本年的第几天的功能，由主函数将结构体数据(年、月、日)传递给 days 函数，结果输出也由主函数输出。
5. 编程实现：编写程序实现用链表存储 10 个整形数据，编写函数 dele(s, x)实现删除链表 s 中值为 x 的节点。
6. 有 5 个学生，每个学生的数据包括学号、姓名、3 门课的成绩。从键盘输入 5 个学生的数据，要求输出 3 门课平均成绩，以及平均分最高的学生的数据(包括学号、姓名、3 门课成绩、平均分)。要求用 input 函数输入数据，用 avg 函数求总平均成绩，用 max 函数找出平均分最高的学生数据，数据输出由主函数实现。
7. 某学校图书馆新购进 1000 种图书，现编写一个 C 程序，将新图书的数据输入，每种图书的信息包括图书编号、书名、作者、单价、出版社、数量，并求出所有新书的总价值，然后编写一个函数实现按书名进行图书的查找操作，在主函数中调用。

10.4 参考程序

1. 参考代码。

```

#include<stdio. h>
struct person{
    char no[20];
    char name[10];

```



```
char sex[3];
char jiguan[50];
char phone[12];
};
void main( ){
    struct person d;
    printf("请输入身份证号:");
    gets(d. no);
    printf("请输入姓名:");
    gets(d. name);
    printf("请输入性别:");
    gets(d. sex);
    printf("请输入籍贯:");
    gets(d. jiguan);
    printf("请输入电话号码:");
    gets(d. phone);
    printf("数据信息为: \n");
    printf("%20s %10s %3s %20s %13s\n", "身份证号", "姓名", "性别", "籍贯", "电话号码");
    printf("%20s %10s %3s %20s %13s\n", d. no, d. name, d. sex, d. jiguan, d. phone);
}
```

2. 参考代码: 注意去掉多余的字符, 保证能正确读取数据。

```
#include<stdio. h>
typedef struct{
    int no;
    char name[10];
    int age;
    char native[20];
}Student;t
void input(Student s[ ], int n){
    int i;
    for(i=0; i<n; i++){
        printf("请输入第%d 个学生数据: \n", i+1);
        printf("学号:");
        fflush(stdin);
        scanf("%d", &s[i]. no);
        printf("姓名:");
        fflush(stdin);
```



```
        gets(s[i]. name);
        fflush(stdin);
        printf("请输入年龄:");
        scanf("%d", &s[i]. age);
        fflush(stdin);
        printf("请输入籍贯:");
        scanf("%s", s[i]. native);
    }
}

void output(Student s[ ], int n)
{
    int i;
    printf("\n 结果数据如下:");
    printf("\n%6s %8s %8s", "学号", "姓名", "成绩");
    for(i=0; i<n; i++){
        printf("\n%6d %8s %8.2f", s[i]. no, s[i]. name,s[i]. score);
    }
    printf("\n");
}

void main( ) {
    student s[5];
    int n=5;
    input(s, n);
    output(s, n);
}
```

3. 参考代码: 注意字符串的比较要用到 strcmp 函数。

```
#include<stdio. h>
#include<string. h>
typedef struct{
    int no;
    char name[10];
    int age;
    char native[20];
}Student;

void main( ){
    student s[10]={
        {1001, "张三", 20, "广西贺州"},
```



```

        {1002, "李四", 25, "广西贺州"},
        {1003, "王五", 22, "广西梧州"},
        {1004, "赵一", 29, "广西贺州"},
        {1005, "张柳", 21, "广西南宁"},
        {1006, "陈义", 30, "广西柳州"},
        {1007, "罗成", 22, "广西南宁"},
        {1008, "李世民", 21, "广西贺州"},
        {1009, "秦英", 40, "广西桂林"},
        {1010, "娜娜", 25, "广西桂林"},
    };

    int i, n = 10, flag = 0;
    char native[20];
    printf("请输入要查找的籍贯:");
    gets(native);
    for(i = 0; i < n; i++)
    {
        //if(strstr(s[i]. native, native))
        if(strcmp(native, s[i]. native) == 0)
        {
            flag = 1;
            printf("%d %s %d %s\n", s[i]. no, s[i]. name, s[i]. age, s[i]. native);
        }
    }
    if(flag == 0){
        printf("没有你要找的人!");
    }
}

```

4. 参考代码：注意对输入数据有效性的检验。

```

#include <stdio. h>
struct Day{
    int year;
    int month;
    int day;
};
int Days(struct Day d)
{
    int i;
    int days = 0;

```



```
    if(d. year% 400 == 0 || (d. year% 4 == 0 && d. year% 100 != 0))
        days++;
    switch(d. month){
        case 12: days += 30;
        case 11: days += 31;
        case 10: days += 30;
        case 9: days += 31;
        case 8: days += 31;
        case 7: days += 30;
        case 6: days += 31;
        case 5: days += 30;
        case 4: days += 31;
        case 3: days += 28;
        case 2: days += 31;
        case 1: days += d. day;
    }
    return days;
}

int main( )
{
    struct Day d;
    int n, flag = 1;
    printf("请输入一个日期:");
    scanf("%d%d%d", &d. year, &d. month, &d. day);
    if(d. month < 1 || d. month > 12){
        flag = 0;
    }
    switch(d. month)
    {
        case 4:
        case 6:
        case 9:
        case 11: if(d. day < 1 || d. day > 30) flag = 0;
                break;
        case 2: if(d. year% 400 == 0 || (d. year% 4 == 0 && d. year% 100 != 0))
                {if(d. day < 1 || d. day > 29) flag = 0;}
                else {if(d. day < 1 || d. day > 28) flag = 0;}
                break;
        default: if(d. day < 1 || d. day > 31) flag = 0;
    }
}
```



```
    }  
    if(flag==0){  
        printf("日期错误!");  
        return 0;  
    }  
    n=Days(d);  
    printf("你输入的日期是今年的第%d 天! \n", n);  
}
```

5. 参考代码：注意链表的基本操作。

```
#include<stdio. h>  
#include<malloc. h>  
typedef struct node{  
    int data;  
    struct node *next;  
} Node;  
void creat(Node *L, int a[ ], int n)  
{  
    int i;  
    Node *p, *q;  
    //将数据以头插入法的方式加到链表中  
    for(i=0; i<n; i++)  
    {  
        q=(Node *)malloc(sizeof(Node));  
        q->data=a[i];  
        q->next=L->next;  
        L->next=q;  
    }  
}  
void display(Node *L)  
{  
    Node *p;  
    p=L->next;  
    printf("链表的数据为: \n");  
    //输出链表数据  
    while(p!=NULL){  
        printf("%d ", p->data);  
        p=p->next;  
    }  
}
```



```
        printf("\n");
    }
    int dele(Node *L, int x)
    {
        Node *p, *q;
        p = L;
        while(p->next != NULL){
            if(p->next->data == x){
                q = p->next;
                p->next = q->next;
                free(q);
            }
            else{
                p = p->next;
            }
        }
    }
}

void main( )
{
    int x, n = 10, a[ ] = {12, 3, 56, 3, 8, 15, 34, 90, 1, 3};
    Node *L;
    L = (Node *)malloc(sizeof(Node));
    L->next = NULL;
    creat(L, a, n);
    display(L);
    printf("请输入要删除的数据:");
    scanf("%d", &x);
    dele(L, x);
    display(L);
}
```

6. 参考代码。

```
#include "stdio. h"
struct student
{ char num[6];
  char name[20];
  float score[3];
  float aver;
}stu[5];
```




```
void input(struct student stu[ ])
{
    int i;
    printf("please enter the information of student:\n");
    for(i=0; i<5;i++)
        scanf("%s %s %f %f %f", &stu[i]. num, &stu[i]. name, &stu[i]. score[0], &stu[i].
score[1], &stu[i]. score[2]);
}
void avg(struct student stu[ ])
{
    int i, j;
    float sum;
    for(j=0; j<3;j++)
    {
        sum=0;
        for(i=0; i<5; i++)
            sum+=(stu[i]. score[j]);
        stu[i]. aver = sum/3. 0;
    }
}
int max(struct student stu[ ])
{
    int i, m;
    for(i=0; i<5; i++)
    {
        m=0;
        stu[i]. aver = (stu[i]. score[0]+stu[i]. score[1]+stu[i]. score[2])/3. 0;
        if(stu[i]. aver>stu[m]. aver)
            m=i;
    }
    return m;
}
void main( )
{
    int i, m;
    input(stu);
    avg(stu);
    m = max(stu);
    for (i=0; i<5; i++)
        printf("第%d 科平均成绩 avg:%f\n", i+1, stu[i]. aver);
    printf("num:%s \nname:%s \nscore:%f %f %f \navg:%f\n", stu[m]. num,
stu[m]. name, stu[m]. score[0], stu[m]. score[1], stu[m]. score[2], stu[m]. aver);
}
```

7. 参考代码。

```
#include "stdio. h"
#include "string. h"
```



```
struct book
{
    int num;
    char shuming[40];
    char zhuzhe[20];
    float danjia;
    char zhuzans[40];
    int count;
};

void locate(struct book b[ ], char ch[ ], int n)
{
    int k;
    for(k=0; k<n; k++)
        if(strcmp(b[k]. shuming, ch)==0) break;
    if(k<n)
    {
        printf("查找成功,此书的信息如下:");
        printf("%d, %s, %s", b[k]. num, b[k]. shuming, b[k]. zhuzhe);
        printf("%f, %s, %d", b[k]. danjia, b[k]. zhuzans, b[k]. count);
    }
    else printf("查无此书");
}

void main( )
{
    struct book b[1000];
    int i;
    char ch[40];
    float sum=0;
    for(i=0; i<1000; i++)
    {
        scanf("%d, %s, %s", &b[i]. num, b[i]. shuming);
        scanf("%s, %f", b[i]. zhuzhe, &b[i]. danjia);
        scanf("%s, %d", b[i]. zhuzans, &b[i]. count);
        sum = sum + b[i]. danjia * b[i]. count;
    }
    scanf("%s", ch);
    locate(b, ch, 1000);
}
```

第 11 章 实验十：文件类型的应用

11.1 实验目的

- (1) 掌握 C 语言中文件和文件指针的概念。
- (2) 掌握文件打开和关闭的概念和方法，理解各种文件的使用方式。
- (3) 掌握有关文件读写、定位等操作函数的使用。

11.2 实验案例分析

1. 编写程序实现将 10 个学生成绩数据从键盘输入并保存在文件 student.txt 中。

分析：可以先定义好结构体数据，直接放在数组中，然后再写入文件中。在文件操作时，要注意先用写的方式(w 格式)打开文件，然后可以选择使用字符流的方式写入，也可以选择块写方式写入，最后记得要关闭文件。

程序参考代码如下。

```
#include<stdio. h>
typedef struct{
    int no;
    char name[10];
    float score;
}Student;
void main( )
{ Student s[10]={ {1001, "张三", 80}, {1002, "李四", 90}, {1003, "王五", 78}, {1004, "赵一", 65}, {1005, "张柳", 60}, {1006, "陈义", 75}, {1007, "罗成", 85}, {1008, "李世民", 80}, {1009, "秦英", 90}, {1010, "娜娜", 80}};
    int i;
    FILE * fp;
    //打开文件
    if((fp = fopen("student. txt", "w+")) == NULL)
    { printf("文件打开失败!");
        return 0;
    }
    //向文件写入数据
    for(i=0; i<10; i++)
```

```

        {fprintf(fp, "%d %s %.2f\n", s[i]. no, s[i]. name, s[i]. score);
        }
        fclose(fp);
    }

```

运行结果文件内容如图 11-1 所示。

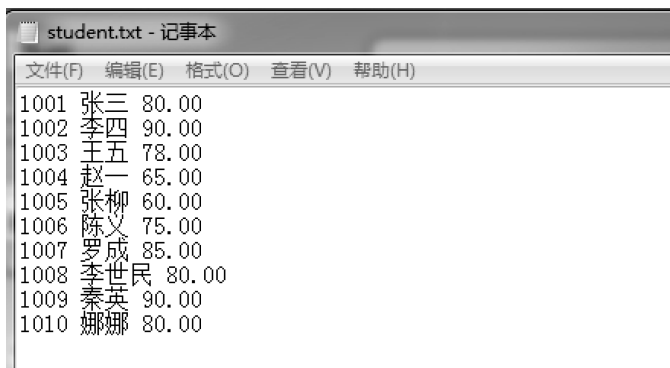


图 11-1 文件内容截图

2. 编写程序实现将文件 student.txt 中的第 n 个学生数据删除。

分析：文件操作需要先打开文件，然后读取数据到数组中，在数组中进行删除，最后再将数组内容写回文件。读取数据的时候因为不知道文件有多少个数据，所以每读取一个数据都需要判断文件是否结束，如果没结束就继续读取下一个数据。

参考程序代码如下。

```

#include<stdio. h>
typedef struct{
    int no;
    char name[10];
    float score;
}Student;
void main( )
{
    Student s[10];
    int i=0, n, j;
    FILE *fp;
    printf("请输入要删除第几个数:");
    scanf("%d",&n);
    if((fp = fopen("student. txt", "r+")) == NULL){
        printf("文件打开失败!");
        return 0;
    }
    fscanf(fp,"%d %s %f\n",&s[i]. no,s[i]. name,&s[i]. score);

```



```

while(!feof(fp))
{
    i++;
    fscanf(fp,"%d %s %f\n",&s[i]. no,s[i]. name,&s[i]. score);
}
j=i;
if(n<1 || n>j+1){
    printf("删除位置错误!");
    return 0;
}
for(i=n;i<j;i++)
{
    s[i-1]=s[i];
}
j--;
rewind(fp);
for(i=0;i<j;i++)
{
    fprintf(fp,"%d %s %f\n",s[i]. no,s[i]. name,s[i]. score);
}
fclose(fp);
}

```

程序运行后文件内容如图 11-2 所示。

文件(F)	编辑(E)	格式(O)	查看(V)	帮助(H)
1001	张三	80.000000		
1002	李四	90.000000		
1003	王五	78.000000		
1004	赵一	65.000000		
1006	陈义	75.000000		
1007	罗成	85.000000		
1008	李世民	80.000000		
1009	秦英	90.000000		
1009	秦英	90.000000		

图 11-2 文件内容截图

11.3 自主编程题

1. 编程实现从键盘输入一行字符串，并将内容保存在文件 data.txt 中。
2. 编程实现：现有一文本文件 student.txt，文件中每行一个学生成绩(包括学号、姓名、高数、英语、C 语言)，编写程序实现将数据从文件中读取并显示在屏幕上。
3. 编程实现：查找文件 student.txt 中姓名为 name(键盘输入)的学生信息。
4. 有 5 个学生，每个学生有 3 门课成绩，从键盘上输入以上数据(包括学号，姓名，

数学、语文、英语三门课成绩), 计算出每个学生的平均成绩。要求: 将原有数据和计算出的平均分数放在磁盘 stu. txt 中, 并且屏幕上可以浏览 5 个学生的成绩。

5. 将上题“stud. txt”文件中的学生数据, 按平均分进行排序处理, 将已排序的学生数据存入一个新文件“stu_ sort”中。

11.4 参考程序

1. 参考代码。

```
#include<stdio. h>
void main( )
{
    char s[100];
    FILE *fp;
    if((fp = fopen("student. txt", "w")) == NULL){
        printf("文件打开失败!");
        return 0;
    }
    printf("请输入一个字符串:");
    gets(s);
    fputs(s, fp);
    fclose(fp);
}
```

2. 参考代码。

```
#include<stdio. h>
typedef struct{
    int no;
    char name[10];
    float score;
}Student;
void main( )
{
    Student s[15];
    int i=0, n=2;
    FILE *fp;
    if((fp = fopen("stu. txt", "r")) == NULL){
        printf("文件打开失败!");
        return 0;
    }
}
```



```
fscanf(fp, "%d%s%f", &s[i]. no, s[i]. name, &s[i]. score);
while(!feof(fp))
{
    i++;
    fscanf(fp, "%d%s%f", &s[i]. no, s[i]. name, &s[i]. score);
}
n=i;
for(i=0; i<n; i++){
    printf("%d %s %.2f\n", s[i]. no,s[i]. name, s[i]. score);
}
fclose(fp);
}
```

3. 参考代码。

```
#include<stdio. h>
#include<string. h>
typedef struct{
    int no;
    char name[10];
    float score;
}Student;

void main( )
{
    Student s[15];
    int i=0, n=2;
    char name[20];
    FILE *fp;
    if((fp = fopen("stu. txt", "r")) == NULL){
        printf("文件打开失败!");
        return 0;
    }
    printf("请输入要查找的姓名:");
    gets(name);
    fscanf(fp, "%d%s%f", &s[i]. no, s[i]. name, &s[i]. score);
    while(!feof(fp))
    {
        i++;
        fscanf(fp, "%d%s%f", &s[i]. no, s[i]. name, &s[i]. score);
    }
}
```



```
    }  
    n=i;  
    for(i=0; i<n; i++){  
        if(strcmp(name, s[i]. name)==0)  
            printf("%d %s %. 2f\n", s[i]. no, s[i]. name, s[i]. score);  
    }  
    fclose(fp);  
}
```

4. 参考代码。

```
#include "stdio. h"  
struct student  
{ char num[6];  
  char name[8];  
  int score[3];  
  float avr;  
} stu[5];  
void main( )  
{ int i, j, sum;  
  FILE *fp;  
  for(i=0; i<5; i++)  
{ printf("\n please input No. %d score:\n", i);  
  printf("stuNo:");  
  scanf("%s", stu[i]. num);  
  printf("name:");  
  scanf("%s", stu[i]. name);  
  sum = 0;  
  for(j=0; j<3; j++)  
{ printf("score %d. ", j+1);  
  scanf("%d", &stu[i]. score[j]);  
  sum += stu[i]. score[j];}  
  stu[i]. avr = sum/3. 0;  
  }  
  fp = fopen("stud. txt", "w");  
  for(i=0; i<5; i++)  
  if(fwrite(stu[i], sizeof(struct student), 1, fp)!=1)  
  printf("file write error\n");  
  fclose(fp);  
}
```

5. 参考代码。



```
#include<stdio. h>
#include<stdlib. h>
#include<string. h>
struct student
{ int num;
  char name[10];
  float score[3];
  float avr;
} stu[5], temp;
void main( )
{ FILE *fp;
  int i, j;
  if((fp = fopen("stud. txt", "rb")) == NULL)
  {printf("can not open file\n");
   exit(0);
  }
  printf("\n 学号 姓名 成绩 1 成绩 2 成绩 3 平均成绩:\n");
  for(i=0; i< 5; i++)
  { fread(&stu[i],sizeof(struct student), 1, fp);
    printf("%d, %s, %. 2f, %. 2f, %. 2f, %. 2f\n", stu[i]. num, stu[i]. name, stu[i]. score[0],
    stu[i]. score[1], stu[i]. score[2], stu[i]. avr);
  }
  fclose(fp);
  for(i=0; i< 4; i++)
  for(j=i+1; j< 4; j++)
  if(stu[i]. avr< stu[j]. avr)
  {temp = stu[i]; stu[i] = stu[j]; stu[j] = temp; }
  printf("\n 排序后:");
  if((fp = fopen("stu_sort. txt", "w")) == NULL)
  { printf("can not open file\n");
   exit(0);
  }
  for(i=0; i< 5; i++)
  if(fwrite(stu[i], sizeof(struct student), 1, fp) != 1)
  { printf("file write error\n");
   fclose(fp);
  }
}
```



```
if((fp=fopen("stu_sort.txt", "r"))==NULL)
{ printf("can not open file\n");
  exit(0);
}
for(i=0; i< 5; i++)
{ fread(&stu[i], sizeof(struct student), 1, fp);
  printf("%d, %s, %.2f, %.2f, %.2f, %.2f\n", stu[i].num, stu[i].name, stu[i].score[0],
  stu[i].score[1], stu[i].score[2], stu[i].avr);
  fclose(fp);
}}
```

第 12 章 综合实训

12.1 实训项目一：学生成绩管理软件

通过编写一个简单的学生成绩管理系统，培养学生初步的独立分析和设计能力，使学生了解应用程序设计解决实际问题的具体过程和实现方法，掌握应用 C 语言进行程序设计的基本规范；重点掌握 C 语言的程序编写、函数设计、算法设计和程序调试方法；初步掌握系统开发过程的问题分析、系统设计、程序编码、测试等基本方法和技能。

12.1.1 问题描述

将 C 语言程序设计中的程序控制结构、函数、数组、结构体、文件等知识串联起来，设计一个简单的学生成绩管理系统，实现学生成绩的增、删、查、改等。

12.1.2 系统分析

对学生成绩管理系统进行分析，根据系统需要实现的学生成绩的输入输出，学生成绩信息的增加、修改、删除、查询，学生成绩总分和平均值的计算以及学生成绩的排序等，将系统划分为成绩录入、成绩修改、成绩删除、成绩查询、成绩统计、成绩排序等 7 个功能模块。系统功能模块如图 12-1 所示。

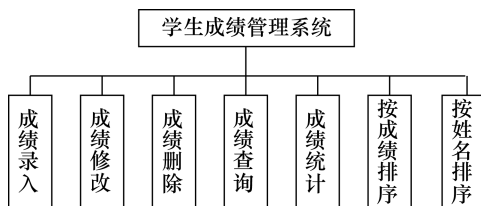


图 12-1 学生成绩管理系统功能模块图

12.1.3 系统设计

1. 数据结构定义

定义学生结构体 Student，结构体中包含学生姓名、三门功课的成绩、总分等数据成员。结构体定义如下。

```
struct Student
{
    char name[20];
    int math, english, chinese, total;
};
```

定义全局结构体数组 st[]，用来存储多个学生的信息，st[] 的定义如下。

```
struct Student st[N];
```

其中，N 表示学生人数的最大值，N 可以根据实际情况修改。

定义全局变量 length 用来表示学生实际人数，length 的定义如下。

```
int length;
```

2. 功能函数设计

运用结构化程序设计“自顶向下，逐步求精”的思想方法，对学生成绩管理系统进行分析，根据系统所要完成的功能，将系统划分为文件读写、添加、删除、修改、查询、统计、输出、排序等函数，并对每个函数进一步分解和细化。

(1) main()：主函数。

main()函数首先调用 ReadFile()函数从 student.txt 文件读入学生信息，然后进入循环，用户输入功能选项，根据用户的输入，利用 switch 多分支选择结构语句，有选择地运行相应功能模块，进入相应功能，输入 0 时，退出循环。系统功能选择如图 12-2 所示。

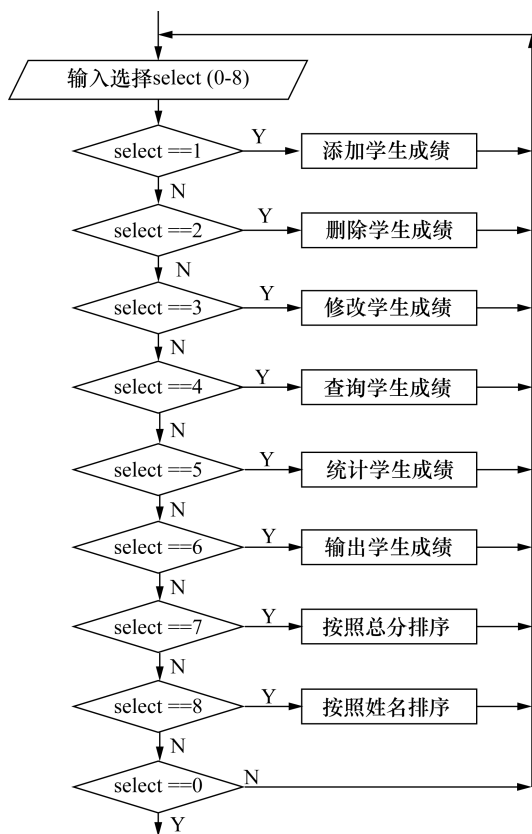


图 12-2 系统功能选择流程图

(2) AddStudent()：添加学生成绩。

AddStudent()中输入学生姓名、成绩等信息保存到数组 st 中，输入完成后，调用函数 SaveFile()将数组 st 中的数据写入 student.txt 文件。

(3) DelStudent()：删除学生成绩。



DelStudent()中输入学生姓名,在数组 st[]中查询该学生,将该学生的数据从数组 st[]中删除,最后调用函数 SaveFile()将数组 st[]中的数据写入 student.txt 文件。

(4) UpdateStudent(): 修改学生成绩。

UpdateStudent()中输入学生姓名,在数组 st 中查询该学生,修改数组 st[]中的数据,最后调用函数 SaveFile()将数组 st[]中的数据写入 student.txt 文件。

(5) SearchStudent(): 查询学生成绩。

SearchStudent()中输入学生姓名,在数组 st[]中查询,并显示该学生的信息。

(6) Aaverage(): 统计学生成绩。

Aaverage()统计数组 st[]中的学生成绩,并输出每门功课的平均成绩。

(7) DispStudent(): 输出学生成绩。

DispStudent()输出保存在数组 st[]中的学生成绩。

(8) SortStudent1(): 按总分降序对学生成绩排序。

SortStudent1()按总分降序对保存在数组 st[]中的学生成绩排序。

(9) SortStudent2(): 按姓名升序对学生成绩排序。

SortStudent2()按姓名升序对保存在数组 st[]中的学生成绩排序。

(10) void SaveFile(): 将学生成绩写入文件 student.txt 中。

SaveFile()将保存在数组 st[]中的学生成绩写入文件 student.txt 中。

(11) void ReadFile(): 从文件 student.txt 中读入学生成绩。

ReadFile()从文件 student.txt 中读入学生成绩,保存到数组 st[]。

12.1.4 程序代码

```
#include<stdio. h>
#include<string. h>
#define N 100
void AddStudent( );           /* 添加学生成绩 */
void DelStudent( );           /* 删除学生成绩 */
void UpdateStudent( );        /* 修改学生成绩 */
void SearchStudent( );        /* 查询学生成绩 */
void Aaverage( );             /* 统计并输出每门功课的平均成绩 */
void DispStudent( );          /* 输出学生成绩 */
void SortStudent1( );          /* 按总分降序 */
void SortStudent2( );          /* 按姓名升序 */
void SaveFile( );             /* 学生成绩写入文件 abc. txt 中 */
void ReadFile( );             /* 从文件 abc. txt 中读入学生成绩 */
struct Student
{
    char name[20];
    int math, english, chinese, total;
```



```
};  
struct Student st[N];  
int length;  
int main( )  
{  
    int select;  
    ReadFile( );  
    printf("\n\t\t 已从文件 d:\\student. txt 读入 %d 个学生的数据\n", length);  
    while ( 1 )  
    {  
        printf("\n\n\t\t===== 欢迎使用学生成绩管理系统 ===== \n");  
        printf("\n");  
        printf("\t\t\t 1. 添加学生成绩\n");  
        printf("\t\t\t 2. 删除学生成绩\n");  
        printf("\t\t\t 3. 修改学生成绩\n");  
        printf("\t\t\t 4. 查询学生成绩\n");  
        printf("\t\t\t 5. 统计学生成绩\n");  
        printf("\t\t\t 6. 输出学生成绩\n");  
        printf("\t\t\t 7. 按照总分降序\n");  
        printf("\t\t\t 8. 按照姓名升序\n");  
        printf("\t\t\t 0. 保存数据、退出系统\n");  
        printf("\n");  
        printf("\t\t 请输入你的选择:");  
        scanf("%d", &select);  
        switch(select)  
        {  
            case 1:  AddStudent( );      break;  
            case 2:  DelStudent( );      break;  
            case 3:  UpdateStudent( );  break;  
            case 4:  SearchStudent( );  break;  
            case 5:  Avearage( );        break;  
            case 6:  DispStudent( );    break;  
            case 7:  SortStudent1( );   break;  
            case 8:  SortStudent2( );   break;  
            case 0:  SaveFile( );      return 0;  
        }  
    }  
}
```



```
/****      添加学生成绩      *****/
void AddStudent( )
{
    struct Student t;
    printf("\n 请输入学生姓名: \n");
    scanf("%s", t. name);
    printf("请输入学生三门功课的成绩: \n");
    scanf("%d%d%d", &t. math, &t. english, &t. chinese);
    t. total=t. math+t. english+t. chinese;
    st[length++]=t;
    SaveFile( );
    printf("\n 添加成功! 目前共%d 条数据\n\n", length);
}
/****      删除学生成绩      ****/
void DelStudent( )
{
    char name[20];
    int i;
    printf("请输入要删除学生的姓名: \n");
    scanf("%s", name);
    for (i=0; i<length; i++)
        if( strcmp(st[i]. name, name)==0 ) break;
    for( ; i<length- 1; i++)
        st[i]=st[i+1];
    length- ;
    SaveFile( );
    printf("\n 删除成功! 目前共%d 条数据\n\n", length);
}
/*****      修改学生成绩      ****/
void UpdateStudent( )
{
    struct Student t;
    int i;
    printf("请输入学生姓名: \n");
    scanf("%s", &t. name);
    printf("请输入学生三门功课的成绩: \n");
    scanf("%d%d%d", &t. math, &t. english, &t. chinese);
    t. total=t. math+t. english+t. chinese;
    for (i=0; i<length; i++)
```



```
        if( strcmp(st[i]. name, t. name)==0 )   break;
st[i]=t;
    SaveFile( );
    printf("\n 修改成功! 目前共%d 条数据\n\n", length);
}
/*****      根据学生姓名进行查询      *****/
void SearchStudent( )
{
    char name[20];
    int i;
    printf("\n 请输入要查找学生的姓名: \n");
    scanf("%s", name);
    for (i=0; i<length; i++)
        if( strcmp(st[i]. name, name)==0 )
        {
            printf("\n 学生姓名: %s\n", st[i]. name);
            printf("三门功课成绩\n");
            printf("数学: %d\n", st[i]. math);
            printf("语文: %d\n", st[i]. chinese);
            printf("英语: %d\n", st[i]. english);
            printf("总分: %d\n\n", st[i]. total);
            break;
        }
}
/****      分别统计并输出每门功课的平均成绩      *****/
void Aaverage( )
{
    double  mathSum, englishSum, chineseSum;
    int i;
    mathSum = englishSum = chineseSum = 0;
    for (i=0; i<length; i++)
    {
        mathSum+= st[i]. math;
        chineseSum+= st[i]. chinese;
        englishSum+= st[i]. english;
    }
    printf("\n \t\t 三门功课的平均成绩: \n");
    printf("\t\t 语文平均分: %5. 2f\n", mathSum/length);
```




```

printf("\t\t 数学平均分:%5.2f\n", chineseSum/length);
printf("\t\t 英语平均分:%5.2f\n\n", englishSum/length);
}
/* 输出学生成绩 */
void DispStudent( )
{
    int i;
    printf("\n");
    printf("\t\t 学生姓名 | 数学 | 语文 | 英语 | 总分 | \n");
    for (i=0; i<length; i++)
    { printf("\t\t %-8s |%4d |%4d |%4d | %4d | \n",
        st[i]. name, st[i]. math, st[i]. chinese, st[i]. english, st[i]. total);
    }
}
/***** 按照总分降序 *****/
void SortStudent1( )
{
    int i, j, max, k;
    struct Student t;
    for (i=0; i<length- 1; i++)
    {
        max=st[i]. total, k=i;
        for (j=i+1; j<length; j++)
            if ( max<st[j]. total) max=st[j]. total, k=j;
        t=st[i]; st[i]=st[k]; st[k]=t;
    }
    printf("\t\t***** 按总分从大到小排序 ***** \n");
    DispStudent( );
    SaveFile( );
}
/***** 按照姓名升序 *****/
void SortStudent2( )
{
    int i, j, k;
    char min[20];
    struct Student t;
    for (i=0; i<length- 1; i++)
    { strcpy(min, st[i]. name), k=i;
        for (j=i+1; j<length; j++)
            if ( strcmp(min, st[j]. name)>0) strcpy(min, st[j]. name), k=j;
        t=st[i]; st[i]=st[k]; st[k]=t;
    }
}

```



```
    }
    printf("\t\t*****      按姓名字母升序      ***** \n");
    DispStudent( );
    SaveFile( );
}
/**      将学生成绩写入文件 student.txt 中      ***/
void SaveFile( )
{
    FILE *fp;
    int i=0;
    fp=fopen("D:\\student.txt", "w");
    for (i=0; i<length; i++)
    {
        fprintf( fp, "%s ", st[i]. name);
        fprintf( fp, "%d ", st[i]. math);
        fprintf( fp, "%d ", st[i]. chinese);
        fprintf( fp, "%d ", st[i]. english);
        fprintf( fp, "%d", st[i]. total);
    }
    fclose(fp);
}
/**      从文件 student.txt 中读入学生成绩      ***/
void ReadFile( )
{
    FILE *fp;
    struct Student t;
    fp=fopen("D:\\student.txt", "r");
    length=0;
    while( !feof(fp) )
    {
        fscanf( fp, "%s", st[length]. name);
        fscanf( fp, "%d", &st[length]. math);
        fscanf( fp, "%d", &st[length]. chinese);
        fscanf( fp, "%d", &st[length]. english);
        fscanf( fp, "%d", &st[length]. total);
        length++;
    }
    fclose(fp);
}
```



12.1.5 调试运行

运行程序，进入系统功能选择界面，如图 12-3 所示。

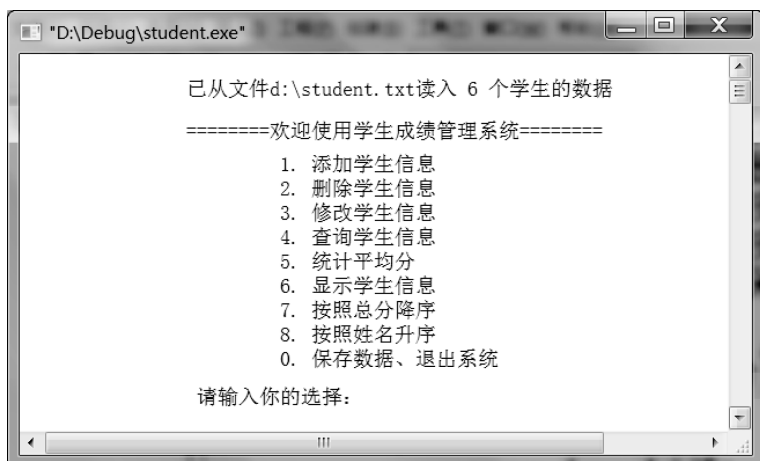


图 12-3 系统功能选择界面

按总分降序排序后，在主界面中选择输出学生成绩信息，如图 12-4 所示。

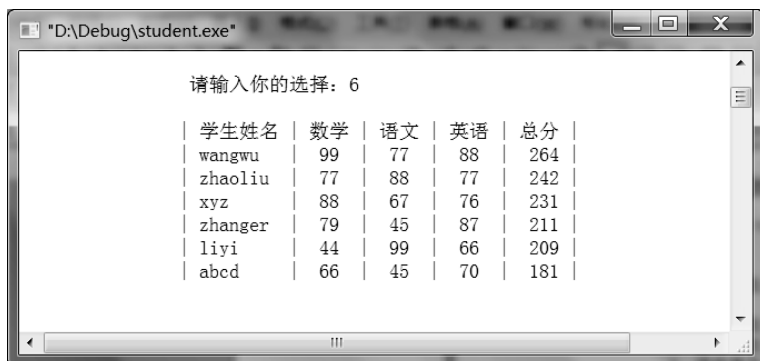


图 12-4 按总分降序输出学生成绩信息

按姓名升序排序后，在主界面中选择输出学生成绩信息，如图 12-5 所示。

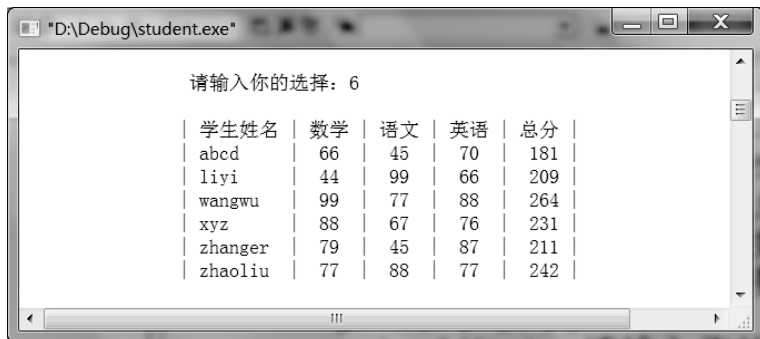


图 12-5 按姓名升序输出学生成绩信息

12.2 实训项目二：贪吃蛇游戏设计

12.2.1 问题描述

贪吃蛇游戏是一个经典小游戏，有一条蛇在一个被围墙封闭的区域内游动，同时区域内会随机出现一个食物，通过键盘上的四个按键可控制蛇分别向上、下、左、右四个方向移动，蛇头碰到食物，则表示食物被吃掉，同时蛇身增长一节，接着又出现食物，等待蛇来吃，如果蛇在移动中碰到墙或蛇头碰到自己的身体，则游戏结束。

设计贪吃蛇游戏的主要目的是让大家夯实 C 语言基础，训练编程思维，培养解决问题的思路，领略多姿多彩的 C 语言。

12.2.2 系统分析

根据贪吃蛇游戏规则和功能需要，系统需要设计以下几方面的内容：一是设计一个游戏区间，中间是空的，四周有墙体；二是需要设计出一条由蛇头和蛇身组成的蛇；三是设计出食物，并且在蛇吃掉食物之后将蛇的身体变长，而且还要重新生成一个食物；四是要实现蛇的移动，使蛇能在键盘控制下向上、下、左、右四个方向移动，当蛇撞到墙或者撞到自己的身体就会死亡；五是要用键盘控制蛇的运动方向；六是要统计并输出最后得分。游戏主要功能模块如图 12-6 所示。

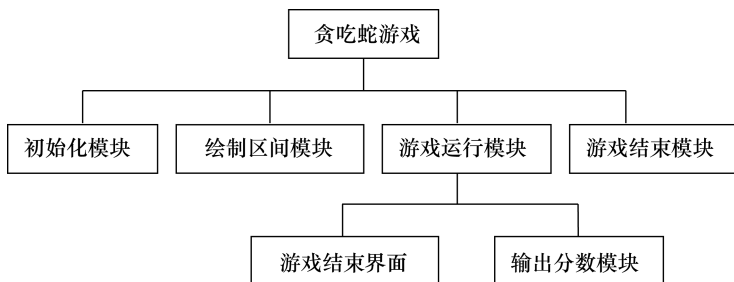


图 12-6 游戏总体功能模块图

12.2.3 系统设计

1. 设计思路

根据系统分析后，可确定贪吃蛇游戏的设计思路。

(1) 首先创建一张地图，中间是空的，四周有墙体，代表游戏的区域，四周的墙体用符号“■”表示，整个地图用一个坐标系建立起来。

(2) 在游戏区域内设计一条蛇，蛇头和蛇身可用符号“★”表示，每个“★”代表一节蛇身，初始化时设定蛇的初始长度(如 5，即用 5 个“★”号表示该条蛇)，以后每吃到一个食物，身体增加一节。

(3) 在游戏区域内设计食物，食物用符号“●”表示，当蛇吃掉一个食物之后要将蛇的



身体增加一节，然后重新生成一个食物。生成食物时，需要产生两个随机数作为食物的位置坐标，而且这个坐标要在一定的范围内，不能在围墙上，也不能和蛇身的坐标重叠。

(4) 设计蛇的运动函数，蛇是通过一列在区域内特定位置显示的“★”表示的，而每个“★”的位置是由其坐标确定的，蛇的运动实际上就是要不断地改变“★”的坐标。蛇运动要从蛇头开始，将蛇头移动到下一个坐标之后，后面的每节蛇身依次覆盖前一节就完成了—个单位的移动。因此，只需要考虑蛇头的运动方向就行了。可设定一个方向变量 status，status = 1 即向上，status 的值 1/2/3/4 分别代表上、下、左、右，方向的改变与否通过 GetAsyncKeyState() 函数来判断。假设蛇向上走，即蛇头向上，其坐标只需 X 不变，Y-1 就好了（注意这里的坐标系是类似数学上第四象限的坐标系，所以 X 轴在最上面），然后将第二节换成前一次蛇头所在的坐标，第三节坐标变成前一次第二节的坐标，以此类推，蛇就往前移动了一个单位。

(5) 设计判断蛇头是否撞到墙或撞到自己身体的函数，只需要判断蛇头的坐标是否和墙的坐标重合，或者和蛇身某一节坐标重合。所以在蛇移动之后可用蛇头的坐标去遍历蛇身的所有坐标，同时判断蛇头坐标的 X 或者 Y 是否大于或者小于地图边界，这样就能知道是否发生碰撞，发生碰撞之后直接用 break 结束游戏。

2. 主要数据结构

(1) 蛇的运动状态用四个常量表示，U：上；D：下；L：左 R：右。

```
#define U 1
#define D 2
#define L 3
#define R 4
```

(2) 蛇用一条单链表表示，每个节点表示一节蛇身，x, y 表示每一节的坐标值，定义如下。

```
typedef struct SNAKE {
    int x;
    int y;
    struct SNAKE *next;
} snake;
```

(3) 程序中需要使用到的全局变量如下。

```
int score=0, add=10;           //总得分与每次吃食物得分。
int status, sleeptime=200;     //每次运行的时间间隔
snake *head, *food;           //蛇头指针,食物指针
snake *q;                      //遍历蛇的时候用到的指针
int endgamestatus=0; //游戏结束的情况,1:撞到墙;2:咬到自己;3:主动退出游戏。
```

3. 主要函数与算法

(1) void color(int b)。

color() 是一个颜色函数，通过函数的参数值 b 的取值(可取 0~15)设置不同的颜色，函

数体主要是通过调用 `SetConsoleTextAttribute` 函数, 这是一个可以在 API 中设置控制台窗口字体颜色和背景色的计算机函数, 靠一个字节的低四位控制前景色, 高四位来控制背景色。

(0=黑色 1=蓝色 2=绿色 3=浅绿色 4=红色 5=紫色 6=黄色 7=白色 8=灰色 9=淡蓝色 10=淡绿色 11=淡浅绿色 12=淡红色 13=淡紫色 14=淡黄色 15=亮白色)

(2) `void pos(int x, int y)`。

`pos()` 函数是一个设置光标位置的函数, 在创建的游戏区域里找一个坐标, 贪吃蛇上下左右移动和食物随机出现都需要定义坐标, 主要是通过调用 `GetStdHandle()` 和 `SetConsoleCursorPosition()` 两个函数来实现光标的坐标定位。`GetStdHandle()` 返回标准的输入、输出或错误的设备的句柄, 而 `SetConsoleCursorPosition()` 是 API 中定位光标位置的函数。

(3) `void creatMap()`。

`creatMap()` 是一个创建游戏地图的函数, 通过在给定的区域四周设置墙体(用符号“■”将四周封闭)形成一个封闭的游戏区间。

算法思路: 通过两条循环语句分别打印上下左右边框。

步骤一: 打印上下边框。

```
for(i=0; i<M; i+=2) { Pos(i, 0); printf("■"); Pos(i, 26); printf("■"); }
```

步骤二: 打印左右边框。

```
for(i=1; i<N; i++) { Pos(0, i); printf("■"); Pos(56, i); printf("■"); }
```

(4) `void initsnake()`。

`initsnake()` 是一个初始化蛇身的函数, 用“★”表示蛇头与蛇身, 初始时设置长度可较小。

算法思路: 利用 `malloc` 函数申报节点空间, 创建一条单链表, 长度自行设定, 同时设置每个节点的坐标, 并在指定位置输出每个节点(用“★”表示节点), 输出的图形表示一条蛇。

(5) `int biteself()`。

`biteself()` 是一个判断是否咬到了自己的函数, 当蛇头坐标与蛇身坐标重合时表示咬到了自己。

算法思路: 游戏中蛇是用一条单链表表示的, 节点的数据就是该节点在游戏区域内的坐标, 因此判断蛇是否咬到自己, 只要判断蛇头节点的坐标是否与蛇身节点的坐标相等就行了。

步骤一: 设一搜索指针 `snake * self` 指向蛇头的下一节点(蛇身)即 `self=head->next`,

步骤二: 利用一个循环语句依次比较蛇头与蛇身节点坐标, 若相等则咬到自己, 若比较到结尾, 没有坐标相等的情况, 则蛇没有咬到自己。

```
while(self != NULL)
{ if(self->x == head->x && self->y == head->y)
  { return 1; }    //返回 1, 表示咬到自身
  self = self->next;
}
return 0;    //返回 0, 表示未咬到自身
```



(6) void createfood()。

createfood()是一个随机产生食物的函数，食物用符号“●”表示，食物出现的位置(坐标 x, y)由随机函数产生，因蛇身也是由坐标决定出现的位置，因此食物也可借用蛇的结构体类型，不再另外定义食物类型。

算法思路：首先，食物用符号“●”表示，类型与蛇节点类似，也是用坐标标识它在游戏区域中的位置，可借用已定义的蛇节点的结构体类型来定义食物节点；其次考虑食物随机出现的位置，需调用随机函数 rand()来产生随机数；最后，食物出现的位置必须有效，即不能与蛇身、墙体四周重叠，因此要对食物节点坐标进行位置控制，可通过模运算控制坐标取值，保证食物不出现在墙体内，再通过食物节点与蛇身节点逐一比较，保证食物不与蛇身重合。

步骤一：生成一个食物节点 snake * food_1; food_1 = (snake *) malloc (sizeof (snake))。

步骤二：随机产生食物节点坐标，并通过模运算控制坐标取值不在墙体内。

```
srand((unsigned)time(NULL));
while((food_1->x%2) != 0) //保证其为偶数,使得食物能与蛇头对齐
    food_1->x = rand() % 52 + 2;
    food_1->y = rand() % 24 + 1;
```

步骤三：搜索蛇身，判定食物节点是否与蛇节点重合，若重合则重新生成食物节点。

```
q = head;
while(q->next == NULL)
{ if(q->x == food_1->x && q->y == food_1->y) //判断蛇身是否与食物重合
    { free(food_1);
      createfood( );
    }
    q = q->next;
}
```

步骤四：确定食物节点坐标，输出食物。

```
Pos(food_1->x, food_1->y);
food = food_1; color(12);
printf("●"); color(11);
```

(7) void cantcrosswall()。

cantcrosswall()是一个判断蛇是否穿墙的函数，若蛇头坐标 x, y 有一个与墙体上下左右边界值相同，则不能继续游戏。

算法思路：只需要判断蛇头坐标的取值不能与墙体四周边界值(x=0, x=56, y=0, y=26)相等就行，若相等就说明蛇已穿墙，此时应结束游戏。

步骤一：判断蛇头坐标是否与墙体边界值相等。

```
if(head->x==0 || head->x==56 || head->y==0 || head->y==26) endgamestatus=1;
```



步骤二：若相等则调用 `endgame()` 函数结束游戏。

(8) `void snakemove()`。

`snakemove()` 是一个控制蛇向上、下、左、右运动的函数，通过判定按键方向，控制蛇的运行方向，在运行过程中碰到食物就吃掉，同时身体增加一节，得分相应增加。

算法思路：蛇是朝上、下、左、右四个方向运动，为了看到蛇的运动轨迹，就要在不同的位置上输出蛇，即打印出表示蛇头与蛇身的符号“★”，同时若蛇在运动中吃了食物，则蛇身增加一节，没吃食物则保持原样。主要步骤如下。

步骤一：生成一个新节点，根据运动方向设置节点坐标

```
snake *nexthead; nexthead = (snake*)malloc(sizeof(snake));
if(status == U)           //判断运动方向, U 为向上运动
{ nexthead->x = head->x;
  nexthead->y = head->y - 1; } //设置新节点坐标
```

步骤二：判断新节点位置是否有食物，若有食物则进行如下操作。

```
if(nexthead->x == food->x && nexthead->y == food->y)
{ nexthead->next = head; //新节点与蛇头连接
  head = nexthead;      //头指针指向新蛇头
  q = head;
  while(q != NULL) //输出整条蛇, 因吃了食物, 蛇身增加了一节
  { Pos(q->x, q->y);
    printf("★");
    q = q->next;
  }
  score = score + add; //分数增加
  createfood();        //重新生成一个食物
}
```

步骤三：判断新节点位置没有食物，进行如下操作。

```
if(!(nexthead->x == food->x && nexthead->y == food->y))
{ nexthead->next = head; //新节点与蛇头连接
  head = nexthead;      //头指针指向新蛇头
  q = head;
  while(q->next->next != NULL)
  { Pos(q->x, q->y);
    printf("★");
    q = q->next;
  } //因增加了新的节点, 但蛇的长度不变, 最后一个节点不输出
  Pos(q->next->x, q->next->y);
  printf(" ");
}
```




```

        free(q->next);
        q->next = NULL; //删除最后一个节点
    }

```

步骤四：判断在运动过程中是否会咬到自己，咬到则结束游戏。

```

        if(biteself() == 1)
        { endgamestatus = 2;
          endgame();
        }

```

(9) void pause()。

pause()是一个游戏暂停的函数，通过 sleep 函数控制蛇的运动速度，按空格键暂停游戏。

(10) void gamecircle()。

gamecircle()是一个控制游戏的函数，在本函数中设置控制运动方向的状态字，同时通过一个无限循环(while(1))调用 snakemove()，控制整个游戏运行，直到结束。

算法思路：首先，为了便于操作游戏，在屏幕右侧显示相应信息，如操控键的提示(上下左右分别用↑、↓、←、→控制，加减速用 F1、F2 控制，空格键控制开始与暂停)，游戏得分等，这类信息通过调用标准函数和输出函数就能在指定位置显示所需要的信息；其次，利用一条无限循环语句[while(1)语句]使得游戏可持续运行，直到结束；最后，在循环语句内需要完成的任务一是实时显示游戏得分，二是通过函数 GetAsyncKeyState()及时获得蛇的运动方向，三是通过函数 GetAsyncKeyState()及时获得加减速控制键信息，调整蛇的运行速度(即增减 sleeptime 的值)，改变吃到一个食物的得分(即增减 add 的值)，速度快得分高，反之减小，四是调用 snakemove()函数进行游戏。主要步骤如下。

步骤一：显示游戏界面提示信息。

```

Pos(64, 15); printf("不能穿墙,不能咬到自己\n");
Pos(64, 16); printf("用↑.↓.←.→分别控制蛇的移动.");
Pos(64, 17); printf("F1 为加速,F2 为减速\n");
Pos(64, 18); printf("ESC :退出游戏.space:暂停游戏.");
Pos(64, 20); printf(" C 语言游戏编程\n");

```

步骤二：循环控制游戏运行。

```

while(1) { 显示游戏实时得分;
            调用 GetAsyncKeyState 函数获取控制键操作信息;
            调用 snakemove 函数进行游戏;
        }

```

(11) void welcometogame()。

welcometogame()是一个显示开始界面的函数。

(12) void endgame()。



endgame() 是一个结束游戏的函数，并显示结束原因和最后得分。

(13) void gamestart()。

gamestart() 是一个游戏初始化的函数，在函数体内调用相关函数进行游戏初始化工作。

12.2.4 程序代码

源程序文件 snake.c，代码如下。

```
#include<stdio.h>
#include<time.h>
#include<windows.h>
#include<stdlib.h>
#define U 1
#define D 2
#define L 3
#define R 4 //蛇的状态,U:上 ;D:下;L:左 R:右

typedef struct SNAKE //定义蛇身的数据结构,每节蛇身为一个节点,x,y 为其坐标
{
    int x;
    int y;
    struct SNAKE * next;
} snake;

//全局变量定义//
int score=0, add=10; //总得分与每次吃食物得分。
int status, sleeptime=200; //每次运行的时间间隔
snake *head, *food; //蛇头指针,食物指针
snake *q; //遍历蛇的时候用到的指针
int endgamestatus=0; //游戏结束的情况,1:撞到墙;2:咬到自己;3:主动退出游戏。

//声明全部函数//
void color( );
void Pos( );
void creatMap( );
void initsnake( );
int biteself( );
void createfood( );
```



```
void cantcrosswall( );
void snakemove( );
void pause( );
void gamecircle( );
void welcometogame( );
void endgame( );
void gamestart( );

void color(int b) //设置颜色函数
{
    HANDLE hConsole = GetStdHandle((STD_OUTPUT_HANDLE));
    SetConsoleTextAttribute(hConsole, b);
}

void Pos(int x, int y) //设置光标位置函数
{
    COORD pos;
    HANDLE hOutput;
    pos.X = x;
    pos.Y = y;
    hOutput = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleCursorPosition(hOutput, pos);
}

void creatMap( ) //创建地图函数
{
    int i;
    for(i = 0; i < 58; i += 2) //打印上下边框
    {
        Pos(i, 0);
        printf("■");
        Pos(i, 26);
        printf("■");
    }
    for(i = 1; i < 26; i++) //打印左右边框
    {
        Pos(0, i);
        printf("■");
        Pos(56, i);
        printf("■");
    }
}
```



```
}
```

```
void initsnake( ) //初始化蛇身函数
```

```
{ snake * tail;
  int i;
  tail=(snake*)malloc(sizeof(snake)); //从蛇尾开始,头插法,以 x, y 设定开始的位置
  tail->x=24;
  tail->y=5;
  tail->next=NULL;
  for(i=1; i<=4; i++) {
    head=(snake*)malloc(sizeof(snake));
    head->next=tail;
    head->x=24+2*i;
    head->y=5;
    tail=head;
  }
  while(tail!=NULL) //从头到尾,输出蛇身
  { Pos(tail->x, tail->y);
    printf("★");
    tail=tail->next;
  }
}
```

```
int biteself( ) //判断是否咬到蛇身的函数
```

```
{ snake *self;
  self=head->next;
  while(self!=NULL) {
    if(self->x==head->x && self->y==head->y) {
      return 1;
    }
    self=self->next;
  }
  return 0;
}
```

```
void createfood( ) //创建随机出现食物的函数
```

```
{ snake *food_1;
```



```

srand((unsigned)time(NULL));
food_1 = (snake*)malloc(sizeof(snake));
while(((food_1->x%2)! = 0)    //保证其为偶数,使得食物能与蛇头对齐
{   food_1->x = rand( )%52+2;
}
food_1->y = rand( )%24+1;
q = head;
while(q->next == NULL) {
    if(q->x == food_1->x && q->y == food_1->y)    //判断蛇身是否与食物重合
    {   free(food_1);
        createfood( );
    }
    q = q->next;
}
Pos(food_1->x, food_1->y);
food = food_1;   color(12);
printf("●");   color(11);
}

void cantcrosswall( )    //判断是否穿墙的函数
{   if(head->x == 0 || head->x == 56 || head->y == 0 || head->y == 26)
{   endgamestatus = 1;
    endgame( );
}
}

void snakemove( )    //控制蛇运行的函数
{   snake *nexthead;
    cantcrosswall( );
    nexthead = (snake*)malloc(sizeof(snake));
    if(status == U) {
        nexthead->x = head->x;
        nexthead->y = head->y - 1;
        if(nexthead->x == food->x && nexthead->y == food->y)    //如果下一个有食物
        {   nexthead->next = head;
            head = nexthead;
            q = head;
        }
    }
}

```



```
        while(q! = NULL) {
            Pos(q->x, q->y);
            printf("★");
            q = q->next;
        }
        score = score + add;
        createfood( );
    } else {                                     //如果没有食物//
        nexthead->next = head;
        head = nexthead;
        q = head;
        while(q->next->next! = NULL) {
            Pos(q->x, q->y);
            printf("★");
            q = q->next;
        }
        Pos(q->next->x, q->next->y);
        printf(" ");
        free(q->next);
        q->next = NULL;
    }
}

if(status == D) {
    nexthead->x = head->x;
    nexthead->y = head->y + 1;
    if(nexthead->x == food->x && nexthead->y == food->y)    //有食物
    { nexthead->next = head;
      head = nexthead;
      q = head;
      while(q! = NULL) {
          Pos(q->x, q->y);
          printf("★");
          q = q->next;
      }
      score = score + add;
      createfood( );
    } else {                                     //没有食物
```



```

        nexthead->next = head;
        head = nexthead;
        q = head;
        while(q->next->next! = NULL)
        {
            Pos(q->x, q->y);
            printf("★");
            q = q->next;
        }
        Pos(q->next->x, q->next->y);
        printf(" ");
        free(q->next);
        q->next = NULL;
    }
}

if(status == L) {
    nexthead->x = head->x - 2;
    nexthead->y = head->y;
    if(nexthead->x == food->x && nexthead->y == food->y) //有食物
    {
        nexthead->next = head;
        head = nexthead;
        q = head;
        while(q! = NULL) {
            Pos(q->x, q->y);
            printf("★");
            q = q->next;
        }
        score = score + add;
        createfood( );
    } else { //没有食物
        nexthead->next = head;
        head = nexthead;
        q = head;
        while(q->next->next! = NULL)
        {
            Pos(q->x, q->y);
            printf("★");
            q = q->next;
        }
    }
}

```



```
        Pos(q->next->x, q->next->y);
        printf(" ");
        free(q->next);
        q->next = NULL;
    }
}

if(status == R) {
    nexthead->x = head->x+2;
    nexthead->y = head->y;
    if(nexthead->x == food->x && nexthead->y == food->y) //有食物
        nexthead->next = head;
        head = nexthead;
        q = head;
        while(q! = NULL) {
            {      Pos(q->x, q->y);
                printf("★");
                q = q->next;
            }
            score = score+add;
            createfood( );
        } else { //没有食物
            nexthead->next = head;
            head = nexthead;
            q = head;
            while(q->next->next! = NULL)
            {      Pos(q->x, q->y);
                printf("★");
                q = q->next;
            }
            Pos(q->next->x, q->next->y);
            printf(" ");
            free(q->next);
            q->next = NULL;
        }
    }

    if(biteself( ) == 1) //判断是否会咬到自己
    { endgamestatus = 2;
```




```

        endgame( );
    }
}

void pause( )    //游戏暂停函数
{
    while(1) {
        Sleep(300);
        if(GetAsyncKeyState(VK_SPACE)) {
            break;
        }
    }
}

void gamecircle( )    //控制游戏函数
{
    Pos(64, 15);
    printf("不能穿墙,不能咬到自己\n");
    Pos(64, 16);
    printf("用↑. ↓. ←. →分别控制蛇的移动.");
    Pos(64, 17);
    printf("F1 为加速, F2 为减速\n");
    Pos(64, 18);
    printf("ESC :退出游戏. space:暂停游戏.");
    Pos(64, 20);
    printf("  C 语言游戏编程\n");
    status = R;
    while(1) {
        Pos(64, 10);
        printf("得分:%d ", score);
        Pos(64, 11);
        printf("每个食物得分:%d 分", add);
        if(GetAsyncKeyState(VK_UP) && status! = D) {    //此函数读取的是物理键状态
            status = U;
        } else if(GetAsyncKeyState(VK_DOWN) && status! = U) {
            status = D;
        } else if(GetAsyncKeyState(VK_LEFT) && status! = R) {
            status = L;
        } else if(GetAsyncKeyState(VK_RIGHT) && status! = L) {

```



```
        status = R;
    } else if(GetAsyncKeyState(VK_SPACE)) {
        pause( );
    } else if(GetAsyncKeyState(VK_ESCAPE)) {
        endgamestatus = 3;
        break;
    } else if(GetAsyncKeyState(VK_F1)) {
        if(sleeptime >= 50) {
            sleeptime = sleeptime - 30;
            add = add + 2;
            if(sleeptime == 320) {
                add = 2;        //防止减到 1 之后再加回来有错
            }
        }
    } else if(GetAsyncKeyState(VK_F2)) {
        if(sleeptime < 350) {
            sleeptime = sleeptime + 30;
            add = add - 2;
            if(sleeptime == 350) {
                add = 1;        //保证最低分为 1
            }
        }
    }
    Sleep(sleeptime);
    snakemove( );
}
}
```

void welcometogame() //开始界面函数

```
{ Pos(40, 12);
  system("title C 语言游戏编程");
  printf("欢迎来到贪食蛇游戏!");
  Pos(40, 25);
  printf("      C 语言游戏编程\n");
  system("pause");
  system("cls");
  Pos(25, 12);
```



```
printf("用↑. ↓. ←. →分别控制蛇的移动, F1 为加速, 2 为减速\n");
Pos(25, 13);
printf("加速将能得到更高的分数。\\n");
system("pause");
system("cls");
}
```

```
void endgame( ) //结束游戏函数
{ system("cls");
  Pos(24, 12);
  if(endgamestatus == 1) {
    printf("对不起,您撞到墙了。游戏结束.");
  } else if(endgamestatus == 2) {
    printf("对不起,您咬到自己了。游戏结束.");
  } else if(endgamestatus == 3) {
    printf("您的已经结束了游戏.");
  }
  Pos(24, 13);
  printf("您的得分是%d\\n",score);
  exit(0);
}
```

```
void gamestart( ) //游戏初始化函数
{ system("mode con cols=100 lines=30");
  welcometogame( );
  creatMap( );
  initsnake( );
  createfood( );
}
```

```
int main( ) //主函数
{ gamestart( );
  gamecircle( );
  endgame( );
  return 0;
}
```



12.2.5 调试运行

运行程序，进入游戏操作界面如图 12-7 所示，按空格键可开始、暂停游戏，四个方向键可控制蛇的运动方向，F1 为加速键、F2 为减速键，可调整蛇的移动速度。

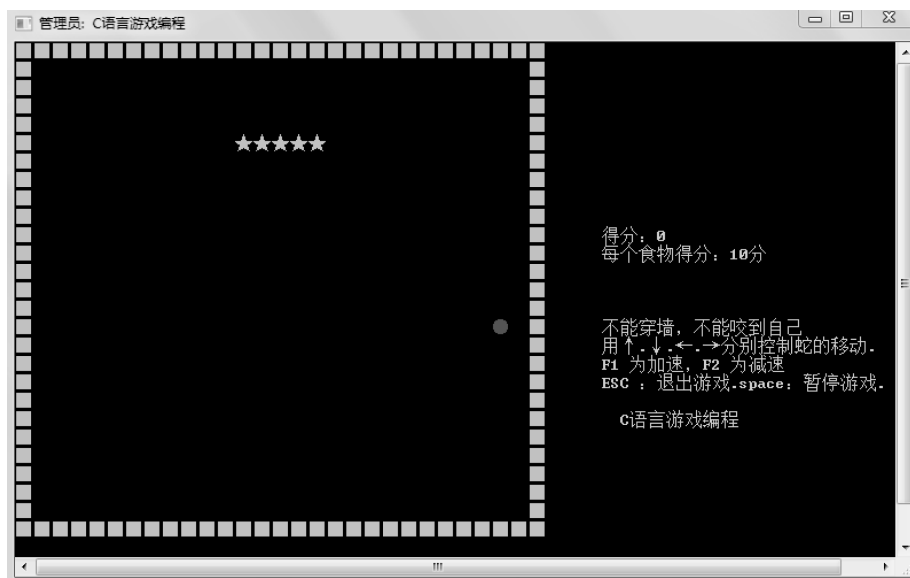


图 12-7 贪吃蛇游戏界面图

第二部分 »»

C 语言程序设计习题与解答

第 1 章 C 语言程序设计概述

一、简答题

1. 什么是程序？什么是程序设计？
2. 程序设计语言的发展经历了哪几个阶段？
3. C 语言程序的基本结构包括哪几个部分？
4. C 语言的特点有哪些？
5. C 语言的源文件、目标文件和可执行文件的后缀名分别是什么？
6. C 语言程序的开发过程包括哪几个步骤？
7. C 语言程序设计课程的学习应掌握哪些知识和能力？

二、选择题

1. 以下不是 C 语言的特点的是()。
A. C 语言简洁、紧凑
B. C 语言能够编制出功能复杂的程序
C. C 语言可以直接对硬件进行操作
D. C 语言移植性好
2. 一个 C 语言程序的执行是从()。
A. `main()` 函数开始，直到 `main()` 函数结束
B. 第一个函数开始，直到最后一个函数结束
C. 第一个语句开始，直到最后一个语句结束
D. `main()` 函数开始，直到最后一个函数结束
3. C 语言程序的基本单位是()。
A. 过程 B. 函数 C. 子程序 D. 标识符
4. 编辑程序的功能是()。
A. 建立并修改程序 B. 将 C 语言源程序编译成目标程序
C. 调试程序 D. 命令计算机执行指定的操作
5. 下列四组选项中，均不是 C 语言关键字的选项是()。



- A. define、lf、type
C. include、scanf、case
6. 以下选项中不合法的用户标识符是()。
A. abc * c B. file11 C. M_ain D. PRINT
7. 下列标识符组中, 合法的用户标识符为()。
A. _ 0123 与 ssiped B. del-word 与 signed
C. list 与 *jer D. keep% 与 wind

第2章 数据类型与表达式

一、选择题

1. C 语言中的标识符只能是字母、数字和下画线且第一个字符()。
A. 必须为字母
B. 必须为下划线
C. 必须是字母、数字或下划线中的任意一种
D. 必须为字母或下划线
2. 下列数据中属于字符串常量的是()。
A. ABC B. "ABC" C. 'abc' D. 'b'
3. 设 $\text{int } a=5$, 结果为 0 的表达式是()。
A. ! a B. 3%a C. a/=a D. ~a
4. 在 C 语言中, 字符型数据在内存中以()形式存放
A. 原码 B. BCD 码 C. 反码 D. ASCII 码
5. 在 C 语言中, 要求参加运算的数必须是整数的运算符是()。
A. / B. * C. % D. =
6. 在定义标识符中, 符合 C 语言规定的是()。
A. for B. 8a C. _ 345 D. * a
7. 设 $a=5, b=6, c=7, d=8$, 则表达式 $a<b? a : c<d? c : d$ 的结果是()。
A. 6 B. 5 C. 8 D. 7
8. 要求当 A 的值为奇数时, 表达式的值为真; A 的值为偶数时, 表达式的值为假。
以下不满足要求的表达式是()。
A. ! (A%2) B. ! (A%2==0)
C. A%2==1 D. A%2
9. 表达式 $0x81\&0x0f$ 的二进制值是()。
A. 00000001 B. 00011001 C. 00110000 D. 00000000
10. 若有定义 $\text{int } a=7, \text{float } x=2.5, y=4.7$ 则表达式 $x+a\%3 * (\text{int})(x+y)\%2/4$ 的值是()。
A. 3.500000 B. 2.750000 C. 0.000000 D. 2.500000



二、填空题

1. $12/5 * 6 =$ _____。
2. $23+16\%3 =$ _____。
3. 设 $a=7$, $x=3.5$, $y=4.7$, 则 $x+a\%3 * (\text{int})(x+y)\%24 =$ _____。
4. 表达式 $(a=3 * 6, a * 8)$, $a+5$ 的值为_____。
5. 设 $x=2$, $y=8$, 则表达式 $++x+y--$ 的值为_____。
6. 若已定义 x 和 y 为 `double` 类型, 且 $x = 1$, 则表达式 $y = x + 3/2$ 的值为_____。
7. 若 x 为 `int` 型变量, 则执行语句 $x=10, x = -2 * x+6$ 后 x 的值为_____。
8. 若 $x=6$, 则表达式 $x * = 2+4$ 的值为_____。

三、程序分析题

1. 根据整型变量的定义与使用, 参考下列程序, 写出程序运行结果。

```
#include<stdio.h>
int main( )
{ int a, b, c, d;
  unsigned m;
  a=15;
  b=- 28;
  m=11;
  c=a+b;
  d=a+m;
  printf("a+b=%d, a+m=%d\n", c, d);
  return 0;
}
```

程序运行结果:

2. 写出程序运行结果。

```
#include<stdio.h>
int main( )
{ int i, j, m, n;
  i=8; j=10;
  m=++i; n=j++;
  printf("%d, %d, %d, %d\n", i, j, m, n);
  return 0;
}
```

程序运行结果:

3. 写出下列程序运行结果。



程序运行结果:

- 程序运行结果:

- 程序运行结果:

一、选择题

- 128



- C. 编译出错 D. 能判断 x 和 y 是否相等

3. 以下有关 switch 语句的说法, 正确的是()。

- A. break 语句是语句中必需的一部分
B. 在 switch 语句中可以根据需要使用或不使用 break 语句
C. break 语句在 switch 语句中不可以使用
D. 在 switch 语句中的每一个 case 都要使用 break 语句

4. 当执行以下程序时, ()。

```
#include<stdio. h>
void main( )。
{
    int a;
    while(a=5)
    printf("%d", a-);
}
```

- A. 循环体将执行 5 次 B. 循环体将执行 0 次
C. 循环体将执行无限次 D. 系统会死机

5. 以下程序的输出结果是()。

```
main( )。
{    int i=010, j=10, k=0x10;
    printf("%d, %d, %d\n", i, j, k);
}
```

- A. 8, 10, 16 B. 8, 10, 10
C. 10, 10, 10 D. 10, 10, 16

6. 以下程序的输出结果是()。

```
void main( )。
{    char c1='6', c2='0' ;
    printf("%d, %d, %d\n", c1, c2, c1-c2);
}
```

- A. 因输出格式不合法, 输出出错信息 B. 54, 48, 6
C. 6, 0, 7 D. 6, 0, 6

7. 下述程序段的输出结果是()。

```
void main( )。
{    int x=10, y=10;
    printf("%d, %d\n", x++, ++y);
}
```

- A. 10, 10 B. 11, 11
C. 11, 10 D. 10, 11



8. 若 m , y , z 均为 int 型变量, 则执行下面语句后 m 值是()。

```
m = 1; x = 2; y = 3; z = 4;
m = (m < x)? m : x;
m = (m < y)? m : y;
m = (m < z)? m : z;
```

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

9. 设已定义整型变量 K , 则下面的程序输出为()。

```
k = 017; s = 111;
printf("%d", ++k);
printf("%x\n", s++);
```

- A. 15 6f B. 16 70
C. 15 71 D. 16 6f

10. 下面程序的运行结果是()。

```
void main( )
{   int c1 = 2, c2 = 0;
    while(!c2 - )
        printf("%d%d", c1 - , c2);
}
```

- A. 陷入死循环 B. 不输出任何内容
C. 输出 2-1 D. 输出 20

二、填空题

1. 请写出以下语句的输出结果_____。

```
printf("%d", strlen("\t\"065\xffab\n"));
```

2. 结构化程序设计中的三种基本结构是_____、_____、_____。

3. 下述程序段的循环次数是_____次, 输出结果是_____。

```
int x = 0, y = 0;
do
{   y++;
    x* = x;
} while(x < 0 && y > 5)
printf("y = %d, x = %d", y, x);
```

4. 下述程序段的运行结果是_____。

```
int x, y;
x = 17;
y = 26;
printf("%d", y / (x % = 6));
```



三、程序分析题

1. 下述程序的输出结果是_____。

```
#include<stdio. h>
void main( )
{   int x=8, y=9;
    int k;
    k=(x++)+(++y);
    printf("%d", k);
}
```

2. 已知字符 A 的 ASCII 码值为 65, 写出以下语句的输出结果。

```
#include<stdio. h>
void main( )
{   char ch=' A' ;
    printf("%c, %d", ch, ch+3);
}
```

输出结果: _____

- 3.

```
void main( )
{   int t=1;
    int i;
    for(i=1; i<5; i++)   t=t*i;
    printf("%d", t);
}
```

输出结果: _____

四、补充程序题

1. 程序的功能是求两个整数的最大公约数, 请填空。

```
#include<stdio. h>
void main( )
{   int a=12, b=8, t;
    while(_____)
    {   t=a%b;
        a=b;
        b=t;
    }
    printf("%3d", a);
}
```



2. 求下列算式 $1+1/2+1/3+\dots+1/100$ 的值, 请填空。

```
#include<stdio. h>

void main(    )。
{    int i;
    float sum=0;
    for(i=1; i<101; i++)
        sum+= _____;
    printf("%f\n", sum);
}
```

3. 求 $2/1, 3/2, 5/3, 8/5, 13/8, 21/13, \dots$ 这个数列的前 20 项之和, 请填空。

```
#include<stdio. h>

int main( )
{ int i, n=20;
  double a=2, b=1, s=0, t;
  for(i=1; i<=n; i++)
  { s= _____;
    t=a;
    a= _____;
    b=t;
  }
  printf("sum = %16. 10f\n", s);
  return 0;
}
```

五、编写程序题

1. 输入一个数, 判断该数是奇数还是偶数。
2. 输出以下图案。

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

3. 输入两个正整数 m 和 n , 求其最大公约数和最小公倍数。
4. 输入一行字符, 分别统计出其中英文字母、空格、数字和其他字符的个数。

6. 编程求 $1!+2!+3!+\cdots+50!$ 的值。

一、选择题

- ```
int a=3, b=4;
```

```
void fun(int x1, int x2)
{ printf("%d, "d\n", x1+x2, b);}
void main()
{ int a=5, b=6; fun(a, b);}
```

- A. 3, 4                      B. 11, 1                      C. 11, 6                      D. 11, 4



- [illegible]

```
void main()
{ int i=5;
 printf("%d \n", sub(i));
}

int sub(int n)
{ int a;
 if(n==1) return 1;
 a=n+sub(n-1);
 return(a);
}
```

- A. 16                      B. 17                      C. 15                      D. 14



11. 以下程序的运行结果是( )。

```
#include<stdio. h>
f(int a[])
{ int i=0;
 while (a[i]<= 10)
 { printf("%4d", a[i]);
 i++;
 }
}
void main()
{ int a[]={1, 5, 10, 9, 11, 7};
 f(a+1);
}
```

A. 5 11 8                      B. 6 12 9                      C. 5 10 9                      D. 5 6 9

12. 以下叙述中不正确的是( )。

A. 预处理命令行都必须以#号开始  
B. 在程序中凡是以#号开始的语句行都是预处理命令行  
C. C 语言程序在执行过程中对预处理命令行进行处理  
D. #define IBM PC100 是正确的宏定义

13. 下面程序的运行结果是( )。

```
#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

14. 以下程序的运行结果是( )。

```
#include<stdio. h>
#define FUDGE(y) 2.84+y
#define PR(a) printf("%d", (int)(a))
#define PRINT1(a) PR(a); putchar(' \n')
main()
{int x=2;
 PRINT1(FUDGE(5)*x);
}
```

A. 11                      B. 12                      C. 13                      D. 15

## 二、编写程序题

1. 编写 search 函数, 功能是利用顺序查找法从数组 a 的 10 个元素中对关键字 m 进行查找。顺序查找法的思路是: 从第一个元素开始, 从前向后依次与关键字比较, 直到找到此元素或查找到数组尾部时结束。若找到, 返回此元素的下标; 若仍未找到, 则返回值-1。
2. 分别编写求其最大公约数 `int gcd(int n, int m)` 和最小公倍数的函数 `int lcs(int n, int m)`, 在主函数中输入两个正整数 m 和 n ( $1 < m, n < 1000000000$ ), 调用函数求出两个数的最大公约数和最小公倍数
3. 已有一个已排好(从小到大)的 n 个元素的整形数组, 输入一个整数, 要求按原来排序的规律将它插入数组中。
4. 编写函数 `del(int a[], int n, int x)`, 它的作用是删除有序数组 a 中的指定元素 x。
5. 编写一个函数用于判断一个整数是否为回文数(回文数是关于中心数对称的, 如 12321, 2345432)。
6. 能够组成直角三角形三个边的最小一组整数是 3、4、5。写程序求出在一定范围内所有可以组成直角三角形三个边的整数组, 输出三个一组的整数。设法避免重复的组。
7. 菲波那契数列定义为:  $f(1)=1; f(2)=1$ ; 当  $n>2$  时,  $f(n)=f(n-1)+f(n-2)$ , 输入 n, 求菲波那契数列的第 n 项。要求: 用递归函数求菲波那契数列的第 n 项。

# 第 5 章 数组与字符串

## 一、选择题

1. 以下定义语句中, 错误的是( )。  
A. `int a[] = {6, 7, 8}`                      B. `int n=5, a[n]`  
C. `char a[] = "string"`                      D. `char a[5] = {'0', '1', '2', '3', '4'}`
2. 以下描述中正确的是( )。  
A. 数组定义中的数组名后面的常量表达式用一对圆括弧括起来  
B. 数组下标从 1 开始  
C. 数组下标的数据类型可以是整型或实型  
D. 数组名的规定与变量名相同
3. 若定义数组并初始化 `char a[10] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'}`; 指出以下正确的语句( )。  
A. `scanf("%c", a[0])`                      B. `scanf("%s", &a);`  
C. `printf("%c", a[3])`                      D. `printf("%s", a);`
4. 若定义数组 `int a[10]`, 其最后一个数组元素为( )。  
A. `a[0]`                      B. `a[1]`                      C. `a[9]`                      D. `a[10]`
5. 若定义数组并初始化 `int a[10] = {1, 2, 3, 4}`, 以下选项哪一个是错误的( )。  
A. `a[8]` 的值为 0                      B. `a[1]` 的值为 1



- ## 二、填空题

3. 执行以下程序的输出结果是 \_\_\_\_\_。

}

- ```
int main( )
```



```
{
    char s[]={"012345"};
    int i=0;
    while( s[i] ){
        s[i]=s[i]- ' 0' +' A' ;
        i++;
    }
    printf("%s\n", s);
}
```

5. 要定义一个 $4 * 4$ 的矩阵, 并初始化主对角线的元素为 1, 其他元素为 0, 则定义并初始化的语名是_____。

6. 有字符数组定义为: `char a[10] = { 'a', 'x', 'b', 'f', '\0', 'r', 'm' }`; 则 `len = strlen(a)`; 的结果 `len` 等于_____。

三、程序分析题

1. 写出下面程序的输出结果。

```
#include<stdio. h>
void main(    )。
{
    int i, a[10];
    for (i=9; i>=0;i- ) a[i]=10- i;
    printf("%4d%4d%4d", a[2], a[5], a[8]);
}
```

程序输出结果:

2. 写出下面程序的输出结果。

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

程序输出结果:

3. 写出下面程序的输出结果。

```
#include<stdio. h>
int f(int b[], int n )
{
    int i, 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);
}
```

4. 写出下面程序的输出结果。

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

程序输出结果:

5. 设从键盘输入字符串: abcde, 写出下面程序的输出结果。

```
#include<stdio. h>
void main( )
{ char a[10];
    int i;
    for(i=1; i<=5; i++)
        scanf("%c", &a[i]);
    printf("%c", a[0]);
}
```

程序输出结果:

四、补充程序题

1. 给定一维整型数组, 输入数据并求第一个值为奇数元素之前的元素和。请将下面的程序补充完整。

```
#include<stdio. h>
int main( )
{ int i, sum=0, a[10];
    printf("请输入 10 个整数:\n");
    for(i=0; i<10; i++)
        scanf("%d", &a[i]);
    for(i=0; i<10; i++)
    {
        if(a[i]%2!=0)
            break;
        _____//补充完整
    }
    printf("sum = %d\n", sum);
}
```



```
return 0;
```

```
}
```

2. 给定一维整型数组，输入数据并对前一半元素升序排序，对后一半元素降序排序。请将下面的程序补充完整。

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

```
#define N 10
```

```
int main( )
```

```
{ int i, j, t, a[N]={23, 67, 31, 28, 49, 37, 80, 46, 19, 21};
```

```
printf("排序前数组为: \n");
```

```
for(i=0; i<N; i++)
```

```
printf("%d", a[i]);
```

```
for(i=0; i<N; i++)
```

```
{ if( i<N/2 )
```

```
{ for( j=i; j<N/2- 1; j++)
```

```
if( a[j]>a[j+1] )
```

```
{t=a[j]; a[j]=a[j+1]; a[j+1]=t; }
```

```
}
```

```
else //请将程序补充完整
```

```
{
```

```
}
```

```
}
```

```
printf("排序前数组为: \n");
```

```
for(i=0; i<N; i++)
```

```
printf("%d", a[i]);
```

```
return 0;
```

```
}
```

3. 给定 $N * N$ 矩阵，输入矩阵元素并互换主次对角线元素值。请将下面的程序补充完整。

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

```
#define N 3
```

```
int main( )
```

```
{ int i, j, t, a[N][N];
```

```
printf("请输入%d x%d 矩阵的数据: \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++)
```



```

{ //将程序补充完整

}

printf("互换主次对角线元素值后,矩阵为:\n");
for(i=0; i<N; i++)
{ for(j=0; j<N; j++)
    printf("%d ", a[i][j]);
    printf("\n");
}
return 0;
}

```

五、编写程序题

1. 菲波那契数列定义为: $f(1)=1$; $f(2)=1$; 当 $n>2$ 时, $f(n)=f(n-1)+f(n-2)$ 。求菲波那契数列的第 n 项。

2. 输入一行字符, 分别统计出其中英文字母、空格、数字和其他字符的个数。

3. 输入一个整数 $n(n<100)$ 和 n 个整数, 将 n 个整数按照从小到大的顺序输出。

4. 输入两个整数 a 和 $b(2\leq a<b<1000)$, 输出 a 和 b 之间的素数。

步骤如下。

第1步: 将 2, 3, 4, 5, ..., 1000 存入数组, 并确定第一个素数 2。

第2步: 删除数组中该素数的倍数。

第3步: 取数组中第一个没有被删除的数作为新素数, 重复第2步。

5. 编写程序, 求一个 $n\times n$ 矩阵对角线元素之和。

6. 已有两个已排序(从小到大)的 n 个元素和 m 个元素的数组, 将它们合并到一个从小到大的数组中。

7. 输入一行英文, 统计每个出现的英文字母的个数。注意: 不区分大小写。

8. 输入一个班 C 语言的考试成绩, 统计 0~59、60~69、70~79、80~89、90~100 每个分数段的人数。注意: 用数组统计, 不能使用 if 和 switch 语句。

9. 输入 $n(1<n<30)$ 个整数, 删除它们中的重复数据。

10. 设 n, m 为矩阵, 编写程序求 $n\times m$ 的积。

11. 读入一行英文, 统计其中的单词个数。注意: 单词之间有 1 个或者多个空格分开。

12. 编程输出下列杨辉三角形。

```

      1
    1 1
  1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

```



13. 幻方是指将 $1 \dots n^2$ 填入 $n \times n$ 方阵中, 使每行元素之和、每列元素之和、两个主对角线上元素之和均相等。编程输出 n 阶幻方(n 为奇数), 如 $n=5$ 时的结果如下, 并验证其结果的正确性。

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

第 6 章 指 针

一、选择题

- 变量的指针, 其含义是指向该变量的()。
A. 值 B. 地址 C. 名字 D. 类型
- 有如下定义: `int sum, *pi=∑` 下列说法不正确的是()。
A. 指针变量 `pi` 的基类型是 `int`
B. 指针变量 `pi` 的类型是 `int *`
C. `pi` 的值是某个整型变量的地址, 或者说 `pi` 指向一个整型变量
D. `sum` 的值的类型是 `int *`
- 若有语句 `int a=7, *p;` 和 `p=&a;` 下面各组选项都代表地址的是()。
A. `&a, &*p, p` B. `a, p, *&a`
C. `&p, *p, &a` D. `&*a, &a, *p`
- 已有定义 `int m=3, int *p, *q;` 且 `p` 和 `q` 均已指向变量 `m`, 下面不能正确执行的赋值语句是()。
A. `m = *p + *q` B. `q = m` C. `p = q` D. `m = *p * (*q)`
- 下面能正确进行字符串赋值操作的是()。
A. `char s[3] = { "CPU" }` B. `char s[3] = { 'C', 'P', 'U' }`
C. `char *s; s = "CPU"` D. `char s[] = { 'C', 'P', 'U' }`
- 若有定义 `char s[10]`, 则下面表达式中不表示 `s[1]` 的地址的是()。
A. `s+1` B. `s++` C. `&s[0]+1` D. `&s[1]`
- 下列代码段, 哪个是正确的?()
A. `int *pointer, x; pointer = x` B. `int *pointer, x * pointer = x`
C. `int *pointer, x; ; pointer = &x` D. `int *pointer, x; ; x = &pointer`
- 若有定义 `int a[5], *p=a;` 则对 `a` 数组元素的不正确引用是()。
A. `int a[5], *p; p=&a;` B. `int a[5], *p; p=a`
C. `int a[5]; int *p=a;` D. `int a[5]; int *p1, *p2=a; *p1 = *p2`



9. 在 C 语言中, 在函数之间进行数据传递的方法除了通过返回值和全程变量(外部变量)外, 还可以采用哪一种方式(将被调用函数中数据传递给主调函数)? ()
- A. 局部静态变量 B. 形参与实参之间传值
C. 局部变量 D. 形参与实参之间传址
10. 设 p 为指针变量, 下面哪组结论不正确()。
- A. *p++与*(p++)等价
B. *(p++)与*(++p)不等价
C. *p++; 与{++*p-1; p++;}等价
D. int *p(int)与int(*p)(int)等价

二、填空题

1. 编写 swap() 函数交换两个变量的值。

请将下面的程序填写完整。

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

```
//swap()函数的定义
```

```
_____  
{  
_____  
.....  
}  
int main( )  
{  
    int m, n;  
    while (scanf("%d%d", &n, &m)!= EOF)  
        { swap(&n, &m); printf("%d %d\n", n, m); }  
    return 0;  
}
```

输入: 包含多组数据, 每组 2 个整数。

数据输入格式: ①多组数据占一行, 每组数据之间用空格键或 TAB 键分开, 每组的两个整数之间用一个空格键或 TAB 键分开; ②每组数据占一行, 两个整数之间用一个空格键或 TAB 键分开, 最后按回车键结束一行数据的输入。

输出: 交换后的结果。

样例输入:

1 2 3 4

样例输出:

2 1

4 3

2. 输入 n ($2 < n < 100$) 和 n 个整数, 编写函数, 计算 n 个整数中的最大值和最小值。



请将下面的程序填写完整。

```
#include<stdio. h>
//下面是函数 f( )的定义

_____
{
_____
.....
}
int main( )
{
    int i, n, max, min, a[100];
    while (scanf("%d", &n)!= EOF)
    {
        for(i=0; i<n; i++)
            scanf("%d", &a[i]);
        f(a, n, &min, &max);
        printf("%d %d\n", min, max);
    }
    return 0;
}
```

输入：包含多组数据，每组数据个数 n 和 n 个整数。

数据输入格式：①多组数据占一行，每组数据之间用空格键或 TAB 键分开，每组的两个整数之间用一个空格键或 TAB 键分开；②每组数据占一行，两个整数之间用一个空格键或 TAB 键分开，最后按回车键结束一行数据的输入。

输出： n 个整数中的最大值和最小值。

样例输入：

4 1 2 3 5 5 2 6 9 4 8

样例输出：

1 5

2 9

3. 分析下面的程序。

```
#include<stdio. h>
int sum(int *p, int n)
{
    int i, j, s=0;
    for(i=0; i<n; i++)
        { p[i]=p[i]+10; s+=p[i]; }
```




```

    for(j=0; j<12; j++)
        printf("%d", *(p+j));
    return s;
}
int main( )
{
    int a[]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
    printf("\n%d\n", sum(a, 5) );
    return 0;
}

```

执行后输出的结果:

4. 分析下面的程序, 比较二维数组元素($*(p+i))[j]$ 与 $*(p+i)[j]$ 的区别。

```

#include<stdio. h>
int main( )
{
    int a[4][3]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
    int (*p)[3]=a;           //a 是二级地址, p 是指向数组的指针
    int i, j, s1=0, s2=0;
    for(i=0; i<2; i++)       //*(p+i)是一级地址,即第 i 行行指针
        for(j=0; j<3; j++)
        {
            s1=s1+(*(p+i))[j];    //运算符“[]”的优先级高于运算符“*”
            s2=s2+*(p+i)[j];
        }                      //先求数组元素的行下标和列下标,再取该数组元素的值
    printf("s1=");
    for(i=0; i<2; i++)
        for(j=0; j<3; j++)
            printf("%d+", (*(p+i))[j] );
    printf("\b= %d\n", s1);
    printf("s2=");
    for(i=0; i<2; i++)
        for(j=0; j<3; j++)
            printf("%d+", *(p+i)[j] );    //*(p+i)[j]与*((p+i)[j])等价
    printf("\b= %d\n", s2);
    return 0;
}

```



执行后输出的结果:

5. 分析下面的程序, 掌握字符串的连接和取字符串的子串的操作。

```
#include<stdio. h>
int  main( )
{
    int i=0;
    char s[100], *p1="program ", *p2,*p;
    p2="code";
    while(*p1!= '\0' )
        { s[i++]=*p1; p1++; } //与 s[i++]=*p1++;等价
    while(*p2!= '\0' )
        s[i++]=*p2++; //与 while(s[i++]=*p2++); 等价, 其中分号为空语句
    s[i]= '\0' ;
    p=s;
    printf("string=%s \nsubstring=%s \n", s, &p[8]); //&p[8]与 &s[8], s+8, p+8 等价
    return 0;
}
```

执行后输出的结果:

6. 分析下面的程序, 从键盘上依次输入 M 个人的姓名和成绩, 设计一个选择排序算法, 按照成绩从高到低的次序排列姓名和成绩, 最后按照成绩从高到低的次序输出每个学生的姓名和成绩。

```
#include<stdio. h>
#include<string. h>
#include<stdlib. h>
#define M 5 //定义常量 M 和 N
void SelectSort(char *a[M], int b[M])
    //算法中对数组参数的操作就是对相应实参数组的操作
{
    int i, j, k, y;
    char x[2*M];
    for(i=1; i<M; i++) //进行 M-1 此选择和交换
    { k=i-1; //给 k 赋初值
        for(j=i; j<M; j++) //选择当前区间内的最大值 b[k]
```



```

        if(b[j]>b[k]) k=j;           //交换 a[i- 1],a[k]和 b[i- 1],b[k]的值
        strcpy(x, a[i- 1]), strcpy(a[i- 1], a[k]); strcpy(a[k], x);
        y=b[i- 1], b[i- 1]=b[k], b[k]=y;    //使成绩和姓名同步交换
    }
}
int main( )
{
    char *ax[M];           //从键盘输入 M 个学生的姓名和成绩到数组 ax 和 bx 中
    int bx[M], i;
    printf("请输入%d 个学生的姓名和成绩:\n", M);
    for(i=0; i<M; i++)      // 用 malloc 函数申请存放 2M 个字符的动态存储空间
    { ax[i]=(char *)malloc(2*M*sizeof(char));
      if(ax[i]==NULL) { printf("分配存储空间失败,退出!"); exit(0);}
      scanf("%s", ax[i]); scanf("%d", &bx[i]);    //输入姓名、成绩
    }
    SelectSort(ax, bx);      //调用选择排序算法对 ax 和 bx 数组按成绩进行排序
    for(i=0; i<M; i++)      //按排序结果依次输出每一个学生的姓名和成绩
    { if((i+1)%3==0) printf("%- 10s%4d\n", ax[i], bx[i]);
      else printf("%- 10s%4d;", ax[i], bx[i]); //每行输出 3 个学生的姓名和成绩
    }
    printf("\n");
    for(i=0; i<M; i++) free(ax[i]);    //释放 malloc 函数分配的动态存储空间
    return 0;
}

```

执行程序,根据提示输入了 5 个学生的信息,屏幕显示如下:

请输入 5 个学生的姓名和成绩:

Zhangyi 60 Lisi 70 Wangwu 80 Zhaoliu 90 Qianqi 100

请按照输出格式写出结果:

三、程序分析题

1. 写出下面程序的输出结果。

```

#include<stdio. h>
void fun(char *c)
{ while(*c)
    { if(*c>=' a' && *c<=' z' ) *c=*c- (' a' - ' A' );
      c++;
    }
}

```



```
}  
int main( )  
{ char s[81]={"abcd1234EFgh"};  
  fun(s);  
  puts(s);  
  return 0;  
}
```

输出结果:

2. 写出下面程序的输出结果。

```
#include<stdio. h>  
void fun(char *s)  
{ while(*s)  
  { if(*s%2)  
    printf("%c", *s);  
    s++;  
  }  
}  
int main( )  
{ char a[]={"abcBYTE"};  
  fun(a);  
  printf("\n");  
  return 0;  
}
```

输出结果:

3. 写出下面程序的输出结果。

```
#include<stdio. h>  
int *f(int *s, int *t)  
{ if(*s<*t)  
  s=t;  
  return s;  
}  
int main( )  
{ int i=3, j=5, *p=&i, *q=&j, *r;  
  r=f(p, q);  
  printf("%d, %d, %d, %d, %d\n", i, j, *p, *q, *r);  
  return 0;  
}
```



输出结果:

4. 写出下面程序的输出结果。程序运行后输入: 57 46 88 ,请分析输出结果。

```
#include<stdio. h>
void swap(float *p, float *q)
{ float t;
  t=*p;
  *p=*q;
  *q=t;
}
int main( )
{ float a, b, c;
  scanf("%f%f%f", &a, &b, &c);
  if(a>b) swap(&a, &b);
  if(a>c) swap(&a, &c);
  if(b>c) swap(&b, &c);
  printf("%. 2f, %. 2f, %. 2f\n", a, b, c);
  return 0;
}
```

输出结果:

5. 分析下面程序的功能, 任意输入一串字符串, 验证输出结果。

```
#include<stdio. h>
void srev(char *p)
{ char t, *q;
  int n;
  n = strlen(p);
  q = p+n- 1;
  while(p<q)
  { t=*p; *p=*q; *q =t;
    p++; q- ;
  }
}
int main( )
{ char a[80];
  gets(a);
  srev(a);
  puts(a);
}
```



四、补充程序题

1. 从键盘输入 10 个整数存入一维数组中, 求出其中的最大数并输出(要求用指针访问数组元素)。请将下面的程序补充完整。

```
#include<stdio. h>
int main( )
{
    int a[10], max, i;
    for(i=0; i<= 9; i++)
        scanf("%d", a+i);
    max=*(a+0);
    for( _____ ) //补充完整
    {   if(*(a+i)>max)
        _____ //补充完整
    }
    printf("max= %d \n", max);
}
```

2. 将一个字符串从第 m 个字符开始的全部字符复制成为另一个字符串。请将下面的程序补充完整。

```
#include<stdio. h>
#include<string. h>
void copystr(char *p1, char *p2, int m)
{
    char *p, *q;
    p=p1+m- 1;
    q=p2;
    while(*p!=' \0' )
    {   /*复制字符 */
        _____
    }
    *q=' \0' ; /*新字符串末尾加上字符串结束标志 */
}
void main( )
{
    int m;
    char s1[80], s2[80];
    printf("请输入一字符串: \n");
    gets(s1);
```



```

printf("请输入开始复制的位置(m):");
scanf("%d", &m);
_____ /*调用函数进行复制 */
puts(s2);
}
3. 有 10 个整数围成一圈，求出相邻三个数之和的最小值。请将下面的程序补充完整。
#define N 10
#include<stdio. h>
int*find_position(int *a, int len)
{
    int *p, *q, sum=0, i=0, j=0, min=0, *pos;
    pos=p=q=a;
    for(i=0; i<3; i++)          //以第一组三位数之和当作最小
        min+=*q++;
    i=0;
    while(i<len)
    {
        sum=0;
        p=a+i;
        for(j=0; j<3; j++)      //取连续的三个数累加
        {
            if(p>a+len- 1)
                p=a;          //若超出了数组的范围,则需要回溯到开始位置
            _____
        }
        if(min>sum)
        {
            _____
        }
        i++;
    }
    return pos;
}
int main( )
{
    int a[10]={5, 2, 6, 1, 4, 1, 9, 2, 3, 8};
    int *p, sum=0;
    p=find_position(a, N);

```



```
printf("\n 最小组为:%d+", *p);
sum += *p;
p++;
if(p > a + N - 1)
    p = a;
printf("%d", *p);
sum += *p;
p++;
if(p > a + N - 1)
    p = a;
printf("+%d =", *p);
sum += *p;
printf("%d\n", sum);
return 0;
}
```

五、编写程序题

1. 输入多个英文单词并保存到字符串数组中, 按照字母顺序从小到大输出。要求编写函数 $f(\text{char} *s[], \text{int } n)$, 对字符指针数组 $*s$ 中的 n 个单词排序。

输入: 多个英文单词(不超过 100 个), 每行一个单词。

输出: 字母顺序从小到大显示。

样例输入:

one

two

three

four

样例输出:

four

one

three

two

2. 将一行英文中指定的字符复制到一个字符数组中并输出。不能使用 `strcpy` 函数。

要求: 编写函数 $f(\text{char} *s1, \text{int } start, \text{int } len, \text{char} *s2)$, 将字符指针 $*s1$ 中从第 $start$ 开始连续的 len 个字符复制到字符指针 $*s2$ 中。

输入: 第 1 行输入一行英文。

第 2 行输入两个整数, 分别为开始位置 $start$ 和字符个数 len 。

输出:

第 $start$ 开始连续 len 个字符。



样例输入：

This problem requires that you write a program to compute the exact value of X. (注意：输入的数据为一行，这里显示为两行)

6 16

样例输出：

problem requires

说明：VC++6.0 环境中，当输入一行英文时，每行只能显示 80 个字符，超出的部分显示在下一行。

3. 将两行英文连接后输出。不能使用 strcat 函数。

要求：编写函数 $f(\text{char} * s1, \text{char} * s2, \text{char} * s3)$ ，将字符指针 $*s1$ 中字符与字符指针 $*s2$ 中字符连接到字符指针 $*s3$ 中。

输入：输入包括 2 行英文。

输出：将 2 行英文连接后输出。

样例输入：

Problems involving the computation of exact values of very large magnitude and precision are common.

For example, the computation of the national debt is a taxing experience for many computer systems.

(注意：连续输入完数据后，按回车键结束)

样例输出：

Problems involving the computation of exact values of very large magnitude and precision are common. For example, the computation of the national debt is a taxing experience for many computer systems.

4. 统计成绩。

一个班有 $N(N < 20)$ 名学生，每个学生修了五门课。编写程序。

① 求每个学生的平均成绩，并输出学生的学号、每门课程的成绩及平均值。

② 求某门课程的平均分。

要求：

① 分别编写 2 个函数实现以上 2 个要求；

② 第 1 个函数用数组名作形式参数，第 2 个函数用指针作形式参数，并在函数体内用指针对数组操作。

输入：

第一个数据：输入 N ，代表 N 名学生。

后面 6 个数据一组，共 N 组。每组有 6 个数据分别为：学号、英语成绩、数学成绩、C++成绩、音乐成绩、美术成绩。

输出：

输出 7 个数据一组，共 N 组。每组数据分别为学生学号、每门成绩和平均成绩(平均

成绩四舍五入保留一位小数)。

最后按顺序输出每门平均成绩(平均成绩四舍五入保留一位小数)。

样例输入:

```
4 20070001 94 92 97 93 90 20070005 84 89 92 81 73 20070004 82
75 94 86 95 20070003 84 86 82 97 91
```

样例输出:

```
20070001 94 92 97 93 90 93.2
20070005 84 89 92 81 73 83.8
20070004 82 75 94 86 95 86.4
20070003 84 86 82 97 91 88.0
86.0 85.5 91.3 89.3 87.3
```

5. 排序与查找。

编写程序:编写函数对 n 个整数从大到小排序,函数有 2 个参数,分别为一个整形指针和整数 n 。排序完成后输入一个数,要求用折半查找法找出该数是数组中第几个元素的值。

输入:

输入共有 3 项:

第 1 项为 n ,第 2 项为 n 个整数,第 3 项为要查找的整数。

输出:

输出你要查找的整数在排好序的数组中是第几个元素。如果该数不在数组中,则输出“wucishu!”。如果输入的数在数组中有多个,则输出此数在排好序后的数组中第一次出现的位置。

样例输入:

```
5 8 6 4 5 7 7
```

样例输出:

```
2
```

第 7 章 结构体

一、选择题

1. 定义以下结构体类型

```
struct s { int a;
           char b;
           float f;
};
```

在 vc6.0 中,则语句 `printf("%d", sizeof(struct s))` 的输出结果为()。



- A. 3 B. 7 C. 9 D. 4
2. 当定义一个结构体变量时,系统为它分配的内存空间是()。

A. 结构中一个成员所需的内存容量
B. 结构中第一个成员所需的内存容量
C. 结构体中占内存容量最大者所需的容量
D. 结构中各成员所需内存容量之和

3. 定义以下结构体类型

```
struct s {  
    int x;  
    float f;  
    } a[3];
```

在vc6.0中,语句 `printf("%d", sizeof(a))` 的输出结果为()。

- A. 24 B. 12 C. 18 D. 6

4. 定义以下结构体数组

```
struct c {  
    int x;  
    int y;  
    } s[2]={1, 3, 2, 7};
```

语句 `printf("%d", s[0].x * s[1].x)` 的输出结果为()。

- A. 14 B. 6 C. 2 D. 21
5. 运行下列程序段,输出结果是()。

```
struct country  
{    int num;  
    char name[10];  
}x[5]={1, "China", 2, "USA", 3, "France", 4, "England", 5, "Spanish"};  
#include "stdio. h"  
void main( )  
{    struct country *p;  
    p=x+2;  
    printf("%d, %c\n", p->num, (*p). name[2]);  
}
```

- A. 3, a B. 4, g C. 2, U D. 5, S

6. 下面程序的运行结果是()。

```
struct KeyWord  
{    char Key[20];  
    int ID;  
}kw[]={ "void", 1, "char", 2, "int", 3, "float", 4, "double", 5};  
void main( )
```



```
{ printf("%c, %d\n", kw[3]. Key[0], kw[3]. ID); }
```

A. i, 3 B. n, 3 C. f, 4 D. l, 4

7. 定义以下结构体类型

```
struct student
{
    char name[10];        // 10
    int score[50];        //200
    float average;        //4
}stud1;
```

则 stud1 占用内存的字节数是()。

A. 64 B. 214 C. 228 D. 7

8. 如果有下面的定义和赋值,则使用()不可以输出 n 中 data 的值。

```
struct SNode {
    unsigned id;
    int data;
}n, *p; p = &n;
```

A. p. data B. n. data C. p->data D. (*p). data

9. 根据下面的定义,能输出 Mary 的语句是()。

```
struct person
{
    char name[9];
    int age;
} struct person class[5]={"John", 17, "Paul", 19, "Mary", 18, "Adam", 16};
```

A. printf("%s\n", class[1]. name);
B. printf("%s\n", class[2]. name);
C. printf("%s\n", class[3]. name);
D. printf("%s\n", class[0]. name);

10. 定义以下结构体数组

```
struct date
{
    int year;
    int month;
    int day;
};

struct s
{
    struct date birthday;
    char name[20];
} x[4]={2008, 10, 1, "guangzhou"}, {2009, 12, 25, "Tianjin"};
```

语句 printf("%s, %d, %d, %d", x[0]. name, x[1]. birthday. year); 的输出结果为()。

A. guangzhou, 2009



- B. guangzhou, 2008
C. Tianjin, 2008
D. Tianjin, 2009
11. 字符‘0’的ASCII码的十进制数为48,且数组的第0个元素在低位,则以下程序的输出结果是()。
- ```
#include "stdio. h"
void main()
{
 union
 {
 int i[2];
 long k;
 char c[4];
 } r, *s = &r;
 s->i[0] = 0x39;
 s->i[1] = 0x38;
 printf("%c", s->c[0]);
}
```
- A. 39                      B. 9                      C. 38                      D. 8
12. 若有以下定义,则语句( )是正确的。
- ```
union data
{
    int i;
    char c;
    float f;
}a;
int n;
```
- A. s = 5 B. a = { 2, 'a', 1. 2 }
C. printf(" %d", a) D. n = a. i
13. 若有下面的说明和定义:
- ```
union test
{
 char ul[5];
 int u2[2];
} ua;
```
- 则 sizeof( union test ) 的值是( )。
- A. 12                      B. 16                      C. 14                      D. 8
14. 下面程序的输出是( )。
- ```
#include "stdio. h"
typedef union
{
    long x[2];
    int y[4];
```



```
char z[8];
} MYTYPE;
void main(int argc, char*argv[])
{MYTYPE them;
printf("%d", sizeof(them));
}
```

A. 32

B. 16

C. 8

D. 24

二、填空题

1. 使多个不同的变量共占用一段内存的结构,称为_____类型。

2. 定义结构体类型的关键字是_____。

3. 设有结构体:

```
struct person{
```

```
    int no;
```

```
    char name[10];
```

} p; 则 p 占用内存空间为_____字节, 将 p 的 no 赋值为 1001 的语句为_____。

4. 设有变量定义如下:

```
struct student{
```

```
    char no[5];
```

```
    char name[10];
```

} s, *p=s; 则将 s 的 no 赋值为“1001”的语句为_____。用指针 p 来赋值 name 成员值为“张三”的语句为_____。

5. 带头节点的链表 L 中, 第一个数据是_____, 其指针域表示为_____。

6. 用户自定义类型说明符为_____。

三、程序分析题

1. 分析下面程序, 写出程序运行结果。

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

```
struct A  {
```

```
    int a;
```

```
    char b[10];
```

```
    double c; };
```

```
struct A f(struct A t)
```

```
{
```

```
    t. a=1002;
```

```
    strcpy(t. b, "ChangRong");
```

```
    t. c=1202. 0;
```

```
    return t;}
```

```
main( )
```



```
{
struct A a={1001, "ZhangDa", 1098. 0};
a=f(a);
jprintf("%d, %s, %6. 1f\n", a. a, a. b, a. c);
}
```

程序运行结果是：

2. 分析下面程序, 写出程序运行结果。

```
#include<stdio. h>
struct ord  {
int x, y;
} dt[2]={1, 2, 3, 4};
main( )
{
struct ord *p=dt;
printf("%d,", ++p->x);
printf("%d\n", ++p->y);
}
```

程序的运行结果是：

3. 分析下面程序, 写出程序运行结果。

```
#include<stdio. h>
#include<string. h>
struct A
{
int a;
char b[10];
double c;
};
void f(struct A *t)
{
strcpy(t->b, "ChangRong");
}
main( )
{
struct A a=(1001, "ZhangDa", 1098, 0);
f(&a);
printf("&d, &s, &6. 1f\n", a. a, a. b, a. c);
}
```

程序的运行结果是：



4. 分析下面程序, 写出程序运行结果。

```
#include<stdio. h>
#include<string. h>
typedef struct {
    char name[9];
    char sex;
    float score[2];
} STU;
void f( STU a)
{
    STU b={"Zhao", ' m' , 85. 0, 90. 0};
    int i;
    strcpy(a. name, b. name);
    a. sex=b. sex;
    for(i=0; i<2; i++)
        a. score[i]=b. score[i];
}
main( )
{
    STU c={"Qian",' p' , 95. 0, 92. 0};
    f(c);
    printf("%s, %c, %2. 0f, %2. 0f\n", c. name, c. sex, c. score[0], c. score[1]);
}
```

程序的运行结果是:

5. 分析下面程序, 写出程序运行结果。

```
#include<stdio. h>
struct tt
{
    int x;
    struct tt *y;
} *p;
struct tt a[4]={20, a+1, 15, a+2, 30, a+3, 17, a};
main( )
{
    int i;
    p=a;
    for(i=1; i<= 2; i++)
        {printf("%d," , p->x);
```




```
p = p->y;
}
}
```

程序的运行结果是：

四、补充程序题

1. 以下程序实现将用指针结构体数组输出，请将程序补充完整。

```
#include<stdio. h>

struct student
{
    int num;
    char name[20];
    char sex;
    int age;
};

struct student stu[3]={{10101, "Li Lin", ' M' , 18}, {10102, "Zhang Fun", ' M' , 19},
                        {10104, "Wang Min", ' F' , 20}};

int main( )
{
    struct student *p;
    printf("No   Name   sex   age\n");
    for (p = stu; _____; p++)
        printf("%5d %-20s %2c %4d\n", _____);
    return 0;
}
```

2. 以下程序中函数 fun 的功能是：统计 person 所指结构体数组中所有性别(sex)为 M 的记录个数，存入变量 n 中，并作为函数值返回。请填空。

```
#include<stdio. h>
#define N 3
typedef struct
{
    int num;
    char nam[10];
    char sex;
}SS;

int fun(SS person[])
{
    int i, n=0;
    for(i=0; i<N; i++)
        if( _____ == ' M' ) n++;
    _____;
}
```



```
}  
void main( )  
{ SS W[N]={1, "AA", ' F' }, {2, "BB", ' M' }, {3, "CC", ' M' }};  
  int n;  
  n=fun(W);  
  printf( "n=%d\n", n);  
}
```

3. 以下程序把三个 NODETYPE 型的变量链接成一个简单的链表,并在 while 循环中输出链表节点数据域中的数据,请填空。

```
#include<stdio. h>  
typedef struct node  {  
    int data;  
    struct node *next;  
} NODETYPE;  
main( )  
{  
    NODETYPE a, b, c, *h, *p;  
    a. data=10;  
    b. data=20;  
    c. data=30;  
    h=&a;  
    a. next=&b;  
    b. next=&c;  
    c. next=NULL;  
    p=h;  
    while(_____)   
    {  
        printf("&d", p->data);  
        _____;  
    }  
}
```

五、编写程序题

1. 定义一个结构体变量,其成员包括:职工号、姓名、性别、身份证号、工资、地址。
2. 对上述定义的变量,从键盘输入一位职工的数据,放到一个结构体变量中,然后用 printf() 函数打印出来。
3. 定义一个结构体数组,存放 6 位职工的信息,并计算最高的工资值及所有职工工资的平均值。



4. 对上述职工的信息, 增加一项“参加工作时间”, 要求含有年、月、日, 试重新定义结构体类型, 并通过一个结构体变量输入输出一位职工的信息。

5. 设学生信息中包括学号、姓名、性别及三门课成绩。试创建一结构体数组, 输入若干学生的数据, 若学号为 0 表示结束输入。

第 8 章 文 件

一、选择题

1. 在 C 程序中, 可以把整形数据以二进制形式写入文件中的函数是()。
A. fprintf B. fread C. fwrite D. fputc
2. 若执行 fopen 函数时发生错误, 则函数的返回值是()。
A. 地址值 B. NULL C. 1 D. EOF
3. 在高级语言中, 对文件的操作的一般步骤是()。
A. 打开文件—操作文件—关闭文件 B. 操作文件—修改文件—关闭文件
C. 读写文件—打开文件—关闭文件 D. 读文件—写文件—关闭文件
4. 当顺利执行了文件关闭操作时, fclose 函数的返回值是()。
A. -1 B. TRUE C. 0 D. 1
5. 若 fp 已定义并指向某文件, 当未遇到文件结束标志时, 函数 feof(fp) 的值为()。
A. 0 B. -1 C. 1 D. 一个非 0 值
6. 将数据从内存中写入文件的操作称为()。
A. 输入 B. 输出 C. 修改 D. 删除
7. 打开一个已存在的非空文件 file.txt, 选择()。
A. fp=fopen("file.txt", "r"); B. fp=fopen("file.txt", "w");
C. fp=fopen("file.txt", "r+"); D. fp=fopen("file.txt", "w+");
8. 下列关于 C 语言文件的叙述中正确的是()。
A. 文件由一系列数据依次排列组成, 只能构成二进制文件
B. 文件由结构序列组成, 可以构成二进制文件或文本文件
C. 文件由数据序列组成, 可以构成二进制文件或文本文件
D. 文件由字符序列组成, 其类型只能是文本文件
9. 有以下程序

```
#include<stdio.h>
main( )
{
    FILE *f;
    f=fopen("filea.txt", "w");
    fprintf(f, "abc");
    fclose(f);
```



```
}
```

若文本文件 filea. txt 中原有内容为 hello, 则运行以上程序后, 文件 filea. txt 中的内容为()。

- A. helloabc B. abclo C. abc D. abchello

10. 有以下程序

```
#include<stdio. h>
main( )
{
    FILE *fp;
    int a[10]={1, 2, 3, 0, 0}, i;
    fp=fopen("d2. dat", "wb");
    fwrite(a, sizeof(int), 5, fp);
    fwrite(a, sizeof(int), 5, fp);
    fclose(fp);
    fp=fopen("d2. dat", "rb");
    fread(a, sizeof(int), 10, fp);
    fclose(fp);
    for(i=0; i<10; i++)
        printf("%d, ", a[i]);
}
```

程序的运行结果是()。

- A. 1, 2, 3, 0, 0, 0, 0, 0, 0, 0
B. 1, 2, 3, 1, 2, 3, 0, 0, 0, 0
C. 123, 0, 0, 0, 0, 0, 123, 0, 0, 0
D. 1, 2, 3, 0, 0, 1, 2, 3, 0, 0

二、填空题

1. C 语言中文件按编码的方式不同可分为_____文件和_____文件两种。
2. C 程序中的文件可以用_____、_____、_____、_____方式打开。
3. C 程序使用文件应遵循一定的规则, 首先调用_____函数打开或建立文件, 再进行读写操作, 且文件操作应调用_____来完成的, 读写操作完成后应该调用_____函数关闭文件。
4. 在打开一个文件时, 如果出错, fopen 函数将返回一个_____。在程序中可以用这一信息来判别是否完成打开文件的工作, 并做相应的处理。正常完成关闭文件操作时, fclose 函数返回值为_____。
5. fputc 函数有一个返回值, 如写入成功则返回写入的字符, 否则返回一个_____, 可用此来判断写入是否成功。
6. fgets(ch, n, fp) 的意义是从 fp 所指的文件中读出_____个字符送入字符数组



ch 中。

7. fread(s, 5, 6, fp) 意义是从 fp 所指的文件中, 每次读 _____ 个字节存入数组 s 中, 连续读 _____ 个数据。

8. 能移动文件内部位置指针进行文件定位的函数主要有 _____。

9. feof 函数(文件指针)判断文件是否处于文件结束位置, 如文件结束, 则返回值为 _____, 否则为 _____。

三、程序分析题

1. 分析下面的程序。

```
#include<stdio. h>
main( )
{
    FILE *f;
    f = fopen("filea. txt", "w");
    fprintf(f, "dear friend!");
    fclose(f);
}
```

若文本文件 filea. txt 中原有内容为 welcome, 则运行以上程序后, 文件 filea. txt 中的内容是:

2. 分析下面的程序。

```
#include<stdio. h>
main( )
{
    FILE *pf;
    char *s1 = "China", *s2 = "Beijing";
    pf = fopen("abc. dat", "wb+");
    fwrite(s2, 7, 1, pf);
    rewind(pf);
    fwrite(s1, 5, 1, pf);
    fclose(pf);
}
```

以上程序执行后 abc. dat 文件的内容是:

3. 分析下面的程序。

```
#include<stdio. h>
main( )
{
    FILE *fp;
    int a[10] = {1, 2, 3}, i, n;
    fp = fopen("dl. dat", "w");
```



```
    for(i=0; i<3; i++)
        fprintf(fp, "%d", a[i]);    fprintf(fp, "\n");
    fclose(fp);
    fp = fopen("dl. dat", "r");
    fscanf(fp, "%d", &n);
    fclose(fp);
    printf("%d\n", n);
}
```

程序的运行结果是_____, 请将程序补充完整。

4. 分析下面的程序。

```
#include<stdio. h>
main( )
{
    FILE *fp;
    char ch;
    fp = fopen("file. txt", "r");
    ch = fgetc(fp);
    while(! feof(fp))
    {
        putchar(ch);
        ch = fgetc(fp);
    }
    putchar(' \n' );
    fclose(fp);
}
```

若文件 file. txt 的内容是 welcome mike! 则程序运行结果是:

5. 有以下程序

```
#include<stdio. h>
main( )
{
    FILE *pf;
    char *s1="China", *s2="Beijing";
    pf=fopen("abc. dat", "at+");
    fwrite(s2, 7, 1, pf);
    rewind(pf);
    fwrite(s1, 5, 1, pf);
    fclose(pf);
}
```



```
}

```

若文件原来的内容是 welcome, 以上程序执行后 abc. dat 文件的内容是:

四、补充程序题

1. 以下程序用来判断指定文件是否能正常打开, 请将程序补充完整。

```
#include<stdio. h>
main( )
{
    _____ ;
    if (((fp = fopen("test. txt", "r")) == _____))
        printf("未能打开文件!\n");
    else
        printf("文件打开成功!\n");
}
```

2. 下面程序的功能是:从键盘输入两个整数,将这两个数写入文件 file. txt 中,然后从该文件中读出这两个整数,计算他们的和并显示在屏幕上。

```
#include "stdio. h"
void main( )
{ _____;
    int x, y, x1, y1, z;
    printf("please input two integer numbers:");
    scanf("%d %d", &x, &y);
    if((_____ ) /* 打开文件, 准备往文件中写入数据 */
    { printf("cann' t open file");
    exit(0);
    }
    fprintf (fp, "%d %d", x, y); /* 将 x, y 的值写入文件 */
    fclose(fp); /* 关闭文件 */
    if((fp = fopen("file. txt", "r")) == NULL) /* 打开文件, 准备从文件中读出数据 */
    {printf("cann' t open file");
    exit(0); ;
    }
    _____; /* 将写入的两个整数读到变量 x1, y1 中 */
    fclose(fp); /* 关闭文件 */
    _____; /* 计算两个数的和 */
    printf("z = %d", z); /* 显示在屏幕上 */
}
```

3. 下面程序是求文件长度_____, 请将程序补充完整。



```
#include "stdio. h"
void main( )
{
    FILE *fp;
    char filename[80];
    long length;
    gets(_____);
    _____;
    if(_____)
        printf("file not found! \n");
    else
    {
        fseek(_____);
        length = _____;
        printf("Length of File is %1d bytes\n", length);
        fclose(fp);
    }
}
```

五、编写程序题

1. 从键盘上输入 n 个字符串，将其保存在文件 `string. txt` 中。
2. 从文件 `word. txt` 中读取内容，并统计单词个数。
3. 设一文本文件 `num. txt` 中放了一组整数。请编程统计并输出文件中正整数、零和负整数的个数。
4. 写入 5 个学生记录到文件“`stu1. dat`”中，记录字段为学生姓名、学号、两科成绩，记录内容自定。写入成功后，随机读取第三条记录，并用第二条记录替换。
5. 设文件 `stu2. dat` 中存放着一年级学生的基本情况，这些情况由以下结构体来描述。

```
struct student
{
    long int num;           /* 学号 */
    char [10];              /* 姓名 */
    int age;                /* 年龄 */
    char sex;               /* 性别 */
    char speciality[20];    /* 专业 */
    char addr[40];          /* 住址 */
};
```

请编写程序，输出年龄在 18~19 岁的学生学号、姓名、年龄和性别。



习题参考答案

第1章 C语言程序设计概述

一、简答题

1. 程序是指为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合。一个程序应该包括以下两方面的内容。

(1) 对数据的描述。在程序中要指定数据的类型和数据的组织形式,即数据结构。

(2) 对操作的描述。即操作步骤,也就是算法。

著名计算机科学家沃思提出了一个公式:数据结构+算法=程序。实际上,一个程序除了以上两个主要的要素外,还应当采用程序设计方法进行设计,并且用一种计算机语言来表示。

程序设计是给出解决特定问题程序的过程,是软件构造活动中的重要组成部分。程序设计往往以某种程序设计语言为工具,给出这种语言下的程序。程序设计过程应当包括分析、设计、编码、测试、排错等不同阶段。为解决某一问题编写的程序不是唯一的,不同的用户编写程序的思路也不会完全一样,因此不同的程序有不同的执行效率,这涉及程序的优化、程序所采用的数据结构以及算法等多方面的因素。

2. 计算机程序设计语言的发展经历了机器语言、汇编语言、高级语言三个阶段。

(1) 机器语言。机器语言指令是由二进制的0和1构成,不同的CPU具有不同的指令系统。机器语言使用的是针对特定型号计算机的语言,它具有灵活、直接执行和运算效率高等特点。

(2) 汇编语言。汇编语言指令是机器指令的符号化,与机器指令存在着直接的对应关系,汇编语言针对计算机特定硬件而编制程序,能准确发挥计算机硬件的功能和特长,程序精练、效率高。

使用汇编语言编写的程序,机器不能直接识别,要由一种程序将汇编语言翻译成机器语言,这种起翻译作用的程序叫汇编程序。汇编程序是语言处理系统软件,汇编程序把汇编语言翻译成机器语言的过程称为汇编。

(3) 高级语言。高级语言与计算机的硬件结构及指令系统无关,在形式上接近于算术语言和自然语言,具有更强的表达能力,可方便地表示数据的运算和程序的控制结构,能更好地描述各种算法,而且容易学习掌握。但高级语言编译生成的程序代码一般比用汇编程序语言设计的程序代码要长,执行的速度也慢。

高级语言经历了从早期语言到结构化程序设计语言,从面向过程的程序设计语言到面向对象的程序设计语言的过程。结构化程序设计语言直接支持结构化的控制结构,具有很强的过程结构和数据结构能力。PASCAL、C、FORTRAN语言就是它们的突出代



表。面向对象程序设计语言以“对象+消息”程序设计范式构成的程序设计语言，称为面向对象语言。比较流行的面向对象语言有 Delphi、Visual Basic、Java、C++等。

3. 一个 C 语言程序的基本结构分为 3 个部分：预处理部分、函数、注释。

(1) 预处理部分。`#include<stdio.h>` 是一条预处理命令，用“#”号开头，后面不能加“;”号。`stdio.h` 是系统提供的头文件，其中包含有关输入输出函数的信息，如程序中用到的 `printf` 输出函数。输出语句中的“%d”为输出格式符，它指定输出结果时的数据类型和格式，程序在执行时，该位置由具体数据替代。

(2) 函数：函数是 C 语言程序的基本单位，用于描述程序所完成的功能。

一个 C 语言程序可以包含任意多个函数，但必须有且只有一个 `main` 主函数。一个 C 语言程序总是从主函数开始执行，最后在主函数结束。主函数的位置在程序中是任意的，其他函数总是通过函数调用语句来执行。主函数可以调用任何其他函数，任何非主函数之间也可以相互调用，但是均不能调用主函数。

一个函数包含两部分：函数首部和函数体。函数首部包括函数类型、函数名和形式参数；函数体需要用花括号括起来，左括号表示函数体的开始，右括号表示函数体的结束。

函数定义形式可表示如下。

函数类型 函数名(形式参数列表)

```
{  
    函数体  
}
```

(3) 注释：为了对程序代码进行说明，可以添加注释，作用是帮助用户阅读程序，它对程序的运行不起作用，在对源程序进行编译时，注释会被忽略。注释内容可以是西文，也可以是中文，注释通常用于说明变量的含义、程序段的功能。注释部分可以放在程序中任意合适的位置，一个好的程序应该有必要的注释，这样可以增加可读性。C 语言程序中用“`/* ..... */`”表示注释一个程序块，注释内容可以是一行或多行。“`/*`”和“`*/`”必须成对出现，且“`/`”和“`*`”之间不能有空格。

4. C 语言的主要特点包括以下方面。

(1) 语言简洁、紧凑，使用方便、灵活，C 语言只有 32 个关键字、9 种控制语句，且程序形式自由。

(2) 运算符丰富。C 语言有算术、关系、逻辑、位运算等运算符 34 种。

(3) 数据类型丰富，具有现代语言的各种数据结构。

(4) 具有结构化的控制语句，是完全模块化和结构化的语言。

(5) 语法限制不太严格，程序设计自由度大。

(6) 允许直接访问物理地址，能进行位运算，能实现汇编语言的大部分功能，可直接对硬件进行操作，兼有高级和低级语言的特点。

(7) 目标代码质量高，程序执行效率高，只比汇编程序生成的目标代码效率低 10%~20%。



(8) 程序可移植性好(与汇编语言相比),基本上不做修改就能用于各种型号的计算机和各种操作系统。

5. C 语言源程序文件的后缀名是 .c, 目标文件的后缀名是 .obj, 可执行文件的后缀名是 .exe。

6. C 语言程序的开发过程可分为以下 5 个步骤。

(1) 通过对问题的分析,设计一种解决问题的可行方案(包括数据结构和算法)。

(2) 通过 C 语言严格描述这个解决方案(编写 C 语言源程序)。

(3) 对源程序进行编译,如果发现程序编译错误,则返回(2)修改源程序,直到正确为止。

(4) 对目标程序进行连接,若正确就可运行程序,如果连接过程中发现连接错误,则返回(2)修改源程序。

(5) 运行程序,得到结果,如果发现程序能运行,但结果不正确,则:

① 如果源程序算法编写错误,则返回(2)重新编写源程序;

② 如果分析问题本身有错误,则返回(1)重新分析问题。

7. C 语言程序设计课程需要学习和掌握的知识和技能很多,包括知识的记忆和灵活掌握,解决问题的思维方法,具体处理的手段、技巧,还有许多实际工作和操作技能问题。

(1) 分析问题的能力,特别是从计算机和程序的角度分析问题的能力。应逐渐学会从问题出发,通过逐步分析和分解,把原问题转化为能用计算机通过程序方式解决的问题。在此过程中构造出一个解决方案。为此,参与者既需要熟悉计算机,也需要了解相关专业知识。

(2) 掌握所学的计算机程序语言,熟悉语言中的各种结构,包括其形式和意义。语言是解决程序问题的工具,要想写好程序,必须熟悉所用语言。应注意,熟悉语言绝不是背诵定义,这个熟悉过程只有在程序设计的实践中才能完成。仅靠看书、读程序、抄程序不可能真正学会写程序。要学会写程序,就需要反复地亲身实践从问题到程序的整个过程,开动脑筋,想办法处理遇到的各种情况。

(3) 学会写程序。虽然写过程序的人很多,但会写程序、能写出好程序的人就少得多了。经过多年的程序设计实践,人们对什么是“好程序”有了许多共同认识。例如,解决同样的问题如何使写出的程序比较简单。这里可能有计算方法的选择问题,有语言的使用问题,其中需要确定适用的程序结构等。除了程序本身是否正确外,人们还特别关注写出的程序是否具有良好的结构,是否清晰,是否易于阅读和理解,当问题中有些条件或要求改变时,是否容易修改程序去满足新的要求,等等。

(4) 检查程序错误的能力。初步写出的程序常会包含一些错误。虽然语言系统能帮我们查出其中的一些,并通告发现错误的位置,但确认实际错误和实际位置,确定应如何改正,这些永远是编程者自己的事。对于系统提出的各种警告、系统无法检查的错误等的认定就更要依靠人的能力。这种能力也需要在学习中有意识地锻炼。

(5) 熟悉所用工具和环境。程序设计要用一些编程工具,要在具体计算机环境中进



行,熟悉工具和环境也是这个学习中很重要的一部分。目前大部分读者可能要用某种集成开发环境做程序开发,熟悉这种环境的使用也很重要,可以大大提高我们的工作效率。

二、选择题

1. B 2. A 3. B 4. A 5. A 6. A 7. A

第2章 数据类型与表达式

一、选择题

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

二、填空题

1. 12 2. 24 3. 11.500000 4. 23
5. 11 6. 2.000000 7. -14 8. 36

三、程序分析题

1. $a+b=-13$, $a+m=26$

2. 9, 11, 9, 10

解析: 因为 $++i$ 是先进行加 1 后运行, 故 $i=9$, $m=9$; 而 $j++$ 是先运行后进行加 1, 故 $n=10$, $j=11$ 。

3. 20

解析: $a+=(a=10)$ 等价于 $a=a+10=10+10=20$ 。

4. 3

解析: $p/10=3$, $3>0$ 为真, 则 $p/10$ 的值就是运行的结果, 故结果为 3。

5. $a=21$, $a=000021$, $a=\square\square\square\square21$, $a=21\square\square\square\square$

$b=2389065$

第3章 C 语言程序控制结构

一、选择题

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

二、填空题

1. 7
2. 顺序结构、选择结构、循环结构
3. 1, $y=1$, $x=0$
4. 5

三、程序分析题

1. 18



2. A, 68

3. 24

四、补充程序题

1. $b! = 0$

2. $1.0/i$

3. $s+a/b;$ $a+b;$

五、编写程序题

1. #include<stdio. h>

void main()

{

int x;

printf("please enter a number:");

scanf("%d", &x);

if(x%2==0)

printf("%d 是一个偶数\n", x);

else

printf("%d 是一个奇数\n", x);

}

2. #include<stdio. h>

void main()

{ int i, j, k;

for(i=0; i<=3; i++)

{

for(j=0; j<=2-i; j++)

printf(" ");

for(k=0; k<=2* i; k++)

printf("*");

printf("\n");

}

for(i=0; i<=2; i++)

{ for(j=0; j<=i; j++)

printf(" ");

for(k=0; k<=4- 2* i; k++)

printf("*");

printf("\n");

}



```
    }  
3. #include<stdio. h>  
   int main( )  
   {   int p, r, n, m, temp;  
       printf("请输入两个正整数 n, m:");  
       scanf("%d%d", &n, &m);  
       if(n<m)  
       {temp = n;  
        n = m;  
        m = temp;  
       }  
       p = n*m;  
       while(m! = 0)  
       {   r = n%m;  
           n = m;  
           m = r;  
       }  
       printf("它们的最大公约数为:%d\n", n);  
       printf("它们的最小公倍数为:%d\n", p/n);  
       return 0;  
   }  
4. #include<stdio. h>  
   int main( )  
   {  
       char c;  
       int letters=0, space=0, digit=0, other=0;  
       printf("请输入一行字符: \n");  
       while((c=getchar( ))! = ' \n' )  
       {   if(c>= ' a'  &&c<=' z'  ||  c>=' A'  && c<=' Z' )  
           letters++;  
           else if(c=="")  
               space++;  
           else if(c>=' 0' &&c<=' 9' )  
               digit++;  
           else  
               other++;  
       }  
   }
```



```
printf("字母数为:%d\n 空格数为:%d\n 数字数为:%d\n 其他字符数为:%d\n", letters, space, digit, other);
return 0;
}
5. #include<stdio. h>
int main( )
{   int day, x1, x2;
    day=9;
    x2=1;
    while(day>0)
    {   x1=(x2+1)*2;
        x2=x1;
        day-;
    }
    printf("total=%d\n", x1);
    return 0;
}
6. #include<stdio. h>
int main( )
{   double s=0, t=1;
    int n;
    for(n=1; n<= 50; n++)
    {   t=t*n;
        s=s+t;
    }
    printf("1!+2!+3!+...+50!=%d\n", s);
    return 0;
}
```

第4章 函数与预处理

一、选择题

1. A 2. D 3. B 4. A 5. D 6. A 7. A 8. D 9. D 10. C 11. C 12. C
13. B 14. B

二、编写程序题

1. 参考程序。



```
#include<stdio. h>
int search(int a[], int m)
{ int i;
  for(i=0; i<=9; i++)  if([a[i]==m]) return(i);
  return(- 1);
}
void main( )
{ int a[10], m, i, no;
  for(i=0; i<10; i++)
    scanf("%d", &a[i]);
    scanf("%d", &m);
    no = search(a, m);
    if(no>=0) printf("\n ok found! %d", no+1);
    else printf("\n Sorry NOT found!");
}
```

2. 参考程序。

```
#include<stdio. h>
int gcd(int n, int m)  //计算 n 和 m 的最大公约数
{ int r;
  r = n%m;
  while(r!=0)
  { n = m;
    m = r;
    r = n%m;
  }
  return m;
}
int lcs (int n, int m)  //计算 n 和 m 的最小公倍数
{ int g, lc;
  g = gcd(n, m);
  lc = n*m
  return lc;
}
int main( )
{ int m, n;
  while(scanf("%d%d", &m, &n)!= EOF)
    printf("%d %d\n", gcd(n, m), lcs(n, m));
}
```




```
    return 0;
}
```

3. 参考程序。

```
#include<stdio. h>
#define N 15
void insert(int a[], int n, int x)
{   int i;
    for(i = n- 1; i >= 0; i- )
        if(a[i] >= x) a[i+1] = a[i];
    a[i+1] = x;
}
void main( )
{   int a[N];
    int x, i;
    scanf("%d", &n);
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    sort(a, n); //本函数需要另外编写
    scanf("%d", &x);
    insert(a, n, x);
    for(i=0; i<=n; i++)
        printf("%4d", a[i]);
    printf("\n");
}
```

4. 参考程序。

```
#include<stdio. h>
#define N 15
int del(int a[], int n, int x)
{   int i, j;
    for(i = n- 1; i >= 0; i- )
        if(a[i] == x) break;
    if(i >= 0)
        for(j = i; j < n; j++)
            a[j] = a[j+1];
    return i;
}
void main( )
```



```
{ int a[N];
  int x, i, k;
  scanf("%d", &n);
  for(i=0; i<n; i++)
    scanf("%d", &a[i]);
  sort(a, n); //本函数需要另外编写
  scanf("%d", &x);
  k=del(a, n, x);
  if(k>=0)
  { printf("删除成功\n");
    n=n-1;
  }
  else printf("数组中无此元素,删除不成功\n");
}
```

5. 参考程序。

```
#include<stdio. h>
int huiwen(int n)
{ int t, s=0;
  t=n;
  while(t)
  {s=s*10+t%10; t/=10; }
  if(s==n) return 1;
  else return 0;
}
void main( )
{ int k, n;
  printf("输入数字:");
  scanf("%d", &n);
  k=huiwen( n);
  if(k==1) printf("是\n");
  else printf("不是\n");
}
```

6. 参考程序。

```
#include<math. h>
#include<stdio. h>
struct Data
{ int a, b, c;
```



```
    } ans[6000];
#define N 5000
int main( )
{   int l, r, flag;
    int i, j, k, cnt=0;
    for(i=3; i<=N; i++){
        for(j=i+1; j<=N; j++){
            k=sqrt(i*i+j*j+0.5);
            if(k>N)
                break;
            if(k*k==i*i+j*j&k>j)
        {   printf("%4d%4d%4d\n", i, j, k);
            ans[cnt]. a = i;
            ans[cnt]. b = j;
            ans[cnt++]. c = k;
        }}}
while(scanf("%d%d", &l, &r)! = EOF)
{   flag=0;
    for(i=0; i<cnt; i++)
        if(ans[i]. a>=l&&ans[i]. c<=r){
            flag=1;
            printf("%d%d%d\n", ans[i]. a, ans[i]. b, ans[i]. c);
        }
    if(flag==0)
        printf("NO\n");
}
return 0;
}
```

7. 参考程序。

```
#include "stdio. h"
int f(int n)
{   int ff;
    if(n==1 || n==2)
        ff=1;
    else ff=f(n-1)+f(n-2);
    return ff;
}
```



```
void main( )
{ int n, t;
  scanf("%d", &n);
  t=f(n);
  printf("%d\n", t);
}
```

第5章 数组与字符串

一、选择题

1. B 2. D 3. D 4. C 5. B 6. A 7. D 8. D 9. A 10. C 11. B

二、填空题

1. 14 5
2. 1
3. 30
4. ABCDEF
5. `int a[4][4] = { {1}, {0, 1}, {0, 0, 1}, {0, 0, 0, 1} };`
6. 4

三、程序分析题

1. 输出结果: 8 5 2
2. 输出结果: 0
3. 输出结果: 120
4. 输出结果: 18
5. 输出结果: 无效的字符, 因为 `a[0]` 没有被赋值

四、补充程序题

```
1. #include<stdio. h>
   int main( )
   {   int i, sum=0, a[10];
       printf("请输入 10 个整数: \n");
       for(i=0; i<10; i++)
           scanf("%d", &a[i]);
       for(i=0; i<10; i++)
       {   if(a[i]%2!=0)
           break;
           sum+=a[i];
       }
   }
```



```
        printf("sum = %d \n", sum);
        return 0;
    }

2. #include<stdio. h>
   #define  N  10
   int  main( )
   {   int i, j, t, a[N]={23, 67, 31, 28, 49, 37, 80, 46, 19, 21};
       printf("排序前数组为: \n");
       for(i=0; i<N; i++)
           printf("%d ", a[i]);
       for(i=0; i<N; i++)
       { if( i<N/2 )
         { for( j=i; j<N/2- 1; j++)
           if( a[j]>a[j+1] )
             { t=a[j]; a[j]=a[j+1]; a[j+1]=t; }
         }
         else{
             for( j=i; j<N- 1; j++)
                 if( a[j]<a[j+1] )
                     { t=a[j]; a[j]=a[j+1]; a[j+1]=t; }
         }
       }
       printf("排序前数组为: \n");
       for(i=0; i<N; i++)
           printf("%d ", a[i]);
       return 0;
   }

3. #include<stdio. h>
   #define  N  3
   int  main( )
   {   int i, j, t, a[N][N];
       printf("请输入 %dx%d 矩阵的数据: \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++)
       {   j=N- 1- i;
```



```
        t=a[i][i]; a[i][i]=a[i][j]; a[i][j]=t;
    }
    printf("互换主次对角线元素值后, 矩阵为: \n");
    for(i=0; i<N; i++)
    { for(j=0; j<N; j++)
        printf("%d ", a[i][j]);
        printf("\n");
    }
    return 0;
}
```

五、编写程序题

1. #include<stdio. h>

```
int main( )
{ int f1, f2, fn, i, n;
  printf("please input n: ");
  scanf("%d", &n);
  f1 = f2 = 1;
  for(i=3; i<=n; i++)
  { fn = f2+f1;
    f2 = f1;
    f1 = fn;
  }
  printf("\n 菲波那契数列的第%d 项是: \n ", n);
  printf("f(%d) = %d \n \n", n, fn);
  return 0;
}
```

2. #include<stdio. h>

```
int main( )
{ int letter, space, digit, other;
  char ch;
  letter = space = digit = other = 0;
  while ((ch = getchar ( )) != '\n' )
  { if (ch>=' a' && ch<=' z' || ch>=' A' &&ch<=' Z' )
      letter++;
    else if (ch>=' 0' && ch<=' 9' )
      digit++;
    else if (ch=="")

```



```
        space++;
    else
        other++;
}
printf ("字母:%d\n", letter);
printf ("空格:%d\n", space);
printf ("数字:%d\n", digit);
printf ("其他字符:%d\n", other);
return 0;
}

3. #include<stdio. h>
int main( )
{ int n, i, j, t, a[200];
  scanf("%d", &n);
  for(i=0; i<n; i++)
      scanf("%d", &a[i]);
  for(i=0; i<n- 1; i++)
      for(j=0; j<n- 1- i; j++)
          if(a[j]>a[j+1])
          { t=a[j];
            a[j]=a[j+1];
            a[j+1]=t;
          }
  for(i=0; i<n; i++)
      printf("%d", a[i]);
  printf("\n");
  return 0;
}

4. #include<stdio. h>
#define MAX 1000
int sign[MAX]={1, 1, 0}; //如果 sign[x]=0, x 为素数
int main( )
{ int a, b, i, j;
  printf("input a, b = \n");
  scanf("%d%d", &a, &b);
  for(i=2; i<=MAX; i++) //从 2 开始到 MAX
  { if(sign[i]==0) //如果 i 为素数
```



```
        { for(j=2*i; j<=b; j+=i) //j=2*i j+=i 保证了 i 是 j 的约数,却 i!=j
            sign[j]=1;          //把约数为 i 的所有值标为 1
        }
    }
    for(i=a; i<=b; i++)
    { if(sign[i]==0) //如果 i 为素数
        printf("%d ", i);
    }
    printf("\n");
    return 0;
}
```

5. #include<stdio. h>

```
int main( )
{ int n, i, j;
  int a[100][100];
  int sum1=0;
  int sum2=0;
  scanf("%d", &n);
  for(i=1; i<=n; i++)
  { for(j=1; j<=n; j++)
    { scanf("%d", &a[i][j]);
      if(i==j)
        sum1+=a[i][j];
      else if(j==n- i+1)
        sum2+=a[i][n- i+1];
    }
  }
  printf("%d %d", sum1, sum2);
  return 0;
}
```

6. #include<stdio. h>

```
int main( )
{ int a[100], b[100], c[200], m, n, tmp, i, j=0, k=0;
  printf("请输入第一个数组的元素个数然后依次输入元素:");
  scanf("%d", &m);
  for(i=0; i<m; i++)
    scanf("%d", &a[i]);
```




```

printf("\n 请输入第 2 个数组的元素个数然后依次输入元素:");
scanf("%d", &n);
for(i=0; i<n; i++)
    scanf("%d", &b[i]);
i=0;
while(j<m&&k<n)
{ if(a[j]<=b[k])
    { c[i]=a[j]; j++; i++;
    }
    else
    { c[i]=b[k]; k++; i++;
    }
}
while( j<m )
{c[i]=a[j]; j++; i++; }
while( k<n )
{c[i]=b[k]; k++; i++; }
for(i=0; i<m+n; i++)
    printf("%d", c[i]);
return 0;
}
7. #include<stdio. h>
int main( )
{ char ch;                //ch 用来每次接收一个字符
  int i, s[26]={0};        //数组 s[]用来统计每个小写字母的个数
  printf("please input a string:\n");
  scanf("%c", &ch);
  while(ch!= '\n' )        //输入一行字符,以回车键结束
  { if(ch>' A' &&ch<' Z' )  //遇到大写字母时,转换成小写来处理
    ch+=32;
    s[ch-' a' ]++;          //0~25 对应 a~z
    scanf("%c", &ch);
  }
  printf("\n");
  for(i=0; i<26; i++)
    if(s[i])                //只输出输入过的字母统计
      printf("%c:%d\n", (char)(i+' a' ), s[i]); //(char)(i+' a' )用强制类型转换,输出

```



对应字母

```
return 0;
```

```
}
```

8. #include<stdio. h>

```
int main( )
```

```
{ int i, k, g1[11]={0}, g2[5]={0}, s[100];
```

```
printf("input the score of C \n");
```

```
for(i=0; i<100; i++)
```

```
{ scanf("%d", &s[i]);
```

```
if( s[i]<0 ) break;
```

```
k=s[i]/10;
```

```
g1[k]++;
```

```
}
```

```
for(i=0; i<6; i++)
```

```
g2[0]=g2[0]+g1[i];
```

```
for(i=1; i<=4; i++)
```

```
g2[i]=g1[i+5];
```

```
g2[4]=g2[4]+g1[10];
```

```
printf("成绩在 0- 59 分的人数是%d\n", g2[0]);
```

```
printf("成绩在 60- 69 分的人数是%d\n", g2[1]);
```

```
printf("成绩在 70- 79 分的人数是%d\n", g2[2]);
```

```
printf("成绩在 80- 89 分的人数是%d\n", g2[3]);
```

```
printf("成绩在 90- 100 分的人数是%d\n", g2[4]);
```

```
return 0;
```

```
}
```

9. #include<stdio. h>

```
int main( )
```

```
{ int n, i, b, flag;
```

```
int a[10];
```

```
int c=0, d[10];
```

```
printf("Enter n: ");
```

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

```
for(i=0; i<n; i++)
```

```
scanf("%d", &a[i]);
```

//输入数组

```
for(b=0; b<n- 1; b++)
```

```
{ flag=0;
```

```
for(i=b+1; i<n; i++)
```



```

        if(a[b]==a[i])
        { a[i]=0;
          flag=1;
        }
        if(flag) a[b]=0;
    }//定义一下重复的为零,这里最后才把第一个比较的变为零!
    for (i=0; i<n; i++)
        if(a[i]!=0)
        { d[c]=a[i];
          c++;
        }//重新定义一个新数组
    for (i=0; i<c; i++)
        printf("%d", d[i]);
    return 0;
}

```

10. #include<stdio. h>

```

int main( )
{ double a[100][100], b[100][100], h[100][100];
  int m, n, p, q, i, j, k, l, c, d;
  double sum;
  scanf("%d %d", &m, &n);           //输入行数和列数
  for(i=0; i<m; i++)
      for(j=0; j<n; j++)
          scanf("%lf", &a[i][j]);    //输入第一个矩阵
  scanf("%d %d", &p, &q);           //输入行数和列数
  for(l=0; l<p; l++)
      for(k=0; k<q; k++)
          scanf("%lf", &b[l][k]);    //输入第二个矩阵
  for(c=0; c<m; c++)                 //行数小于等于第一个矩阵的行数
      { for(d=0; d<q; d++)            //列数小于等于第二个矩阵的列数
          { sum=0;
            for(k=0; k<n; k++)         //第一个矩阵行数和第二个矩阵列数
                sum+=a[c][k]*b[k][d]; //计算
            h[c][d]=sum;
          }
      }
  }
  for(c=0; c<m; c++)

```



```
{ for(d=0; d<q; d++)
    printf("%.0f ", h[c][d]);    //输出
printf("\n");
}
return 0;
}
```

11. #include<stdio. h>

```
int main( )
{ char string[81];                //定义长度为 81
  int i, num=0, word=0;
  char c;
  gets(string);                  //输入字符串
  for(i=0; (string[i])!= '\0' ; i++) //直到字符串的末尾
  { if(string[i]=="")
    word=0;                      //如果为空格 那么将知道有单词
    else if(word==0)             //二重判断直到不为空格
    { word=1;                    //word 状态置为 1
      num++;                    //累加单词
    }
  }
  printf("There are %d words in the line. \n", num); //输出有几个单词
}
```

12. #include<stdio. h>

```
#include<stdlib. h>
int main( )
{ int i, j, k, n;
  int a[100]={1};
  int pre, sum;
  printf("Enter the YangHui Triangle Levels:");
  scanf("%d", &n);
  if (n<1) {
    printf("The Levels %d is invalid\n", n);
    return 0;
  }
  for (i=0; i<n; i++) {
    for (k=0; k<n- i; k++) {
      printf(" ");
    }
  }
}
```



```
    }
    pre = a[0];
    for (j=0; j<i; j++) {
        sum = pre;
        pre = a[j+1];
        a[j+1] = sum+pre;
        printf("%3d ", a[j]);
    }
    printf("%3d \n", a[j]);
}
return 0;
```

```
}
```

```
13. #include<stdio. h>
#include<stdlib. h>
#include<memory. h>
void magic(int);
void magic1(int);
void magic4(int);
void magic42(int);
void magic(int n)
{
    if(n==1)
    { printf("1 \n");
      return;
    }
    if(n==2)
    { printf("不存在 \n");
      return;
    }
    if(n%2==1)
    { magic1(n);
      return;
    }
    else if(n%4==0)
    { magic4(n);
      return;
    }
    else
```



```
{magic42(n);
    return;
}
}

void magic1(int n)
{   int **a=(int **)malloc(sizeof(int*)*n);
    int i;
    int j;
    int k;
    for(i=0; i<n; i++)
    {   a[i]=(int *)malloc(sizeof(int)*n);
    }
    for(i=0; i<n; i++)
    {   for(j=0; j<n; j++)
        {   a[i][j]=0;
        }
    }
    //首先确定 1 的位置,在第一行中间列。
    j=n/2;
    a[0][j]=1;
    for(k=2; k<=n* n; k++)
    {   i-;
        j++;
        //保证行列数的循环。
        if(i<0)
        {   i=n-1;
        }
        if(j>n-1)
        {   j=0;
        }
        if(a[i][j]==0)
        {   a[i][j]=k;
        }
        else
        {   i=(i+2)%n;
            j=(j-1+n)%n;
            a[i][j]=k;
        }
    }
}
```



```

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

void magic4(int n)
{   return;
}

void magic42(int n)
{   return;
}

int main( )
{ int n;
  while(scanf("%d", &n)!= EOF)
  {   magic(n);
  }
}

```

第6章 指 针

一、选择题

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

二、填空题

1. void swap(int *x, int *y)
{



```
int temp = *x;
*x = *y;
*y = temp;
}
```

说明：当输入 Ctrl+Z 键时，表示结束数据输入。

2. void f(int a[], int n, int *min, int *max)

```
{
    int i;
    *min = *max = a[0];
    for(i = 1; i < n; i++)
    {
        if(*max < a[i]) *max = a[i];
        else if(*min > a[i]) *min = a[i];
    }
}
```

说明：当输入 Ctrl+Z 键时，表示结束数据输入。

3. 11, 12, 13, 14, 15, 6, 7, 8, 9, 10, 11, 12 65

4. $s_1 = 1+2+3+4+5+6 = 21$

$s_2 = 1+4+7+4+7+10 = 33$

5. string = program code

substring = code

6. Qianqi 100; Zhaoliu 90; Wangwu 80;

Lisi 70; Zhangyi 60;

三、程序分析题

1. 分析：函数通过指针形参访问主函数中的字符串，通过指针移动来访问字符串中的每一个字符，并对该字符进行处理，如果是小写字母，则转换为大写字母。

输出结果：ABCD1234EFGH

2. 分析：通过检查字符串中的字母的 ASCII 码是否被 2 整除，如果不被 2 整除，则输出该字母。

输出结果：acYE

3. 输出结果：3, 5, 3, 5, 5

4. 输出结果：46.00, 57.00, 88.00

5. 分析：程序利用 p, q 两个指针一个指向字符串首部，一个指向字符串的尾部，将指针往中间移动，实现字符串反序存放。如输入：“fdsFDSFdsf”，则输出：“dsfDFSDFdsf”。

四、补充程序题

1. #include <stdio. h>

int main()



```
{
    int a[10], max, i;
    for(i=0; i<=9; i++)
        scanf("%d", &a[i]);
    max = a[0];
    for(i=1; i<=9; i++)
    {
        if(a[i]>max)
            max = a[i];
    }
    printf("max = %d\n", max);
}

2. #include<stdio. h>
#include<stdio. h>
void copystr(char *p1, char *p2, int m)
{
    char *p, *q;
    p = p1 + m - 1;
    q = p2;
    while(*p != '\0')
    {
        *q = *p;
        p++;
        q++;
    }
    *q = '\0'; /*新字符串末尾加上字符串结束标志 */
}

int main( )
{
    int m;
    char s1[80], s2[80];
    printf("请输入一字符串: \n");
    gets(s1);
    printf("请输入开始复制的位置(m):");
    scanf("%d", &m);
    copystr(s1, s2, m); /* 调用函数进行复制 */
    puts(s2);
}
```



```
}  
3. #define N 10  
#include<stdio. h>  
int*find_position(int *a, int len)  
{  
    int *p, *q, sum=0, i=0, j=0, min=0, *pos;  
    pos=p=q=a;  
    for(i=0; i<3; i++)//以第一组三位数之和当作最小  
        min+=*q++;  
    i=0;  
    while(i<len)  
    {  
        sum=0;  
        p=a+i;  
        for(j=0; j<3; j++)//取连续的三个数累加  
        {  
            if(p>a+len-1)  
                p=a; //若超出了数组的范围,则需要回溯到开始位置  
            sum=sum+(*p);  
            p++;  
        }  
        if(min>sum)  
        {  
            pos=a+i;  
            min=sum;  
        }  
        i++;  
    }  
    return pos;  
}  
int main( )  
{  
    int a[10]={5, 2, 6, 1, 4, 1, 9, 2, 3, 8};  
    int *p, sum=0;  
    p=find_position(a, N);  
    printf("\n 最小组为:%d+", *p);  
    sum+=*p;
```



```

    p++;
    if(p>a+N-1)
        p=a;
    printf("%d", *p);
    sum+=*p;
    p++;
    if(p>a+N-1)
        p=a;
    printf("+%d=", *p);
    sum+=*p;
    printf("%d\n", sum);
    return 0;
}

```

五、编写程序题

1. 解法一:

```

#include<stdio. h>
#include<string. h>
void sort(char *b[], int n);
int main( )
{
    int i=0, q;
    char p[100][20], *a[100];
    while(gets(p[i]))
        { a[i]=p+i; i++;}          //p[i]等价 p+i
    q=i;
    sort(a, q);
    for (i=0; i<q; i++)
        printf("%s\n", *(a+i) );   /*(a+i)等价 a[i]
    return 0;
}

void sort(char * b[],int n)
{
    int i , k;
    char *temp;
    for (i=n; i>0; i- )
        for(k=0; k<i- 1; k++)
            if(strcmp(b[k], b[k+1])>0)

```



```
{ temp = b[k]; b[k] = b[k+1]; b[k+1] = temp; }
```

```
}
```

说明:

输入多个英文单词(不超过 100 个), 每行一个单词;

最后一行输入 Ctrl+Z 键, 表示数据输入结束, 输出所有的单词(按字母顺序从小到大)。

解法二:

```
#include<stdio. h>
#include<string. h>
void gdy(char (*q)[20], int n)
{ int i, j; char p[100][20];
  for(i=0; i<n; i++)
    for(j=1+i; j<n; j++)
      if(strcmp(q[i], q[j])>0)
        { strcpy(p[i], q[i]);
          strcpy(q[i], q[j]);
          strcpy(q[j], p[i]);}
  for(i=0; i<n; i++)
    printf("%s\n", q[i]);
}
int main( )
{
  char a[100][20]; int n=0;
  while(scanf("%s", &a[n])!=EOF)
    n++;
  gdy(a, n);
  return 0;
}
```

说明:

输入多个英文单词(不超过 100 个), 每行一个单词;

若结束数据输入, 需要在最后两行输入 Ctrl+Z 键(即每行输入 Ctrl+Z 键后再按回车键), 最后输出所有的单词(按字母顺序从小到大)。

2. 解法一:

```
#include<stdio. h>
int f(char *s1, int start, int len, char*s2)
{
  int i;
```



```

    s1 += start - 1;
    for(i = 0; i < len; i++)
        { *s2 = *s1; s1++; s2++; } //也可以使用*(s2++) = *(s1++);
    *s2 = 0;                        //也可以使用*s3 = ' \0' ;
    return 0;
}
int main( )
{
    char s1[1024], s2[1024];
    int start, len;
    gets(s1);
    scanf("%d %d", &start, &len);
    f(s1, start, len, s2);
    puts(s2);                        //也可以使用 printf("%s\n", s2);
    return 0;
}

```

解法二:

```

#include<stdio. h>
#include<string. h>
void f(char *s1, int start, int len, char *s2)
{
    char ch; s2 = s1 + start - 1;
    if(start + len < strlen(s1))
        { ch = s2[len]; s2[len] = ' \0' ; }
    printf("%s\n", s2);
}
int main( )
{
    int start, len;
    char s[200], s1[200];
    gets(s);
    scanf("%d%d", &start, &len);
    f(s, start, len, s1);
    return 0;
}

```

解法三:

```

#include<stdio. h>

```



```
void f(char *s1, int start, int len, char *s2)
{
    int i, j;
    for(i=(start-1), j=0; i<(start+len-1); j++, i++)
        s2[j]=s1[i];
}

int main( )
{
    char a[100], b[100]={' \0' };
    int start, len;
    gets(a);
    scanf("%d %d", &start, &len);
    f(a, start, len, b);
    printf("%s\n", b);
    return 0;
}
```

说明:VC++6.0 环境中,当输入一行英文时,每行只能显示 80 个字符,超出的部分显示在下一行。

3. 解:

```
#include<stdio. h>
void f(char *s1, char *s2, char *s3)
{
    while(*s1)           /*s1 等价*s1!=0
        *s3++=*s1++;
    while(*s2)           /*s2 等价*s2! =0
        *s3++=*s2++;
    *s3=' \0' ;          //也可以使用*s3=0;
}

int main( )
{
    char s1[512], s2[512], s3[1024];
    gets(s1);
    gets(s2);
    f(s1, s2, s3);
    puts(s3);             //也可以使用 printf("%s\n", s3);
    return 0;
}
```



解法一:

```
#include<stdio. h>
struct xueshen
{
    int a[6];
}x[20];
void hanshu1(struct xueshen x[], int m)
{
    int i, j, z;
    for(i=0; i<m; i++)
    {
        z=0;
        for(j=0; j<6; j++)
        {
            printf("%d ", x[i]. a[j]);
            if(j>0)z=z+x[i]. a[j];
        }
        printf("%. 1f\n", (float)z/5);
    }
}
void hanshu2(struct xueshen *p, int m)
{
    int i, j, z;
    struct xueshen *q;
    for(j=1; j<6; j++)
    {
        z=0;
        for(i=0; i<m; i++)
        {
            q=p+i;
            z=z+q->a[j];
        }
        if(j<2)printf("%. 1f", (float)z/m);
        else printf(" %. 1f", (float)z/m);
    }
}
int main( )
```



```
{
    int i, j, m, n;
    scanf("%d", &m);
    for(i=0; i<m; i++)
        for(j=0; j<6; j++)
            scanf("%d", &x[i].a[j]);
    hanshu1(x, m);
    hanshu2(x, m);
    return 0;
}
```

解法二:

```
#include<stdio.h>
void print(int a[20][5], int b[20], int n)
{
    int i, j, c[20]={0};
    for(i=0; i<n; i++)
    {
        printf("%d ", b[i]);
        for(j=0; j<5; j++)
        {
            printf("%d ", a[i][j]);
            c[i]+=a[i][j];
        }
        printf("%.1lf\n", c[i]*1.0/5);
    }
}

void aver(int (*p)[5], int n)
{
    int sum, i, j;
    for(j=0; j<5; j++)
    {
        sum=0;
        for(i=0; i<n; i++)
            sum+=*(*(p+i)+j);
        if(j<4)
            printf("%.1lf ", sum*1.0/n);
        else
```




```

        printf("%. 1lf\n", sum*1. 0/n);
    }
}
int main( )
{
    int a[20][5], b[20], n, i, j;
    scanf("%d", &n);
    for(i=0; i<n; i++)
    {
        scanf("%d", &b[i]);
        for(j=0; j<5; j++)
            scanf("%d", &a[i][j]);
    }
    print(a, b, n);
    aver(a, n);
    return 0;
}

```

解法一:

```

#include<stdio. h>
fun(int *p, int n)
{
    int i, j, temp, k;
    for(i=0; i<n- 1; i++)
    {k=i;
        for(j=i+1; j<n; j++)
            if(*(p+k)<*(p+j))
                k=j;
        if(i!=k)
            { temp=*(p+i); *(p+i)=*(p+k); *(p+k)=temp; }
    }
}
int main( )
{
    int a[100], *p, n, i, m, t=0, low=0, hig    h, mid;
    scanf("%d", &n);
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    p=a;

```



```
fun(p, n);
scanf("%d", &m);
high = n;
while(low <= high)
{
    mid = (low+high)/2;
    if(m == a[mid])
    {
        t = 1;
        while(a[mid-1] == a[mid] && mid >= 0)
            mid--;
        break;
    }
    else if(m < a[mid])
        low = mid+1;
    else
        high = mid-1;
}
if(t == 1)
    printf("%d\n", mid+1);
else
    printf("wucishu");
}
```

解法二:

```
#include<stdio.h>
void paixu(int *p, int n)
{
    int i, j, m, t, max;
    for(i=0; i<n-1; i++)
    {
        max = p[i]; m = i;
        for(j=i+1; j<n; j++)
            if(p[j]>max){ max = p[j]; m = j; }
        t = p[i]; p[i] = p[m]; p[m] = t;
    }
}

int chazhao(int x[], int n, int m)
{
    int i, low, high, biao = 0;
```



```
    low = 0; high = n - 1;
    while(high >= low)
    {
        i = (high + low) / 2;
        if(x[i] > m)
            low = i + 1;
        elseif(x[i] < m) high = i - 1;
        else
            { biao = i + 1; break; }
    }
    return biao;
}
int main(int argc, char *argv[])
{
    int x[100], n, m, i, shu;
    scanf("%d", &n);
    for(i = 0; i < n; i++)
        scanf("%d", &x[i]);
    paixu(x, n);
    scanf("%d", &m);
    shu = chazhao(x, n, m);
    if(shu)
        printf("%d", shu);
    else
        printf("wucishu");
    return 0;
}
```

解法三:

```
#include <stdio.h>
int n;
int order(int *a, int x)
{
    int j, i, t, *q;
    q = a;
    for(i = 0; i < n; i++)
    {
        a = q;
```



```
        for(j=0; j<n; j++)
        {
            if(*a<*(a+1))
                { t=*a; *a=*(a+1);          *(a+1)=t; }
            a++;
        }
    }
    a=q;
    for(a; a<a+n; a++)
        if(*a==x) return (a-q);
    return 0;
}
int main(int argc, char *argv[])
{
    int i, x, j, a[100]={ 0 };
    scanf("%d", &n);
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    scanf("%d", &x);
    j=order(a, x);
    if(j==0)
        printf("wucishu!");
    else
        printf("%d\n", j+1);
    return 0;
}
```

解法四:

```
#include<stdio. h>
char a[100]; int n, m, l=0, x, t, i, j;
void sort(char *a, int n)
{
    for(i=0; i<n; i++)
        for(j=i+1; j<n; j++)
            if(a[i]<a[j])
                { t=a[i]; a[i]=a[j]; a[j]=t; }
}
void f(char *a, int n, int x)
```



```
{
    m = n/2;
    if(a[m]==x)
    { printf("%d\n", m+1); }
    else if(a[m]<x)
        for(i=0; i<m; i++)
            if(a[i]==x){ printf("%d\n", i+1); break; }
    else if(a[m]>x)
        for(i=m; i<n; i++)
            if(a[i]==x){ printf("%d\n", i+1); break; }
    for(i=0; i<n; i++)
        if(a[i]!=x){ l++; }
    if(l==n){ printf("wucishu!"); printf("\n"); }
}
int main( )
{
    scanf("%d", &n);
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    scanf("%d", &x);
    sort(a, n);
    f(a, n, x);
    return 0;
}
```

第 7 章 结构体

一、选择题

1. C 2. D 3. A 4. C 5. A 6. C 7. B 8. A 9. B 10. A
11. B 12. D 13. D 14. B

二、填空题

- 共用体
- struct
- 14 p. no = 1001
- strcpy(s. no, "1001") strcpy(p->name, "张三")
- L->next->data L->next->next



6. typedef

三、程序分析题

1. 因为函数将值改变了,所以运行结果是:1002, ChangRong, 1202. 0
2. 因为成员运算符的优先级高, ++p->x 就是++(p->x), 所以运行结果:2, 3
3. 因为函数传递的是指针,修改的值会返回来,所以运行结果是:1001, ChangRong, 1098, 0
4. 函数参数是值传递的方式,赋值结果不会返回来,运行结果是:Qian, p, 95. 0, 92. 0
5. p=a; 那么第一次输出 p->x 就是 20,然后 p=p->y, 则 p=a+1, 所以第二次输出 15, 运行结果为:20, 15,

四、补充程序题

1. p<stu+3 p->num, p->name, p->sex, p->age。第一个空要求填写条件,因为数组大小是 3,根据前面指针输出数组的基本知识,这里应该填 p<stu+3, 第二个空是输出数据值,所以每个数据都用指针使用成员。
2. person[i]. sex return n 第一个空是填写符合要求的条件,所以是 person[i]. sex, 第二个空因为题目要求返回符合条件个数,所以是 return n。
3. p!=NULL p=p->next. 第一个空是循环条件,根据链表结束标志,所以应该填写 p!=NULL, 或 p, 第二个空是将指针移动到下一个数据,填写 p=p->next。

五、编写程序题

1. struct worker

```
{    long num;
    char name[20];
    char sex;
    char Idnumber[12];
    float wages;
    char add[50];
} employee;
```

2. #include "stdio. h"

```
void main( )
{    struct worker employee;
    printf("输入一个员工信息:职工号、姓名、性别、身份证号、工资、地址\n");
    scanf("%ld%s", &employee. num, employee. name);
    scanf("%*c%c %s", &employee. sex, employee. Idnumber);
    scanf("%f%s", &employee. wages, employee. add);
    printf("员工信息:职工号、姓名、性别、身份证号、工资、地址\n");
    printf ("%ld,%s", employee. num, employee. name);
    printf ("%c,%s", employee. sex, employee. Idnumber);
```



```
printf ("%f,%s\n", employee. wages, employee. add);
}

3. #include "stdio. h"
void main( )
{   struct  worker employee[6];
    int i, m;
    float max, ave =0;
    printf("输入 6 个员工信息:职工号、姓名、性别、身份证号、工资、地址\n");
    for(i=0; i<6; i++)
    {scanf("%ld%s", &employee[i]. num, employee[i]. name);
      scanf("%*c%c %s", &employee[i]. sex, employee[i]. Idnumber);
      scanf("%f%s", &employee[i]. wages, employee[i]. add);
      ave = ave+employee[i]. wages;
    }
    ave = ave/6;
    m =0;
    for(i=1; i<6; i++)
    if(employee[i]. wages> employee[m]. wages) m =i;
        printf ("总平均工资是:%. 2f \n", ave);
    printf("工资最高的员工信息:职工号、姓名、性别、身份证号、工资、地址\n");
    printf ("%ld,%s", employee[m]. num, employee[m]. name);
    printf ("%c、 %s", employee[m]. sex, employee[m]. Idnumber);
    printf ("%f,%s\n", employee[m]. wages, employee[m]. add);
}

4. #include "stdio. h"
struct date
{   int year; int month; int day; };
    struct  worker
    { long num;
      char name[20];
      char sex;
      char Idnumber[12];
      float wages;
      char add[50];
      struct date birthday;
    };
void main( )
```



```
{    struct  worker employee;
    printf("输入一个员工信息:职工号、姓名、性别、身份证号、工资、地址、生日\n");
    scanf("%ld%s", &employee. num, employee. name);
    scanf("%*c%c %s", &employee. sex, employee. Idnumber);
    scanf("%f%s", &employee. wages, employee. add);
    scanf("%d%d%d", &employee. birthday. year, &employee. birthday. month
    &employee. birthday. day);
    printf("员工信息:职工号、姓名、性别、身份证号、工资、地址、生日\n");
    printf ("%ld,%s", employee. num, employee. name);
    printf ("%c,%s", employee. sex, employee. Idnumber);
    printf ("%f,%s\n", employee. wages, employee. add);
    printf("%d- %d- %d\n", employee. birthday. year, employee. birthday. month
    employee. birthday. day);
}

5. #include<stdio. h>
    #include<string. h>
    struct student
    {    int num;
        char name[20];
        float score[3];
    };
    int main( )
    {    int i, j;
        struct student std[10];
        puts("Please enter information of student : ");
        i=0;
        scanf("%s", &std[i]. num);
        while(std[i]. num!= 0&&i<10)
        {    scanf("%s", std[i]. name);
            for(j=0; j<3; ++j)
                scanf("%f", &std[i]. score[j]);
            i++;
            scanf("%s", &std[i]. num);
        }
        for(i=0; i<10; ++i)
        {    printf("Num=%d Name=%-6s ", std[i]. num, std[i]. name);
```




```
printf("Score1=%0.2f Score2=%0.2f Score3=%0.2f\n", std[i].score[0],  
std[i].score[1], std[i].score[2]);  
}  
return 0;  
}
```

第8章 文 件

一、选择题

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

二、填空题

1. 文本 二进制
2. 只读 只写 读写 追加
3. fopen 读写函数 fcolse
4. 空地址(NULL) 0
5. 写入失败标志 EOF
6. n-1
7. 5 6
8. rewind, fseek 两个
9. 1 0

三、程序分析题

1. dear friend!, 文件以"W"方式打开时, 写入会从头开始写, 原来的内容没有了。
2. Chinang, rewind(pf)将文件指针指向开始位置, 再写入时覆盖前面的内容, 但未覆盖部分仍然存在。
3. 123。循环写入文件时数据间没有分隔符, 所以文件内容是 123, 因此读入数据 n 的值是 123。
4. welcome mike! (回车), 程序将文件的内容一个一个字符地读取并输出。
5. welcomeBeijingChina, 注意文件是以追加的方式打开的, 所以内容都是追加在后面。

四、补充程序题

1. FILE *fp NULL
2. FILE *fp fp=fopen("file.txt", "w+")==NULL fscanf(fp, "%d %d", &x1, &y1) z=x1+y1
3. filename fp=fopen(filename, "rb"); fp==NULL fp, 0 L, 2 ftell(fp)

五、编写程序题

1. 要注意每输入一个字符串后加上回车符。参考代码如下。



```
#include "stdio. h"
int main( )
{
    FILE *fp;
    char s[80];
    int n, i;
    printf("请输入要输入的字符串个数:");
    scanf("%d", &n);
    if((fp = fopen("string. txt", "w")) == NULL)
    {
        printf("文件打开失败! \n");
        return 0;
    }
    getchar( );
    for(i=0; i<n; i++)
    {
        printf("请输入%d 个字符串:", i+1);
        gets(s);
        fputs(s, fp);
        fputs("\n", fp);
    }
    fclose(fp);
}
```

2. 注意单词的开始和结束,这里使用一个标记 flag, flag=0 是代表读取的字母是一个新单词的开始,flag=1 时读取的字母是在一个单词内,碰到不是字母时,一个单词结束。参考代码如下。

```
#include "stdio. h"
int main( )
{
    FILE *fp;
    char s[80];
    int n=0, i=0, flag=0;
    if((fp = fopen("abc. dat", "r")) == NULL)
    {
        printf("文件打开失败! \n");
        return 0;
    }
}
```



```

    }
    *s = fgetc(fp);
    while(! feof(fp))
    {
        i++;
        *(s+i) = fgetc(fp);
    }
    s[i+1] = ' \0' ;
    puts(s);
    for(i=0; s[i] != ' \0' ; i++)
    {
        if(flag == 0 && ((s[i] >= ' a' && s[i] <= ' z' ) || (s[i] >= ' A' && s[i] <= ' Z' )))
        {
            n++;
            flag = 1;
        }
        else if(flag == 1 && !((s[i] >= ' a' && s[i] <= ' z' ) || (s[i] >= ' A' && s[i] <= ' Z' )))
        {
            flag = 0;
        }
    }
    printf("n = %d", n);
    fclose(fp);
}

```

3. 注意不知道数据个数时,要用 `feof(fp)` 判断是否读到文件末尾,是则结束读取数据,否则读取下一个数据。参考代码如下。

```

int main( )
{
    FILE *fp;
    int zn=0, fn=0, zr=0, x;
    if((fp = fopen("num. txt", "r")) == NULL)
    {
        printf("文件打开失败! \n");
        return 0;
    }
    fscanf(fp, "%d", &x);
}

```



```
while(!feof(fp))
{
    if(x>0)  zn++;
    else if(x==0) zr++;
    else fn++;
    fscanf(fp, "%d", &x);
}
if(x>0)  zn++;
else if(x==0) zr++;
else fn++;
printf("正整数:%d 个,正整数:%d 个,零:%d 个", zn, fn, zr);
fclose(fp);
}
```

4. 要注意读写数据时位置指针在哪里,最好用绝对定位的方式设好位置指针,再进行文件操作。参考代码如下。

```
#include"stdio. h"
struct student{
    char name[10];
    int no;
    float score[2];
}a[5]={
    {"mike", 1, 56, 78},
    {"mary", 2, 66, 68},
    {"rose", 3, 86, 68},
    {"nike", 4, 80, 75},
    {"piter", 5, 86, 88}
};
int main( )
{
    FILE *fp;
    int zn, i;
    struct student b;
    if((fp=fopen("stu1. dat", "w"))==NULL)
    {
        printf("文件打开失败! \n");
        return 0;
    }
}
```



```
    }
    zn = sizeof(struct student);
    fwrite(a, zn, 5, fp);
    fclose(fp);
    if((fp = fopen("stu1. dat", "r+")) == NULL)
    {
        printf("文件打开失败! \n");
        return 0;
    }
    fseek(fp, zn, 0);
    fread(&b, zn, 1, fp);
    fseek(fp, 2*zn, 0);
    fwrite(&b, zn, 1, fp);
    fclose(fp);
}
```

5. 可以先读取全部数据并放入到数组中,然后进行判断,也可以每读取一个判断一个。

参考代码如下。

```
#include "stdio. h"
struct student{
    long int num;
    char name[10];
    int age;
    char sex;
    char speciality[20];
    char addr[40];
};
int main( )
{
    FILE *fp;
    struct student b;
    if((fp = fopen("stu2. dat", "r+")) == NULL)
    {
        printf("文件打开失败! \n");
        return 0;
    }
    fread(&b, zn, 1, fp);
```



```
while(! feof(fp)){  
    if(b. age == 18 || b. age == 19)  
        printf("%d %s %d %c\n", b. num, b. name, b. age, b. sex);  
    fread(&b, zn, 1, fp);  
}  
if(b. age == 18 || b. age == 19)  
    printf("%d %s %d %c\n", b. num, b. name, b. age, b. sex);  
fclose(fp);  
}
```



参考文献

- [1] 戴经国, 庄景明. C 语言程序设计[M]. 成都: 电子科技大学出版社, 2017.
- [2] 谭浩强. C 语言程序设计[M]. 3 版. 北京: 清华大学出版社, 2005.
- [3] 李峰, 谢中科. C 语言程序设计上机指导与习题解答[M]. 上海: 复旦大学出版社, 2015.
- [4] 徐翠霞. C 语言程序设计实训[M]. 北京: 中国水利水电出版社, 2006.
- [5] 冯克鹏, 李涛. C 语言程序设计基础[M]. 成都: 电子科技大学出版社, 2014.
- [6] 张书云. C 语言程序设计习题解答与上机指导[M]. 北京: 中国铁道出版社, 2008.
- [7] 凌云, 吴海燕. C 语言程序设计与实践[M]. 北京: 机械工业出版社, 2011.

附录 C 语言程序常见错误分析

在实际的 C 语言编程过程中，一个 C 程序编写完成后，在编译、链接、运行等环节可能会出现各种各样的错误，出现错误的原因多样，如算法设计错误、语法不熟悉造成的编译错误、数据录入错误、函数调用造成的链接错误等，因此还要对程序进行调试和测试，尽可能地发现这些错误，并进行修改，只有当程序正确无误后才能正常运行，得到正确结果。

一、C 语言程序编制过程中常见错误

C 语言程序编制过程的各个阶段都可能发生错误，根据不同阶段和出错原因可以将程序设计中的错误分成以下五类。

一是语法错误，这是指在程序的编译过程中由编译程序识别或检查出来的错误，称为“语法错误”。主要原因是程序中出现了不符合语法规则的语句格式，各种标识符(常量、变量、类型、宏名等)说明与使用不一致，使用了不正确的分隔符或不存在的标号，对变量的初始化错误，漏写了各种括号或标点，不恰当的循环嵌套等。在编译时，编译程序会发现这些错误，并给出相应的出错信息。

二是链接错误，这是指链接程序在将程序链接成可执行程序时发现的错误，主要出错原因是函数调用错误(写错函数名、参数格式错误、库函数缺少包含文件或包含文件的路径错误)等原因引起的。

三是运行错误，这是指 C 程序通过编译、链接后，在运行过程中出现的错误。主要原因可能是：在表达式中出现遇到了除数为零的错误、求一个负数的平方根、实数比较相等、数据录入格式错误等。

四是逻辑性错误，主要原因是算法设计出错、表达式书写错误(如把 $x > 2 \& \& x < 10$ 写成 $2 < x < 10$)等。编译程序无法查出此类错误，造成运行结果错误。

五是警告性错误，这是指编译程序在编译阶段发现程序中有一些可疑的或含混不清的地方，如源程序中发现了一个定义过但从未使用过的变量，或出错变量类型与数据不一致的情况。这类情况从语法上讲是正确的，因此一般不会停止编译，在大多数情况下不会阻止目标程序与可执行程序的生成、链接和运行。但是对这类错误不应掉以轻心，应仔细检查程序，这往往存在着某种潜在的运行期错误。

二、程序的调试与测试

程序调试的任务是排除程序中的错误，使程序能顺利地运行并得到预期的效果。程序的调试阶段不仅要发现和消除语法上的错误，还要发现和消除逻辑错误和运行错误。除了可以利用编译时提示的“出错信息”来发现和改正语法错误外，还可以通过程序的测试来发现逻辑错误和运行错误。



测试的目的是检查程序有无“漏洞”。对于一个简单的程序,要找出其运行时全部可能执行到的路径,并正确地准备数据并不困难。但是,如果需要测试一个复杂的大程序,要找到全部可能的路径,并准备出所需的测试数据并非易事。例如,有两个非嵌套的 if 语句,每个 if 语句有 2 个分支,它们所形成的路径数目为 $2 \times 2 = 4$; 如果一个程序包含 100 个 if 语句,则可能的路径数目为 $2^{100} = 1.267651 \times 10^{30}$ 。实际上进行测试的只是其中一部分(执行概率最高的部分)。因此,经过测试的程序一般还不能轻易宣布为“没有问题”,而只能说“经过测试的部分无问题”。

三、常见错误分析

1. 漏写地址符或多写地址符。

`scanf("%d%d", x, y);` //漏写地址符,正确为:`scanf("%d%d", &x, &y);`

`printf("%d%d", &x, &y)` //多写地址符,正确:`printf("%d%d", x, y);`

2. 没给格式控制字符串加双引号,或双引号引用位置错误。

如 `scanf(%d%d, &a, &b);` 或 `scanf("%d%d, &a, &b");` 或 `printf("a+b=%d\n", "a+b);`

正确为:`scanf("%d%d", &a, &b); printf("a+b=%d\n", a+b);`

3. 函数 `printf()` 欲输出一个表达式的值,但格式控制字符串中却没有与其对应的格式转换字符。

如:`printf("a+b=\n", a+b);` 正确为:`printf("a+b=%d\n", a+b);`

4. 函数 `printf()` 欲输出一个表达式的值,写了格式转换字符,但是这个格式转换字符对应的表达式却忘记写在 `printf()` 中。

如:`printf("a+b=%d\n");` 正确为:`printf("a+b=%d\n", a+b);`

5. 函数 `scanf()` 或 `printf()` 的格式的类型控制符与要输入/输出的变量类型不一致。

如:`int x;` 函数 `printf` 写成"`printf("x=%f\n", x);`", 正确为:`printf("x=%d\n", x);`

6. 用户从键盘输入的数据格式与函数 `scanf()` 中格式控制字符串要求的格式不一致。

如:`scanf("%d%d", &a, &b);` 错误输入 3, 4 中间加了逗号“,” 正确输入 3 4 ,中间用空格隔开两个数据。

7. 函数 `scanf()` 格式控制字符串中含有“\n”等转义字符,导致数据输入不能以正常方式终止。

如:`scanf("%d%d\n", &a, &b);` 正确为:`scanf("%d%d", &a, &b);`

8. 函数 `scanf()` 输入实型数据时,不能在格式控制字符串中规定精度。

如:`scanf("%6.2f", &a);` 正确为:`scanf("%f", &a);`

9. 混淆“=”和“==”,在 C 语言中,“=”赋值运算符,“==”是关系运算符。

如:`if(a==3) a=b;`

前者是进行比较, a 是否和 3 相等,后者表示如果 a 和 3 相等,把 b 值赋给 a。由于习惯问题,初学者往往会犯这样的错误。

10. 在不该出现分号的地方加了分号。

例如:`if(x>y);`



```
printf("x is larger than y. \n");
```

11. 对于复合语句，忘记加花括号。

例如：`i=1; a=0;`

```
while (i<= 10)
```

```
    a+=i; i++;    //应该写成{ a+=i; i++; }
```

```
    printf("a=%d\n", a);
```

12. 误把数组说明时所定义的元素个数作为最大下标值使用。C 语言规定，引用数组元素时下标从 0 开始，即下标值的下限为 0，而下标的上限值是数组定义时元素个数减 1。

13. 在 switch 语句的各分支中未使用 break 语句。

例如：根据考试成绩的等级打印出百分制数段。

```
switch(grade)
```

```
{ case ' A' :printf("85~100\n");
```

```
  case ' B' :printf("70~84\n");
```

```
  case ' C' :printf("60~69\n");
```

```
  case ' D' :printf("<60\n");
```

```
  default:printf("error\n");
```

由于漏写了 break 语句，case 只起标号的作用，而不起判断作用。因此，当 grade 值为 A 时，printf 函数在执行完第一个语句后接着执行第二、三、四、五个 printf 函数语句。正确写法应在每个分支后再加上“break;”。

14. 混淆 break 语句和 continue 语句的作用。

例如：用键盘输入一批数，输出其中大于 60 的数，当输入 0 时结束循环。

```
do
```

```
{ scanf("%d", &x);
```

```
  if(x<60) break; //错误,应该使用 continue
```

```
  printf("%d\n", x);
```

```
}while(x!=0);
```

题目要求是 x 等于 0 时结束循环，而不是小于 60，使用 break 使循环提前结束了，不符题目要求，正确的是使用 continue 继续输入下一个数。

15. 在使用变量前未定义。

例如：`main( )`

```
{ a=1;
```

```
  b=2;
```

```
  printf("%d\n", a+b);
```

```
}
```

错误：变量 a, b 没有定义，应该加上 `int a, b;`

16. 定义或引用数组的方式不对。



C 语言规定,在对数组进行定义或对数组元素进行引用时必须要用方括号。

例如:以下写法都将造成编译时出错。

```
int a(10); int b[5, 4];
printf("%d\n",
b[1+2, 2]);
```

17. 混淆字符和字符串。

C 语言中的字符常量是由一对单引号括起来的单个字符;而字符串常量是用一对双引号括起来的字符序列。字符常量存放在字符型变量中,而字符串常量只能存放在字符型数组中。例如:假设已说明 num 是字符型变量,则以下赋值语句是非法的。

```
num = "1"; //正确写成 num = ' 1' ;
```

18. 在引用数组元素或指针变量之前没对其赋初值或赋值错误。

例如:

```
int a[6], b;
b = a[5]; //错误 a[5]没有赋值,应先给元素赋值。
int *ptr, i = 1;
*ptr = i; //错误,应先有 ptr = &i;
```

19. 混淆数组名与指针变量。

在 C 语言中,数组名代表数组的首地址,它的值是一个常量,不能被修改。例如,在以下程序段中,用 a++是不合法的。

```
main( )
{ int i, a[10];
  for (i=0; i<10; i++)
    scanf("%d", a++);
  :
}
```

20. 混淆不同类型的指针。

若有以下语句:

```
int *p1, a=1;
float *p2;
p1 = &a;
```

则赋值语句 p2=p1 是非法的。

21. 所调用的函数在调用前未定义。

```
main( )
{ float a=10, b=20, c;
  c = fun(a, b); //函数没有定义
  :
}
```



22. 混淆结构体类型名和结构体变量名。

若定义了以下结构体类型 student:

```
struct student
{ long int num;
  char name[20];
  int age;
  float score;
};
```

则赋值语句: student. num = 199401; 是错误的, student 是类型名, 不是变量名。

四、C 语言错误提示

(一) 语法类错误

Ambiguous operators need parentheses——不明确的运算需要用括号括起

Ambiguous symbol "xxx"——不明确的符号

Argument list syntax error——参数表语法错误

Array bounds missing——丢失数组界限符

Array size toolarge——数组尺寸太大

Bad character in paramenters——参数中有不适当的字符

Bad file name format in include directive——包含命令中文件名格式不正确

Bad ifdef directive synatax——编译预处理 ifdef 有语法错

Bad undef directive syntax——编译预处理 undef 有语法错

Bit field too large——位字段太长

Call of non-function——调用未定义的函数

Call to function with no prototype——调用函数时没有函数的说明

Cannot modify a const object——不允许修改常量对象

Case outside of switch——漏掉了 case 语句

Case syntax error——Case 语法错误

Code has no effect——代码不可述不可能执行到

Compound statement missing { ——分程序漏掉" {"

Conflicting type modifiers——不明确的类型说明符

Constant expression required——要求常量表达式

Constant out of range in comparison——在比较中常量超出范围

Conversion may lose significant digits——转换时会丢失意义的数字

Conversion of near pointer not allowed——不允许转换近指针

Could not find file "xxx"——找不到 xxx 文件

Declaration missing; ——说明缺少";"

Declaration syntax error——说明中出现语法错误



Default outside of switch——Default 出现在 switch 语句之外

Define directive needs an identifier——定义编译预处理需要标识符

Division by zero——用零作除数

Do statement must have while——Do-while 语句中缺少 while 部分

Enum syntax error——枚举类型语法错误

Enumeration constant syntax error——枚举常数语法错误

Error directive : xxx——错误的编译预处理命令

Error writing output file——写输出文件错误

Expression syntax error——表达式语法错误

Extra parameter in call——调用时出现多余错误

File name too long——文件名太长

Function call missing——函数调用缺少右括号

Function definition out of place——函数定义位置错误

Function should return a value——函数必须返回一个值

Goto statement missing label——Goto 语句没有标号

Hexadecimal or octal constant too large——16 进制或 8 进制常数太大

Illegal character "x"——非法字符 x

Illegal initialization——非法的初始化

Illegal octal digit——非法的 8 进制数字

Illegal pointer subtraction——非法的指针相减

Illegal structure operation——非法的结构体操作

Illegal use of floating point——非法的浮点运算

Illegal use of pointer——指针使用非法

Improper use of a typedef symbol——类型定义符号使用不恰当

In-line assembly not allowed——不允许使用行间汇编

Incompatible storage class——存储类别不相容

Incompatible type conversion——不相容的类型转换

Incorrect number format——错误的数据格式

Incorrect use of default——Default 使用不当

Invalid indirection——无效的间接运算

Invalid pointer addition——指针相加无效

Irreducible expression tree——无法执行的表达式运算

Lvalue required——需要逻辑值 0 或非 0 值

Macro argument syntax error——宏参数语法错误

Macro expansion too long——宏的扩展以后太长

Mismatched number of parameters in definition——定义中参数个数不匹配

Misplaced break——此处不应出现 break 语句



- Misplaced continue——此处不应出现 continue 语句
- Misplaced decimal point——此处不应出现小数点
- Misplaced elseif directive——不应编译预处理 elseif
- Misplaced else——此处不应出现 else
- Misplaced else directive——此处不应出现编译预处理 else
- Misplaced endif directive——此处不应出现编译预处理 endif
- Must be addressable——必须是可以编址的
- Must take address of memory location——必须存储定位的地址
- No declaration for function "xxx"——没有函数 xxx 的说明
- No stack——缺少堆栈
- No type information——没有类型信息
- Non-portable pointer assignment——不可移动的指针(地址常数)赋值
- Non-portable pointer comparison——不可移动的指针(地址常数)比较
- Non-portable pointer conversion——不可移动的指针(地址常数)转换
- Not a valid expression format type——不合法的表达式格式
- Not an allowed type——不允许使用的类型
- Numeric constant too large——数值常太大
- Out of memory——内存不够用
- Parameter "xxx" is never used——能数 xxx 没有用到
- Pointer required on left side of——的左边必须是指针
- Possible use of "xxx" before definition——在定义之前就使用了 xxx(警告)
- Possibly incorrect assignment——赋值可能不正确
- Redeclaration of "xxx"——重复定义了 xxx
- Redefinition of "xxx" is not identical——xxx 的两次定义不一致
- Register allocation failure——寄存器定址失败
- Repeat count needs an lvalue——重复计数需要逻辑值
- Size of structure or array not known——结构体或数组大小不确定
- Statement missing; ——语句后缺少";"
- Structure or union syntax error——结构体或联合体语法错误
- Structure size too large——结构体尺寸太大
- Sub scripting missing]——下标缺少右方括号
- Superfluous & with function or array——函数或数组中有多余的"&"
- Suspicious pointer conversion——可疑的指针转换
- Symbol limit exceeded——符号超限
- Too few parameters in call——函数调用时的实参少于函数的参数
- Too many default cases——Default 太多(switch 语句中一个)
- Too many error or warning messages——错误或警告信息太多



Too many type in declaration——说明中类型太多

Too much auto memory in function——函数用到的局部存储太多

Too much global data defined in file——文件中全局数据太多

Two consecutive dots——两个连续的句点

Type mismatch in parameter xxx——参数 xxx 类型不匹配

Type mismatch in redeclaration of "xxx"——xxx 重定义的类型不匹配

Unable to create output file "xxx"——无法建立输出文件 xxx

Unable to open include file "xxx"——无法打开被包含的文件 xxx

Unable to open input file "xxx"——无法打开输入文件 xxx

Undefined label "xxx"——没有定义的标号 xxx

Undefined structure "xxx"——没有定义的结构 xxx

Undefined symbol "xxx"——没有定义的符号 xxx

Unexpected end of file in comment started on line xxx: 从 xxx 行开始的注解尚未结束文件不能结束

Unexpected end of file in conditional started on line xxx: 从 xxx 开始的条件语句尚未结束文件不能结束

Unknown assemble instruction——未知的汇编结构

Unknown option——未知的操作

Unknown preprocessor directive: "xxx"——不认识的预处理命令 xxx

Unreachable code——无路可达的代码

Unterminated string or character constant——字符串缺少引号

User break——用户强行中断了程序

Void functions may not return a value——Void 类型的函数不应有返回值

Wrong number of arguments——调用函数的参数数目出错

"xxx" not an argument——xxx 不是参数

"xxx" not part of structure——xxx 不是结构体的一部分

xxx statement missing (——xxx 语句缺少左括号

xxx statement missing)——xxx 语句缺少右括号

xxx statement missing; ——xxx 缺少分号

"xxx" declared but never used——说明了 xxx 但没有使用

"xxx" is assigned a value which is never used——给 xxx 赋了值但未用过

Zero length structure——结构体的长度为零

(二)警告类错误

"xxx" declare but never used 变量 xxx 已定义但从未用过。

"xxx" is assigned a value which is never used 变量 xxx 已赋值但从未用过。

Code has no effect 程序中含有没有实际作用的代码。

Non-portable pointer conversion 不适当的指针转换,可能是在应该使用指针的地方用

了一个非 0 的数值。

Possible use of "xxx" before definition 表达式中使用了未赋值的变量

Redeclaration of "main" 一个程序文件中主函数 main 不止一个。

Suspicious pointer conversion 可疑的指针转换。通常是使用了基本类型不匹配的指针。

Unreachable code 程序含有不能执行的代码。

Compound statement missing } in function main 程序结尾缺少括号}。

} "expected; "(" "expected 等复合语句或数组初始化的结尾缺少")"; "("。

Case outside of switch case 不属于 switch 结构, 多由于 switch 结构中的花括号不配对所致。

Case statement missing ':' switch 结构中的某个 case 之后缺少冒号。

Constant expression required 定义数组时指定的数组长度不是常量表达式。

Declaration syntax error 结构体或联合类型的定义后缺少分号。

Declaration was expected 缺少说明, 通常是因为缺少分界符如逗号、分号、右圆括号等所引起的。

Default outside switch Default 部分放到了 switch 结构之外, 一般是因为花括号不匹配而引起的。

do statement must have while do 语句中缺少相应的 while 部分。

Expression syntax 表达式语法错, 如表达式中含有两个连续的运算符。

Extra parameter in call 'fun' 调用函数 fun 时给出了多余的实参。

Function should return a value 函数应该返回一个值, 否则与定义时的说明类型不匹配。

Illegal use of pointer 指针被非法引用, 一般是使用了非法的指针运算。

Invalid pointer addition 指针相加非法。一个指针(地址)可以和一个整数相加, 但两个指针不能相加。

Lvalue required 赋值运算的左边是不能寻址的表达式。

Misplaced else 程序遇到了没有配对的 else。

No matching 表达式中的括号不配对。

Pointer required on left side of _ > 在"_ >"运算的左边只能允许一个指针而不能是一个一般的结构体变量或联合类型的变量。

Statement missing; 程序遇到了后面没有分号的语句。

Too few parameters in call 调用某个函数时实参数目不够。

Unable to open include file 'xxxxxxx. xxx' 头文件找不到。

Unexpected '}' or: '{' 在不希望的地方使用了'}'或:'{'。

Undefined symbol 'x' in function fun 函数 fun 中的变量 x 没有定义。

(三)连接中的常见错误

主要错误类似于“undefined symbol _ print in modula xxx”(print 没有定义), 通常是函数名书写错误。