西安交通大学实验报告

课程 计算机程序设计 实验名称 函数与指针 第 1 页 共 13 页

系 别 钱 学 森 学 院 实 验 日 期 2017年 4 月 13 日

专业班级 钱62 组别\_\_\_\_\_\_\_\_\_\_\_ 实 验 报 告 日 期 2017 年 4 月13日

姓 名 周宇晨 学号 2160405046 报 告 退 发 ( 订正 、 重做 )

同 组 人\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 教 师 审 批 签 字

一、实验目的

1.指向函数的指针

2.递归函数

3.函数重载

二、实验内容

题目一 迭代求

1.要点分析

迭代法，.

2.程序源码

#include<iostream>

#include<cmath>

#include<Windows.h>

using namespace std;

double power(double x, int k)

{

if (k == 1)

{

return x;

}

else

{

return x\*power(x, k - 1);

}

}

int main()

{

double x;

int k;

cout << "请输入实数x和整数k:" << endl;

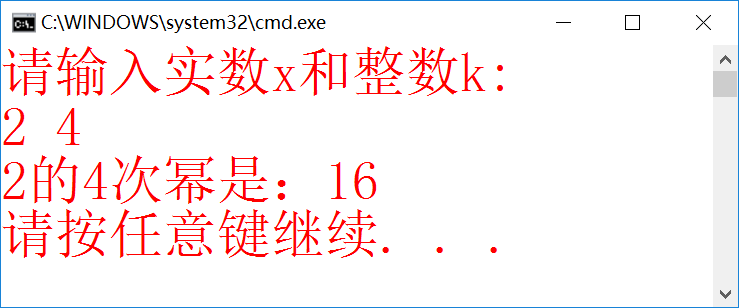
cin >> x >> k;

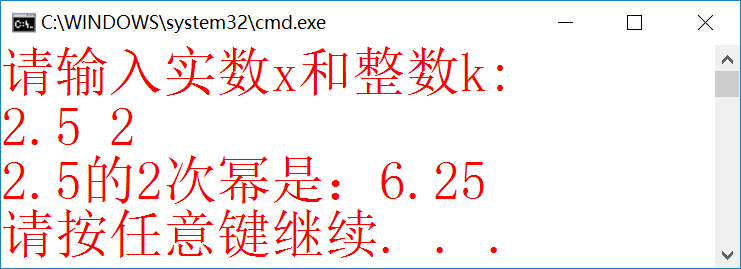
cout << x<<"的"<<k<<"次幂是：" << power(x, k) << endl;

return 0;

}

3.实验结果





题目二 之斐波那契数列

1.要点分析

略

2.程序源码

#include<iostream>

#include<cmath>

#include<Windows.h>

using namespace std;

int fabonaci(int n)

{

if (n == 1)return 0;

if (n == 2)return 1;

if (n == 3)return 1;

if (n >= 3)

{

return fabonaci(n - 1) + fabonaci(n - 2);

}

}

int main()

{

int n;

cout << "请输入n：";

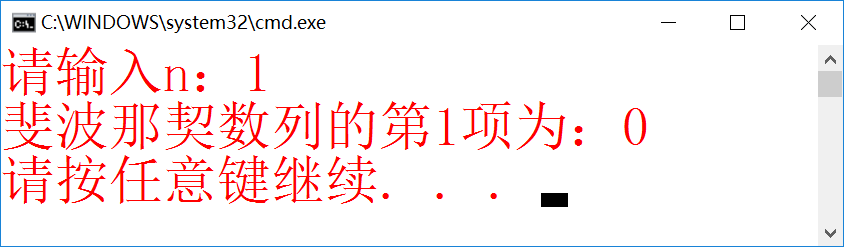
cin >> n;

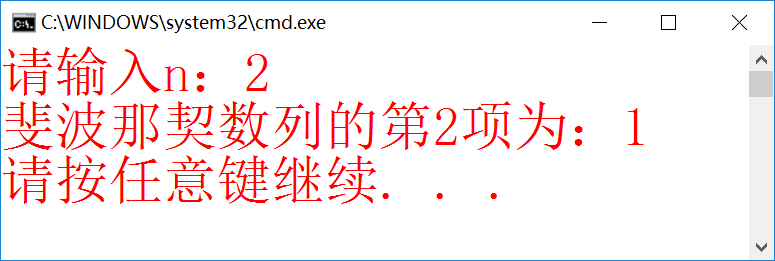
cout << "斐波那契数列的第" << n << "项为：" << fabonaci(n) << endl;

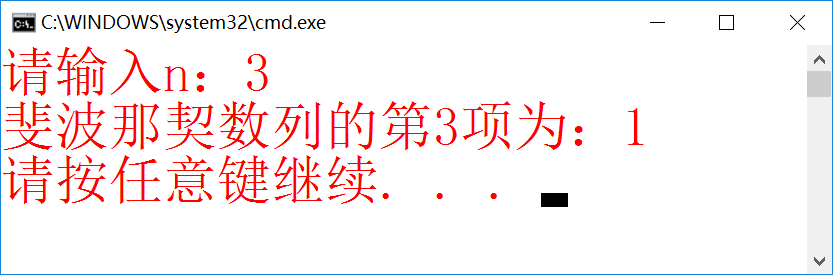
return 0;

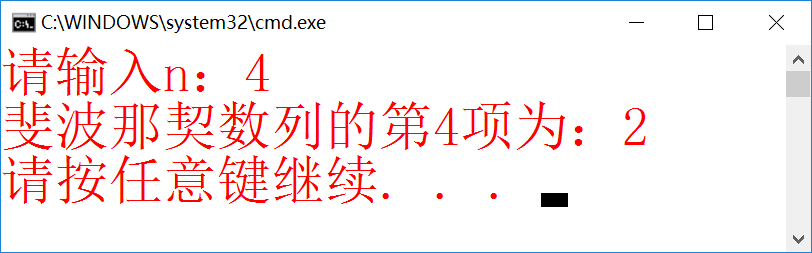
}

3.实验结果









题目二 之平方和

1.要点分析

利用函数fabonaci（）求出各项的值，存入新开辟的数组，然后求和。

2.程序源码

#include<iostream>

#include<cmath>

#include<Windows.h>

using namespace std;

int fabonaci(int n)

{

if (n == 1)return 0;

if (n == 2)return 1;

if (n == 3)return 1;

if (n >= 3)

{

return fabonaci(n - 1) + fabonaci(n - 2);

}

}

int sum\_f(int \*pi, int n)

{

int s = 0;

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

{

s += pow(\*pi, 2);

pi++;

}

return s;

}

int main()

{

int n;

cout << "请输入n：";

cin >> n;

int \*pi = new int[n];

for (int i = 1; i <= n; i++)

{

\*pi = fabonaci(i);

pi++;

}

pi -= n;

cout << sum\_f(pi, n) << "=";

for (int j = 1; j <= n; j++)

{

cout << fabonaci(j)<<"^2"<<"+";

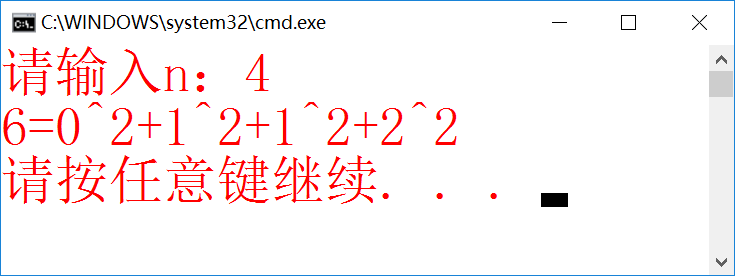
}

cout << "\b " << endl;

return 0;

}

3.实验结果



题目三 重载函数求最大值

1.要点分析

略

2.程序源码

#include<iostream>

#include<cstring>

#include<windows.h>

using namespace std;

int maximum(int a, int b, int c)

{

int t;

if (b<c) { t = b; b = c; c = t; }

return a>b ? a : b;

}

float maximum(float a, float b, float c)

{

float t;

if (b<c) { t = b; b = c; c = t; }

return a>b ? a : b;

}

float myatof(char \*a)

{

int i = 0;

float y = 0;

if (\*(a + 0) == '-' || \*(a + 0) == '+') i++;

while (\*(a + i) != '\0'&&\*(a + i) != '.')

{

y = y \* 10 + (\*(a + i) - '0');

i++;

}

if (\*(a + i) == '.')

{

float s = 0.1;

while (\*(a + i + 1) != '\0')

{

y = y + s\*(a[i + 1] - '0');

s = s / 10;

i++;

}

}

if (a[0] == '-') y = -y;

return y;

}

float maximum(char \*a, char \*b, char \*c)

{

float aa, bb, cc, t;

aa = myatof(a);

bb = myatof(b);

cc = myatof(c);

if (bb<cc) { t = bb; bb = cc; cc = t; }

return aa>bb ? aa : bb;

}

int main()

{

char a3[20], b3[20], c3[20];

cout << "请输入三个字符型数" << endl;

cin >> a3 >> b3 >> c3;

cout << "最大数是" << maximum(a3, b3, c3) << endl;

int a1, b1, c1;

cout << "请输入三个整形数" << endl;

cin >> a1 >> b1 >> c1;

cout << "最大数是" << maximum(a1, b1, c1) << endl;

float a2, b2, c2;

cout << "请输入三个浮点型数" << endl;

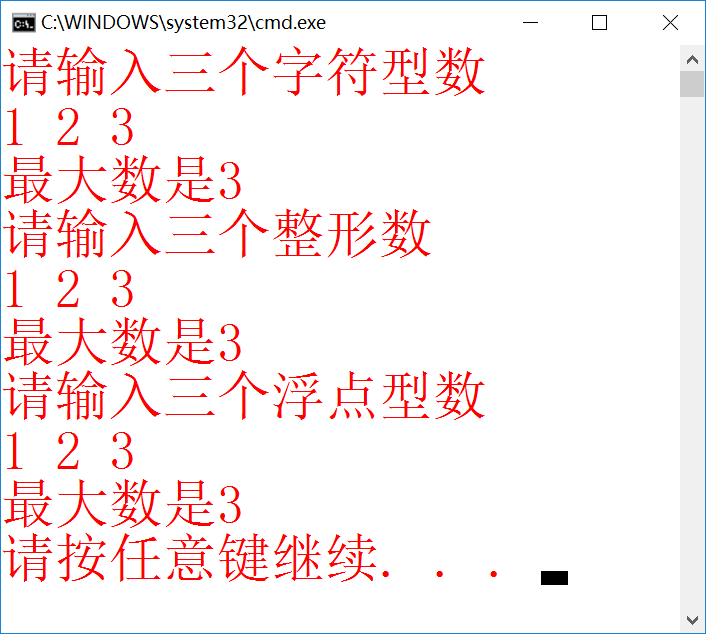
cin >> a2 >> b2 >> c2;

cout << "最大数是" << maximum(a2, b2, c2) << endl;

return 0;

}

3.实验结果



题目四 去空格

1.要点分析

将指针移至第一个不为空格的字符处。

2.程序源码

#include<iostream>

#include<cmath>

#include<Windows.h>

using namespace std;

char \*myltrim(char \*string)

{

while (\*string == 32)string++;

return string;

}

int main()

{

cout << "请输入一个字符串(100个字符以内）：" << endl;

char s[100];

cin >> s;

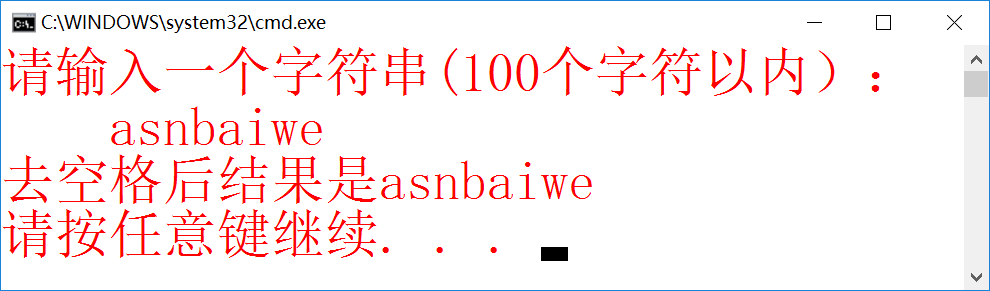
cout << "去空格后结果是";

cout << myltrim(s) << endl;

return 0;

}

3.实验结果



题目五 用牛顿迭代法求任意一元方程

1.要点分析

创建函数fun（）和differ（）分别求多项式及其导数的值；利用迭代逼近求根；控制输出格式。

2.程序源码

#include<iostream>

#include<cmath>

using namespace std;

double fun(double \*xs, int count, double x)

{

double s = 0;

for (int i = count; i >= 0; i--)

{

s += (\*xs)\*pow(x, i);

\*xs++;

}

return s;

}

double differ(double \*xs, int count, double x)

{

double s = 0;

for (int i = count; i > 0; i--)

{

s += (\*xs)\*i\*pow(x, i - 1);

\*xs++;

}

return s;

}

double sol(double \*xs, int count, double x)

{

double e = .00001; //精度

double x1 = x - fun(xs, count, x) / differ(xs, count, x);

if (x1 - x < e && fun(xs, count, x1) < e)

{

return x1;

}

else

{

return sol(xs, count, x1);

}

}

int main()

{

int n;

double x0;

cout << "请输入方程阶数：";

cin >> n;

double \*p = new double[n + 1];

for (int i = n; i >0; i--)

{

cout << "请输入x^" << i << "次方的系数：";

cin >> \*p;

p++;

}

cout << "请输入常数项系数：";

cin >> \*p;

p = p - n; //指针归位

cout << "请输入根的基准：";

cin >> x0;

cout << "方程";

double temp;

for (int j = n; j > 0; j--)

{

temp = \*p;

if (j != n)

{

if (temp > 0)cout << "+";

}

if (temp != 0)

{

if (temp == 1)

{

cout << "x^" << j;

}

else

{

cout << \*p << "x^" << j;

}

}

if (j == 1)cout << "\b\b";

p++;

}

if (\*p > 0)cout << "+";

cout << \*p << "=0在" << x0 << "附近的根为";

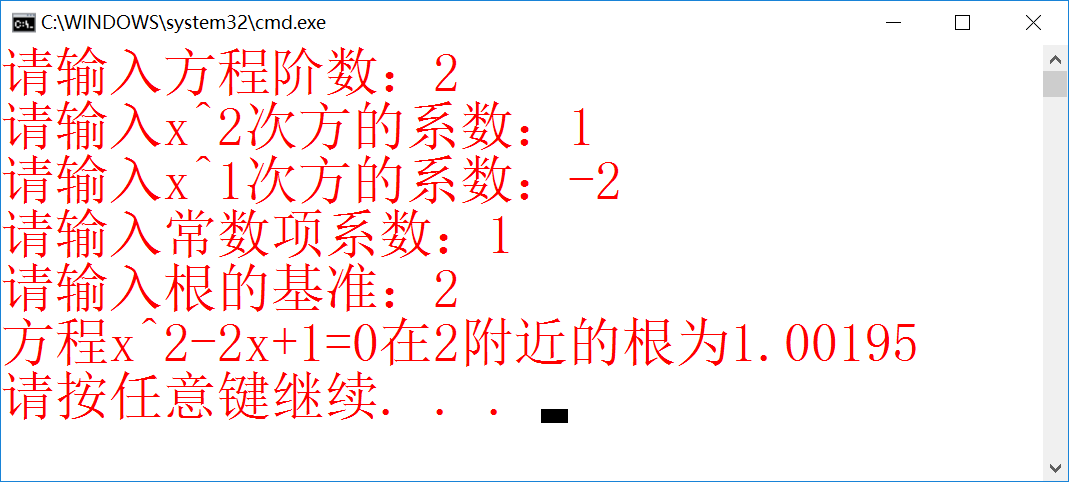
p = p - n; //指针归位

cout << sol(p, n, x0) << endl;

return 0;

}

3.实验结果



题目六 字符串转数

1.要点分析

以小数点为界，依次把每项按级数累和。

2.程序源码

#include<iostream>

#include<cmath>

using namespace std;

double myatof(char \*num)

{

double s = 0;

int t = 1, i = 0, k = 0;

if (\*num == 45)

{

t = -1;

num++;

}

while (\*num != 46)

{

i++;

num++;

}

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

{

num--;

s += ((\*num)-48)\*pow(10.0, j);

}

num = num + i + 1;

while (\*num != 0)

{

k++;

s += ((\*num)-48)\*pow(10.0, -k);

num++;

}

return s\*t;

}

int main()

{

char x[100];

cout << "请输入一个数：";

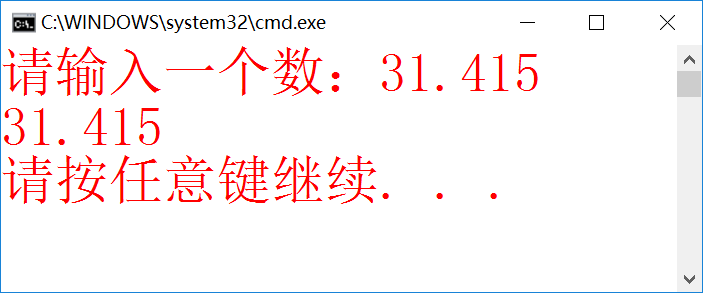
cin >> x;

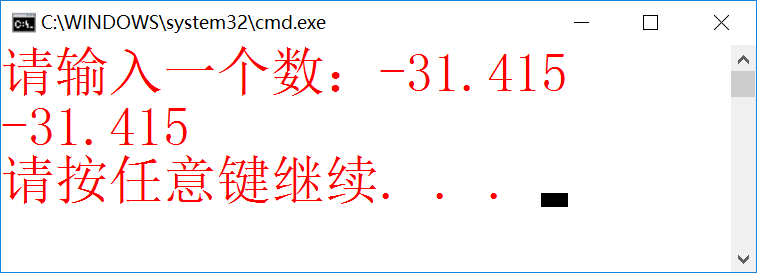
cout << myatof(x) << endl;

return 0;

}

3.实验结果





题目七 查找字串个数

1.要点分析

寻找字串是否存在；查找个数。

2.程序源码

#include<iostream>

#include<cstring>

#include<windows.h>

using namespace std;

int f(char \*p, char \*s)

{

int np, ns, i, j, k, n = 1;

int r = 0;

np = strlen(p);

ns = strlen(s);

for (i = 0; i<np - ns; i++)

{

if (\*(p + i) == \*s)

{

for (j = 0; j<ns; j++)if (\*(p + j + i) != \*(s + j)) n = 0;

if (n == 1)r++;

}

n = 1;

}

return r;

}

int main()

{

char a[100], b[100];

cout << "请输入主字符串（100个字符以内）：" << endl;

cin >> a;

cout << "请输入目标字串：" << endl;

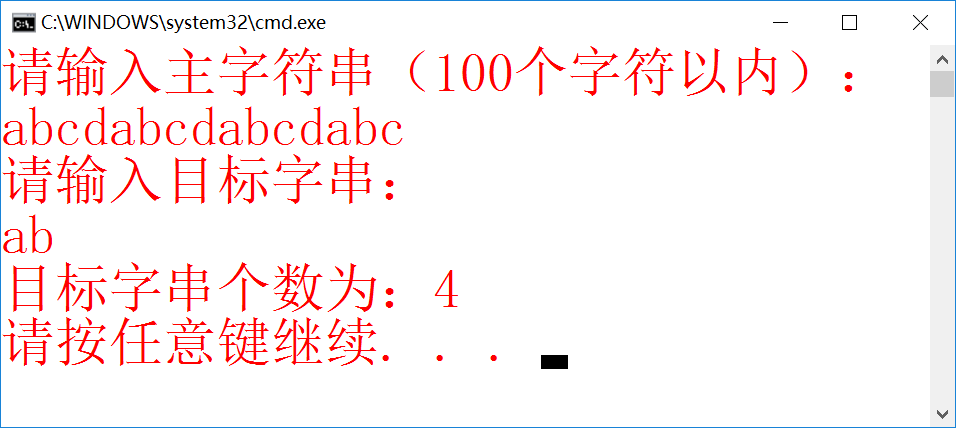
cin >> b;

cout << "目标字串个数为：" << f(a, b) << endl;

return 0;

}

3.实验结果



三、个人小结

经过上周考试的残忍洗礼，这周决定洗心革面好好做人，作业好好做了，还拓展上网查证了不少。老实说，滋味不错。

本周学习了函数的迭代，以及指向函数的指针。过了指针一关，这周内容并不难理解消化。当这周编了牛顿迭代法求根的时候，我才意识到自己已经能编出真正能做事的程序了。心情很复杂，主要是激动。多学多练，总有回报。

好好学习，功德无量啊。