专业名称：计算机技术、软件工程　科目名称：程序设计实践

说明：（1）本次考试共包括五道编程题，每题25分。程序完全符合要求且得到正确结果，则可能得到满分；如果编程思路正确但没有得到正确的结果，则总得分不超过20分。

（2）如不能上机完成编成任务，请在答题纸上写出编程思路；

（3）上机时间两个小时；

1、编写一个程序，输入a、b、c三个值,输出其中最大值。

2、设圆半径r=1.5，圆柱高h=3，求圆周长、圆面积、圆球表面积、圆球体积、圆柱体积,取小数点后两位数字，请编程序。

3、对于一个给定的，不超出long int范围的正数，只使用一次循环，计算以下三件事并输出结果：

1) 求出它是几位数。

2) 将他的每一位数字分别输出，数字间用连字符分割。

3) 最后按照逆序输出每一位数字。

例如，如果给定的数字是“123”，则应输出以下结果：

3

1-2-3

321

4、建立一个用数字作为密码，存放一个字符串的保险箱类，他具有以下成员函数：

1) 一个构造函数，接受一个数字作为初始密码。

2) 另一个构造函数，没有指定初始密码，此时初始密码默认为零（此处不允许使用默认参数）。

3) 一个开箱门函数，给定一个密码，如果密码正确，则保险箱开箱。否则保持状态不变。

4) 一个锁箱门函数，没有参数，将保险箱锁定。

1. 编写一个程序，输入a、b、c三个值,输出其中最大值。

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

int main() {
```

```
    int a, b, c;

    // 输入三个值
    cout << "请输入三个整数: ";
    cin >> a >> b >> c;

    // 找出最大值
    int max_value = a;
    if (b > max_value) {
        max_value = b;
    }
    if (c > max_value) {
        max_value = c;
    }

    // 输出最大值
    cout << "最大值是: " << max_value << endl;

    return 0;
}
```

2. 设圆半径r为1.5，圆柱高h=3，求圆周长、圆面积、圆球表面积、圆球体积、圆柱体积,取小数点后两位数字，请编程序。

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

int main() {
    double r = 1.5; // 半径
    double h = 3.0; // 圆柱高
    double pi = 3.141592653589793; // 手动定义π值

    // 计算圆周长
    double circumference = 2 * pi * r;

    // 计算圆面积
    double area = pi * r * r;

    // 计算圆球表面积
    double sphere_surface_area = 4 * pi * r * r;

    // 计算圆球体积
    double sphere_volume = (4.0 / 3.0) * pi * r * r * r;

    // 计算圆柱体积
    double cylinder_volume = pi * r * r * h;

    // 输出结果，保留两位小数
    printf("圆周长: %.2f\n", circumference);
    printf("圆面积: %.2f\n", area);
    printf("圆球表面积: %.2f\n", sphere_surface_area);
    printf("圆球体积: %.2f\n", sphere_volume);
    printf("圆柱体积: %.2f\n", cylinder_volume);

    return 0;
}
```

3. 对于一个给定的，不超出 long int 范围的正数，只使用一次循环，计算以下三件事并输出结果:

   1)求出它是几位数。

   2)将他的每一位数字分别输出，数字间用连字符分割。

3)最后按照逆序输出每一位数字。

例如，如果给定的数字是“123”，则应输出以下结果:

3

1-2-3

321

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

int main() {
    long int num;
    cout << "请输入一个正整数: ";
    cin >> num;

    // 处理特殊情况：如果输入是0
    if (num == 0) {
        cout << "1\n"; // 0是1位数
        cout << "0\n"; // 输出0
        cout << "0\n"; // 逆序输出0
        return 0;
    }

    int digitCount = 0; // 记录数字的位数
    long int reverseNum = 0; // 用于存储逆序数字
    string digits = ""; // 用于存储每一位数字并用连字符分割

    // 一次循环完成所有操作
    while (num > 0) {
        int digit = num % 10; // 获取当前最低位数字
        reverseNum = reverseNum * 10 + digit; // 构建逆序数字
        digits = to_string(digit) + (digits.empty() ? "" : "-") + digits; // 构建连字符分割的字符串
        num /= 10; // 去掉最低位
        digitCount++; // 位数加1
    }

    // 输出结果
    cout << digitCount << endl; // 输出位数
    cout << digits << endl; // 输出连字符分割的每一位数字
    cout << reverseNum << endl; // 输出逆序数字

    return 0;
}
```

4. 建立一个用数字作为密码，存放一个字符串的保险箱类，它具有以下成员函数：
    1. 一个构造函数，接受一个数字作为初始密码。
    2. 另一个构造函数，没有指定初始密码，此时初始密码默认为零（此处不允许使用默认参数）。
    3. 一个开箱门函数，给定一个密码，如果密码正确，则保险箱开箱。否则保持状态不变。
    4. 一个锁箱门函数，没有参数，将保险箱锁定。
    5. 一个更新内容函数，在开箱的情况下，更新保险箱内存放的字符串。
    6. 一个取出内容函数，在开箱的情况下，在屏幕上输出存放的字符串。
    7. 一个更改密码函数，在开箱的情况下，更新保险箱的密码。

    根据以上要求，完善成员变量，建立一个保险箱类并生成一个该类的实例，简单调用演示各函数。

```cpp
#include <iostream>
#include <string>

```

```cpp
class SafeBox {
private:
    int password;
    bool isOpen;
    std::string boxContent;

public:
    // 构造函数，接受一个数字作为初始密码
    SafeBox(int p) : password(p), isOpen(false), boxContent("") {}

    // 构造函数，没有指定初始密码，初始密码默认为零
    SafeBox() : password(0), isOpen(false), boxContent("") {}

    // 开箱门函数，给定一个密码，如果密码正确，则保险箱开箱
    void open(int p) {
        if (p == password) {
            isOpen = true;
            std::cout << "保险箱已打开" << std::endl;
        } else {
            std::cout << "密码错误，保险箱未打开" << std::endl;
        }
    }

    // 锁箱门函数，将保险箱锁定
    void lock() {
        isOpen = false;
        std::cout << "保险箱已锁定" << std::endl;
    }

    // 更新内容函数，在开箱的情况下，更新保险箱内存放的字符串
    void updateContent(const std::string& newContent) {
        if (isOpen) {
            boxContent = newContent;
            std::cout << "保险箱内容已更新" << std::endl;
        } else {
            std::cout << "保险箱未打开，无法更新内容" << std::endl;
        }
    }

    // 取出内容函数，在开箱的情况下，在屏幕上输出存放的字符串
    void retrieveContent() {
        if (isOpen) {
            std::cout << "保险箱中的内容为: " << boxContent << std::endl;
        } else {
            std::cout << "保险箱未打开，无法取出内容" << std::endl;
        }
    }

    // 更改密码函数，在开箱的情况下，更新保险箱的密码
    void changePassword(int newPassword) {
        if (isOpen) {
            password = newPassword;
            std::cout << "保险箱密码已更改" << std::endl;
        } else {
            std::cout << "保险箱未打开，无法更改密码" << std::endl;
        }
    }
};

int main() {
    SafeBox mySafe(1234);
```

```
    mySafe.open(1234);
    mySafe.updateContent("这是保险箱内的初始内容");
    mySafe.retrieveContent();
    mySafe.changePassword(5678);
    mySafe.lock();
    mySafe.open(5678);
    mySafe.retrieveContent();
    return 0;
}
```

学硕

1. **输出从 2-500 之间的质数。**

方式一：简单暴力法

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

int main() {
    for (int i = 2; i <= 500; ++i) {
        bool isPrime = true;
        // 从 2 到该数的平方根进行检查
        for (int j = 2; j <= sqrt(i); ++j) {
            if (i % j == 0) {
                isPrime = false;
                break;
            }
        }
        if (isPrime) {
            cout << i << " ";
        }
    }
    cout << endl;
    return 0;
}
```

方式二：埃拉托斯特尼筛法（Sieve of Eratosthenes）

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

int main() {
    vector<bool> isPrime(501, true);
    // 0 和 1 不是质数
    isPrime[0] = isPrime[1] = false;

    for (int i = 2; i * i <= 500; ++i) {
        if (isPrime[i]) {
            // 将 i 的倍数标记为非质数
            for (int j = i * i; j <= 500; j += i) {
                isPrime[j] = false;
            }
        }
    }

    for (int i = 2; i <= 500; ++i) {
```

```
        if (isPrime[i]) {
            cout << i << " ";
        }
    }
    cout << endl;
    return 0;
}
```

2. **先建立一个“房间类”，用纯虚函数定义计算房间面积函数，再建立一个“方形房间类”和“圆形房间类”，继承“房间类”，并分别对虚函数重新定义实现各自的面积计算。再建立一个“房租缴费合同的类”，其中有变量：（单位面积的租金，单位面积的佣金，租住期限）。该类有个构造函数，分别实现对上面三个变量（租金，佣金，期限）的初始化，还需建立一个公用函数，实现计算房租缴费的总额【计算公式：（单位面积租金 * 租住期限+单位面积佣金）* 房子面积）】，其中房子面积的参数接收来自房间类的实例。 请分别建立以上各类的实例，并计算出方形和圆形房间应缴纳费用的总额。**

```
#include <iostream>
#include <cmath>

// 房间类，作为基类
class Room {
public:
    // 纯虚函数，用于计算房间面积
    virtual double calculateArea() const = 0;
    virtual ~Room() {}
};

// 方形房间类，继承自房间类
class SquareRoom : public Room {
private:
    double sideLength;  // 方形房间的边长
public:
    // 构造函数，初始化边长
    SquareRoom(double side) : sideLength(side) {}

    // 重写计算面积的虚函数
    double calculateArea() const override {
        return sideLength * sideLength;
    }
};

// 圆形房间类，继承自房间类
class CircularRoom : public Room {
private:
    double radius;  // 圆形房间的半径
public:
    // 构造函数，初始化半径
    CircularRoom(double r) : radius(r) {}

    // 重写计算面积的虚函数
    double calculateArea() const override {
        return M_PI * radius * radius;
    }
};

// 房租缴费合同类
class RentContract {
private:
    double rentPerUnitArea;  // 单位面积的租金
    double commissionPerUnitArea;  // 单位面积的佣金
```

```
    int rentalPeriod;  // 租住期限

public:
    // 构造函数，初始化租金、佣金和期限
    RentContract(double rent, double commission, int period)
        : rentPerUnitArea(rent), commissionPerUnitArea(commission), rentalPeriod(period)
{}

    // 计算房租缴费总额的函数
    double calculateTotalCost(const Room& room) const {
        double area = room.calculateArea();
        return (rentPerUnitArea * rentalPeriod + commissionPerUnitArea) * area;
    }
};

int main() {
    // 创建方形房间实例，边长为 5
    SquareRoom squareRoom(5);
    // 创建圆形房间实例，半径为 3
    CircularRoom circularRoom(3);
    // 创建房租缴费合同实例，单位面积租金为 10，单位面积佣金为 2，租住期限为 3 个月
    RentContract contract(10, 2, 3);

    // 计算方形房间的房租缴费总额
    double squareRoomCost = contract.calculateTotalCost(squareRoom);
    // 计算圆形房间的房租缴费总额
    double circularRoomCost = contract.calculateTotalCost(circularRoom);

    // 输出方形房间的房租缴费总额
    std::cout << "方形房间应缴纳费用的总额: " << squareRoomCost << std::endl;
    // 输出圆形房间的房租缴费总额
    std::cout << "圆形房间应缴纳费用的总额: " << circularRoomCost << std::endl;

    return 0;
}
```