

- 
1. 下面程序可以找出两个整数的较大数。

```
#include <iostream>
using namespace std;
int Max_int(int x, int y); //函数声明
int main()
{
    int a, b, Max;
    cout << " Please input two integer numbers: "; cin >> a >> b;
    Max = Max_int(a, b); //函数调用
    cout << " The Large number: " << Max << endl; return 0;
}
int Max_int(int x, int y)
{ if (x > y) {return x;} else { return y; } } //函数返回结果
```

2. 下面程序可以动态计算圆形面积和圆柱的表面积。

The circle:

Area:28.2743

The cylinder:

Area:131.947

Press any key to continue

注: 以上为计算结果。

```
#include <iostream>
using namespace std;
#define PI 3.1415 //1) 常量 PI 的定义语句 (2 分)。
class Tshape
{
public:
    virtual double Perimeter()=0; // 周长
    virtual double Area(); //2) 纯虚函数的定义 (2 分)。
};
class Tpoint : public Tshape //3) 继承方式 (2 分)。
{
public:
    Tpoint(int m=0,int n=0){ x = m, y = n; }
    int Get_x() { return x; } //4) 请写出函数体 (3 分)。
    int Get_y() { return y; }
    double Perimeter () { return 0; }
    double Area() { return 0; }
private:
    int x,y;
};
class Tcircle :public Tpoint
{
public:
```

---

```

Tcircle(int m=0,int n=0,double l=0):Tpoint(m,n) {   Radius = l;  }
int Get_Radius () {   return Radius;  }
double Perimeter () {   return 2*PI* Radius;  }
double Area() //5) 请写出函数体 (3 分)。
{   return PI * Radius * Radius;  }

protected:
    double Radius; //6) 圆半径定义 (2 分)。
};

class Tcylinder :public Tcircle
{
public:
    Tcylinder() { h = 0; }
    Tcylinder(int m,int n,double l,double k):Tcircle(m,n,l) { h = k; }
    ~Tcylinder() { }
    double Area() //7) 请写出函数体 (4 分)。
    {
        return 2 * PI * Radius * (Radius + h);
    }

private:
    double h; //8) 圆柱体高定义 (2 分)。
};

class TMyProgram
{
public://接口
    void run(Tpoint *p) { cout << p->Area(); }
};

int main()
{
    Tcircle c2(1,1,3);    Tcylinder cy(1,2,3,4);    TMyProgram my;
    cout<<"The circle:\nArea:";    my.run(&c2);    cout<<endl;
    cout<<"The cylinder:\nArea:";    my.run(&cy);    cout<<endl;
    return 0;
}

```

1) 程序的功能是什么? (2 分)

输出 23-49 之间的所有素数

2) 写出程序的输出结果。 (4 分)

23

29

31

37

41

43

47

```

#include<iostream>
using namespace std;
#include<cmath>
int main ()
{
    int i,j,k;
    for(i = 23; i < 50; i = i + 2)

```

---

```

    {
        k=( int ) sqrt ( i );
        for(j = 2; j <= k; j++)
            if(i % j == 0) break;
        if(j >= k+1) cout<<i<<endl;
    }
    return 0;
}

```

1) 请阐述左右程序的区别; (2 分)

左侧程序中的 f(x) 是一个宏, 它在编译之前就被替换, 不会进行类型检查。

右侧程序中的 f(x) 是一个内联函数, 它在编译时被插入到调用点, 可以进行类型检查。

2) 分别写出左右两侧程序的运行结果。 (4 分)

**4\*4=16**

**4+1\*4+1=9**

**4\*4=16**

**(4+1) \* (4+1) =25**

```
//左侧程序
#include<iostream>
using namespace std;
#define f(x) x*x
int main ()
{
    int x(4);
    cout<<f(x)<<endl;
    cout<<f(x+1)<<endl;
    return 0;
}
```

```
//右侧程序
#include<iostream>
using namespace std;
inline int f(int x)
{ return x*x; }
int main ()
{
    int x(4);
    cout<<f(x)<<endl;
    cout<<f(x+1)<<endl;
    return 0;
}
```

### 3、阅读程序回答下面问题。

1) 程序中是否存在抽象类? 如果有是哪一个? (3 分)

**存在, Base 类**

2) 写出程序的输出结果。 (3 分)

**x=20 y=400**

**x=200 y=50**

**x=x/y=4**

```
#include<iostream>
using namespace std;
class Base
{
public:
    virtual void getxy( int i,int j = 0 ) { x = i; y = j; }
    virtual void fun() = 0 ;
protected:
    int x , y;
};
class A: public Base
{
public:
```

---

```
void fun()      {   cout<<"x = "<<x<<'\t'<<"y = x * x = "<<x*x<<endl; }
};

class B:public Base
{
    public:
        void fun()
        {
            cout<< "x = " << x << '\t' << "y = " << y << endl;
            cout << "y = x / y = " << x / y << endl; 
        }
};

int main()
{
    Base * pb; A obj1; B obj2;
    pb = &obj1;          pb -> getxy( 20 );          pb -> fun();
    pb = &obj2;          pb -> getxy( 200, 50 );       pb -> fun();
    return 0;
}
```

#### 4、阅读程序回答下面问题。

1) 程序的功能是什么？（3分）

将数组 a 中的元素写入文件 file.dat，并从文件中读取这些数，计算输出它们的最大值和最小值。

2) 写出程序的输出结果。（3分）

max=95.7

min=11

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    fstream fst("file.dat",ios::out|ios::in|ios::binary);
    if(!fst)
    {   cout<<"file.dat file not open";     exit(1); }
    double a[]={56.8,32,16,54.4,23.2,67,73,80,95.7,11};
    for(int i=0;i<10;i++)
        fst.write((char *)(&a[i]),sizeof(double));
    double x, max,min;
    int n=fst.tellg()/sizeof(double);
    fst.seekg(0);
    fst.read((char *)(&x),sizeof(double));
    max=min=x;
    for(int i=1;i<n;i++)
    {
        fst.read((char *)(&x),sizeof(double));
        if(x>max)      max=x;
        else if(x<min) min=x;
    }
}
```

---

```
    cout<<"max="<<max<<endl;
    cout<<"min="<<min<<endl;
    return 0;
}
```

5、编写一函数，求数列 1, 2, 3/2, 5/3, 8/5, .....前 n 项之和，n 由主函数中给定。

```
#include<iostream>
using namespace std;
double f1(int i){
    int a=i;
    double t1=1,t2=2,t3=0;
    if(a==1)
        return 1;
    else if(a==2)
        return 2;
    else
    {
        for(int j=2;j<a;j++){
            t3=t1+t2;
            t1=t2;
            t2=t3;
        }
        return t3;
    }
}
double f2(int n){
    int k = n;
    double s=3;
    if(k==1)
        return 1;
    else if(k==2)
        return 3;
    else{
        for(int i = 3;i<=k;i++){
            s+=f1(i)/(i-1);
        }
        return s;
    }
}
int main()
{
    int n;
    double t;
    cin>>n;
```

---

```
t=f2(n);
cout<<t;
return 0;
}
```

6、同学们，快过年了，有没有想念你的父母，还记得他们的生日吗？知道其生日是当年的第几天吗？请编写程序实现给定某个年、月、日的值，例如：2020 年 12 月 17 日。计算出这一天属于该年的第几天。要求如下：

- 1) 用面向对象的方法实现； 提示：可定义日期类 Tdate。
- 2) 闰年判断和计算第几天等需为成员函数。

```
#include <iostream>
using namespace std;
class Tdate {
private:
    int year, month, day;
public:
    int ifleap(int year) {
        if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0) {
            return 1;
        }
        else {
            return 0;
        }
    }
    int coutdata(int year, int month, int day) {
        int sum = 0;
        int a[12] = { 31,28,31,30,31,30,31,31,30,31,30,31 };
        if (ifleap(year) == 1) {
            a[1] = 29;
        }
        for (int i = 0; i < month - 1; i++) {
            sum += a[i];
        }
        sum += day;
        return sum;
    };
};
int main() {
    Tdate date;
    int year, month, day;
    cin >> year >> month >> day;
    cout << date.coutdata(year, month, day) << endl;
    return 0;
}
```