

注意：请在 D:盘建立以：“学号\_姓名”为名字的文件夹，在此文件夹下答题，最后只保留该文件夹下面的 1.cpp, 2.cpp, 3.cpp 三个文件，提交整个文件夹即可。

1. （30 分）现有文本文件 num.txt，其中有若干行整数，每行中的整数用空格隔开，行末的结束数字为 0 或 1（这两个数字只出现在行末作为特殊含义），0 表示需要对该行数字求和（SUM），1 表示需要对该行数字求平均值（AVG）。编写程序将文件 num.txt 中的数据逐个读入，根据每行结束数字 0 或 1 的要求进行计算，并将结果按行依次存放到文件 output.txt 中，实例如下图所示。（注意只能用 C++的文件库函数，不能用 C 的库函数）

输入文件 num.txt	输出文件 output.txt
2 3 4 0	SUM: 9
5 6 7 1	AVG: 6
2 4 3 0	SUM: 9
8 10 12 12 1	AVG: 10.5

num.txt

23 34 5445 6 64 55 35 0  
3434 43 4 5 4 6 6 1  
878 789 9 9 7 8 6 8 5757 0  
6757 775 76 66 464 6 1  
34434 4 2 42 3 0  
343 3443 633 4 3 1  
344 342 4 0

2. (35 分) 设有一个字符串类 MyString 的定义如下:

```
class MyString {
public:
    MyString(const char* s); //构造字符串, 使count加一
    MyString(const MyString& str); //构造字符串, 使count加一
    ~MyString(); //析构字符串, 使count减一
    char *c_str(); //访问字符串 (获取当前字符串指针)
    bool operator==(const MyString& str); //两个字符串相等比较
    char &operator[](int i); //下标访问第i个字符
    static void printCount(); //打印count的值
private:
    char* sstring; //用于记录字符串内容, 需要分配内存
    static int count; //用于记录字符串对象的个数, 初始值为0
};
```

请完成类中所有成员函数和重载运算符的实现代码, 并编写主函数对每个函数进行测试。

3. (35 分) 编写程序计算正方体、圆形和圆柱体的表面积和体积。要求:

(1) 设计一个抽象类 Object, 包含一个保护数据成员 double data (它可以作为正方体的边长、圆形的半径或圆柱体底面圆的半径), 构造函数 Object(double data), 用于求表面积 area() 和体积 volume() 的纯虚函数。

(2) 由抽象类 Object 派生出描述正方体的 Cube 类和描述圆形的 Circle 类, 再由 Circle 类派生出描述圆柱体的 Cylinder 类 (增加 height 表示高度), 在这 3 个类中都有构造函数、以及用于计算表面积和体积的虚函数的重载实现。要求 Cylinder 类的 area() 和 volume() 函数实现需要调用其父类 Circle 的 area() 函数实现 (即获得底面圆的面积, 然后继续计算)。

(3) 设计 main 函数对上述功能进行测试, 计算并输出正方体、圆形和圆柱体 3 种不同类型对象的表面积和体积。其中对派生类中 area() 和 volume() 函数的调用要求使用基类 Object 的指针进行调用。

1.

```
#include <iostream>
#include <fstream>
using namespace std;

int main() {
    ifstream infile("num.txt", ios::in);

    if (!infile) {
        cout << "Input file open error!" << endl;
        exit(0);
    }

    ofstream outfile("output.txt", ios::out);
    if (!outfile) {
        cout << "Output file open error!" << endl;
        exit(0);
    } //----- 至此10分

    int i, sum = 0, count = 0;
    while (!infile.eof()) {
        infile >> i;
        if (i == 0)
        {
            outfile << "SUM: " << sum << "\n";
            sum = 0; count = 0;
        }
        else if (i == 1)
        {
            outfile << "AVG: " << (double)sum/count << "\n";
            sum = 0; count = 0;
        }
        else
        {
            sum += i;
            count++;
        }
    } //----- 至此25分

    outfile.close();
    infile.close(); //----- 至此30分

    return 0;
}
```

2.

```
#include <iostream>
using namespace std;

class MyString {
public:
    MyString(const char* s); //构造字符串，使count加一
    MyString(const MyString& str); //构造字符串，使count加一
    ~MyString(); //析构字符串，使count减一
    char *c_str(); //访问字符串（获取当前字符串）
    bool operator==(const MyString& str); //两个字符串相等比较
    char &operator[](int i); //下标访问第i个字符
    static void printCount(); //打印count的值
private:
    char* sstring; //用于记录字符串内容
    static int count; //用于记录字符串对象的个数
};

int MyString::count = 0; //----- 3分

MyString::MyString(const char* s) { //----- 4分
    if (!s) {
        sstring = 0;
    }
    else {
        sstring = new char[strlen(s) + 1];
        strcpy(sstring, s);
    }
    count++;
}

MyString::MyString(const MyString& str) { //----- 4分
    if (!str.sstring) {
        sstring = 0;
    }
    else {
        sstring = new char[strlen(str.sstring) + 1];
        strcpy(sstring, str.sstring);
    }
    count++;
}

MyString::~MyString() { //----- 4分
```

```

        if(sstring) delete[] sstring;
        count--;
    }

char* MyString::c_str() {/------- 4分
    return sstring;
}

bool MyString::operator==(const MyString& str) {/------- 4分
    return (strcmp(sstring, str.sstring) ? false : true);
}

char &MyString::operator[](int i) {/-------4分
    return sstring[i];
}

void MyString::printCount() {/------- 2分
    cout << "Count is " << count << "\n";
}

int main() {/------- 6分
    MyString a("this is a string.");
    MyString b("I like C++.");
    MyString c(b);
    MyString::printCount();
    if (a == b) {
        cout << "a == b" << " " << endl;
    }
    else {
        cout << "a != b" << " " << endl;
    }

    if (b == c) {
        cout << "b == c" << " " << endl;
    }
    else {
        cout << "b != c" << " " << endl;
    }

    c[8] = c[9] = 'P';
    cout << "c is " << c.c_str() << "\n";
    if (b == c) {
        cout << "b == c" << " " << endl;
    }
}

```

```

        else {
            cout << "b != c" << " " << endl;
        }

        return 0;
    }
}

```

3.

```

#include <iostream>
using namespace std;

class Object { //----- 6分
protected:
    double data;

public:
    Object(double data) {
        Object::data = data;
    }
    virtual double area() = 0;
    virtual double volume() = 0;
};

class Cube :public Object { //----- 6分

public:
    Cube(double data) :Object(data) {};
    double area() {
        return data * data * 6;
    }

    double volume() {
        return data * data * data;
    }
};

class Circle : public Object { //-----6分

public:
    Circle(double radius) :Object(radius) {};
    double area() {
        return 3.1416 * data * data;
    }
    double volume() {

```

```

        return 0;
    }
};

class Cylinder :public Circle { //----- 10分
    double height;

public:
    Cylinder(double radius, double height) :Circle(radius) {
        Cylinder::height = height;
    }

    double area() {
        return 2 * 3.1416 * data * height + 2 * Circle::area(); //----注意此处得分点
    }

    double volume() {
        return Circle::area() * height; //----注意此处得分点
    }
};

int main() { //-----7分
    Object* p;
    Cube obj1(1);
    Circle obj2(1);
    Cylinder obj3(1, 1);
    p = &obj1;
    cout << "the area of the cube is: " << p->area() << endl;
    cout << "the volume of the cube is: " << p->volume() << endl;

    p = &obj2;
    cout << "the area of the Circle is: " << p->area() << endl;
    cout << "the volume of the Circle is: " << p->volume() << endl;

    p = &obj3;
    cout << "the area of the Cylinder is: " << p->area() << endl;
    cout << "the volume of the Cylinder is: " << p->volume() << endl;

    return 0;
}

```