**《面向对象程序设计语言》作业（3.19）**

地信班109092023XXX 许愿

1. 写一个函数，分别求两个整数的最大公约数和最小公倍数，用主函数调用两个函数，并输出结果，两个整数由键盘输入。

#include <iostream>

using namespace std;

int gcd(int a, int b){ // 求最大公约数

    int t;

    while(b != 0){ // 辗转相除法

        t = a % b;

        a = b;

        b = t;

    }

    return a;

}

int lcm(int a, int b){ // 求最小公倍数

    return a \* b / gcd(a, b);

}

int main(){

    int a, b;

    cout << "请分别输入a和b: ";

    cin >> a >> b;

    cout << "最大公约数: " << gcd(a, b) << endl;

    cout << "最小公倍数: " << lcm(a, b) << endl;

    return 0;

}



1. 求方程ax^2+bx+c=0的根，用三个函数分别求当b^2-4ac大于0、等于0、小于0时的根，并输出结果。从主函数输入a,b,c的值。

#include <iostream>

#include <cmath>

using namespace std;

int main(){

    double a,b,c;

    cout << "分别输入a b c三个数:";

    cin >> a >> b >> c;

    double pbs = b\*b-4\*a\*c; // 计算判别式的值

    if(pbs>0){

        double x1 = (-b+sqrt(pbs))/(2\*a);

        double x2 = (-b-sqrt(pbs))/(2\*a);

        cout << "方程的两个根 x1=" << x1 << " x2=" << x2 << endl; // 求根公式

    }else if(pbs==0){

        double x = -b/(2\*a);

        cout << "方程的根 x=" << x << endl; // 求根公式

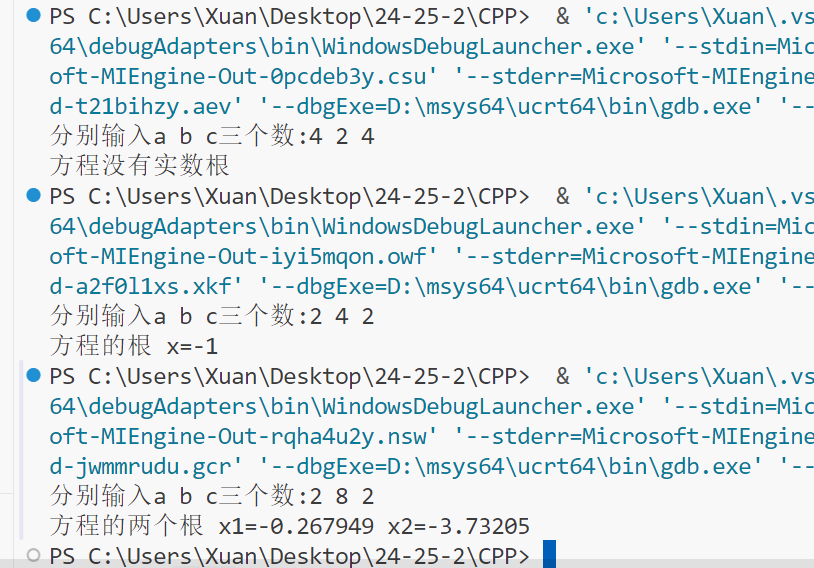
    }else{

        cout << "方程没有实数根";

    }

    return 0;

}



1. 写一个判别素数的函数，在主函数中输入一个整数，输出是否为素数的信息。

#include <iostream>

using namespace std;

bool is\_prime(int a){ // 返回布尔值，真或假

    for(int i=2;i<a;i++){ // 从2开始判断是否整除，若整除则不是素数

        if(a%i==0) return false;

    }

    return true; // 否则是素数

}

int main(){

    int a;

    cout << "输入一个整数: ";

    cin >> a;

    if(is\_prime(a)){

        cout << a << "是素数" << endl;

    }else{

        cout << a << "不是素数" << endl;

    }

    return 0;

}



1. 求a!+b!+c!的值，用一个函数fac(n)求n!。a,b,c的值由主函数输入，最终得到的值在主函数中输出。

#include <iostream>

using namespace std;

int fac(int n){

    if(n==1){

        return 1;

    }

    return n\*fac(n-1); // 递归调用

}

int main(){

    int a, b, c;

    cout << "请分别输入a, b, c: ";

    cin >> a >> b >> c;

    cout << "a!+b!+c!=" << fac(a) + fac(b) + fac(c) << endl;

    return 0;

}



1. 写一个函数求sinh(x)的值，求sinh(x)的近似公式为sinh(x)=(e^x-e^(-x))/2。其中，用一个函数求e^x。

#include <iostream>

#include <cmath>

using namespace std;

double get\_e\_x(double x){

    return exp(x); // cmath库内置的计算e的x次方的值的函数

}

int main(){

    double x;

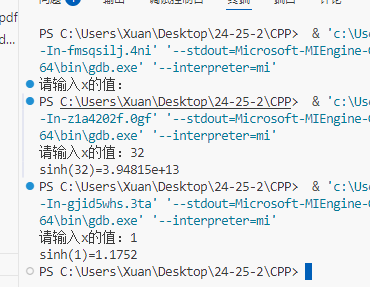
    cout << "请输入x的值：";

    cin >> x;

    cout << "sinh(" << x << ")=" << (get\_e\_x(x) - get\_e\_x(-x)) / 2 << endl;

    return 0;

}



1. 用牛顿迭代法求根。方程为ax^3+bx^2+cx+d=0。系数a,b,c,d的值依次为1,2,3,4，由主函数输入。求x在1附近的一个实根。求出根后由主函数输出。

#include <iostream>

using namespace std;

double niudundiedaifa(double a, double b, double c, double d){

    double x = 1, x0; // 初始化x值为1

    do

    {

        x0 = x;

        x = x0-(((a\*x0+b)\*x0+c)\*x0+d)/((3\*a\*x0+2\*b)\*x0+c);

    } while (abs(x - x0) >= 1e-3); // 若求得的x值和1的差值大于1e-3则继续迭代

    return x;

}

int main(){

    double a, b, c, d;

    cout << "请输入a,b,c,d的值：";

    cin >> a >> b >> c >> d;

    double x = niudundiedaifa(a, b, c, d);

    cout << "x = " << x << endl;

    return 0;

}



1. 写一个函数验证哥德巴赫猜想：一个不小于6的偶数可以表示为两个素数之和。在主函数中输入一个不小于6的偶数n，然后调用函数gotbaha，在gotbaha函数中再调用prime函数，prime函数的作用是判别一个数是否为素数。在gotbaha函数中输出以下形式的结果：34=3+31。

#include <iostream>

using namespace std;

bool prime(int n){ // 判断是否为素数

    for(int i=2;i<n;i++){ // 从2开始判断是否为素数

        if(n%i==0) return false; // 若不是素数则返回false

    }

    return true; // 遍历完成，是素数，返回true

}

void gotbaha(int n){ // 在这个函数中输出信息所以无需设置返回值

    for(int i=2;i<=n/2;i++){ // 从2开始遍历到n/2

        if(prime(i) && prime(n-i)){ // 若i和n-i都是素数

            cout << n << "=" << i << "+" << n-i << endl; // 输出结果

            break; // 只需要输出一条所以判断出结果后直接跳出即可

        }

    }

}

int main(){

    int n;

    while(true){ // 获取数字, 若数字不满足要求则重新获取

        cout << "请输入一个不小于6的偶数n: ";

        cin >> n;

        if(n>6 && n%2==0){

            break;

        };

        cout << "输入错误，请重新输入" << endl;

    }

    gotbaha(n); // 调用函数

    return 0;

}



1. 用递归方法求n阶勒让德多项式的值。

#include <iostream>

using namespace std;

double get\_p(int n,double x){

    if(n==0){

        return 1;

    }else if(n==1){

        return x;

    }else{

        return ((2\*n-1)\*x\*get\_p(n-1,x)-(n-1)\*get\_p(n-2,x))/n;

    }

}

int main(){

    int n;

    double x;

    cout << "请输入n和x:";

    cin >> n >> x;

    cout << "P" << n << "(" << x << ")=" << get\_p(n,x) << endl;

    return 0;

}



1. 汉诺塔问题。

#include <iostream>

using namespace std;

void hannuota(int n, char a, char b, char c){ // 后三个参数为盘的名字

    if(n == 1){

        cout << "将盘子" << n << "从" << a << "移动到" << c << endl;

    }else{ // 分治思想

        hannuota(n-1, a, c, b);

        cout << "将盘子" << n << "从" << a << "移动到" << c << endl;

        hannuota(n-1, b, a, c);

    }

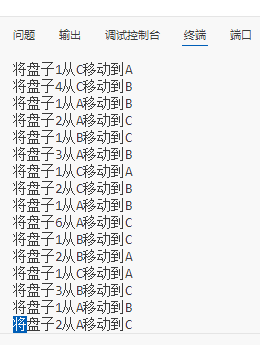
}

int main(){

    hannuota(64, 'A', 'B', 'C');

    return 0;

}



1. 用递归法将一个整数n转换为字符串。

#include <iostream>

using namespace std;

string n\_to\_string(int n){

    int n1 = n / 10; // n1为去除最末尾数字后的数字

    int n2 = n % 10; // n2为最末尾的数字

    if(n1 == 0){ // 已经到最后

        return string(1, '0' + n2); // 直接返回n2对应的数字的string形式

    }

    return n\_to\_string(n1) + string(1, '0' + n2); // 递归调用返回

}

int main(){

    int n;

    cout << "请输入数字n: ";

    cin >> n;

    cout << n\_to\_string(n) << endl;

    return 0;

}



1. 用递归法求f(x)=i^2（对i求和）。

#include <iostream>

using namespace std;

int getfx(int n){

    if(n==1){

        return 1;

    }

    return getfx(n-1)+n\*n;

}

int main(){

    int n;

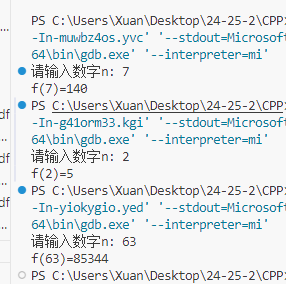
    cout << "请输入数字n: ";

    cin >> n;

    cout << "f(" << n << ")=" << getfx(n) << endl;

    return 0;

}



1. 三角形的面积为area=s\*(s-a)\*(s-b)\*(s-c)，其中s=(a+b+c)/2。定义两个带参数的宏，一个用来求s，一个用来求area。编写程序，在程序中用带实参的宏名来求面积area。

#include <iostream>

#include <cmath>

using namespace std;

double get\_s(double a,double b,double c){

    return (a+b+c)/2;

}

double get\_area(double a,double b,double c){

    double s=get\_s(a,b,c);

    return sqrt(s\*(s-a)\*(s-b)\*(s-c)); // 调用cmath库的内置函数求平方根

}

int main(){

    double a,b,c;

    cout << "请分别输入三角形的三边a,b,c: ";

    cin >> a >> b >> c;

    cout << "三角形的面积为: " << get\_area(a,b,c) << endl;

    return 0;

}

