

## 《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;
}
```

```
问题 输出 调试控制台 终端 端口 评论

● 请分别输入a和b: 9 16
最大公约数: 1
最小公倍数: 144
● PS C:\Users\Xuan\Desktop\24-25-2\CPP> & 'c:\Users\
-In-zvgv3rpc.ayx' '--stdout=Microsoft-MIEngine-Out-
64\bin\gdb.exe' '--interpreter=mi'
请分别输入a和b: 82 144
最大公约数: 2
最小公倍数: 5904
● PS C:\Users\Xuan\Desktop\24-25-2\CPP> & 'c:\Users\
-In-wzvyb4ii.rnn' '--stdout=Microsoft-MIEngine-Out-
64\bin\gdb.exe' '--interpreter=mi'
请分别输入a和b: 4 8
最大公约数: 4
最小公倍数: 8
○ PS C:\Users\Xuan\Desktop\24-25-2\CPP>
```

2. 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;
}
```

```

● PS C:\Users\Xuan\Desktop\24-25-2\CPP> & 'c:\Users\Xuan\.vs64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-Out-0pcdeb3y.csu' '--stderr=Microsoft-MIEngine-d-t21bihzy.aev' '--dbgExe=D:\msys64\ucrt64\bin\gdb.exe' '--
分别输入a b c三个数:4 2 4
方程没有实数根

● PS C:\Users\Xuan\Desktop\24-25-2\CPP> & 'c:\Users\Xuan\.vs64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-Out-iyi5mqon.owf' '--stderr=Microsoft-MIEngine-d-a2f011xs.xkf' '--dbgExe=D:\msys64\ucrt64\bin\gdb.exe' '--
分别输入a b c三个数:2 4 2
方程的根 x=-1

● PS C:\Users\Xuan\Desktop\24-25-2\CPP> & 'c:\Users\Xuan\.vs64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-Out-rqha4u2y.nsw' '--stderr=Microsoft-MIEngine-d-jwmmrdu.gcr' '--dbgExe=D:\msys64\ucrt64\bin\gdb.exe' '--
分别输入a b c三个数:2 8 2
方程的两个根 x1=-0.267949 x2=-3.73205

○ PS C:\Users\Xuan\Desktop\24-25-2\CPP>

```

3. 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;
}

```

```

PS C:\Users\Xuan\Desktop\24-25-2\CPP> & 'c:\64\debugAdapters\bin\WindowsDebugLauncher.exe
oft-MIEngine-Out-ltwycgcd.wgr' '--stderr=Micro
d-lz2pootm.bh1' '--dbgExe=D:\msys64\ucrt64\bi
● 输入一个整数: 33
33不是素数
PS C:\Users\Xuan\Desktop\24-25-2\CPP> & 'c:\64\debugAdapters\bin\WindowsDebugLauncher.exe
oft-MIEngine-Out-uiaasaghf.mjr' '--stderr=Micro
d-lpqmgumb.h3z' '--dbgExe=D:\msys64\ucrt64\bi
● 输入一个整数: 7
7是素数
○ PS C:\Users\Xuan\Desktop\24-25-2\CPP>

```

4. 求  $a!+b!+c!$  的值，用一个函数  $\text{fac}(n)$  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;
}

```

```

PS C:\Users\Xuan\Desktop\24-25-2\CPP> & 'c:\U
-In-pblnytu4.13s' '--stdout=Microsoft-MIEngine
64\bin\gdb.exe' '--interpreter=mi'
● 请分别输入a, b, c: 3 6 8
a!+b!+c!=41046
● PS C:\Users\Xuan\Desktop\24-25-2\CPP> & 'c:\U
-In-dke0zcmi.vro' '--stdout=Microsoft-MIEngine
64\bin\gdb.exe' '--interpreter=mi'
请分别输入a, b, c: 2 2 2
a!+b!+c!=6
○ PS C:\Users\Xuan\Desktop\24-25-2\CPP>

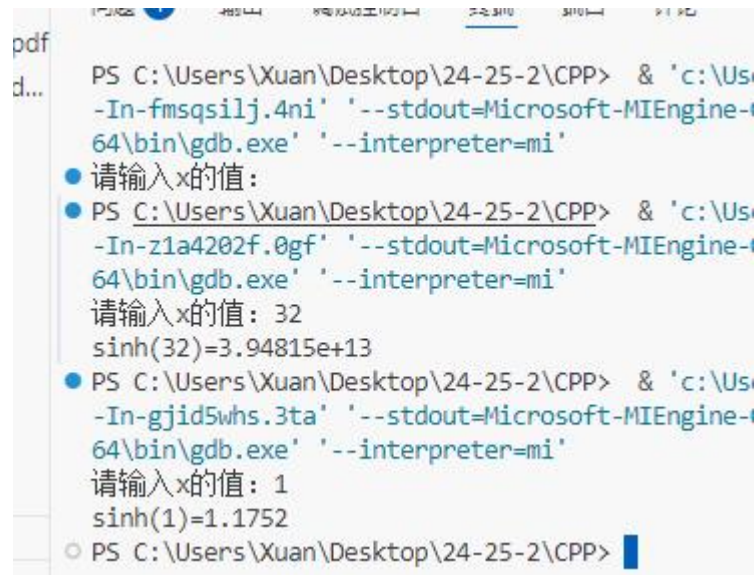
```

5. 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;
}
```



```
PS C:\Users\Xuan\Desktop\24-25-2\CPP> & 'c:\Us
-In-fmsqsilj.4ni' '--stdout=Microsoft-MIEngine-1
64\bin\gdb.exe' '--interpreter=mi'
● 请输入x的值:
● PS C:\Users\Xuan\Desktop\24-25-2\CPP> & 'c:\Us
-In-z1a4202f.0gf' '--stdout=Microsoft-MIEngine-1
64\bin\gdb.exe' '--interpreter=mi'
请输入x的值: 32
sinh(32)=3.94815e+13
● PS C:\Users\Xuan\Desktop\24-25-2\CPP> & 'c:\Us
-In-gjid5whs.3ta' '--stdout=Microsoft-MIEngine-1
64\bin\gdb.exe' '--interpreter=mi'
请输入x的值: 1
sinh(1)=1.1752
○ PS C:\Users\Xuan\Desktop\24-25-2\CPP>
```

6. 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;
}
```

```
-In-fj1myykr.ca3' '--stdout=Microsoft-MIEngine-C
64\bin\gdb.exe' '--interpreter=mi'
请输入a,b,c,d的值: 1 2 3 4
x = -1.65063
PS C:\Users\Xuan\Desktop\24-25-2\CPP> 
```

7. 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;
}

```

```

PS C:\Users\Xuan\Desktop\24-25-2\CPP> & '64\debugAdapters\bin\WindowsDebugLauncher.oft-MIEngine-Out-rhs2d10j.bzo' '--stderr=Md-bnscfsj3.mr1' '--dbgExe=D:\msys64\ucrt64
● 请输入一个不小于6的偶数n: 35
  输入错误，请重新输入
  请输入一个不小于6的偶数n: 36
  36=5+31
● PS C:\Users\Xuan\Desktop\24-25-2\CPP> & '64\debugAdapters\bin\WindowsDebugLauncher.oft-MIEngine-Out-u4o04mrt.0u2' '--stderr=Md-uk3rbmtu.5q5' '--dbgExe=D:\msys64\ucrt64
  请输入一个不小于6的偶数n: 999990
  999990=7+999983
○ PS C:\Users\Xuan\Desktop\24-25-2\CPP>

```

8. 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;
}

```



```

问题 输出 调试控制台 终端 端口 评论
PS C:\Users\Xuan\Desktop\24-25-2\CPP> & 'c:\
-In-3ymrc1mw.32n' '--stdout=Microsoft-MIEngi
64\bin\gdb.exe' '--interpreter=mi'
● 请输入n和x:6 8
P6(8)=3.70448e+06
● PS C:\Users\Xuan\Desktop\24-25-2\CPP> & 'c:\
-In-ndmx0ct3.ft1' '--stdout=Microsoft-MIEngi
64\bin\gdb.exe' '--interpreter=mi'
请输入n和x:6 3
P6(3)=8989
○ PS C:\Users\Xuan\Desktop\24-25-2\CPP>

```

## 9. 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;
}

```



```
问题 输出 调试控制台 终端 端口

将盘子1从C移动到A
将盘子4从C移动到B
将盘子1从A移动到B
将盘子2从A移动到C
将盘子1从B移动到C
将盘子3从A移动到B
将盘子1从C移动到A
将盘子2从C移动到B
将盘子1从A移动到B
将盘子6从A移动到C
将盘子1从B移动到C
将盘子2从B移动到A
将盘子1从C移动到A
将盘子3从B移动到C
将盘子1从A移动到B
将盘子2从A移动到C
```

10. 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;
}
```

```
PS C:\Users\Xuan\Desktop\24-25-2\CPP> &
-In-prfazibz.vkh' '--stdout=Microsoft-MI
64\bin\gdb.exe' '--interpreter=mi'
● 请输入数字n: 965
965
○ PS C:\Users\Xuan\Desktop\24-25-2\CPP>
```

11. 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;
}

```

```

PS C:\Users\Xuan\Desktop\24-25-2\CPP\
- In-muwbz4os.yvc' '--stdout=Microsoft
64\bin\gdb.exe' '--interpreter=mi'
● 请输入数字n: 7
f(7)=140
● PS C:\Users\Xuan\Desktop\24-25-2\CPP\
- In-g41orm33.kgi' '--stdout=Microsoft
64\bin\gdb.exe' '--interpreter=mi'
df 请输入数字n: 2
f(2)=5
df ● PS C:\Users\Xuan\Desktop\24-25-2\CPP\
- In-yiokygio.yed' '--stdout=Microsoft
64\bin\gdb.exe' '--interpreter=mi'
df 请输入数字n: 63
f(63)=85344
○ PS C:\Users\Xuan\Desktop\24-25-2\CPP\

```

12. 三角形的面积为  $\text{area} = s(s-a)(s-b)(s-c)$ , 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  $\text{area}$ 。编写程序，在程序中用带实参的宏名来求面积  $\text{area}$ 。

```

#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;
}

```

```

PS C:\Users\Xuan\Desktop\24-25-2\CPP> g++ -In-x32ksjra.vy3' '--stdout=Microsoft-MIEngi
64\bin\gdb.exe' '--interpreter=mi'
请分别输入三角形的三边a,b,c: 6 8 10
三角形的面积为: 24
PS C:\Users\Xuan\Desktop\24-25-2\CPP> g++ -In-gg0dard5.pyy' '--stdout=Microsoft-MIEngi
64\bin\gdb.exe' '--interpreter=mi'
请分别输入三角形的三边a,b,c: 6 9 10
三角形的面积为: 26.6634
PS C:\Users\Xuan\Desktop\24-25-2\CPP>

```