

目 录

第 1 章 程序设计与 C 语言.....	1
1.2 实验 1: 熟悉 C 语言集成开发环境.....	1
第 2 章 C 语言程序设计基础.....	5
2.2 实验 2: 数据类型、C 语言运算符与表达式.....	5
第 3 章 顺序结构程序设计.....	11
3.2 实验 3: 顺序结构程序设计.....	11
第 4 章 选择结构程序设计.....	17
4.2 实验 4: 选择结构程序设计 (1)	17
4.3 实验 5: 选择结构程序设计 (2)	21
第 5 章 循环结构程序设计.....	27
5.2 实验 6: 循环结构程序设计(1)	27
5.3 实验 7: 循环结构程序设计(2)	32
第 6 章 同一类型多个元素的集合-数组.....	39
6.2 实验 8: 数组程序设计(1)	39
6.3 实验 9: 数组程序设计(2)	44
第 7 章 模块化程序设计-函数.....	52
7.2 实验 10: 函数程序设计 (1)	52
7.3 实验 11: 函数程序设计 (2)	58
第 8 章 C 语言的精华-指针	64
8.2 实验 12: 指针程序设计 (1)	64
8.3 实验 13: 指针程序设计 (2)	70
第 9 章 用户自己建立数据类型.....	75
9.2 实验 14: 结构体程序设计.....	75
第 10 章 编译预处理.....	83
10.2 实验 15: 编译预处理.....	83
第 11 章 文件.....	88
11.2 实验 16: 文件.....	88
第 12 章 位运算.....	94
12.2 实验 17: 位运算.....	94

第 1 章 程序设计与 C 语言

1.2 实验 1：熟悉 C 语言集成开发环境

1.2.1 实验目的

1. 熟悉 C 语言集成开发环境的使用方法。
2. 熟悉 C 语言程序从编辑、编译、链接到最后运行得到结果的过程及各过程的作用。
3. 了解 C 语言程序的结构特征与书写规则，能够编写简单的 C 程序。
4. 初步理解程序调试的思想，能找出并改正 C 程序中的语法错误。

1.2.2 实验内容

1. 程序示例

【程序 1.1】上机运行程序，比较 C99 标准和 C89 标准的 C 程序，用不同的开发环境验证运行结果与自己分析的结果是否相符，并记录运行结果。

(1) C89 标准：建议把 main 函数返回指定为 void 型，并返回 void，可省略。

```
#include <stdio.h>

void main ( )
{   printf ("Hello,This is  my first c programme!\n");   }
```

(2) C99 标准：建议把 main 函数返回指定为 int 型，并通过 return 语句返回 0。

```
#include <stdio.h>

int main ( )
{   printf ("Hello,This is  my first c programme!\n");
    return 0;
}
```

2. 程序阅读

【程序 1.2】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>

int main()
{   float x,y,z;
    x=5.6; y=9.9;
    if (x > y) z = y;
    else z = x;
    printf("最小值=%f\n",z);
    return 0;
}
```

3. 程序改错

【程序 1.3】程序中有 4 处错误，上机调试程序，修改有错误的语句行，并输出正确的

运行结果。

```
第 1 句: #include<stdio.h>
第 2 句: int  main()
第 3 句: {  inta,b,sum;
第 4 句:    a=789;
第 5 句:    b=456;
第 6 句:    sum=a+b
第 7 句:    printf(“sum is %d\n,sum”);
第 8 句:    return  0;
第 9 句: }
```

4. 程序完善

【程序 1.4】上机运行程序，实现求 2 个整数中的较大值，补充完善程序内容，并验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
int main( )
{
    int x,y,z;           //定义 3 个整型 (int) 变量
    x=88 y=90;           //将 88 赋值给变量 a, 将 90 赋值给变量 b
    if (x>y)  ①           //比较变量 x 和 y, 变量 z 为变量 x 和 y 中的最大值
    else      ②
    printf(“最大值=%d\n”, ③); //输出最大值=变量 z
    return 0;
}
```

5. 程序设计

【程序 1.5】编写程序，实现输出以下三行信息。

```
*****
我要成为一个优秀的 C 程序员!
*****
```

【程序 1.6】编写程序，实现分两行输出自己的姓名及联系电话，如李明。

姓名：王芳
电话：15312345678

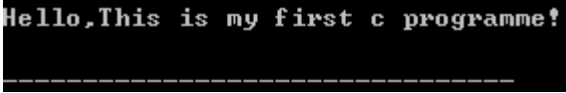
1.2.3 实验指导

1. 程序示例

【程序 1.1】上机运行程序，比较 C99 标准和 C89 标准的 C 程序，用不同的开发环境验证运行结果与自己分析的结果是否相符，并记录运行结果。

①VC++6.0 集成开发环境:用 C99 标准编写的程序功能与 C89 标准相同，验证运行正确，运行结果相同。

②DEV-C++ 集成开发环境：用 C99 标准编写的程序，验证运行正确，程序运行结果如图 1-1 所示。但用 C89 标准编写的程序，则会报错，因为 DEV-C++不可以在 main 前加 void。



```
Hello, This is my first c programme!
-----
```

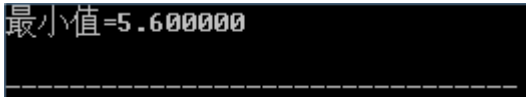
图 1-1 程序 1.1 运行结果

2. 程序阅读

【程序 1.2】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
int main()
{
    float x,y,z;        //定义 3 个实型（float）变量
    x=5.6; y=9.9;        //将 5.6 赋值给变量 x，将 9.9 赋值给变量 y
    if (x > y) z = y;     //比较变量 x 和 y，变量 z 为变量 x 和 y 中的最小值
    else z = x;
    printf("最小值=%f\n", z); //输出最小值=变量 z
    return 0;
}
```

程序运行结果如图 1-2 所示。



```
最小值=5.600000
-----
```

图 1-2 程序 1.2 运行结果

3. 程序改错

【程序 1.3】程序中有 4 处错误，上机调试程序，修改有错误的语句行，并输出正确的运行结果。

```
#include<stdio.h>
int main()
{
    int a,b,sum; //定义 3 个整型（int）变量
    a=789;        //将 789 赋值给变量 a
    b=456;        //将 456 赋值给变量 b
    sum=a+b;      //计算变量 a 与变量 b 的和，赋给变量 sum
    printf("sum is %d\n", sum); //输出变量 sum 的值
    return 0;
}
```

程序运行结果如图 1-3 所示。



```
sum is 1245
-----
```

图 1-3 程序 1.3 的运行结果

4. 程序完善

【程序 1.4】上机运行程序，实现求 2 个整数中的较大值，补充完善程序内容，并验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
int main( )
{
    int x,y,z;        //定义 3 个整型（int）变量
    x=88 ;y=90;        //将 88 赋值给变量 a，将 90 赋值给变量 b
```

```

    if (x>y)    z=x;        //比较变量 x 和 y，变量 z 为变量 x 和 y 中的最大值
    else      z=y;
    printf("最大值=%d\n",  z );    //输出最大值=变量 z
    return 0;
}

```

程序运行结果如图 1-4 所示。



图 1-4 程序 1.4 运行结果

5. 程序设计

【程序 1.5】编写程序，实现输出以下三行信息。

```

*****
我要成为一个优秀的 C 程序员!
*****

#include<stdio.h>
int main()
{ printf("*****\n");
  printf("我要成为一个优秀的 C 程序员!\n");
  printf("*****\n");
  return 0;
}

```

程序运行结果如图 1-5 所示。



图 1-5 程序 1.5 运行结果

【程序 1.6】编写程序，实现分两行输出自己的姓名及联系电话，如李明。

姓名：李明

电话：15388881234

```

#include<stdio.h>
int main( )
{ printf("姓名：王芳\n");
  printf("电话：15312345678\n");
  return 0;
}

```

程序运行结果如图 1-6 所示。

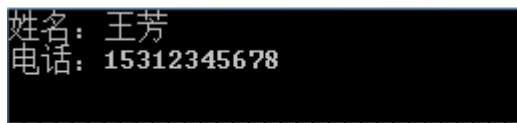


图 1-6 程序 1.6 运行结果

第 2 章 C 语言程序设计基础

2.2 实验 2：数据类型、C 语言运算符与表达式

2.2.1 实验目的

1. 掌握 C 语言基本数据类型的概念。
2. 掌握 C 语言运算符和表达式的使用。
3. 能够将各种数学表达式转换成 C 语言表达式。
4. 理解 C 语言的自动类型转换和强制类型转换和赋值的概念。

2.2.2 实验内容

1. 程序示例

【程序 2.1】上机运行程序，练习整型、实型数据的使用，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```
#include<stdio.h>
int main( )
{
    int a=10,b=012,c=0xa;
    float f1=5.4321f;
    double f2=5432.1;
    printf("a=%d,b=%d,c=%d\n",a,b,c);
    printf("a=%o,b=%o,c=%o\n",a,b,c);
    printf("a=%x,b=%x,c=%x\n",a,b,c);
    printf("f1=%f,f2=%f\n",f1,f2);
    printf("f1=%e,f2=%e\n",f1,f2);
    return 0;
}
```

【程序 2.2】上机运行程序，练习整型、字符型数据的使用，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```
#include <stdio.h>
int main ( )
{
    char c1,c2;
    c1='A';
    c2=c1+32;
    printf("c1=%c,c1=%d\n",c1,c1);
    printf("c2=%c,c2=%d\n",c2,c2);
    return 0;
}
```

2. 程序阅读

【程序 2.3】在阅读并理解程序思路的基础上，练习强制类型转换运算符的使用，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include <stdio.h>
int main( )
{ int d;
  d=(int)((double)(3/2)+0.5+(int)1.99*2);
  printf("d=%d\n",d);
  return 0;
}
```

【程序 2.4】在阅读并理解程序思路的基础上，练习自增、自减运算符的使用，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include <stdio.h>
int main( )
{ int x=012, y, z;
  printf("x=%d\n", x);
  y=++x+3;
  printf("x=%d, y=%d\n", x, y);
  z=x++;
  printf("x=%d, z=%d\n", x, z);
  return 0;
}
```

3. 程序改错

【程序 2.5】程序中有 4 处错误，上机调试程序，修改有错误的语句行，并输出正确的运行结果。

```
第 1 句: #include<stdio.h>
第 2 句: int main()
第 3 句: { float x=2, y ;
第 4 句: y=3;
第 5 句: z=2x +y;
第 6 句: printf("%d\n", y);
第 7 句: printf("%f\n, z");
第 8 句: return 0;}
```

4. 程序完善

【程序 2.6】上机运行程序，实现计算两个整数的商数和余数并输出结果。根据注释，填写相应语句，补充完善程序内容，并验证运行结果与自己分析的结果是否相符，记录下运行结果。

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

```

int main( )
{
    _____ ①           //定义两个 int 型变量
    a=15,b=5;           //为两个整型变量赋值
    _____ ②           //计算并输出两个整数的商数
    _____ ③           //计算并输出两个整数的余数
    return 0;
}

```

5. 程序设计

【程序 2.7】编写一个程序，实现输入任意十进制整型常量，转化成八进制和十六进制输出。

【程序 2.8】编写一个程序，实现从键盘输入两个整数，分别输出这两个数的和、差、积、商、余数。

2.2.3 实验指导

1. 程序示例

【程序 2.1】上机运行程序，练习整型、实型数据的使用，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```

a=10,b=10,c=10
a=12,b=12,c=12
a=a,b=a,c=a
f1=5.432100,f2=5432.100000
f1=5.432100e+000,f2=5.432100e+003

```

图 2-1 例题 2.1 运行结果

【程序 2.2】上机运行程序，练习整型、字符型数据的使用，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```

c1=A,c1=65
c2=a,c2=97

```

图 2-2 例题 2.2 运行结果

2. 程序阅读

【程序 2.3】在阅读并理解程序思路的基础上，练习强制类型转换运算符的使用，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

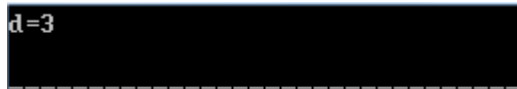
```

#include <stdio.h>
int main( )
{
    int d; //定义 (int) 整型变量 d
    d=(int)((double)(3/2)+0.5+(int)1.99*2);
    // int((double)(3/2)+0.5+(int)1.99*2)=int(1.0+0.5+2)=int(3.5)=3。
    printf("d=%d\n",d); //输出强制类型转换的整型变量 d 的值
    return 0;
}

```

```
}
```

程序运行结果如图 2-3 所示。



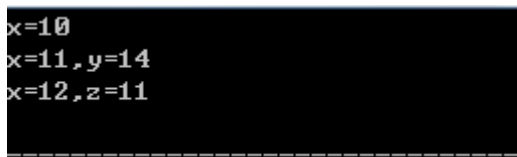
```
d=3
```

图 2-3 例题 2.3 运行结果

【程序 2.4】在阅读并理解程序思路的基础上，练习自增、自减运算符的使用，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include <stdio.h>
int main( )
{ int x=012,y,z; //定义整型变量 x、y、z, 其中 012 是一个八进制的常数
  printf("x=%d\n",x); //以%d 格式（十进制）输出（int）整型变量 x
  y=++x+3;           //先 x+1=11, 后 y=x+3=11+3=14
  printf("x=%d,y=%d\n",x,y); //以%d 格式（十进制）输出的结果是 x=11,y=14
  z=x++;             //先 x=11, z=x=11, 后 x=x+1=11+1=12
  printf("x=%d,z=%d\n",x,z); //以%d 格式（十进制）输出的结果是 x=12,z=11
  return 0;
}
```

程序运行结果如图 2-4 所示。



```
x=10
x=11,y=14
x=12,z=11
```

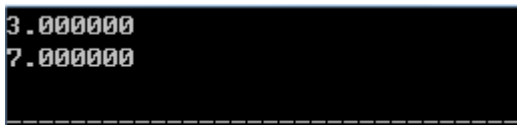
2-4 例题 2.4 运行结果

3. 程序改错

【程序改错 2.5】程序中有 4 处错误，上机调试程序，修改有错误的语句行，并输出正确的运行结果。

```
#include<stdio.h>
int main()
{ float x=2,y,z ;
  y=3;
  z=2*x +y;
  printf("%f\n",y);
  printf("%f\n",z);
  return 0;
}
```

程序运行结果如图 2-5 所示。



```
3.000000
7.000000
```

图 2-5 程序 2.5 运行结果

4. 程序完善

【程序 2.6】上机运行程序，实现计算两个整数的商数和余数并输出结果。根据注释，填写相应语句，补充完善程序内容，并验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
int main( )
{   int a,b;           //定义两个 int 整型变量
    a=15,b=5;          //为两个整型变量赋值
    printf("两个整数的商数=%d\n",a/b);    //计算并输出两个整数的商数
    printf("两个整数的余数=%d\n",a%b);    //计算并输出两个整数的余数
    return 0;
}
```

程序运行结果如图 2-6 所示。

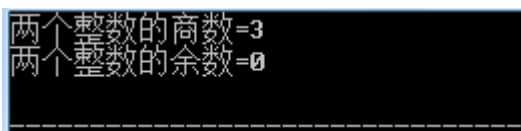


图 2-6 程序 2.6 运行结果

5. 程序设计

【程序 2.7】编写一个程序，实现输入任意十进制整型常量，转化成八进制和十六进制输出。

```
#include<stdio.h>
int main( )
{   int a;              //定义 int 整型变量 a
    scanf("a=%d",&a);    //以%d 格式（十进制）输入整型变量 a
    printf("十进制=%d, 八进制=%o, 十六进制=%x\n", a, a, a);
    //以%d（十进制），以%o（八进制），以%x（十六进制）格式输出整型变量 a
    return 0;
}
```

程序运行结果如图 2-7 所示。



图 2-7 例题 2.7 运行结果

【程序 2.8】编写一个程序，实现从键盘输入两个整数，分别输出这两个数的和、差、积、商、余数。

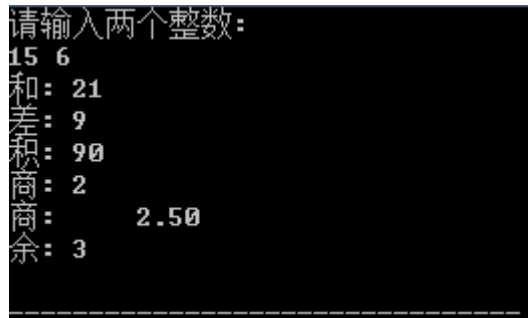
```
#include<stdio.h>
int main( )
{   int num1, num2;    //定义 int 型变量 num1 和 num2
    printf("请输入两个整数: \n");
```

```

scanf ("%d%d", &num1, &num2);    //从键盘输入两个整数
printf("和: %d\n", num1+num2);    //输出了两个数的和
printf("差: %d\n", num1-num2);    //输出了两个数的差
printf("积: %d\n", num1*num2);    //输出了两个数的积
printf("商: %d\n", num1/num2);    //输出了两个数的商，注意除数不能为 0
printf("商: %8.2f\n", (float)num1 / (float)num2);
//输出了余数，注意强制类型转换
printf("余: %d\n", num1%num2) ;
return 0;
}

```

程序运行结果如图 2-8 所示。



```

请输入两个整数:
15 6
和: 21
差: 9
积: 90
商: 2
商:      2.50
余: 3

```

图 2-8 例题 2.8 运行结果

第 3 章 顺序结构程序设计

3.2 实验 3：顺序结构程序设计

3.2.1 实验目的

1. 掌握顺序结构程序设计的基本方法。
2. 掌握不同类型变量的定义及使用。
3. 掌握 C 语言中各种数据的输入/输出方法，能正确使用各种格式符。
4. 能够编写、调试和修改简单的顺序结构程序。

3.2.2 实验内容

1. 程序示例

【程序 3.1】上机运行下面的程序，验证运行结果与自己预习分析结果是否相符，并记录运行结果。

```
#include <stdio.h>
int main( )
{ putchar(getchar()+1);
  putchar(getchar()+1);
  putchar(getchar()+1);
  putchar('\n');
  return 0;
}
```

2. 程序阅读

【程序 3.2】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

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

3. 程序改错

【程序 3.3】程序中有 5 处错误，上机调试程序，修改有错误的语句行，并输出正确的运行结果。

```
# include<stdio.c>
int mian( )
{ char c1, c2;
  c1=97;
  c2=98;
  print("%c %c\n", c1-32, c2-32);
  printf( '%d %d\n' , c1;c2)
  return 0;
}
```

4. 程序完善

【程序 3.4】阅读程序，用下面程序的 scanf 函数输入数据，在键盘上将如何输入，才能输出 a=1, b=2, x=3.5, y=67.89, c1=' A' , c2=' b' , 并将程序补充完整。并验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
# include<stdio.h>
int main( )
{ int a,b;
  float x,y;
  char c1,c2;
  scanf("a=%d, b=%d",&a,&b);
  scanf("%f%e",&x,&y);
  scanf("%c%c",&c1,&c2);
  _____ //输出结果
}
```

5. 程序设计

【程序设计 3.5】编写一个程序，实现输入矩形的两个边长，求其面积和周长。

【程序设计 3.6】编写一个程序，实现输入一个华氏温度 f，要求输出摄氏温度 c。

二者间的转换公式为 $c = \frac{5}{9}(f - 32)$ ，输出取 2 位小数。

【程序设计 3.7】编写一个程序，实现输入任意的 3 个数，求它们的和及平均值。

【程序设计 3.8】编写一个程序，实现输入任意大写字母，转化为小写输出，要求分别用字符输入输出函数和格式输入输出函数完成。（选做）

3.2.3 实验指导

1. 程序示例

【程序验证 3.1】上机运行程序，验证运行结果与自己预习结果是否相符，并记录运行结果。



图 3-1 程序 3.1 运行结果

2. 程序阅读

【程序 3.2】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
int main( )
{ float a, b, c;
  scanf("a=%f,b=%f",&a, &b); //要求输入交换前变量 a,b 的值
  c=a;                        //交换 a,b 的值第 1 步, 将 a 的值赋给 c
  a=b;                        //交换 a,b 的值第 2 步, 将 b 的值赋给 a
  b=c;                        //交换 a,b 的值第 3 步, 将 c 的值赋给 b
  printf("a=%f,b=%f\n", a, b); //输出交换后变量 a, b 的值;
  return 0;
}
```

程序运行结果如图 3-2 所示。

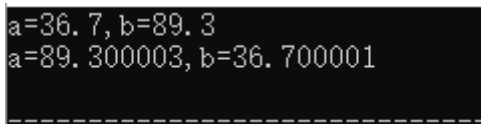


图 3-2 程序 3.2 运行结果

3. 程序改错

【程序 3.3】程序中有 5 处错误，上机调试程序，修改有错误的语句行，并输出正确的运行结果。

```
# include<stdio.h>
int main( )
{ char c1, c2; //定义字符变量 c1,c2
  c1=97;
  c2=98;
  printf("%c %c\n", c1-32, c2-32); //计算 c1-32=65 (字母 A), c2-32=66 (字母 B)
  printf( "%d %d\n" , c1,c2); //输出字符变量 c1, c2
  return 0;
}
```

改完后程序运行结果如图 3-3 所示。

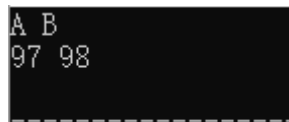


图 3-3 程序 3.3 运行结果

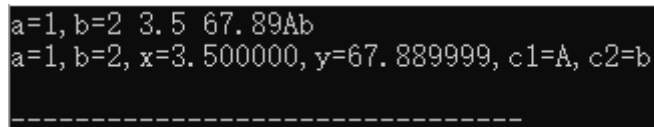
4. 程序完善

【程序 3.4】阅读程序，用下面程序的 scanf 函数输入数据，在键盘上将如何输入，才

能输出 a=1, b=2, x=3.5, y=67.89, c1=' A' , c2=' b' , 并将程序补充完整。并验证运行结果与自己分析的结果是否相符, 记录下运行结果。

```
# include<stdio.h>
int main( )
{ int a,b;//定义整型变量 a,b
  float x,y;//定义实型变量 x,y
  char c1,c2; //定义字符变量 c1,c2
  scanf("a=%d,b=%d",&a,&b); //键盘输入 a,b 的值
  scanf("%f%f",&x,&y); //键盘输入 x,y 的值
  scanf("%c%c",&c1,&c2); // 键盘输入 c1,c2 的值
  printf("a=%d,b=%d,x=%f,y=%f,c1=%c,c2=%c\n",a,b,x,y,c1,c2); //输出结果
}
```

程序运行结果如图 3-4 所示。



```
a=1,b=2 3.5 67.89Ab
a=1,b=2,x=3.500000,y=67.889999,c1=A,c2=b
-----
```

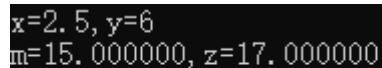
图 3-4 程序 3.4 运行结果

5. 程序设计

【程序设计 3.5】编写一个程序, 实现输入矩形的两个边长, 求其面积和周长。

```
# include<stdio.h>
int main( )
{ float x,y,m,z;
  scanf("x=%f,y=%f",&x,&y);
  m=x*y;
  z=2*x+2*y;
  printf("m=%f,z=%f\n",m,z);
  return 0;
}
```

程序运行结果如图 3-5 所示。



```
x=2.5,y=6
m=15.000000,z=17.000000
-----
```

图 3-5 程序 3.5 运行结果

【程序 3.6】编写一个程序, 实现输入一个华氏温度 f, 要求输出摄氏温度 c。

二者间的转换公式为 $c = \frac{5}{9}(f - 32)$, 输出取 2 位小数。

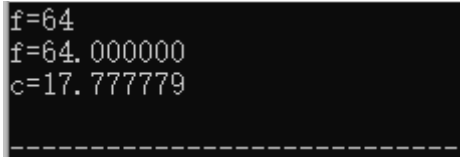
```
#include <stdio.h>
int main ( )
{ float f,c; // 定义 f 和 c 为单精度浮点型变量
  scanf("f=%f",&f); // 输入 f 的值
  c=(5.0/9)*(f-32); // 利用公式计算 c 的值
  printf("f=%f\nc=%f\n",f,c); // 输出 c 的值
}
```

```

    return 0;
}

```

程序运行结果如图 3-6 所示。



```

f=64
f=64.000000
c=17.777779

```

图 3-6 程序 3.6 运行结果

【程序设计 3.7】编写一个程序，实现输入任意的 3 个数，求它们的和及平均值。

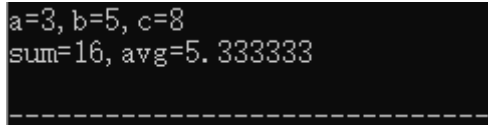
```

#include <stdio.h>

int main ( )
{
    int a, b, c, sum;
    float avg;
    scanf("a=%d, b=%d, c=%d", &a, &b, &c);
    sum=a+b+c;
    avg=sum/3.0;
    printf("sum=%d, avg=%f\n", sum, avg);
    return 0;
}

```

程序运行结果如图 3-7 所示。



```

a=3, b=5, c=8
sum=16, avg=5.333333

```

图 3-7 程序 3.7 运行结果

【程序 3.8】编写一个程序，实现输入任意大写字母，转化为小写输出，要求分别用字符输入输出函数和格式输入输出函数完成。

方法一：用格式输入输出函数完成

```

#include <stdio.h>

int main ( )
{
    char x;
    scanf("%c", &x);
    printf("%c\n", x+32);
    return 0;
}

```

方法二：用字符输入输出函数完成

```

#include <stdio.h>

int main ( )
{
    char x;
    x=getchar();
    putchar(x+32);
    putchar('\n');
    return 0;
}

```

程序运行结果如图 3-8 所示。



图 3-8 程序 3.8 运行结果

第 4 章 选择结构程序设计

4.2 实验 4：选择结构程序设计（1）

4.2.1 实验目的

1. 正确使用关系表达式和逻辑表达式，熟练使用输入和输出语句。
2. 熟练掌握使用 if 语句，能够正确使用 if 语句和 if 语句的嵌套来编写选择结构程序，并正确理解程序的执行流程。
3. 能够熟练运用选择结构进行综合编程。

4.2.2 实验内容

1. 程序示例

【程序 4.1】编程从键盘输入 3 个整数 a、b、c，输出其中最大的数。上机运行程序，验证运行结果与自己预习分析结果是否相符，并记录运行结果。

```
#include<stdio.h>
int main( )
{ int a,b,c,max;
  printf("请输入 3 个整数:");
  scanf("%d,%d,%d",&a,&b,&c);
  if(a>b) max=a;
  else    max=b;
  if(c>max) printf("3 个数中的最大数=%d\n",c);
  else    printf("3 个数中的最大数=%d\n",max);
  return 0;
}
```

2. 程序阅读

【程序 4.2】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
int main( )
{ int a,b,c,t;
  printf("输入 3 个整数的值:");
  scanf("%d,%d,%d",&a,&b,&c);
  if(a>b)
  { t=a;a=b;b=t; }
  if(a>c)
  { t=a;a=c;c=t; }
```

```

if(b>c)
{ t=b;b=c;c=t; }
printf("3 个数按照从小到大排列为: %d, %d, %d\n", a, b, c);
return 0;
}

```

3. 程序改错

【程序 4.3】编程实现从键盘输入一个整数，判断其能否既被 3 整除又被 5 整除，若能，输出“yes!”；否则，输出“no!”。下面调试程序中有 5 处错误，上机调试程序，修改有错误的语句行，并输出正确的运行结果。

```

#include<stdio.h>
int main( )
{   int x
    printf("请输入一个整数:");
    scanf("%d", x);
    if(x%3=0&& x%5=0) printf("yes!\n");
    else printf("no!\b");
    return 0;
}

```

4. 程序完善

【程序 4.4】编写鸡兔同笼问题程序，已知鸡兔同笼总数为 h，总脚数为 f，求鸡兔各有多少只？阅读程序后并将程序补充完整。并验证运行结果与自己分析的结果是否相符，记录下运行结果。

```

#include<stdio.h>
int main( )
{   int h, f, x, y;
    printf("输入:");
    scanf("%d, %d ", &h, &f);
    if(h>0&&f>0)
    {
        _____ ① _____ //计算鸡的数量
        _____ ② _____ //计算兔的数量
        printf( ③ ); //显示鸡兔的数量
    }
    else
        printf("输入错误! \n");
}

```

5. 程序设计

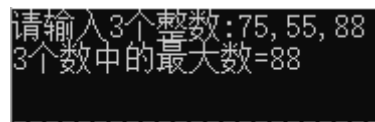
【程序4.5】编写一个程序实现这样的功能:商店卖软盘，每片定价3.5元，按购买的数

量可给予如下的优惠：购买满100片，优惠5%；购买满200片，优惠6%；购买满300片，优惠8%；购买满400片，优惠10%；购买500片以上，优惠15%。根据不同的购买量，打印应付货款。

4.2.3 实验指导

1. 程序示例

【程序 4.1】编程从键盘输入 3 个整数 a、b、c，输出其中最大的数。上机运行程序，验证运行结果与自己预习分析结果是否相符，并记录运行结果。

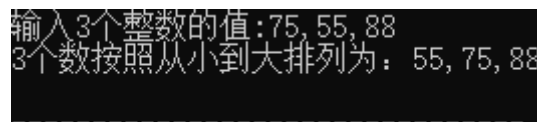


```
请输入3个整数:75, 55, 88
3个数中的最大数=88
```

图 4-1 程序 4.1 运行结果

2. 程序阅读

【程序 4.2】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。



```
输入3个整数的值:75, 55, 88
3个数按照从小到大排列为: 55, 75, 88
```

图 4-2 程序 4.2 运行结果

3. 程序改错

【程序 4.3】编程实现从键盘输入一个整数，判断其能否既被 3 整除又被 5 整除，若能，输出“yes!”；否则，输出“no!”。此程序中有 5 处错误，上机调试程序，修改有错误的语句行，并输出正确的运行结果。

```
#include<stdio.h>
int main( )
{   int x;
    printf(“请输入一个整数:”);
    scanf(“%d”,&x);
    if(x%3==0&& x%5==0) printf(“yes!\n”);
    else printf(“no!\n”);
    return 0;
}
```

修改后程序运行结果如图 4-3 所示。



```
请输入一个整数:30
yes!
```

图 4-3 程序 4.3 运行结果

4. 程序完善

【程序 4.4】编写鸡兔同笼问题程序，已知鸡兔同笼总数为 h，总脚数为 f，求鸡兔各有多少只？阅读程序后并将程序补充完整。

```

#include<stdio.h>
int main( )
{ int h, f, x, y;
  printf("输入:"); //输入鸡兔总数, 鸡兔脚的总数
  scanf("%d, %d", &h, &f); //从键盘输入两个数
  if(h>0&&f>0)
  { x=(4*h-f)/2; //计算鸡的数量
    y=(f-2*h)/2; //计算兔的数量
    printf("鸡: %d 兔: %d\n", x, y); //显示鸡兔的数量
  }
  else
    printf("输入错误!\n");
}

```

程序运行结果如图 4-4 所示。

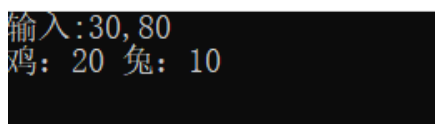


图 4-4 程序 4.4 运行结果

5. 程序设计

【程序4.5】编写一个程序实现这样的功能:商店卖软盘, 每片定价3.5元, 按购买的数量可给予如下的优惠: 购买满100片, 优惠5%; 购买满200片, 优惠6%; 购买满300片, 优惠8%; 购买满400片, 优惠10%; 购买500片以上, 优惠15%。根据不同的购买量, 打印应付货款。

```

#include<stdio.h>
int main( )
{ int n;
  float p, total;
  printf("输入购买软盘的数量:");
  scanf("%d", &n);
  if(n<0) printf("您的输入有误, 应输入正整数或 0!\n");
  else{
    if(n>=0&&n<100) p=0.0;
    else if(n>=100&&n<200) p=0.05;
      else if(n>=200&&n<300) p=0.06;
      else if(n>=300&&n<400) p=0.08;
        else if(n>=400&&n<500) p=0.10;
          else p=0.15;
    }
  total=3.5*n*(1-p);
  printf("应付货款=%f\n", total);
  return 0;
}

```

程序运行结果如图 4-5 所示。



```
输入购买软盘的数量:240
应付货款=789.599976
```

图 4-5 程序 4.5 运行结果

4.3 实验 5：选择结构程序设计（2）

4.3.1 实验目的

1. 正确使用关系表达式和逻辑表达式，熟练使用输入和输出语句。
2. 熟练掌握使用 if 语句及 switch 语句的格式与功能，并能够在 switch 语句中正确使用 break 语句编程。
3. 能够熟练运用选择结构进行综合编程。

4.3.2 实验内容

1. 程序示例

【程序 4.6】假设今天是星期日，程序求 20201212 天后是星期几。上机运行程序，验证运行结果与自己预习分析结果是否相符，并记录运行结果。

```
#include <stdio.h>
int main( )
{ int n=123456;
  n=n%7;
  printf("\n星期");
  switch(n)
  { case 0:printf("天");break;
    case 1:printf("一");break;
    case 2:printf("二");break;
    case 3:printf("三");break;
    case 4:printf("四"); break;
    case 5:printf("五");break;
    case 6:printf("六");break;
  }
  printf("\n\n");
  return 0;
}
```

2. 程序阅读

【程序 4.7】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

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

```

int main( )
{ long int num;
  int indiv, ten, hundred, thousand, place;
  printf("请输入一个 0-9999 之间的正整数: ");
  scanf("%d", &num);
  if(num>999) place=4;
  else if(num>99) place=3;
  else if(num>9) place=2;
  else place=1;
  printf("输入的正整数是%d 位数\n", place);
  /*输出位数*/
  /*以下四行分别求千位、百位、十位、个位数字 */
  thousand=num/1000;
  hundred=num/100%10;
  ten=num/10%10;
  indiv=num%10;
  switch(place) /*根据位数判断应该输出哪几位数字*/
  { case 4:
      printf("正序输出四位数分为:%d, %d, %d, %d", thousand, hundred, ten, indiv);
      printf("\n 反序输出各位数字为: ");
      printf("%d, %d, %d, %d\n", indiv, ten, hundred, thousand);
      break;
    case 3: printf("正序输出三位数分别为:%d, %d, %d", hundred, ten, indiv);
      printf("\n 反序输出各位数字为: ");
      printf("%d, %d, %d \n", indiv, ten, hundred);
      break;
    case 2: printf("正序输出两位数分别为: %d, %d", ten, indiv);
      printf("\n 反序输出各位数字为: ");
      printf("%d, %d \n", indiv, ten);
      break;
    case 1: printf("正序输出一位数分别为: %d", indiv);
      printf("\n 反序输出各位数字为: ");
      printf("%d\n", indiv);
      break;
  }
  return 0;
}

```

3. 程序完善

【程序 4.8】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: 当 $s < 250$ 时没有折扣, 当 $250 \leq s < 500$ 时有 2% 的折扣, 当 $500 \leq s < 1000$ 时有 5% 的折扣, 当 $1000 \leq s < 2000$ 时有 8% 的折扣, 当 $2000 \leq s < 3000$ 时有 10% 的折扣, 当 $3000 \leq s$ 时有 15% 的折扣。在阅读并理解程序思路的基础上, 说明程序功能, 补充横线上的内容。并验证运行结果与自己分析的结果是否相符, 记录下运行结果。

```
#include <stdio.h>
int main( )
{
    int c, s;
    _____ ① _____ p, w, d, f;
    printf("please enter price, weight, discount:");
    scanf("%f, %f, %d", &p, &w, &s);
    if(s >= 3000) c = 12;
    else c = s / 250;
    _____ ② _____
    { 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 = _____ ③ _____;
    printf("freight=%10.2f\n", f);
    return 0;
}
```

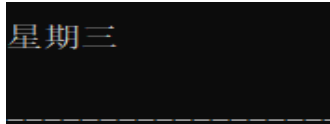
4. 程序设计

【程序 4.9】编写一个程序, 输入年号和月份, 判断该年是否为闰年, 并根据给出的月份判断是什么季节和该月有多少天?

4.3.3 实验指导

1. 程序示例

【程序 4.6】假设今天是星期日, 程序求 20201212 天后是星期几。上机运行程序, 验证运行结果与自己预习分析结果是否相符, 并记录运行结果。

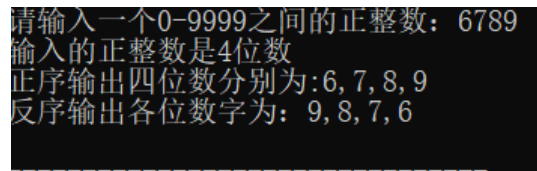


星期三

图 4-6 程序 4.6 运行结果

2. 程序阅读

【程序 4.7】本程序实现的是，从键盘输入一个不多于 4 位的正整数，求出它是几位数，并分别按正序和反序打印出每位数字。



请输入一个0-9999之间的正整数: 6789
输入的正整数是4位数
正序输出四位数分别为:6, 7, 8, 9
反序输出各位数字为: 9, 8, 7, 6

图 4-7 程序 4.7 运行结果

3. 程序完善

【程序 4.8】理解程序功能，阅读程序后并将程序补充完整。

```
#include <stdio.h>

int main( )
{
    int c, s;
    float p, w, d, f;
    printf("please enter price, weight, discount:");
    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 * (1 - d / 100);
    printf("freight=%10.2f\n", f);
    return 0;
}
```

程序运行结果如图 4-8 所示。

```
please enter price,weight,discount:100,20,300
freight= 588000.00
```

图 4-8 程序 4.8 运行结果

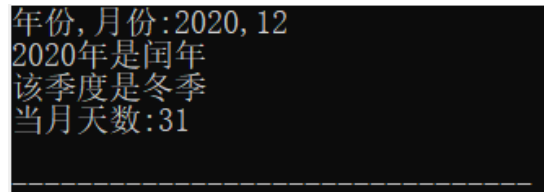
4. 程序设计

【程序 4.9】编写一个程序，输入年号和月份，判断该年是否为闰年，并根据给出的月份判断是什么季节和该月有多少天？

```
#include <stdio.h>
int main( )
{ int y,m, leap, season, days;
  printf("年份, 月份:");
  scanf("%d,%d",&y,&m);
  if ((y%4==0 && y%100!=0) || (y%400==0))
      leap=1;
  else
      leap=0;
  if (m>=3 && m<=5) season=1;
  else if (m>=6 && m<=8) season=2;
  else if (m>=9 && m<=11) season=3;
  else season=4;
  switch (m)
  { case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12: days=31; break;
    case 4:
    case 6:
    case 9:
    case 11: days=30; break;
    case 2:
      if(leap==1) days=29;
      else days=28;
  }
  printf("%d年%s闰年\n",y, (leap==1 ? "是" : "不是"));
  printf("该季度是");
  switch(season)
  { case 1: printf("春季\n"); break;
    case 2: printf("夏季\n"); break;
```

```
        case 3:printf("秋季\n");break;
        case 4:printf("冬季\n");break;
    }
    printf("当月天数:%d\n", days);
    return 0;
}
```

程序运行结果如图 4-9 所示。



```
年份, 月份:2020, 12
2020年是闰年
该季度是冬季
当月天数:31
-----
```

图 4-9 程序 4.9 运行结果

第 5 章 循环结构程序设计

5.2 实验 6：循环结构程序设计(1)

5.2.1 实验目的

1. 熟练掌握使用 while、do-while、for 语句实现循环程序设计的方法。
2. 掌握循环结构中常用的一些算法，如穷举法、迭代法、递推法等。

5.2.2 实验内容

1. 程序示例

【程序 5.1】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>

int main( )
{   int  i;

    float sum=0;

    for(i=1;i<=100;i++)

        sum=sum+1.0/i;

    printf("sum=%f\n", sum);

    return 0;

}
```

请分析，该程序完成的功能是：_____。

【程序 5.2】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include <stdio.h>

int main( )
{ char c;

  int  letters=0, space=0, digit=0, others=0;

  printf("please input some characters:");

  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
            others++;
    }

    printf("letters=%d, space=%d\n", letters, space);
    printf("digit=%d, others=%d\n", digit, others);
    return 0;
}

```

请分析，该程序完成的功能是：_____。

2. 程序改错

【程序 5.3】程序中有 3 处错误，上机调试程序，修改有错误的语句行，并输出正确的运行结果。求 $\text{sum} = 1! + 2! + 3! + \dots + n!$ 。

```

第1句: #include<stdio.h>
第2句: int main( )
第3句: { int i,n;
第4句:     long int s,sum;
第5句:     printf("请输入n值:");
第6句:     scanf("%d",&n);
第7句:     for(i=1;i<n;i++)
第8句:         {s=s *i;
第9句:             sum=sum+s;
第10句:         }
第11句:     printf("1!+2!+3!+...+n!=%ld\n",sum);
第12句:     return 0;
第13句: }

```

3. 程序完善

【程序 5.4】请将输入的正整数按逆序输出若输入 135 则输出 531，请填写。

```

#include<stdio.h>
int main( )
{
    int n,s;
    printf(" Enter a number: ");
    scanf("%d",&n);
    printf("Output: ");
    do
    {
        s=n%10;
        printf(" %d",s);
        _____ ① _____ //取出变量 n 中未输出的位
    } while(_____ ② _____); //设定循环条件
    return 0;
}

```

【程序 5.5】某幼儿园有若干小朋友，需要分组参加运动会（100~200 人），如果分成 4 人一组，则多两人；若分成 7 人一组则多 3 人；若分成 9 人一组，则多 5 人。求幼儿园小朋友的总数。

```

#include<stdio.h>
int main( )
{   int i;
    for(i=100;i<200;i++)
    if((i-2)%4==0)
    if(_____①)           //设计条件表达式，分成7人一组则多3人
    if(!(i-5)%9)
    printf("%d",i);
    return 0;
}

```

4. 程序设计

【程序5.6】编写程序，小猴子摘下一些桃子，卖掉一半，又吃了一只；第二天卖掉剩下的一半，又吃了一只；第三天、第四天、第五天都如此办理，第六天一看，发现就剩下一只桃子了，求小猴子共摘了多少只桃子。

【程序 5.7】编写程序，用一张五角的纸币换 5 分和 1 角的硬币（至少各有一枚），有哪些换法？

5.2.3 实验指导

1. 程序示例

【程序 5.1】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```

#include<stdio.h>
int main( )
{   int i;
    float sum=0;
    for(i=1;i<=100;i++)      //设定循环变量的初值，循环的条件，循环变量的增值
    sum=sum+1.0/i;           //循环体，完成累加
    printf("sum=%f\n",sum); //输出所求的和
    return 0;
}

```

程序的运行结果如图 5-1 所示。

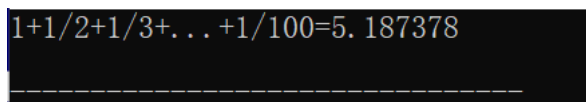


图5-1 例题5.1运行结果

【程序 5.2】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```

#include <stdio.h>

```

```

int main( )
{ char c;
  int letters=0, space=0, digit=0, others=0;
  printf("please input some characters:");
  while((c=getchar())!='\n')    //设定循环条件
  { if((c>='a' && c<='z') || (c>='A' && c<='Z')) letters++;
    //判断是否是字母，统计字母的数量
    else if(c==' ')
      space++; //判断是否是空格，统计空格的数量
    else if(c>='0' && c<='9')
      digit++; //判断是否是数字，统计数字的数量
    else
      others++; //统计其他字符的数量
  }
  printf("letters=%d, space=%d\n", letters, space);
  printf("digit=%d, others=%d\n", digit, others);
  return 0;
}

```

程序的运行结果如图 5-2 所示。

```

please input some characters:I love c programmer.
letters=16, space=3
digit=0, others=1

```

图5-2 例题5.2运行结果

2. 程序改错

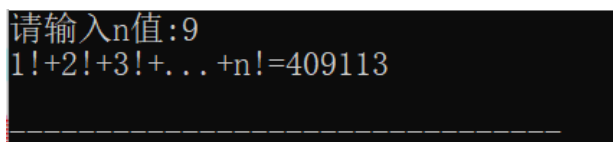
【程序 5.3】程序中有 3 处错误，上机调试程序，修改有错误的语句行，并输出正确的运行结果。求 $\text{sum} = 1! + 2! + 3! + \dots + n!$ 。

```

#include<stdio.h>
int main( )
{ int i,n;
  long int s,sum;    //定义变量
  s=1;sum=0;         //变量赋初值
  printf("请输入n值:"); //提示行
  scanf("%d",&n);    //此处应注意取地址符&
  for(i=1;i<=n;i++)  //确定循环的要素
  { s=s *i;          //用表达式表示n!
    sum=sum+s;       //求和
  }
  printf("1!+2!+3!+...+n!=%ld\n",sum); //输出结果
  return 0;
}

```

程序运行结果如图5-3所示。



```

请输入n值:9
1!+2!+3!+...+n!=409113

```

图5-3 例题5.3运行结果

3. 程序完善

【程序 5.4】请将输入的正整数按逆序输出若输入 369 则输出 963，请填空。

```

#include<stdio.h>
int main( )
{   int n,s;
    printf(" Enter a number: ");    //提示行
    scanf("%d",&n);                //从键盘输入原数据
    printf( "Output: ");            //提示行
    do
    { s=n%10;                        //取出变量 n 中的最低位
      printf(" %d" ,s);
      n/=10;                        //取出变量 n 中未输出的位
    } while(n!=0 );                //设定循环条件
    return 0;
}

```

程序运行结果如图5-4所示。



```

Enter a number: 369
Output: 963

```

图5-4 例题5.4运行结果

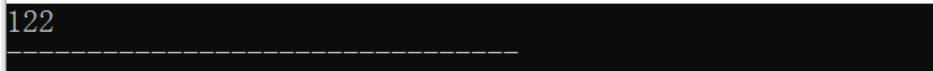
【程序 5.5】某幼儿园有若干小朋友，需要分组参加运动会（100~200 人），如果分成 4 人一组，则多两人；若分成 7 人一组则多 3 人；若分成 9 人一组，则多 5 人。求幼儿园小朋友的总数。

```

#include<stdio.h>
int main( )
{   int i;                          //定义变量
    for(i=100;i<200;i++)            //循环变量赋初值，设定循环条件及循环变量的增量
    if((i-2)%4==0)                  //设计 if 表达式，除以 4 的余数为 2，即 (i-2)%4==0
    if((i-3)%7==0 )                //设计 if 表达式，除以 7 的余数为 3，即 (i-3)%7==0
    if(!((i-5)%9))                 //设计 if 表达式，除以 9 的余数为 5，即 !((i-5)%9)
    printf("%d",i);                //输出符合条件的值
    return 0;
}

```

程序运行结果如图5-5所示。



```

122

```

图5-5 例题5.5运行结果

4. 程序设计

【程序5.6】编写程序，小猴子摘下一些桃子，卖掉一半，又吃了一只；第二天卖掉剩下的一半，又吃了一只；第三天、第四天、第五天都如此办理，第六天一看，发现就剩下一只桃子了。编写一个程序，求小猴子共摘了多少只桃子。

```
#include <stdio.h>
int main( )
{ int n=1,i;
  for (i=1;i<=5;i++)
    n=(n+1)*2;
  printf("桃子总数:%d\n",n);
  return 0;
}
```

程序运行结果如图5-6所示。



图5-6 例题5.6运行结果

【程序 5.7】编写程序，用一张五角纸币换 5 分和 1 角的硬币（至少各有一枚），有哪些换法？

```
#include<stdio.h>
int main( )
{ int x,y;
  for(y=1;y<5;y++)
    for(x=1;x<10;x++)
      if(5*x+10*y==50)
        printf("x=%d, y=%d\n", x, y);
  return 0;
}
```

程序运行结果如图5-7所示。

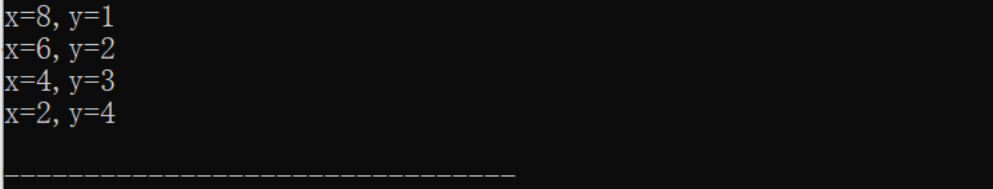


图5-7 例题5.7运行结果

5.3 实验 7：循环结构程序设计(2)

5.3.1 实验目的

1. 熟练掌握 break 语句和 continue 语句的用法。

2. 熟练掌握循环嵌套的程序设计方法。

5.3.2 实验内容

1. 程序示例

【程序 5.8】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include <stdio.h>
int main( )
{   int i;
    for(i=100;i<=120;i++)
        { if(i%3!=0)
            continue;
          printf( "%d", i);
        }
    return 0;
}
```

请分析，该程序完成的功能是：_____。

【程序 5.9】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include <stdio.h>
#define PI 3.14159
int main( )
{   int r ;
    float area;
    for(r=1;r<=10;r++)
        { area=PI*r*r ;
          if(area>100) break;
          printf("r=%d, area=%f\n", r, area);
        }
    return 0;
}
```

请分析，该程序完成的功能是：_____。

2. 程序完善

【程序 5.10】本程序的功能是求 100~200 间的全部素数，请根据注释填空。

```
#include <stdio.h>
#include <math.h>
int main( )
{   int m, k, i, n=0;
    for( _____ ① ) /*设置循环的要素*/
        { k=sqrt(m);
```

```

        for(i=2;i<=k;i++)
            if(m%i==0) break;
        if(i>=k+1) {printf("%d ",m);n=n+1;}
        if(n%10==0) printf("\n");
    }
    printf("\n");
    return 0;
}

```

3. 程序设计

【程序 5.11】编写程序，从键盘输入 6 名学生的 5 门成绩，分别统计出每个学生的平均成绩。

【程序5.12】编写程序，输出两位正整数的“平方镜反数”。平方镜反数是指该数的平方与该数的反序数的平方互为反序数。例如 $13^2=169$ ， $31^2=961$ ，则13就是一个平方镜反数。

【程序 5.13】编写程序，输出如图形。

```

    *
   **
  ***
 ****
 ***
  *

```

5.3.3 实验指导

1. 程序示例

【程序 5.8】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```

#include <stdio.h>
int main( )
{
    int i;
    for(i=100;i<=120;i++)        //设定循环要素
    {
        if(i%3==0)                //判定 i 能否被 3 整除
            continue;            //表达式成立，结束本次循环
        printf( "%d",i);          //表达式不成立，即不能被 3 整除，输出该数
    }
    return 0;
}

```

程序的运行结果如图 5-8 所示。

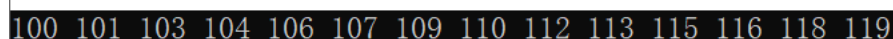


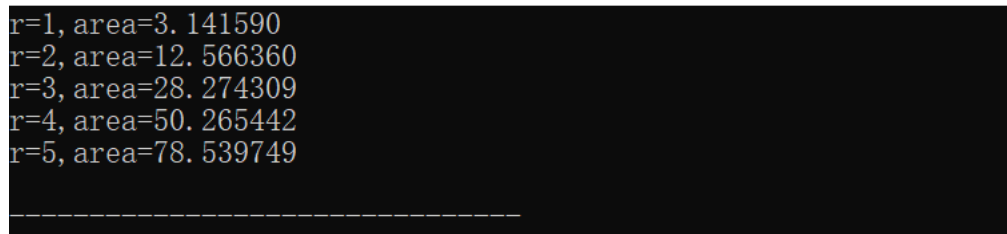
图5-8 例题5.8运行结果

【程序 5.9】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机

运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include <stdio.h>
#define PI 3.14159
int main( )
{   int  r ;           //定义圆的半径为整形变量
    float area;        //定义圆的面积为实型变量
    for(r=1;r<=10;r++) //设定循环的要素
    {   area=PI*r*r ;
        if(area>100) break;           //当面积大于 100 时，结束循环
        printf("r=%d,area=%f\n",r,area); //输出圆的半径和面积
    }
    return 0;
}
```

程序的运行结果如图 5-9 所示。



```
r=1, area=3.141590
r=2, area=12.566360
r=3, area=28.274309
r=4, area=50.265442
r=5, area=78.539749
```

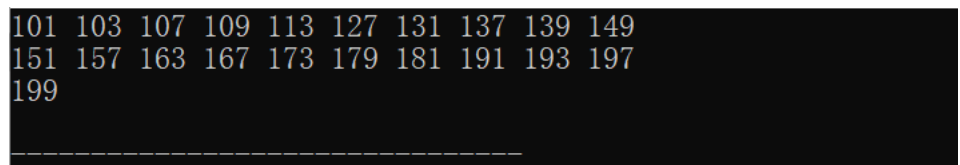
图5-9 例题5.9运行结果

2. 程序完善

【程序 5.10】本程序的功能是求 100~200 间的全部素数，请根据注释填空。

```
#include <stdio.h>
#include <math.h>           //使用了平方根sqrt函数，需要包含该头文件
int main( )
{   int m, k, i, n=0;       //定义变量
    for(m=101;m<=200;m=m+2) //设计循环的要素
    {   k=sqrt(m);           //求m的开方值
        for(i=2;i<=k;i++)   //设计内层循环的要素
        {   if(m%i==0) break; //表达式成立，不是素数，跳出循环
            if(i>=k+1) {printf("%d ",m);n=n+1;} //输出满足条件的数
            if(n%10==0) printf("\n");           //每行十个数据，写满十个，换行输出
        }
    }
    printf("\n");
    return 0;
}
```

分析：程序运行结果如图5-10所示。



```
101 103 107 109 113 127 131 137 139 149
151 157 163 167 173 179 181 191 193 197
199
```

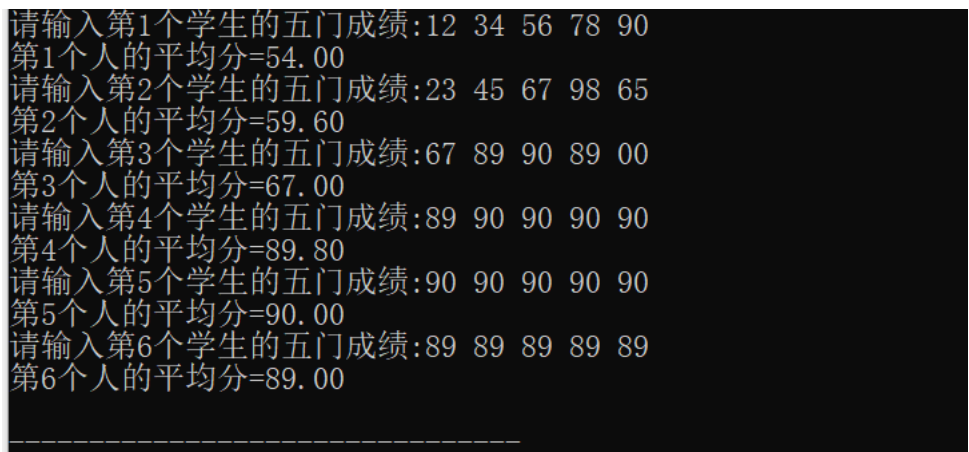
图5-10 例题5.10运行结果

3. 程序设计

【程序 5.11】编写程序，从键盘输入 6 名学生的 5 门成绩，分别统计出每个学生的平均成绩。

```
#include<stdio.h>
int main( )
{ int i, j;
  float score, sum, ave;
  for(i=1; i<=6; i++)
  { sum=0;
    printf("请输入第%d 个学生的五门成绩:", i);
    for(j=1; j<=5; j++)
    { scanf("%f", &score);
      sum=sum+score;
    }
    ave=sum/5;
    printf("第%d 个人的平均分=%5.2f\n", i, ave);
  }
  return 0;
}
```

程序运行结果如图 5-11 所示。



```
请输入第1个学生的五门成绩:12 34 56 78 90
第1个人的平均分=54.00
请输入第2个学生的五门成绩:23 45 67 98 65
第2个人的平均分=59.60
请输入第3个学生的五门成绩:67 89 90 89 00
第3个人的平均分=67.00
请输入第4个学生的五门成绩:89 90 90 90 90
第4个人的平均分=89.80
请输入第5个学生的五门成绩:90 90 90 90 90
第5个人的平均分=90.00
请输入第6个学生的五门成绩:89 89 89 89 89
第6个人的平均分=89.00
```

图5-11 例题5.11运行结果

【程序5.12】编写程序，输出两位正整数的“平方镜反数”。平方镜反数是指该数的平方与该数的反序数的平方互为反序数。例如 $13^2=169$ ， $31^2=961$ ，则13就是一个平方镜反数。

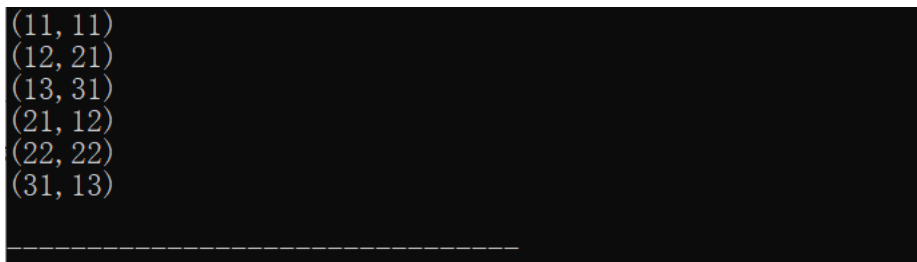
```
#include <stdio.h>
int main( )
{ int i, j, n, m, t;
  for(i=10; i<100; i++)
  { j=i%10*10+i/10;
    m=i*i;
  }
```

```

n=j*j;
t=0;
while(n)
    { t=t*10+n%10; n=n/10; }
if(m==t)
printf("(d, %d)\n", i, j);
}
return 0;
}

```

程序运行结果如图5-12所示。



```

(11, 11)
(12, 21)
(13, 31)
(21, 12)
(22, 22)
(31, 13)
-----

```

图5-12 例题5.12运行结果

【程序 5.13】编写程序，输出如图形。

```

      *
     ***
    *****
   ***
  *
#include<stdio.h>
int main( )
{ int i, j;
  for(i=1; i<=3; i++)
  { for(j=1; j<=3-i; j++)
    printf(" ");
    for(j=1; j<=2*i-1; j++)
      printf("*");
    printf("\n");
  }
  for(i=4; i<=5; i++)
  { for(j=1; j<=i-3; j++)
    printf(" ");
    for(j=1; j<=2*(5-i)+1; j++)
      printf("*");
    printf("\n");
  }
  return 0;
}

```

}

程序运行结果如图5-13所示。

```
*  
***  
*****  
***  
*  
-----
```

图5-13 例题5.13运行结果

第 6 章 同一类型多个元素的集合-数组

6.2 实验 8：数组程序设计(1)

6.2.1 实验目的

1. 掌握一维数组的定义、赋值和输入输出的方法。
2. 会用一维数组编程解决实际生活中的问题。
3. 熟练掌握使用二维数组进行程序设计的方法。

6.2.2 实验内容

1. 程序示例

【程序 6.1】已知 10 个人的成绩存放在 score 数组中，统计低于平均分的人数，并将低于平均分的成绩输出显示。

```
#include <stdio.h>
#define N 10
int main( )
{ float score[10];
  float aver=0;
  int i, j=0;
  for(i=0; i<N; i++)
  { scanf("%f", &score[i]);
    aver+=score[i];
  }
  aver/=N;
  for(i=0; i<N; i++)
  if(score[i]<aver)
  { j++;
    printf("score[%ld]=%.1f\n", i, score[i]);
  }
  printf("j=%2d\n", j);
  return 0;
}
```

2. 程序阅读

【程序 6.2】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
#define N 10
int main( )
```

```

{ int i, j, k=0, t, d, a[N];
long n;
printf("Please enter a long integer:");
scanf("%d", &n);
while(n>0)
{ t=n%10; a[k++]=t; n/=10; }
for(i=0; i<k-1; i++)
{ d=i;
for(j=i+1; j<k; j++)
if(a[j]>a[d]) d=j;
if(d!=i) {t=a[d]; a[d]=a[i]; a[i]=t;
}
}
for(i=0; i<k; i++)
printf("%ld ", a[i]);
return 0;
}

```

【程序 6.3】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```

#include<stdio.h>
int main( )
{ int tt[3][4]={{1, 2, 3, 4}, {8, 7, 6, 5}, {5, 4, 1, 3}}, pp[4]; //赋初值
int i, j, min;
for(j=0; j<N; j++)
{ min=tt[0][j];
for(i=0; i<M; i++)
{ if (tt[i][j]<min)
min=tt[i][j];
}
pp[j]=min;
}
for(i=0; i<N; i++)
printf("pp[%d]=%d, ", i, pp[i]);
return 0;
}

```

3. 程序设计

【程序 6.4】已知班 5 名学生期末 4 门课程的考试成绩。编制程序计算各门课的平均成绩和每位同学的平均绩和总成绩，并输出显示。

【程序 6.5】找出一个已知二维数组的鞍点，即该位置上的元素在该行上最大，在该列上最小。也可能没有鞍点。

6.2.3 实验指导

1. 程序示例

【程序 6.1】10 个人的成绩存放在 score 数组中，请编写程序。它的功能是：统计低于平均分的人数，并将低于平均分的成绩输出显示。

```
78.5 77.4 65.5 63.4 72.9 91.2 45.7 85.3 45.9 56.9
score[2]=65.5
score[3]=63.4
score[6]=45.7
score[8]=45.9
score[9]=56.9
aver=68.3,j= 5
-----
```

图 6-1 程序 6.1 运行结果

2. 程序阅读

【程序 6.2】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
#define N 10
int main( )
{ int i,j,k=0,t,d,a[N];
  long n;
  printf("Please enter a long integer:");
  scanf("%d",&n);
  while(n>0)      // 依次取出每位数，并将数存入数组
  { t=n%10;      // 取出一位数
    a[k++]=t;    // 将数存入数组
    n/=10;      // 为取下一位数做准备
  }
  for(i=0;i<k-1;i++)      // 对取出的数进行降序排序，共k-1轮
  { d=i; // 记录每轮起始元素的下标
    for(j=i+1;j<k;j++)
      if(a[j]>a[d]) // 后续元素依次与起始元素进行比较
        d=j; // 记录比起始元素大的元素的下标
    if(d!=i)      // 一轮比较后判断记录的大小是否发生了变化
      {t=a[d];a[d]=a[i];a[i]=t;} // 若发生了变化，则交换元素值
  }
  for(i=0;i<k;i++)
    printf("%1d ",a[i]);      /*输出排序结果*/
  return 0;
}
```

程序运行结果如图6-2所示。

```
Please enter a long integer:9872156
9 8 7 6 5 2 1
-----
```

图 6-2 程序 6.2 运行结果

【程序 6.3】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
int main( )
{ int tt[3][4]={ {1, 2, 3, 4}, {8, 7, 6, 5}, {5, 4, 1, 3}}, pp[4]; //赋初值
int i, j, min;
for(j=0; j<4; j++)
{ min=tt[0][j];
for(i=0; i<3; i++)
{ if (tt[i][j]<min)
min=tt[i][j]; //求出每列的最小值
}
pp[j]=min; //存入pp数组中
}
for(i=0; i<4; i++)
printf("pp[%d]=%d, ", i, pp[i]); //按顺序输出每列的最小值
return 0;
}程序的运行结果如图6-3所示。
```

```
pp[0]=1, pp[1]=2, pp[2]=1, pp[3]=3,
-----
```

图 6-3 程序 6.3 运行结果

3. 程序设计

【程序 6.4】已知班5名学生期末4门课程的考试成绩。编制程序计算各门课的平均成绩和每位同学的平均绩和总成绩，并输出显示。

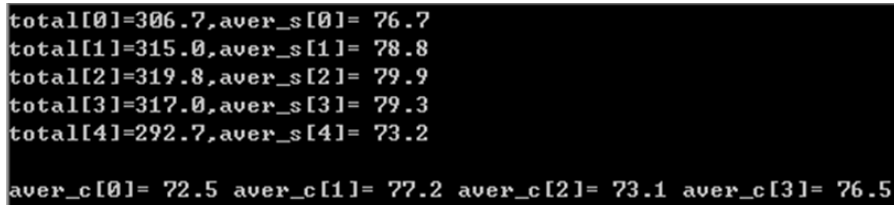
```
#include <stdio.h>
#define M 5
#define N 4
int main( )
{float
tt[M][N]={ {70.0, 80.0, 78.9, 88.5}, {80.5, 75.5, 69.5, 50.5}, {75.0, 74, 81, 93},
{81.2, 85.5, 90.4, 85.0}, {56.0, 70.8, 45.5, 65.5}};
float total_M[M], aver_s[M], aver_c[N], total_N[N];
int i, j;
for(j=0; j<M; j++)
{ total_M[j]=0;
for(i=0; i<N; i++)
```

```

        total_M[j]+=tt[i][j]; //求每学生的总成绩
        aver_s[j]=total_M[j]/N; //求每学生的平均成绩
    }
    for(j=0;j<N;j++)
    { total_N[j]=0;
      for(i=0;i<M;i++)
          total_N[j]+=tt[i][j]; //求每门课的总成绩
      aver_c[j]=total_N[j]/M; //求每门课的平均成绩
    }
    for(i=0;i<M;i++)
        printf("total[%d]=%5.1f, aver_s[%d]=%5.1f\n", i, total_M[i], i, aver_s[i]);
    /*输出每个学生的总成和平均成绩 */
    printf("\n");
    for(i=0;i<N;i++)
        printf("aver_c[%d]=%5.1f ", i, aver_c[i]); /*输出每门课的平均成绩 */
    printf("\n");
    return 0;
}

```

程序的运行结果如图6-4所示。



```

total[0]=306.7,aver_s[0]= 76.7
total[1]=315.0,aver_s[1]= 78.8
total[2]=319.8,aver_s[2]= 79.9
total[3]=317.0,aver_s[3]= 79.3
total[4]=292.7,aver_s[4]= 73.2

aver_c[0]= 72.5 aver_c[1]= 77.2 aver_c[2]= 73.1 aver_c[3]= 76.5

```

图6-4 程序6.4的运行结果

【程序 6.5】找出一个已知二维数组的鞍点，即该位置上的元素在该行上最大，在该列上最小。也可能没有鞍点。

```

#include<stdio.h>
#define M 4
#define N 4
int main( )
{ int a[M][N]={ {8, 2, 3, 4}, {7, 6, 5, 4}, {6, 2, 3, 4}, {8, 6, 5, 4} };
  int i, j, k, max, col=0, row=0, flag=0, n=0;
  for(i=0;i<M;i++)
  { max=a[i][0]; //预设第1列的值为最大值
    for(j=0;j<N-1;j++)
    { if (max<a[i][j+1])
        { max=a[i][j+1]; //找出每一行的最大值
          col=j+1; //记录此元素的列号
          row=i;
        }
    }
  }
}

```

```

        }
    else
        { col=0;row=i; }
    }
    flag=1;
    for(k=0;k<M;k++)
    {
        if(max>a[k][col])    {flag=0;break;}
    }
    if (flag==1)
    { printf("鞍点: row=%d, col=%d", row, col); //输出鞍点的位置
      n++;
    }
}
if(n==0) printf("无鞍点");
return 0;
}

```

程序的运行结果如图6-5所示。

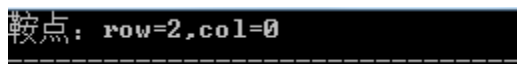


图6-5 程序6.5的运行结果

6.3 实验 9：数组程序设计(2)

6.3.1 实验目的

1. 掌握字符数组的输入、输出方式。
2. 掌握字符数组及字符串函数的应用。

6.3.2 实验内容

1. 程序示例

【程序 6.6】使用 scanf(), printf() 函数配合 %s 格式输入中间含有空格的字符串的输入输出。

```

#include<stdio.h>
int main( )
{ int i;
  char str1[15],str2[15],str3[15],str4[15];
  printf("请输入一串字符: ");
  scanf ("%s%s%s%s", str1,str2, str3, str4);
  printf("%s %s %s %s\n", str1, str2, str3, str4);
  return 0;
}

```

2. 程序阅读

【程序 6.7】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
int main( )
{ int i,n=0;
  char s[20];
  printf("请输入一个字符串:");
  for(i=0;(s[i]=getchar( ))!='\n';i++)
  if(s[i]>='A' && s[i]<='Z') n++;
  printf("n=%d\n",n);
  for(i=0;s[i]!='\n';i++)
    if(s[i]>='A' && s[i]<='Z')
      printf("%c",s[i]+32);
    else
      printf("%c",s[i]);
  printf("\n");
  return 0;
}
```

【程序 6.8】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
#include<string.h>
int main( )
{ char str1[20],str2[20];
  int i=0,j=0;
  printf("请输入第一个字符串: ");
  gets(str1);
  while(str1[i]!='\0')
  {
    if((str1[i]>='a' && str1[i]<='z') || (str1[i]>='A' && str1[i]<='Z'))
      { str2[j]=str1[i];j++; }
    i++;
  }
  str2[j]='\0';
  printf("新串为: ");
  puts(str2);
  printf("新串长度为: %d\n",strlen(str2));
  return 0;
}
```

3. 程序完善

【程序 6.9】删除一个字符串中某个字符。

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

```

#include<string.h>
int main( )
{ int i,j=0;
  char ch[]="this is a string.",c;
  puts(ch);
  printf("请输入一个字符: ");
  c=getchar();
  for(i=0;ch[i]!='\0';i++)
    if(ch[i]!=c)
      ① _____ ;
      ② _____ ;
  printf("%s\n",ch);
  return 0;
}

```

4. 程序设计

【程序6.10】运用字符串处理函数，将已知的5种英文字符串按英文字符的顺序从小到大排序。

【程序 6.11】编写统计一个长度为 2 的字符串在另一个字符串中出现的次数。

【程序 6.12】输入一个字符串，判断是否为回文串。所谓回文串，是指一个字符串顺读和反读是一样的，如“123321”和“abcba”都是回文串。

6.3.3 实验指导

1. 程序示例

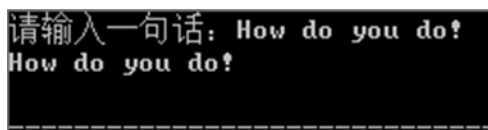
【程序 6.6】使用 scanf(), printf() 函数配合%s 格式输入中间含有空格的字符串的输入输出。

```

#include<stdio.h>
int main( )
{ int i;
  char str1[15],str2[15],str3[15],str4[15]; //定义四个字符数组
  printf("请输入一串字符: ");
  scanf ("%s%s%s%s", str1,str2,str3,str4); //输入带空格的四个字符串
  printf("%s %s %s %s\n", str1,str2,str3,str4); //输出带空格的四个字符串
  return 0;
}

```

程序运行结果如图6-6所示。



```

请输入一句话: How do you do!
How do you do!

```

图 6-6 程序 6.6 运行结果

2. 程序阅读

【程序 6.7】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
int main( )
{ int i,n=0;
  char s[20];
  printf("请输入一个字符串:");
  for(i=0;(s[i]=getchar())!='\n';i++) //输入字符直到遇见回车符
  if(s[i]>='A' && s[i]<='Z') n++; //所有的大写字母个数
  printf("n=%d\n",n);
  for(i=0;s[i]!='\n';i++)
    if(s[i]>='A' && s[i]<='Z')
      printf("%c",s[i]+32); //大写转换为小写输出
    else
      printf("%c",s[i]); //其它字符不变输出
  printf("\n");
  return 0;
}
```

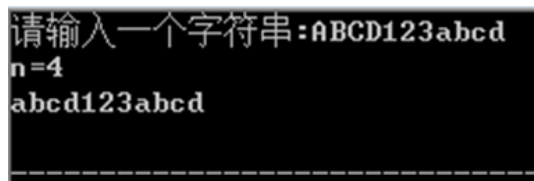


图6-7 程序6.7运行结果

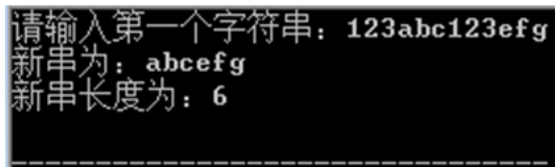
【程序 6.8】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
#include<string.h>
int main( )
{char str1[20],str2[20];
//定义二个字符串数组，一个保存输入的字符串，另一个存储删除数字后的字符串
  int i=0,j=0;
  printf("请输入第一个字符串:");
  gets(str1); //输入字符串
  while(str1[i]!='\0')
  { if((str1[i]>='a' && str1[i]<='z') || (str1[i]>='A' && str1[i]<='Z'))
    { str2[j]=str1[i]; //保留字符
      j++; //记录字符个数
    }
  }
```

```

    }
    i++;
}
str2[j]='\0';          //尾部增加结束符变成字符串
printf("新串为: ");
puts(str2);           //输出字符串
printf("新串长度为: %d\n", strlen(str2)); //输出字符串长度值
return 0;
}

```



```

请输入第一个字符串: 123abc123efg
新串为: abcefg
新串长度为: 6

```

图6-8 程序6.8运行结果

3. 程序完善

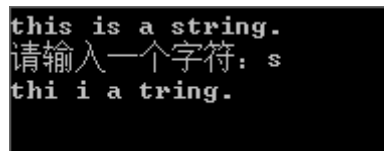
【程序6.9】 删除一个字符串中某个字符。

```

#include<stdio.h>
#include<string.h>
int main( )
{ int i,j=0;
  char ch[]="this is a string.",c;
  puts(ch);
  printf("请输入一个字符: ");
  c=getchar();
  for(i=0;ch[i]!='\0';i++)
    if(ch[i]!=c)
      ch[j++]=ch[i];    //将不是要删除的字符存在数组中
  ch[j]='\0';          //字符串加上结束符
  printf("%s\n",ch);
  return 0;
}

```

程序运行结果如图6-9所示。



```

this is a string.
请输入一个字符: s
thi i a tring.

```

图6-9 程序6.9运行结果

4. 程序设计

【程序6.10】 运用字符串处理函数，将已知的5种英文字符串按英文字符的顺序从小到大排序。

```

#include<stdio.h>
#include<string.h>
int main( )
{ char str[5][30]={"Batang","Arial","Dotum","Gulim","Calibri"};
  char ct[30];
  int i=0,j=0,k;
  for (i=0;i<5;i++)
  {k=i;
   strcpy(ct, str[i]);
   for(j=i+1;j<5;j++)
   if(strcmp(str[j], ct)<0)
   { k=j; strcpy(ct, str[j]); }
   if(k!=i)
   { strcpy(ct, str[i]);
     strcpy(str[i], str[k]);
     strcpy(str[k], ct);
   }
   puts(str[i]);
 }
 return 0;
}

```

程序运行结果如图6-10所示。



```

Arial
Batang
Calibri
Dotum
Gulim

```

图6-10 程序6.10运行结果

【程序6.11】编写统计一个长度为2的字符串在另一个字符串中出现的次数。

```

#include<stdio.h>
#define N 30
int main( )
{ int n=0,i=0,j=0;
  char str[N], substr[3];
  printf("输入一个字符串:");
  gets(str);
  printf("输入一个子串: ");
  gets(substr);
  while(str[i]!='\0')
  { if(str[i]==substr[j])
    { i++;j++;}
    else

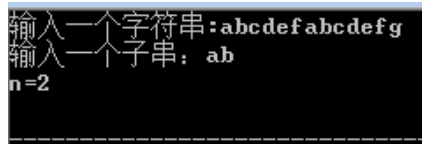
```

```

        { i++;j=0;}
    if(substr[j]=='\0')
        { n++; j=0;}
    }
    printf("n=%d\n",n);
    return 0;
}

```

程序运行结果如图6-11所示。



```

输入一个字符串:abcdefg
输入一个子串: ab
n=2

```

图6-11 程序6.11运行结果

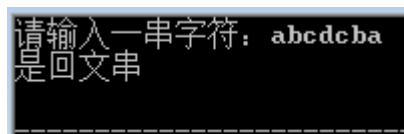
【程序6.12】输入一个字符串，判断是否为回文串。所谓回文串，是指一个字符串顺读和反读是一样的，如“12344321”和“andna”都是回文串。

```

#include<stdio.h>
int main( )
{ char s[10];
  int i,j,n;
  printf("请输入一串字符: ");
  gets(s);
  for(i=0; s[i]!='\0'; i++);
  n=i;
  for(i=0, j=n-1; i<j; i++, j--)
    if(s[i]!=s[j]) break;
  if(i<j)
    printf("不是回文串\n");
  else
    printf("是回文串\n");
  return 0;
}

```

程序运行结果如图6-12, 6-13所示。



```

请输入一串字符: abcdba
是回文串

```

图6-12 程序6.12运行结果（1）

```
请输入一串字符: 12341234  
不是回文串
```

图6-13 程序6.12运行结果（2）

第 7 章 模块化程序设计-函数

7.2 实验 10：函数程序设计（1）

7.2.1 实验目的

1. 掌握自定义函数的声明、定义和调用方法。
2. 理解主调函数和被调函数之间的参数传递方式。

7.2.2 实验内容

1. 程序示例

【程序 7.1】上机运行程序，练习函数定义、调用和声明，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```
#include <stdio.h>
int main( )
{ long factorial (int n);    //函数原型声明
  int m, n;
  long c;
  scanf("m=%d,n=%d", &m, &n);
  c=factorial(m)/factorial(n);    //调用语句 c=factorial(m)/factorial(n)
  c=c/factorial(m-n);            //调用语句 c=c/factorial(m-n)
  printf("%ld\n",c);
  return 0;
}
long factorial (int n)    //自定义函数 factorial 的功能是求 n!
{ long fact;
  int i;
  for (fact=1, i=2; i<=n; i++)
    fact *= i ;
  return ( fact );
}
```

【程序 7.2】上机运行程序，练习各种函数调用的方法，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```
#include <stdio.h>
int main( )
{ int max(int x,int y) ;    //有参函数 max 的函数声明
  int a,b,c,c1,c2,c3,c4,c5;
  scanf("a=%d,b=%d,c=%d",&a,&b,&c);
```

```

    c1=max(a, b);
    c2=max(4, 9);
    c3=6+max(4, 9);
    c4=max(a, max(b, c));
    c5=max(6, max(4, 9));
    printf("c1=max(a, b)=%d\n", c1);
    printf("c2=max(4, 9)=%d\n", c2);
    printf("c3=6+max(4, 9)=%d\n", c3);
    printf("c4=max(a, max(b, c))=%d\n", c4);
    printf("c5=max(6, max(4, 9))=%d\n", c5);
    printf("max=%d\n", max(6, max(4, 9)));
    return 0;
}

int max(int x, int y)    //定义有参函数 max，求出两个整数中的较大数。
{ int z;
  if (x > y) z = x;
  else z = y;
  return(z);
}

```

2. 程序阅读

【程序 7.3】在阅读并理解程序思路的基础上，理解函数的值传递，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```

#include <stdio.h>
void fun(int p)
{ int d=3;
  p=d++;
  printf("fun 函数中 p=%d\n", p);
}

int main( )
{ int a=1;
  fun(a);
  printf("main 函数中 a=%d\n", a);
  return 0;
}

```

3. 程序改错

【程序 7.4】程序中有 4 处错误，上机调试程序，修改有错误的语句行，并输出正确的运行结果。

```

第 1 行: #include<stdio.h>
第 2 行: int main( )
第 3 行: { int max(int x, int y);
第 4 行: int a, b, c;

```

```

第 5 行: scanf("%d,%d",&a,b);
第 6 行: max(a,b);
第 7 行: printf("max is %d\n,c");
第 8 行: return 0; }
第 9 行: int max (int x,int y);
第 10 行: { int z ;
第 11 行: z=x>y?x:y;
第 12 行: return (z);
第 13 行: }

```

4. 程序完善

【程序 7.5】已知求班级学生 C 语言成绩的平均分函数 aver 和最高分函数 max，请根据注释，填写相应语句，上机运行程序，实现一个学生班级 C 语言成绩处理程序。

```

#include <stdio.h>
#define N 10
float aver(float x[], int n) //求平均分函数
{ int i; float sum=0;
  for(i=0;i<n;i++) sum+=x[i];
  return sum/n;
}
float max (float x[], int n) //求最高分函数
{ int i ; float m=x[0];
  for(i=1;i<n;i++) if(m<x[i]) m=x[i];
  return m ;
}

```

(1) 设计统计班级学生 C 语言成绩不及格人数的函数。

```

int fault(float x[], int n) //统计不及格人数函数
{ ① //定义循环变量和统计变量
  ② //查询不及格分数，统计人数
  return f; //返回不及格人数
}

```

(2) 设计主函数：从键盘输入学生 C 语言成绩，调用平均分、最高分和统计不及格人数的函数，输出平均分、最高分和不及格人数。

```

int main()
{ float a[N]; //定义成绩数组
  int i; //定义循环变量
  printf("请输入 10 个学生的成绩:"); //提示输入成绩
  ③ // 输入成绩
  printf("平均分=%.1f\n", aver(a,N)); //调用平均分函数并输出
  printf("最高分=%.1f\n", max(a,N)); //调用最高分函数并输出
  ④ //调用统计不及格人数函数并输出
}

```

```
return 0;
}
```

5. 程序设计

【程序 7.6】设计三个函数，分别实现计算圆周长，圆面积和圆球体积的功能，通过主函数由键盘键入圆半径 r 的值，调用三个函数实现具体的计算值输出。

【程序 7.7】用非递归方法编写一个求 $n!$ 的函数，在主函数中实现计算：
 $1!+2!+3!+4!+5!+\dots+20!$

7.2.3 实验指导

1. 必做内容

【程序 7.1】上机运行程序，练习函数定义、调用和声明，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```
m=6,n=3
20
```

图 7-1 例题 7.1 运行结果

【程序 7.2】上机运行程序，验证运行结果与预习结果是否相符，并记录运行结果。

```
a=6,b=4,c=9
c1=max(a,b)=6
c2=max(4,9)=9
c3=6+max(4,9)=15
c4=max(a,max(b,c))=9
c5=max(6,max(4,9))=9
max=9
```

图 7-2 例题 7.2 运行结果

2. 程序阅读

【程序 7.3】在阅读并理解程序思路的基础上，理解函数的值传递，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
fun函数中p=3
main函数中a=1
```

图 7-3 例题 7.3 运行结果

3. 程序改错

【程序 7.4】程序中有 4 处错误，上机调试程序，修改有错误的语句行，并输出正确的运行结果。

```
#include<stdio.h>
int main( )
{ int max(int x,int y);
  int a,b,c;
  scanf("%d,%d",&a,&b);
  c=max(a,b);
```

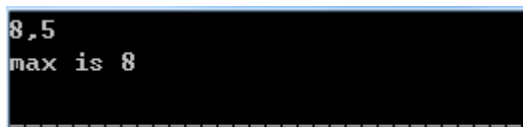
```

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

int max (int x,int y)
{ int z ;
  z=x>y?x:y;
  return (z);
}

```

程序运行结果如图 7-4 所示。



```

8.5
max is 8

```

图 7-4 例题 7.4 运行结果

4. 程序完善

【程序 7.5】已知求班级学生 C 语言成绩的平均分函数 aver 和最高分函数 max，请根据注释，填写相应语句，上机运行程序，实现一个学生班级 C 语言成绩处理程序。

```

#include <stdio.h>
#define N 10
float aver(float x[], int n) //求平均分函数
{ int i; float sum=0;
  for(i=0;i<n;i++) sum+=x[i];
  return sum/n;
}

float max (float x[], int n) //求最高分函数
{ int i ; float m=x[0];
  for(i=1;i<n;i++) if(m<x[i]) m=x[i];
  return m ;
}

int fault(float x[], int n) //统计不及格人数函数
{ int i,f=0; //定义循环变量和统计变量
  for(i=0;i<n;i++) if(x[i]<60) f++; //查询不及格分数，统计人数
  return f; //返回不及格人数
}

int main()
{ float a[N]; //定义成绩数组
  int i; //定义循环变量
  printf("请输入 10 个学生的成绩:"); //提示输入成绩
  for(i=0;i<N;i++) scanf("%f",&a[i]); // 输入成绩
  printf("平均分=%.1f\n",aver(a,N)); //调用平均分函数并输出
  printf("最高分=%.1f\n", max(a,N)); //调用最高分函数并输出
  printf("不及格人数=%d\n", fault(a,N)); //调用统计不及格人数函数并输出
}

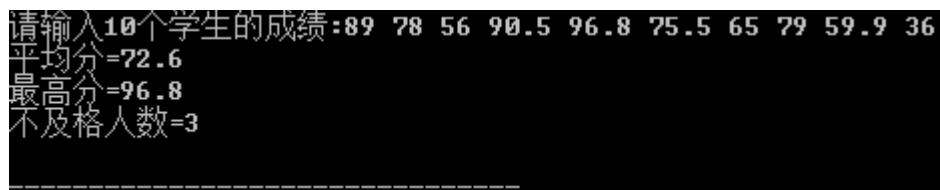
```

```

    return 0;
}

```

程序运行结果如图 7-5 所示。



```

请输入10个学生的成绩:89 78 56 90.5 96.8 75.5 65 79 59.9 36
平均分=72.6
最高分=96.8
不及格人数=3

```

图 7-5 例题 7.5 运行结果

5. 程序设计

【程序 7.6】设计三个函数，分别实现计算圆周长，圆面积和圆球体积的功能，通过主函数由键盘键入圆半径 r 的值，调用三个函数实现具体的计算值输出。

```

#include <stdio.h>
#define pi 3.14
float zc(int n)  ///计算圆周长
{ float l;
  l=2*pi*n;      //计算圆周长 l
  return l;
}
float mj (int n)  //计算圆面积
{ float s;
  s=n*n*pi;      //计算圆面积 s
  return s ;
}
float tj(int n)  //计算矩形的面积功能
{ float v;
  v=4.0/3*pi*n*n*n;  //计算圆球体积 v
  return v;
}
int main ()
{ float r;
  printf("请输入圆半径 r: ");
  scanf("%f",&r);          //要求输入圆半径 r
  printf("圆周长为:      l=%6.2f\n", zc(r));  //输出圆周长 l，保留 2 位小数
  printf("圆面积为:      s=%6.2f\n", mj (r));  //输出圆面积 s，保留 2 位小数
  printf("圆球体积为:    v=%6.2f\n", tj(r));  //输出圆球体积 v，保留 2 位小数
  return 0;
}

```

程序运行结果如图 7-6 所示。

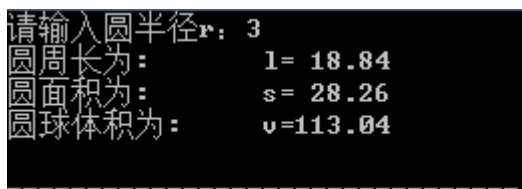


图 7-6 例题 7.6 运行结果

【程序 7.7】用非递归方法编写一个求 $n!$ 的函数，在主函数中实现计算：
 $1!+2!+3!+4!+5!+\dots+10!$

```
#include <stdio.h>
int main()
{ double fac(int n);
  int n;
  double s=0;
  for(n=1;n<=10;n++)
    s=s+fac(n);
  printf("1!+2!+3!+4!+5!+\dots+10!=%lf\n", s);
  return 0;
}

double fac(int n)
{
  double f=1;
  int i;
  for(i=1;i<=n;i++)
    f=f*i;
  return(f);
}
```

程序运行结果如图 7-7 所示。

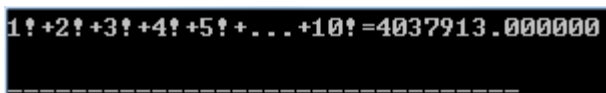


图 7-7 例题 7.7 运行结果

7.3 实验 11：函数程序设计（2）

7.3.1 实验目的

1. 掌握函数的嵌套调用和递归调用的方法。
2. 理解嵌套调用和递归调用之间的相同和不同点。
3. 了解全局变量和局部变量、动态变量、静态变量的概念和使用方法。

7.3.2 实验内容

1. 程序示例

【程序 7.8】上机运行程序，理解函数的值传递，验证运行结果与自己分析的结果是否

相符，记录下运行结果。

```
#include <stdio.h>
int main( )
{   int max2(int a,int b); //对 max2 函数声明
    int a,b,c,d,max;
    printf("请输入 4 个整数:");
    scanf("%d%d%d%d",&a,&b,&c,&d); //输入 4 个整数
    max=max2(max2(max2(a,b),c),d); //调用函数 max2, 求出 4 个数中最大值
    printf("max=%d \n",max);
    return 0;
}

int max2(int a,int b) //定义函数 max2, 求 2 个数中的最大值
{   if(a>=b)
        return a;
    else
        return b;
}
```

【程序 7.9】上机运行程序，理解函数的嵌套调用，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```
#include<stdio.h>
int f(int x,int y)
{ return ((y-x)*x);}
int main( )
{int a=3,b=4,c=5,d;
 d=f(f(a,b),f(a,c));
 printf("d=%d\n",d);
 return 0;
}
```

【程序 7.10】上机运行程序，理解函数的递归调用，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```
#include <stdio.h>
fun(int x)
{if(x/2>0) fun(x/2); printf("%d ",x); }
int main( )
{   fun(6);
    printf("\n");
    return 0;
}
```

2. 程序阅读

【程序 7.11】上机运行程序，练习静态变量的使用，验证运行结果与自己分析的结果

是否相符，并分析如果是动态变量结果会如何改变。

```
#include <stdio.h>
int fun()
{
    static int x=1;
    x*=2;
    return x;
}
int main( )
{
    int i,s=1;
    for(i=1;i<=3;i++)
        s*=fun();
    printf("%d\n",s);
    return 0;
}
```

【程序 7.12】上机运行程序，练习函数的递归调用，验证运行结果与预习结果是否相符，并分析该程序有什么不妥的地方。

```
#include <stdio.h>
int fac(int n)    //定义递归函数 fac
{
    int f;
    if(n<0)
        printf("n<0,data error!");
    else if(n==0||n==1)    //表示 1!=1 或者 0!=1
        f=1;
    else f=fac(n-1)*n;    //表示 n!=n * (n-1)!
    return(f);
}
int main( )
{
    int n; int y;
    printf("请输入数字，求阶乘:");
    scanf("%d",&n);
    y=fac(n);    //调用 fac 函数，求 n!
    printf("%d!=%d\n",n,y);
    return 0;
}
```

3. 程序设计

【程序 7.13】用递归方法编写一个求 $n!$ 的函数，在主函数中实现计算：

$1!+2!+3!+4!+5!+\dots+10!$

【程序 7.14】用函数嵌套和递归方法编写函数，在主函数中实现求两个整数的最大公约数和最小公倍数。

7.3.3 实验指导

1. 程序示例

【程序 7.8】上机运行程序，理解函数的值传递，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
请输入4个整数:6 9 10 3
max=10
```

图 7-8 例题 7.8 运行结果

【程序 7.9】上机运行程序，理解函数的嵌套调用，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```
d=1
```

图 7-9 例题 7.9 运行结果

【程序 7.10】上机运行程序，理解函数的递归调用，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```
1 3 6
```

图 7-11 例题 7.10 运行结果

2. 程序阅读

【程序 7.11】上机运行程序，练习静态变量的使用，验证运行结果与自己分析的结果是否相符，并分析如果是动态变量结果会如何改变。

```
64
```

图 7-12 例题 7.11 运行结果（1）

```
8
```

图 7-13 例题 7.11 运行结果（2）

【程序 7.12】上机运行程序，练习函数的递归调用，验证运行结果与预习结果是否相符，并分析该程序有什么不妥的地方。

①若求 $0 \sim 31$ 的阶乘，此程序才验证正确，程序运行结果如图 7-14 所示。

```
请输入数字，求阶乘:5
5!=120
Press any key to continue
```

图 7-14 例题 7.12 运行结果（1）

②若求大于等于 32 的阶乘，则会越界溢出，程序运行结果如图 7-15 所示。

```
请输入数字，求阶乘:32
32!=-2147483648
Press any key to continue
```

图 7-15 例题 7.12 运行结果（2）

③若将函数 fac 的返回值，变量 f，变量 y，设计成双精度实型 double。

```
请输入数字，求阶乘:32
32!=2631308369336935200000000000000000.000000
Press any key to continue
```

图 7-16 例题 7.12 运行结果（3）

④若求负数的阶乘，则会报错，不输出错误结果，程序运行结果如图 7-17 所示。

```
请输入数字，求阶乘:-7
n<0,data error!
Press any key to continue
```

图 7-17 例题 7.12 运行结果（4）

5. 程序设计

【程序 7.13】用递归方法编写一个求 $n!$ 的函数，在主函数中实现计算：

$1!+2!+3!+4!+5!+\dots+10!$

```
#include <stdio.h>
int main()
{ double fac(int n);
  int i;
  double s=0;
  for(i=1;i<=10;i++)
    s=s+fac(i);
  printf("1!+2!+3!+4!+5!+\dots+10!=%lf\n", s);
  return 0;
}

double fac(int n)
{ double f;
  if(n==0||n==1)
    f=1;
  else f=fac(n-1)*n;
  return(f);
}
```

程序运行结果如图 7-18 所示。

```
1!+2!+3!+4!+5!+\dots+10!=4037913.000000
```

图 7-18 例题 7.13 运行结果

【程序 7.14】用函数嵌套和递归方法编写函数，在主函数中实现求两个整数的最大公约数和最小公倍数。

方法一：函数嵌套调用

```
#include <stdio.h>
int divisor (int a,int b) /*自定义函数求两数的最大公约数*/
{ int temp; /*定义整型变量*/
```

```

    if(a<b)                /*通过比较求出两个数中的最大值和最小值*/
    { temp=a;a=b;b=temp;} /*设置中间变量进行两数交换*/
    while(b!=0)            /*通过循环求两数的余数，直到余数为 0*/
    { temp=a%b; a =b; b=temp; } /*变量数值交换*/
    return (a);            /*返回最大公约数到调用函数处*/
}

int multiple (int a,int b) /*自定义函数求两数的最小公倍数*/
{ int divisor (int a,int b); /*自定义函数返回值类型*/
  int temp;
  temp=divisor(a,b); /*再次调用自定义函数，求出最大公约数*/
  return (a*b/temp); /*返回最小公倍数到主调函数处进行输出*/
}

main()
{ int m,n,t1,t2;          /*定义整型变量*/
  printf("请输入两个整数:"); /*提示输入两个整数*/
  scanf("%d%d",&m,&n);      /*通过终端输入两个数*/
  t1=divisor(m,n);         /*自定义主调函数*/
  t2=multiple(m,n);        /*自定义主调函数*/
  printf("最大公约数= %d\n",t1); /*输出最大公约数*/
  printf("最小公倍数= %d\n", t2); /*输出最小公倍数*/
}

```

方法二：函数递归调用

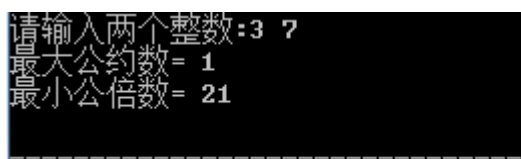
```

int gcd (int a,int b)
{ if(a%b==0)
    return b;
  else
    return gcd(b,a%b);
}

#include <stdio.h>
main()
{ int m,n,t1;
  printf("请输入两个整数:");
  scanf("%d%d",&m,&n);
  t1=gcd(m,n);
  printf("最大公约数= %d\n",t1);/*最大公约数*/
  printf("最小公倍数= %d\n",m*n/t1);/*最小公倍数*/
}

```

程序运行结果如图 7-19 所示。



```

请输入两个整数:3 ?
最大公约数= 1
最小公倍数= 21

```

图 7-19 例题 7.14 运行结果

第 8 章 C 语言的精华-指针

8.2 实验 12：指针程序设计（1）

8.2.1 实验目的

1. 理解指针与指针变量的概念。
2. 能正确使用数组的指针和指向数组的指针变量。
3. 能正确使用字符串的指针和指向字符串的指针变量。

8.2.2 实验内容

1. 程序示例

【程序 8.1】上机运行程序，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```
#include <stdio.h>
int main( )
{ int x=9, y=5, c, *a=&x, *b=&y, *w=&c;
  *a=*a+*a;
  *w=*a;
  *a=*b;
  *b=*w;
  printf("%d, %d\n", x, y);
  return 0;
}
```

【程序 8.2】上机运行程序，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```
#include <stdio.h>
int main( )
{ char a[]="programming", b[]="language" ;
  char *p1, *p2;
  int i;
  p1=a; p2=b;
  for(i=0; i<7; i++)
  if(*(p1+i)==*(p2+i))
  printf("%c", *(p1+i));
  printf("\n");
  return 0;
}
```

```
}
```

2. 程序阅读

【程序 8.3】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include <stdio.h>
int main( )
{ int a=28,b;
  char s[10],*p;
  p=s;
  do{b=a%16;
    if(b<10) *p=b+48;
    else     *p=b+55;
    p++;
    a=a/5;
  }while(a>0);
  *p='\0';
  puts(s);
  return 0;
}
```

3. 程序改错

【程序 8.4】函数实现的功能是将长整型 s 中的每一位上为偶数的数依次取出，构成一个新数放在 p 中，高位仍在高位，低位仍在低位。例如，当 s 中的数为 87653142 时，p 中的数为：8642。请改正程序中的 5 处错误，使它能得出正确的结果。

```
第 1 句: #include <stdio.h>
第 2 句: int main( )
第 3 句: { long s,p,*t=p;
第 4 句:   int d;
第 5 句:   long s1=1;
第 6 句:   printf("\nPlease enter s:");
第 7 句:   scanf("%ld",s);
第 8 句:   *t=0;
第 9 句:   while(s>0)
第 10 句: { d=s%10;
第 11 句:   if(d%2=0)
第 12 句:   { t=d*s1+*t;
第 13 句:     s1*=10;
第 14 句:   }
第 15 句:   s/=10;
第 16 句: }
第 17 句: printf("The result is: %ld\n",p);
```

第 18 句: return 0;
第 19 句: }

4. 程序完善

【程序 8.5】上机运行程序，利用指针实现两个字符串的连接，补充完善程序内容，并验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include <stdio.h>
#include <string.h>
int main( )
{ char str1[100],str2[100],*p,*q;
  printf("请输入两个字符串: \n");
  gets(str1);
  gets(str2);
  p=str1+strlen(str1);
  q=str2;
  while(____①____)
  { *p=*q;
    ____②____;
    ____③____;
  }
  ____④____;
  printf("连接后新串为: \n");
  puts(str1);
  return 0;
}
```

5. 程序设计

【程序 8.6】编写函数，输入一个八进制数，输出为十进制数。

【程序 8.7】编写函数，计算一个字符在一个字符串中出现的次数。

8.2.3 实验指导

1. 程序示例

【程序 8.1】上机运行程序，验证运行结果与自己分析的结果是否相符，并记录运行结果。



图 8-1 程序 8.1 运行结果

【程序 8.2】上机运行程序，验证运行结果与自己分析的结果是否相符，并记录运行结果。



图 8-2 程序 8.2 运行结果

2. 程序阅读

【程序 8.3】在阅读并理解程序思路的基础上，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include <stdio.h>
int main( )
{ int a=28, b;      //定义 2 个整型变量 a, b
  char s[10], *p;   //定义一个字符数组 s 和一个 char *指针 p
  p=s;              //将字符数组 s 首地址赋给指针 p
  do{ b=a%16;       //将 a 除以 16 的余数赋给 b
    if(b<10) *p=b+48; //当 b<10 时，将 b+48 的值赋给*p
    else     *p=b+55; //当 b>=10 时，将 b+55 的值赋给*p
    p++;
    a=a/5;         //a/5 整除的结果赋给 a
  }while(a>0);
  *p='\0';         //加字符串结束标志
  puts(s);         //输出字符串 s
  return 0;
}
```

程序运行结果如图 8-3 所示。



图 8-3 程序 8.3 运行结果

3. 程序改错

【程序 8.4】函数实现的功能是将长整型 s 中的每一位上为偶数的数依次取出，构成一个新数放在 p 中，高位仍在高位，低位仍在低位。例如，当 s 中的数为 87653142 时，p 中的数为：8642。请改正程序中的 5 处错误，使它能得出正确的结果。

- ①第 3 句：long s, p, *t=p; 改成： long s, p, *t=&p;
- ②第 7 句：scanf("%ld", s); 改成： scanf("%ld", &s);
- ③第 11 句：if(d%2=0) 改成： if(d%2==0)
- ④第 12 句： t=d*s1+*t; 改成： *t=d*s1+*t;
- ⑤第 15 句： s/=10; 改成： s/=10;

程序运行结果如图 8-4 所示。

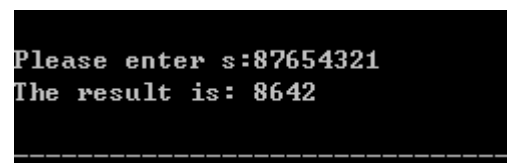


图 8-4 程序 8.4 运行结果

4. 程序完善

【程序 8.5】上机运行程序，利用指针实现两个字符串的连接，补充程序完善内容，并验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include <stdio.h>
#include <string.h>
int main( )
{ char str1[100],str2[100],*p,*q;
  printf("请输入两个字符串: \n");
  gets(str1);
  gets(str2);
  p=str1+strlen(str1); /*p 指向第一个串的末尾*/
  q=str2; /*q 指向第二个串的首部*/
  while(*q!='\0') /*如果第二个串未结束, 继续执行*/
  { *p=*q;
    p++; /*指针p 向后移动*/
    q++; /*指针q 向后移动*/
  }
  *p='\0'; /*串末尾加上结束标志*/
  printf("连接后新串为: \n");
  puts(str1);
  return 0;
}
```

程序运行结果如图 8-5 所示。

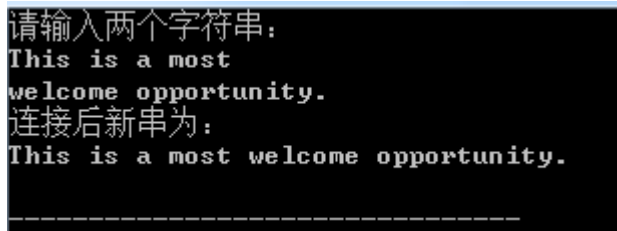


图 8-5 程序 8.5 运行结果

5. 程序设计

【程序 8.6】编写函数，输入一个八进制数，输出为十进制数。

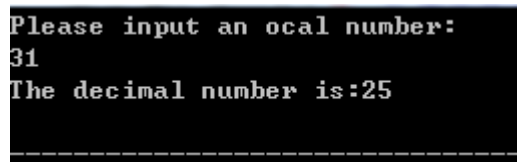
```
#include <stdio.h>
int main( )
{ char *p,s[6];int n;
  p=s;
  printf("Please input an ocal number: \n");
  gets(p);
  n=0;
  while(*(p)!='\0')
```

```

    { n=n*8+*p-'0';
      p++;
    }
    printf("The decimal number is:%d\n",n);
    return 0;
}

```

程序运行结果如图 8-6 所示。



```

Please input an ocal number:
31
The decimal number is:25

```

图 8-6 程序 8.6 运行结果

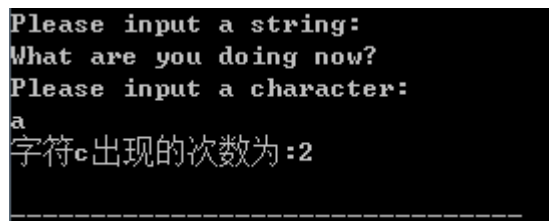
【程序 8.7】编写函数，计算一个字符在一个字符串中出现的次数。
程序代码如下：

```

#include<stdio.h>
#include<string.h>
int main( )
{ char str[100];
  char *s=str, c;
  int count=0;
  printf("Please input a string:\n");
  gets(s);
  printf("Please input a character:\n");
  scanf("%c",&c);
  while(*s)
  { if(*s==c)
    { count++;
      s++;
    }
  }
  printf("字符 c 出现的次数为:%d\n",count);
  return 0;
}

```

程序运行结果如图 8-7 所示。



```

Please input a string:
What are you doing now?
Please input a character:
a
字符c出现的次数为:2

```

图 8-7 程序 8.7 运行结果

8.3 实验 13：指针程序设计（2）

8.3.1 实验目的

1. 学会用指针作为函数参数的方法。
2. 学会使用函数的指针和指向函数的指针变量。

8.3.2 实验内容

1. 程序示例

【程序 8.8】分析下列程序的运行结果。程序代码如下，分析该程序的运行结果。

```
#include<stdio.h>
void swap(int *x,int *y)
{ int t;
  t=*x;*x=*y;*y=t;
}
int main( )
{ int a[9]={1,2,3,4,5,6,7,8,9}, i, *p, *q;
  p=a;q=&a[8];
  while(*p!=*q)
    { swap(p,q); p++; q--; }
  for(i=0;i<9;i++)
    printf("%d",a[i]);
  return 0;
}
```

2. 程序阅读

【程序 8.9】分析下列程序的运行结果。程序代码如下，分析该程序的运行结果。

```
#include<stdio.h>
#include<string>
int main( )
{ char str[]="cdalb";
  void abc(char *p);
  abc(str);
  puts(str);
  return 0;
}

void abc(char *p)
{ int i,j;
  for(i=0,j=0;*(p+i)!='\0';i++)
    if(*(p+i)>='d')
    {
```

```

        *(p+j)=*(p+i);
        j++;
    }
    *(p+j)='\0';
}

```

3. 程序完善

【程序 8.10】下面程序的功能是将一个整数字符串转换为一个整数，如“-3267”转换为-3267。补充程序完善内容，并验证运行结果与自己分析的结果是否相符，记录下运行结果。

```

#include<stdio.h>
#include<string.h>
int main( )
{ char s[6];
  int n;
  int chnum(char *p);
  printf("Please input a string:\n");
  gets(s);
  if(*s=='-') n=-chnum(s+1);
  else n=chnum(s);
  printf("%d\n",n);
  return 0;
}

```

```

int chnum(char *p)
{ int num=0,k,len,j;
  len=strlen(p);
  for( ; ① ; p++)
  { k=②;
    len--;
    j=len;
    while(③)
    { k=k*10;j--;}
    num=num+k;
  }
  return num;
}

```

【程序 8.11】下面程序的功能是统计子串 substr 在母串 str 中出现的次数。补充完善程序内容，并验证运行结果与自己分析的结果是否相符，记录下运行结果。

```

#include<stdio.h>
int count(char *str,char *substr)
{ int i, j, k, num=0;
  for(i=0; ① ;i++)

```

```

for(____②____, k=0; substr[k]==str[j]; k++, j++)
    if(substr[____③____]=='\0' )
        { num++;break; }
return num;
}
int main( )
{ char str[80], substr[80];
  int n;
  printf("Please input a string:\n");
  gets(str);
  printf("Please input a substring:\n");
  gets(substr);
  printf("The result is %d\n", count(str, substr));
  return 0;
}

```

4. 程序设计

【程序 8.12】编写函数实现，计算字符串的长度（包括 ‘\0’ 所占的长度）。

【程序 8.13】编程实现从键盘输入一个大于 1 的正整数 n, 计算 $1+2+3+\dots+n$, 要求用函数指针完成。

8.3.3 实验指导

1. 程序示例

【程序 8.8】分析下列程序的运行结果。程序代码如下，分析该程序的运行结果。

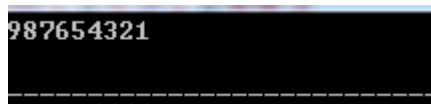


图8-8 程序8.8的运行结果

2. 程序阅读



图8-9 程序8.9的运行结果

3. 程序完善

【程序 8.10】下面程序的功能是将一个整数字符串转换为一个整数，如 “-3267” 转换为 -3267。补充完善程序内容，并验证运行结果与自己分析的结果是否相符，记录下运行结果。

- ①: *p!='\0'
 ②: *p-'0'
 ③: j!=0

调试该程序，运行结果如图 8-10 所示。

```

Please input a string:
3267
3267
-----

```

图8-10 程序8.10的运行结果

【程序 8.11】下面程序的功能是统计子串 substr 在母串 str 中出现的次数。补充完善程序内容，并验证运行结果与自己分析的结果是否相符，记录下运行结果。

①: str[i]!='\0'

②: j=i

③: k+1

调试该程序，运行结果如图 8-11 所示。

```

Please input a string:
If you tried,you could do the work better.
Please input a substring:
you
The result is 2
-----

```

图8-11 程序8.11的运行结果

4. 程序设计

【程序 8.12】编写函数实现，计算字符串的长度（包括 ‘\0’ 所占的长度）。

```

#include<stdio.h>
int myLength(char *ps)
{ int length=0;
  while(*ps)
  { length++;
    ps++;
  }
  length++;
  return length;
}
int main( )
{ char str[100],*p;
  int length=0;
  printf("请输入100字符以内的字符串: \n");
  gets(str);
  p=str; /*p 指向字符串的开始*/
  length=myLength(p);
  printf("字符串长度为: %d\n",length);
  return 0;
}

```

调试该程序，运行结果如图 8-12 所示。

```

请输入100字符以内的字符串:
How can one stand by and do noing?
字符串长度为: 35
-----

```

图8-12 程序8.12的运行结果

【程序 8.13】编程实现从键盘输入一个大于 1 的正整数 n, 计算 $1+2+3+\dots+n$, 要求用函数指针完成。

```

#include<stdio.h>
long fun(int n);
int main( )
{ int n;
  long (*p)(int), s;
  printf("input n:\n");
  scanf("%d", &n);
  p=fun;
  s=(*p)(n);
  printf("sum=%ld", s);
  return 0;
}
long fun(int n)
{ int i;
  long sum=0;
  for(i=1; i<=n; i++)
    sum=sum+i;
  return sum;
}

```

程序运行结果如图 8-13 所示。

```

input n:
100
sum=5050
-----

```

图8-13 程序8.13运行结果

第 9 章 用户自己建立数据类型

9.2 实验 14：结构体程序设计

9.2.1 实验目的

1. 熟练掌握结构体类型变量的定义和使用方法。
2. 熟练掌握结构体数组的定义和使用方法。

9.2.2 实验内容

1. 程序示例

【程序 9.1】在阅读并理解程序思路的基础上，上机运行程序，补充注释内容，验证运行结果与自己分析的结果是否相符，记录下运行结果。该程序定义一个结构体变量，其成员项包括学号、姓名、年龄、分数；通过键盘输入所需的具体数据，然后输出。

```
#include<stdio.h>
#include<string.h>
struct student
{ int num;
  char name[20];
  char sex;
  int age;
  float score;
};
int main( )
{ struct student stu1;
  printf("请输入学号:");
  scanf("%d",&stu1.num);
  printf("请输入姓名:");
  scanf("%s",stu1.name);
  printf("请输入年龄:");
  scanf("%d",&stu1.age);
  printf("请输入分数:");
  scanf("%f",&stu1.score);
  printf("输出学号、姓名、年龄、分数为:\n");
  printf("%d,%s,%d,%.2f\n",stu1.num,stu1.name,stu1.age,stu1.score);
  printf("\n");
  return 0;
}
```

【程序 9.2】在阅读并理解程序思路的基础上，上机运行程序，补充注释内容，验证运行结果与自己分析的结果是否相符，记录下运行结果。该程序利用结构体数组编制程序，班级 10 名同学推举三名同学竞选班长，统计候选人选票。

```
#include <stdio.h>
#include <string.h>
struct person
{ char name[20];
  int count;
} leader[3]={"lilei",0,"hanmeimei",0,"wangtao",0};
int main( )
{   int i,j;
    char leader_name[20];
    for(i=1;i<=10;i++)
    {   scanf("%s", leader_name);
        for(j=0;j<3;j++)
            if(strcmp(leader_name, leader[j].name)==0)
                leader[j].count++;
    }
    printf("\n");
    for(i=0;i<3;i++)
        printf("%5s:%d\n", leader[i].name, leader[i].count);
    return 0;
}
```

【程序 9.3】在阅读并理解程序思路的基础上，上机运行程序，补充注释内容，验证运行结果与自己分析的结果是否相符，记录下运行结果。该程序建立结构体数组，并输入 3 名学生的序号、姓名、性别基本信息，并输出，要求使用指针变量来指向结构体数组及其元素。

```
#include <stdio.h>
struct student
{ int num;
  char name[20];
  char sex;
  int age;
} stu[3]={ {10101,"Li Lin",'M',18},
           {10102,"Zhang Fun",'M',19},
           {10104,"Wang Min",'F',20} };
int main( )
{   struct student *p;
    for(p=stu;p<stu+3;p++)
```

```

    printf("%d %s %c %d\n", p->num, p->name, p->sex, p->age);
    return 0;
}

```

2. 程序纠错

【程序 9.4】程序中有 2 处错误，上机调试程序，修改有错误的语句行，并输出正确的运行结果。

```

第 1 句: #include<stdio.h>
第 2 句: struct employee
第 3 句: { char name[10];
第 4 句: float salary;
第 5 句: };
第 6 句: int main()
第 7 句: { struct employee a;
第 8 句: scanf("%s", a.name);
第 9 句: printf("请输入工资:");
第 10 句: scanf("%f", a.salary);
第 11 句: printf("输出姓名, 工资为:\n");
第 12 句: printf("%s, %.2f\n", a.name, a.salary);
第 13 句: return 0;
第 14 句: }

```

【程序 9.5】程序中有 2 处错误，上机调试程序，修改有错误的语句行，并输出正确的运行结果。

```

第 1 句: #include<stdio.h>
第 2 句: int main()
第 3 句: { struct student ;
第 4 句: { long int num;
第 5 句: char name[30];
第 6 句: char sex;
第 7 句: } a={202002, "Zhang Shan", "M"};
第 8 句: printf("NO. :%ld\nname:%s\nsex:%c\n", a.num, a.name, a.sex);
第 9 句: return 0;
第 10 句: }

```

3. 程序设计

【程序 9.6】编写程序，建立 3 个学生的结构记录，包括学号、姓名、成绩，然后计算其平均成绩，并输出。

【程序 9.7】试利用指向结构体的指针编制程序，输入 3 个学生的学号以及语文、数学、英语成绩，然后计算其平均成绩，并输出成绩表。

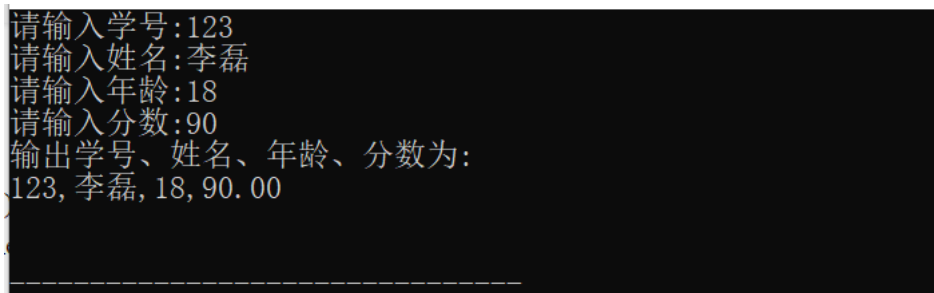
9.2.3 实验指导

1. 程序示例

【程序 9.1】在阅读并理解程序思路的基础上，上机运行程序，补充注释内容，验证运行结果与自己分析的结果是否相符，记录下运行结果。该程序定义一个结构体变量，其成员项包括学号、姓名、年龄、分数；通过键盘输入所需的具体数据，然后输出。

```
#include<stdio.h>
#include<string.h>
struct student //声明结构体类型
{ int num; //以下五行为结构体成员
  char name[20];
  char sex;
  int age;
  float score;
};
int main( )
{ struct student stu1; //定义结构体变量 stu1
  printf("请输入学号:");
  scanf("%d",&stu1.num); //根据提示输入该变量对应的成员信息
  printf("请输入姓名:");
  scanf("%s",stu1.name);
  printf("请输入年龄:");
  scanf("%d",&stu1.age);
  printf("请输入分数:");
  scanf("%f",&stu1.score);
  printf("输出学号、姓名、年龄、分数为:\n");
  printf("%d,%s,%d,%.2f\n",stu1.num,stu1.name,stu1.age,stu1.score); /*输出
stu1 的信息*/
  printf("\n");
  return 0;
}
```

程序运行结果如图 9-1 所示。



```
请输入学号:123
请输入姓名:李磊
请输入年龄:18
请输入分数:90
输出学号、姓名、年龄、分数为:
123, 李磊, 18, 90.00
```

图 9-1 例题 9.1 运行结果

【程序 9.2】在阅读并理解程序思路的基础上，上机运行程序，补充注释内容，验证运行结果与自己分析的结果是否相符，记录下运行结果。该程序利用结构体数组编制程序，班级 10 名同学推举三名同学竞选班长，统计候选人选票。

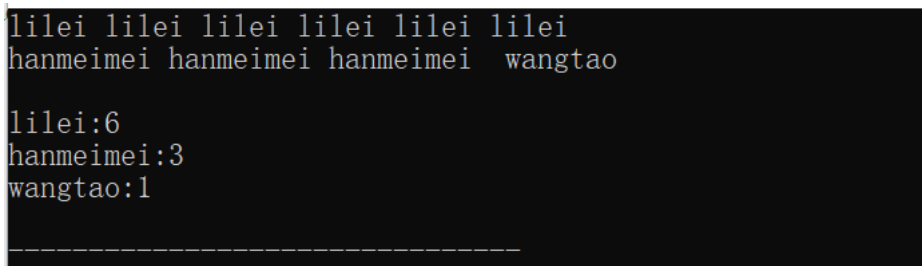
```
#include <stdio.h>
```

```

#include <string.h>
struct person                                //声明结构体类型
{
    char name[20];
    int count;
} leader[3]={"lilei",0,"hanmeimei",0,"wangtao",0};
/*定义结构体数组 leader 并初始化*/
int main( )
{
    int i,j;
    char leader_name[20];
    for(i=1;i<=10;i++)                      //设定循环条件，投票人数 10 人
    {
        scanf("%s",leader_name);           //输入得票人姓名
        for(j=0;j<3;j++)                   //设定循环条件，确定被计票人
            if(strcmp(leader_name,leader[j].name)==0)//比对得票是否是该候选人
                leader[j].count++;          //统计该候选人的票数
    }
    printf("\n");
    for(i=0;i<3;i++)
        printf("%5s:%d\n",leader[i].name,leader[i].count); //输出候选人的票数
    return 0;
}

```

程序运行结果如图 9-2 所示。



```

lilei lilei lilei lilei lilei lilei
hanmeimei hanmeimei hanmeimei wangtao

lilei:6
hanmeimei:3
wangtao:1

```

图 9-2 例题 9.2 的运行结果

【程序 9.3】在阅读并理解程序思路的基础上，上机运行程序，补充注释内容，验证运行结果与自己分析的结果是否相符，记录下运行结果。该程序建立结构体数组，并输入 3 名学生的序号、姓名、性别基本信息，并输出，要求使用指针变量来指向结构体数组及其元素。

```

#include <stdio.h>
struct student                                //声明结构体类型
{
    int num;
    char name[20];
    char sex;
    int age;
} stu[3]={10101,"Li Lin",'M',18},
          {10102,"Zhang Fun",'M',19},
          {10104,"Wang Min",'F',20}};      //定义结构体数组并初始化
int main( )

```

```

{ struct student *p;           //定义指向 struct student 结构体变量得指针变量
  for(p=stu;p<stu+3;p++)
    printf("%d %s %c %d\n", p->num, p->name, p->sex, p->age); //输出结果
  return 0;
}

```

程序运行结果如图 9-3 所示。

```

10101 Li Lin M 18
10102 Zhang Fun M 19
10104 Wang Min F 20
-----

```

图 9-3 例题 9.3 的运行结果

2. 程序改错

```

#include<stdio.h>
struct employee
{ char name[10];
  float salary;
};           //声明结构体类型
int main( )
{ struct employee a;      //定义结构体变量 a
  scanf("%s", a.name);    //输入 a 的姓名
  printf("请输入工资:");
  scanf("%f",&a.salary); //输入 a 的工资
  printf("输出姓名, 工资为:\n");
  printf("%s, %.2f\n", a.name, a.salary); //输出 a 的姓名和工资
  return 0;
}

```

程序运行结果如图9-4所示。

```

李华
请输入工资:7896
输出姓名, 工资为:
李华, 7896.00
-----

```

图9-4 例题9.4运行结果

【程序 9.5】程序中有 2 处错误，上机调试程序，修改有错误的语句行，并输出正确的运行结果。

```

#include <stdio.h>
int main()
{ struct student           //声明结构体类型 struct student
  { long int num;          //以下 3 行为结构体成员
    char name[30];
    char sex;
  }a={202002,"Zhang Shan",'M'}; //定义变量 a 并初始化
}

```

```
printf("NO. :%ld\nname:%s\nsex:%c\n", a. num, a. name, a. sex); //输出 a 的姓名和工资
return 0;
}
```

程序运行结果如图9-5所示。

```
NO. :202002
name:Zhang Shan
sex:M
```

图9-5 例题9.5运行结果

3. 程序设计

【程序 9.6】编写程序，建立 3 个学生的结构记录，包括学号、姓名、成绩，然后计算其平均成绩，并输出。

```
#include <stdio.h>
struct student
{
    int num;
    char name[20];
    int score;
};
int main( )
{
    int i;double sum=0;
    struct student s[3];
    printf("依次输入三个学生的信息:\n");
    for(i=0;i<3;i++)
    {
        scanf("%d%s%d",&s[i].num,&s[i].name,&s[i].score);
        sum+=s[i].score;
    }
    printf("平均成绩如下%f:",sum/3);
    printf("\n");
    return 0;
}
```

程序运行结果如图 9-6 所示。

```
依次输入三个学生的信息:
10 lilei 90
11 hanmei 92
12 wangtao 93
平均成绩如下91.666667:
```

图 9-6 例题 9.6 的运行结果

【程序 9.7】试利用指向结构体的指针编制程序，输入 3 个学生的学号以及语文、数学、英语成绩，然后计算其平均成绩，并输出成绩表。

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

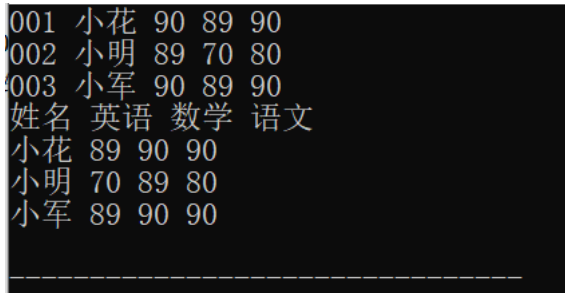
```

#include<stdlib.h>
struct student{  char num[10],name[10];
                 int  cn,eng,math;
                 }; //设置一个学生信息的结构体

int main( )
{ int i, j;
  FILE *fp; //设置文件指针
  struct student stu[3],temp;
  if((fp=fopen("c:\\data.txt","w"))==NULL)
  { printf("can not open");
    exit(0);
  } //确保文件成功打开
  for(i=0;i<3;i++)
  { scanf("%s%d%d%d",stu[i].num,stu[i].name,
    &stu[i].math,&stu[i].eng,&stu[i].cn);
  }
  printf("姓名 英语 数学 语文 \n");
  for(i=0;i<3;i++)
  { printf("%s %d %d %d\n",stu[i].name,stu[i].eng,
    stu[i].math,stu[i].cn);
    fprintf(fp,"%s %d %d %d\n",stu[i].name,stu[i].eng,
    stu[i].math,stu[i].cn);
  }
  return 0;
}

```

程序运行结果如图 9-7 所示。



```

001 小花 90 89 90
002 小明 89 70 80
003 小军 90 89 90
姓名 英语 数学 语文
小花 89 90 90
小明 70 89 80
小军 89 90 90
-----

```

图 9-7 例题 9.7 的运行结果

第 10 章 编译预处理

10.2 实验 15：编译预处理

10.2.1 实验目的

1. 掌握带参数的宏定义与不带参数的宏定义的使用。
2. 掌握文件包含的使用。
3. 了解条件编译。

10.2.2 实验内容

1. 程序示例

【程序 10.1】 将宏定义应用在输出格式中，验证如下程序并记录输出结果。

```
#include <stdio.h>
#define PR printf
#define student "%d,%s,%.1f"
#define AnyKey PR("\nPress any key to quit!\n");
int main( )
{ int num=25;
  char name[]="张三";
  float score=98.0;
  PR(student,num,name,score);
  AnyKey
  return 0;
}
```

2. 程序改错

【程序 10.2】 如下程序有 1 处错误，修改如下程序，使输入值为 5 时，结果为 2。

```
第 1 句: #define MN(x) x*x
第 2 句: int main( )
第 3 句: { int n,m;
第 4 句:   printf("Input a number:");
第 5 句:   scanf("%d",&n);
第 6 句:   m=72/MN(n+1);
第 7 句:   printf("m=%d\n",m);
第 8 句:   return 0;
第 9 句: }
```

3. 程序阅读

【程序 10.3】 在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上

机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
#ifndef COMPUTER_A
#define COMPUTER_A
#endif
#ifndef COMPUTER_A
typedef unsigned int u32;
u32 a,b,c;
#else
typedef unsigned short int u16;
u16 a,b,c;
#endif
int main( )
{ scanf("a=%d,b=%d",&a,&b);
  c=a+b;
  printf("c=%d",c);
  return 0;
}
```

【程序 10.4】已知头文件 exp10_3.h 在 E:\headfiler 的文件夹下，上机运行程序，验证文件包含预处理结果是否正确，并记录运行结果。

4. 程序设计

【程序 10.5】采用条件编译，使给定的二个整数做加法或减法输出。

【程序 10.6】包含一个 exp10_61.c 判断是否构成三角形的文件，如果构成则求出三角形的面积，否则输出不构成三角形。

10.2.3 实验指导

1. 程序示例

【程序 10.1】将宏定义应用在输出格式中的，验证如下程序并记录输出结果。

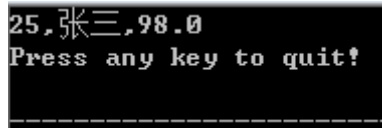


图 10-1 程序 10.1 运行结果

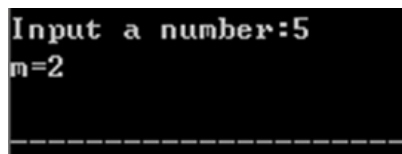
2. 程序改错

【程序 10.2】程序如下 1 处错误，修改如下程序，使输入值为 5 时，结果为 2。

```
#define MN(x) ((x)*(x))
int main( )
{int n,m;
printf("Input a number:");
scanf("%d",&n);
m=72/MN(n+1);
printf("m=%d\n",m);
return 0;
}
```

```
}
```

程序运行结果如图 10-2 所示。



```
Input a number:5
m=2
```

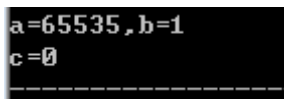
图 10-2 程序 10.2 运行结果

3. 程序阅读

【程序验证 10.3】在阅读并理解程序思路的基础上，说明程序功能，补充注释内容，并上机运行程序，验证运行结果与自己分析的结果是否相符，记录下运行结果。

```
#include<stdio.h>
#ifndef COMPUTER_A
#define COMPUTER_A
#endif
#ifndef COMPUTER_A
    typedef unsigned int u32; //类型定义 32 位无符号数
    u32 a,b,c;
#else
    typedef unsigned short int u16;//类型定义 16 位的无符号整数
    u16 a,b,c;
#endif
int main( )
{
    scanf("a=%d,b=%d",&a,&b);
    c=a+b;
    printf("c=%d",c); //c 结果为 16 位无符号整数
    return 0;
}
```

程序运行结果如图 10-3 所示。



```
a=65535,b=1
c=0
```

图 10-3 程序 10.3 运行结果

【程序 10.4】已知头文件 exp10_3.h 在 E:\headfiler 的文件夹下，上机运行程序，验证文件包含预处理结果是否正确，并记录运行结果。

```
#include<stdio.h>
#include"E:\headfiler\exp10_3.h" //包含头文件
int main( )
{
    int a,b,c;
    printf("请输入两个数: ");
    scanf("%d,%d",&a,&b);
    c=fmax(a,b); //调用头文件中的函数
    printf("max= %d\n",c);
    return 0;
}
```

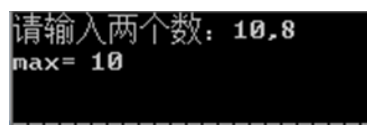
```
}
```

头文件

exp10_3.h 内容如下:

```
int fmax(int a,int b);  
int fmax(int a,int b)  
{   int max;  
    if(a>b) max=a;  
    else max=b;  
    return max;  
}
```

程序运行结果如图 10-4 所示。



```
请输入两个数: 10,8  
max= 10
```

图 10-4 程序 10.4 运行结果

4. 程序设计

【程序 10.5】 采用条件编译，使给定的二个整数做加法或减法输出。

```
#include<stdio.h>  
#define ADD  
int main( )  
{   int a,b,c;  
    printf("请输入两个数: ");  
    scanf("%d,%d",&a,&b);  
    #ifdef ADD  
        c=a+b;  
        printf("c= %d\n",c);  
    #else  
        c=a-b;           //未被编译  
        printf("c=%d\n",c); //未被编译  
    #endif  
    return 0;  
}
```

程序运行结果如图 10-5 所示。



```
请输入两个数: 10,8  
c=18
```

图 10-5 程序 10.5 运行结果

【程序 10.6】 包含一个 exp10_61.cpp 判断是否构成三角形的文件，如果构成则求出三角形的面积，否则输出不构成三角形。

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

```

#include"E:\headfile\ex2\exp10_61.cpp" //包含其它文件
float area; //全局变量声明
int flag; //全局变量声明
void triangle();
int main( )
{
    triangle();
    if(flag==0)
        printf("不构成三角形\n");
    else
        printf("三角形的面积是: %f\n", area);
    return 0;
}

```

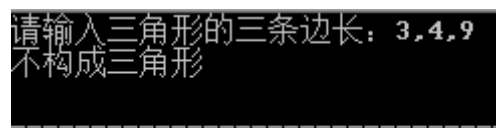
被包含程序: exp10_61.cpp

```

extern int flag; //外部变量声明
extern float area; //外部变量声明
void triangle()
{
    int a,b,c;
    double s;
    printf("请输入三角形的三条边长: ");
    scanf("%d,%d,%d",&a,&b,&c);
    s=(a+b+c)/2.0;
    if(a+b<=c || b+c<=a || a+c<=b)
        flag=0;
    else
    {
        flag=1;
        area=sqrt(s*(s-a)*(s-b)*(s-c));
    }
}

```

程序运行结果如图 10-6 和图 10-7 所示。

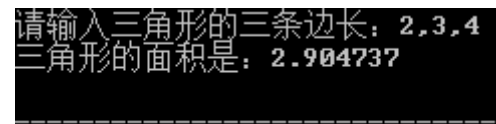


```

请输入三角形的三条边长: 3,4,9
不构成三角形

```

图 10-6 程序 10.6 运行结果 1



```

请输入三角形的三条边长: 2,3,4
三角形的面积是: 2.904737

```

图 10-7 程序 10.6 程序运行结果 2

第 11 章 文件

11.2 实验 16：文件

11.2.1 实验目的

1. 掌握文件以及缓冲文件系统、文件指针的概念。
2. 学会使用文件打开、关闭、读、写等文件操作函数。
3. 学会用缓冲文件系统对文件进行简单的操作。

11.2.2 实验内容

1. 程序阅读

【程序 11.1】上机运行程序，验证运行结果与预习结果是否相符，请给以下程序做分析和注释，并记录运行结果。

```
#include <stdio.h>
#include <stdlib.h>
int main( )
{   char ch;
    FILE *fp;
    if((fp=fopen("11-1.txt", "a+"))==NULL)
    {   printf("Cannot open the file! ");
        exit(0);
    }
    do
    {   ch=getchar();
        fputc(ch, fp);
    }while(ch!='\n');
    fclose(fp);
    return 0;
}
```

【程序 11.2】上机运行程序，验证运行结果与预习结果是否相符，请给以下程序做分析和注释，并记录运行结果。

```
#include <stdio.h>
int main( )
{   FILE *fp1,*fp2;
    char str[100];
    fp1=fopen("f1.txt", "r");
    fp2=fopen("f2.txt", "w");
```

```

        while(!feof(fp1))
        { if(fgets(str, 100, fp1))
            fputs(str, fp2);
        }
    fclose(fp1);
    fclose(fp2);
    return 0;
}

```

2. 程序完善

【程序 11.3】编写一个程序，实现将文件中的数据求和并写入文本文件尾。

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    ① _____ //定义文件指针 fp
    int n, sum;
    if((fp=fopen("f3.txt", "r+"))==NULL) //判断是否能打开 f3.txt 文件
    { printf("can not open file.\n");
      exit(0);
    }
    sum=0;
    while(② _____) sum=sum+n;
    //用循环结构通过 fscan 函数来遍历文件中的数据，每读一个数据就计算一次累加和
    ① _____ ; //将各数据的和 sum 写入 f1.txt 文件
    fclose(fp);
    return 0;
}

```

3. 程序设计

【程序 11.4】编写一个程序，实现从键盘输入一行字符串，将其中的小写字母全部转换成大写字母，然后输出到一个记事本文件 f4.txt 中保存，并检验文件中的内容。

【程序 11.5】编写一个程序，实现统计文本文件中各类字符个数。

11.2.3 实验指导

1. 程序阅读

【程序 11.1】上机运行程序，验证运行结果与预习结果是否相符，请给以下程序做分析和注释，并记录运行结果。

```

#include <stdio.h>
#include <stdlib.h>
int main( )
{
    char ch;
    FILE *fp; //定义文件指针

```

```

    if((fp=fopen("11-1.txt", "a+"))==NULL) //文件打开不成功，结束程序
    {   printf("Cannot open the file! ");
        exit(0); //关闭所有文件，终止正在执行的程序，表示正常退出
    }
do
{   ch=getchar();
    fputc(ch, fp);
}while(ch!='\n'); //从键盘上输入一行字符，并通过 fputc 函数成功写入
fclose(fp); //关闭 fp 所指向的文件
return 0;
}

```

程序运行结果如图 11-1 所示。

(1) 从键盘上输入前文件 (2) 从键盘上输入一行字符 (3) 从键盘上输入后文件

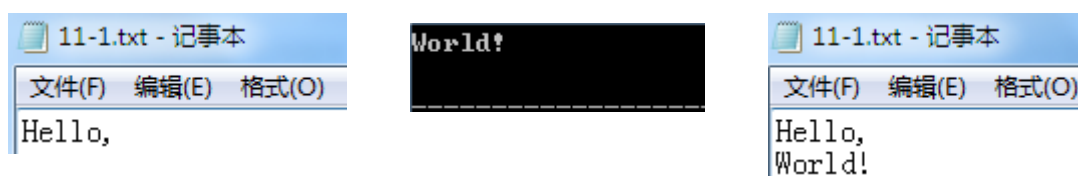


图 11-1 例题 11.1 运行结果

【程序 11.2】上机运行程序，验证运行结果与预习结果是否相符，请给以下程序做分析和注释，并记录运行结果。

```

#include <stdio.h>
int main( )
{ FILE *fp1,*fp2; //分别指向两个文件的指针 fp1 和 fp2
  char str[100]; //数组大小的设定与文件长度相关
  fp1=fopen("f1.txt", "r"); //为输入打开文本文件 f1.txt
  fp2=fopen("f2.txt", "w"); //为输出打开文本文件 f2.txt
  while(!feof(fp1)) //判断 fp1 指向的 f1.txt 文件是否结束
  { if(fgets(str, 100, fp1)) fputs(str, fp2);
    //从 fp1 所指向的文件读取字符串到 fp2 所指向的文件
  }
  fclose(fp1); //关闭 fp1 所指向的文件
  fclose(fp2); //关闭 fp2 所指向的文件
  return 0;
}

```

程序运行结果如图 11-2 所示。

(1) 复制前 f1 文件

(2) 复制后 f2 文件

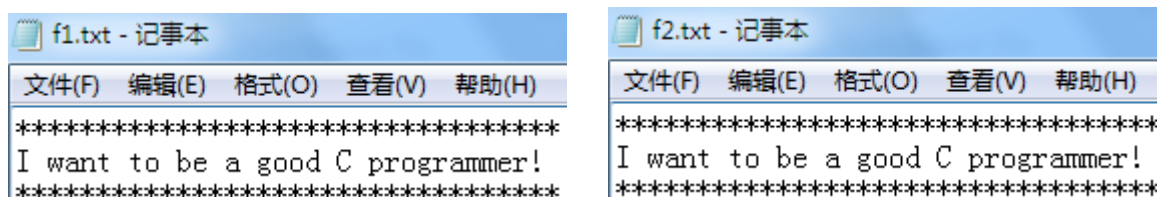


图 11-2 例题 11.2 运行结果

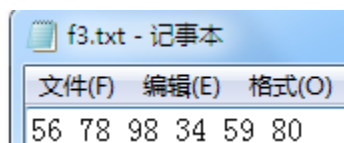
2. 程序完善

【程序 11.3】编写一个程序，实现将文件中的数据求和并写入文本文件尾。

```
#include <stdio.h>
#include <stdlib.h>
int main()
{ FILE *fp; //定义文件指针 fp
  int n, sum;
  if((fp=fopen("f3.txt", "r+"))==NULL) //判断是否能打开 f3.txt 文件
  { printf("can not open file.\n");
    exit(0);
  }
  sum=0;
  while(fscanf(fp, "%d", &n)!=EOF) sum=sum+n;
  //用循环结构通过 fscanf 函数来遍历文件中的数据，每读一个数据就计算一次累加和
  fprintf(fp, "\nsum=%d", sum); //将各数据的和 sum 写入 f1.txt 文件
  fclose(fp);
  return 0;
}
```

程序运行结果如图 11.3 所示。

(1) 求和前的文本文件 f3.txt



(2) 求和后的文本文件 f3.txt

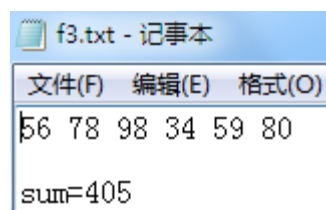


图 11-3 例题 11.3 运行结果

3. 程序设计

【程序 11.4】编写一个程序，实现从键盘输入一行字符串，将其中的小写字母全部转换成大写字母，然后输出到一个文本文件 f4.txt 中保存，并检验文件中的内容。

```
#include <stdio.h>
#include <stdlib.h>
int main()
{ FILE *fp;
  char str[100], filename[10];
  int i;
  if((fp=fopen("f4.txt", "w"))==NULL) //以写方式打开文本文件
  { printf("can not open file.\n");
    exit(0);
  }
  printf("Input a string: ");
```

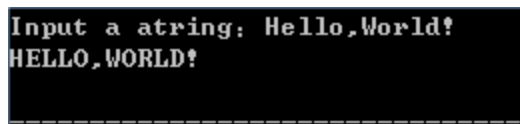
```

gets(str);                // 读入一行字符串
for (i=0;str[i];i++)      //处理该行中的每一个字符
{ if(str[i]>='a' &&str[i]<='z')    //若是小写字母
    str[i]=str[i]-32;          //将小写字母转换为大写字母
  fputc(str[i], fp);          //将转换后的字符写入文件
}
fclose(fp);               //关闭文件
fp=fopen("f4.txt", "r");   //以读方式打开文本文件
fgets(str, 100, fp);      //从文件中读入一行字符串
printf("%s\n", str);
fclose(fp);               //关闭文件
return 0;
}

```

程序运行结果如图 11.4 所示。

(1) 从键盘上输入一行字符并输出

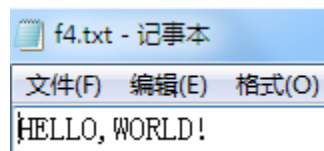


```

Input a atring: Hello,World!
HELLO, WORLD!

```

(2) 写入文本文件



```

f4.txt - 记事本
文件(F) 编辑(E) 格式(O)
HELLO, WORLD!

```

图 11-4 例题 11.4 运行结果

【程序设计 11.5】编写一个程序，实现统计文本文件中各类字符个数。

```

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int main( )
{ FILE *fp;    //定义文件指针 fp
  int a=0, d=0, b=0, o=0;
    //定义字符个数 a，数字个数 d，空格、跳格和回车个数 b、其它字符个数 o
  char c;
  if((fp=fopen("f5.txt", "r"))==NULL)
  //判断以读方式打开 f5.txt 文件是否成功
  { printf("can not open file.\n");
    exit(0);
  }
  while((c=fgetc(fp))!=EOF)
  //用循环结构来处理该文本文件中的每一个字符，判断并统计
  { if(isalpha(c)) a++;          //判断并统计字符个数 a
    else if(isdigit(c)) d++;     //判断并统计数字个数 d
      else if(c==' ' || c=='\t' || c=='\n') b++;
        //判断并统计空格、跳格和回车个数 b
      else o++;                 //判断并统计其它字符个数 o
  }
}

```

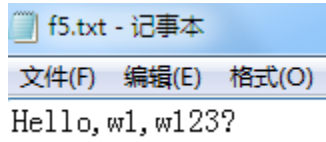
```

    }
    printf("字符 a=%d 数字 d=%d\n 空格、跳格、回车 b=%d 其它 o=%d\n", a, d, b, o);
    fclose(fp);
    return 0;
}

```

程序运行结果如图 11-5 所示。

(1) 文本文件 f5



(2) 统计文本文件中各类字符个数

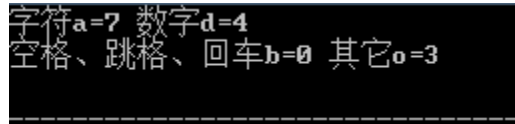


图 11-5 例题 11.5 运行结果

第 12 章 位运算

12.2 实验 17：位运算

12.2.1 实验目的

1. 掌握位运算的概念。
2. 掌握 6 种位运算的使用方法。

12.2.2 实验内容

1. 程序示例

【程序 12.1】上机运行程序，练习按位与&、按位或|、按位异或^的使用，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```
#include<stdio.h>
int main( )
{   int x,y;
    x=3; y=5;
    printf("x&y:%d\n", x&y);
    printf("3&5=%d, 3&4=%d\n", 3&5, 3&4);
    printf("3|5=%d, 3|4=%d\n", 3|5, 3|4);
    printf("3^5=%d, 3^4=%d\n", 3^5, 3^4);
    return 0;
}
```

【程序 12.2】上机运行程序，练习位运算的混合运算的使用，验证运行结果与自己分析的结果是否相符，并记录运行结果。

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

【程序 12.3】上机运行程序，练习位运算的混合运算的使用，验证运行结果与自己分析的结果是否相符，并记录运行结果。

```
#include <stdio.h>
int main( )
{   unsigned a, b, d;
    scanf("%o", &a);
```

```

        b=a<<1;
        d=b&a;
        printf("a:%o,%d\n",a,a);
        printf("b:%o,%d\n",b,b);
        printf("d:%o,%d\n",d,d);
        return 0;
}

```

2. 程序完善

【程序 12.4】阅读程序，若要使程序的运行结果为 248，应在下划线处填入什么，验证运行结果与分析结果是否相符，并记录运行结果。

```

#include <stdio.h>
int main( )
{ short c=124;
  c=___①___;
  printf("%d\n",c);
  return 0;
}

```

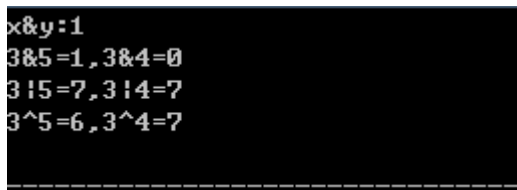
3. 程序设计

【程序设计 12.5】编写一个程序，实现将一个整数的低 4 位翻转。要求用八进制数输入和输出。

2.3.3 实验指导

1. 程序示例

【程序 12.1】上机运行程序，练习按位与&、按位或|、按位异或^的使用，验证运行结果与自己分析的结果是否相符，并记录运行结果。



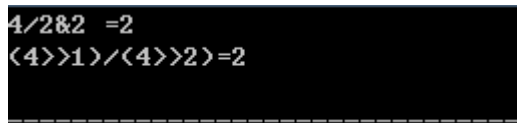
```

x&y:1
3&5=1,3&4=0
3|5=7,3|4=7
3^5=6,3^4=7

```

图 12-1 例题 12.1 运行结果

【程序 12.2】上机运行程序，练习位运算的混合运算的使用，验证运行结果与自己分析的结果是否相符，并记录运行结果。



```

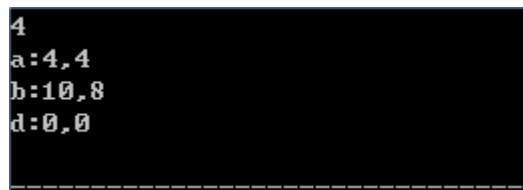
4/2&2 =2
(4>>1)>(4>>2)=2

```

图 12-2 例题 12.2 运行结果

【程序 12.3】上机运行程序，练习位运算的混合运算的使用，验证运行结果与自己分

析的结果是否相符，并记录运行结果。



```
4
a: 4.4
b: 10.8
d: 0.0
```

图 12-3 例题 12.3 运行结果

2. 程序完善

【程序 12.4】阅读程序，若要使程序的运行结果为 248，应在下划线处填入什么，验证运行结果与分析结果是否相符，并记录运行结果。

```
#include <stdio.h>
int main( )
{ short c=124;
  c=c<<1;
  printf("%d\n",c);
  return 0;
}
```

程序运行结果如图 12-4 所示。



```
248
```

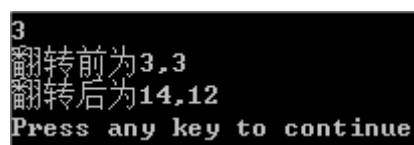
图 12-4 例题 12.4 运行结果

3. 程序设计

【程序 12.5】编写一个程序，实现将一个整数的低 4 位翻转。要求用八进制数输入和输出。

```
#include<stdio.h>
int main( )
{ int a,b;
  scanf("%o",&a);
  printf("翻转前为%o,%d\n",a,a);
  b=a^15 ;
  printf("翻转后为%o,%d\n",b,b);
  return 0;
}
```

程序运行结果如图 12-5 所示。



```
3
翻转前为3.3
翻转后为14.12
Press any key to continue
```

图 12-5 例题 12.5 运行结果