**实验五 函数**

**一、 目的**

1. 掌握函数的定义、声明和调用的方法；

2. 掌握函数实参与形参的对应关系的，以及“值传递”的方式；

3. 掌握递归函数的用法；

4. 按实验内容要求完成全程程序设计后才允许上机。 **二、 实验内容**

1、

#include<stdio.h>

main()

{

int m1, n1, a, b;

int f(double), g(int, int), l(int, int);

double m, n;

/\*首先分别验证输入m,n的合法性\*/

printf("请输入正整数m:");

scanf("%lf", &m);

m1 = f(m);

printf("请输入正整数n:");

scanf("%lf", &n);

n1 = f(n);

a = g(m1, n1);; b = l(m1, n1);

printf("两个正整数的最大公约数为%d\n", a);

printf("两个正整数的最小公倍数为%d\n", b);

}

/\*以下定义一个检验输入是否为正整数的函数\*/

int f(double n)

{

while ((int)n != n || n <= 0)

{

printf("请输入正整数！：");

scanf("%lf", &n);

}

return((int)n);

}

/\*以下定义一个求两数最大公约数的函数\*/

int g(int m, int n)

{

int i;

i = m % n;

while (i != 0)

{

m = n;

n = i;

i = m % n;

}

return(n);

}

/\*以下定义一个求两数最小公倍数的函数\*/

int l(int m, int n)

{

/\*先求出两数的最大公约数\*/

int i, k, p, q;

i = m % n;

p = m; q = n;

while (i != 0)

{

m = n;

n = i;

i = m % n;

}

/\*最后原两数相乘再除以最大公约数即为所求\*/

k = (p \* q) / n;

return(k);

}



2、

#include<stdio.h>

main()

{

int x;

void wrt(int);

printf("请输入一个整数：");

scanf("%d", &x);

printf("转换为字符串后：");

/\*判断输入的整数是否为负数\*/

if (x < 0)

{

x = -x;//转为整数再转换

printf("-");//负号转换为字符

}

wrt(x);

printf("\n");

}

/\*下面定义一个转化整数为字符的函数\*/

void wrt(int x)

{

char c;

/\*判断是否为个位数进行递归\*/

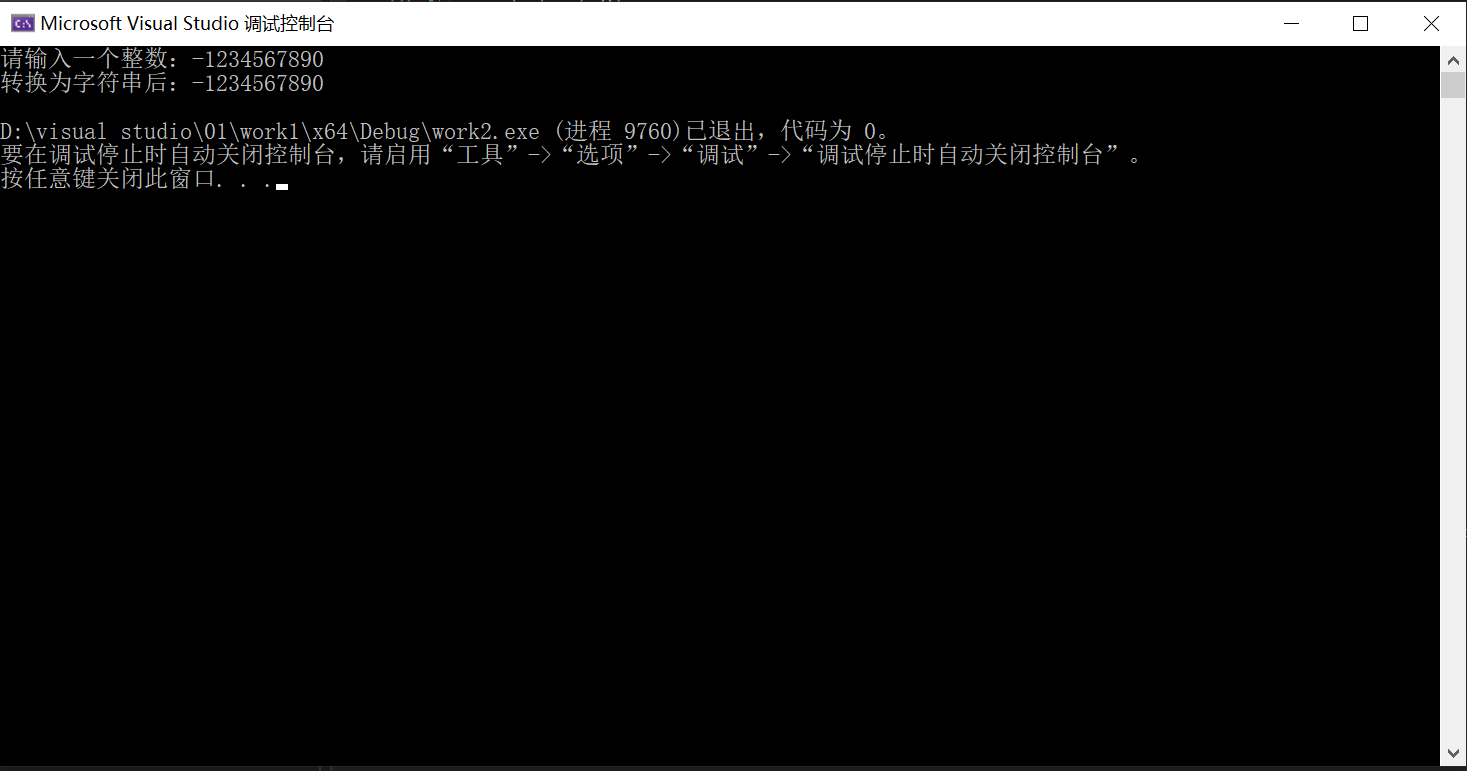
if (x / 10 != 0)

wrt(x / 10);

c = x % 10 + '0';

putchar(c);

}



3、

#include<stdio.h>

main()

{

float temp[96] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 1.46667, 11.4667, 31.6, 52.7333, 80.3333, 116.333, 156.6, 199.4, 242.2,

283.4, 329.2, 379.333, 431.333, 482.6, 541, 594.4, 643.533, 692.133, 736.267,

772.667, 810.133, 841.867, 868.2, 892.4, 917.667, 939.8, 954.667, 969, 976.8,

983.4, 987.467, 994.933, 1023.67, 875.2, 873.933, 758.8, 678.2, 515.867, 782.533,

908.8, 779.2, 831.4, 645.533, 734.067, 679.533, 610.267, 565.067, 512.467, 462,

405.2, 354.133, 302, 247.8, 191.533, 140, 94.2667, 57.5333, 25.9333, 4, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0 };

float tempy[96] = { 0.595, 0.595, 0.595, 0.595, 0.595, 0.595, 0.595, 0.595, 0.595,

0.595, 0.595, 0.595, 0.595, 0.595, 0.595, 0.595, 0.595, 0.595, 0.595, 0.595,

0.595, 0.595, 0.595, 0.595, 0.595, 0.595, 0.595, 0.595, 0.595, 0.595, 0.595,

0.595, 1.785, 2.57833, 3.927, 5.79233, 7.379, 9.48133, 11.1473, 12.4167,

13.6627, 16.193701, 18.248699, 19.042, 19.042, 19.105301, 16.6383,

17.240999, 14.631, 11.8217, 11.663, 12.155, 15.488, 21.859301, 19.32,

19.042, 19.6133, 21.105, 22.9937, 20.827299, 23.858299, 23.0333, 19.2883,

15.6937, 21.5893, 23.802999, 20.518299, 21.5893, 17.907301, 17.971001,

17.574301, 16.781, 15.5513, 12.3773, 10.2747, 8.60867, 6.86333, 5.39567,

3.88767, 2.856, 2.142, 2.142, 0.952, 0.952, 0.952, 0.952, 0.952, 0.952,

0.952, 0.952, 0.952, 0.952, 0.952, 0.952, 0.952, 0.952 };

float f(float[], float[], int);

float g(float[], float[], int);

float a, b;

b = f(temp, tempy, 96); a = g(temp, tempy, 96);

printf("a的值为%f\nb的值为%f", a, b);

}

float f(float temp[],float tempy[],int n)

{

int i;

float b, p = 0.0, q = 0.0, j = 0.0, k = 0.0;

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

{

p += temp[i] / n;

q += tempy[i] / n;

j += (temp[i] \* temp[i]) / n;

k += (temp[i] \* tempy[i]) / n;

}

b = (j \* q - p \* k) / (n \* j - p \* p);

return (b);

}

float g(float temp[], float tempy[], int n)

{

int i;

float a, p = 0.0, q = 0.0, j = 0.0, k = 0.0;

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

{

p += temp[i] / n;

q += tempy[i] / n;

j += (temp[i] \* temp[i]) / n;

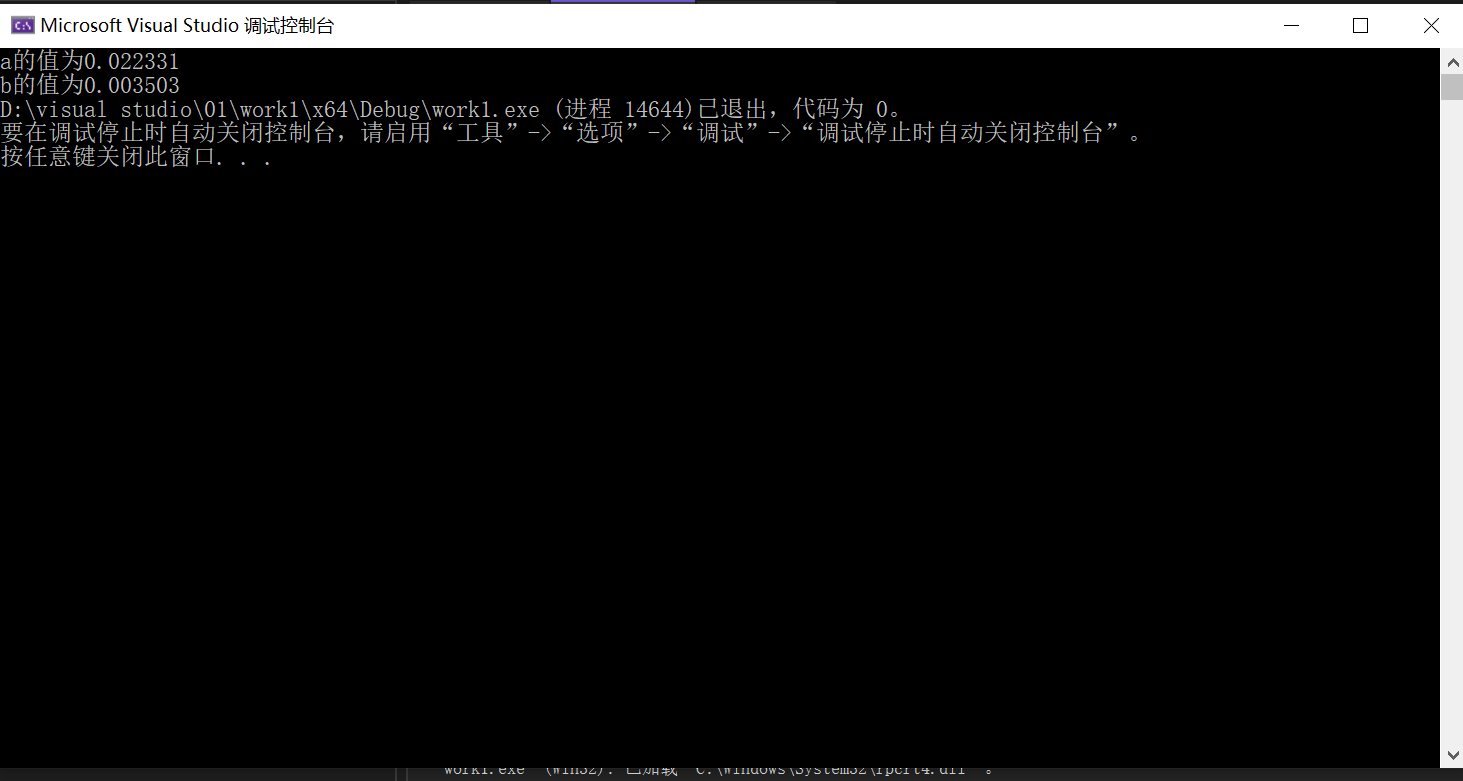
k += (temp[i] \* tempy[i]) / n;

}

a = (n\*k-p\*q) / (n \* j - p \* p);

return (a);

}



**三、 心得总结**

1、vs编译环境中，为类型为float的数组初始化小数时会警告：

“初始化”从“double”到“float”截断

对数值精度会有影响，编程时尽量用double定义实型。

2、编写递归函数时要明确语句的顺序，明确语句是在if语句外还是if语句内，结果是否会缺失。

3、检验输入合法性可分开输入，分开检验避免错位