题目描述：

本次实验实现一个涉及结算功能的停车场系统，以用户、车辆和停车卡为核心进行 管理，接收进车和出车，并进行计费。本系统有以下特性：

（1）不同车辆价格不同： 汽车：五座及以下汽车每小时五元，五座以上每小时十元； 自行车：停车前两小时免费，之后每小时一元。

（2）使用不同停车卡消费可享受不同折扣：普通卡不享受折扣，月卡享受 8 折，年卡 享受 7 折。每次结算都从停车卡中扣除相应金额，不考虑卡内余额不足的情况。

（3）车与卡都与唯一用户关联，为简单起见，假设不存在重名用户，每人有且仅有一张 停车卡。 **（但是每个人可能有多辆vehicle）**

注意事项：

- 请根据要求自行设计所有的类，并满足下述接口要求。
- 请注意类的初始化和析构以保证程序运行安全。
- 请注意动态绑定，抽象类和虚基类的使用。
- 请提交提交 Card.h,Card.cpp,Vehicle.h,Vehicle.cpp, ParkSystem.h, ParkSystem.cpp 文件。
- 注意编码格式为 utf-8，需 zip 打包提交。
- 注意提交的文件中不要包含 main 函数。
- **前两个测试样例分别测试Card和Vehicle。**
- 如果需要用到基本数据类型转换成string，可以使用**to_string(int/double/...)**

第一部分：获取卡的折扣和余额

请实现 Card 类，将其作为基类，并实现 MonthCard、YearCard 两个派生类，分别对应 普通卡、月卡和年卡三种类型，并满足以下要求：

1. **卡对象的构造**需要依次提供卡号、姓名、余额三个变量，其中卡号和姓名是 string 类型，余额为 double 类型。
2. 实现 **get_balance()**函数获取卡内余额。
3. 实现**get_discount()**函数获取不同类型的卡对应的折扣率（三种卡的折扣分别取1， 0.8，0.7）。

调用接口示例：

```
    Card* card1 = new Card("card001", "xiaobing", 100);
    Card* monthcard1 = new MonthCard("monthcard001", "xiaohong", 50);
    Card* yearcard1 = new YearCard("yearcard001", "xiaoming", 80);
    cout << card1->get_balance() << endl;
    cout << monthcard1->get_balance() << endl;
    cout << yearcard1->get_balance() << endl;
    cout << card1->get_discount() << endl;
    cout << monthcard1->get_discount() << endl;
    cout << yearcard1->get_discount() << endl;
    delete card1;
    delete monthcard1;
    delete yearcard1;
```

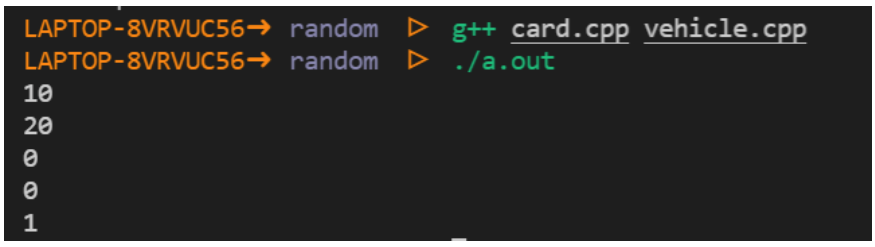
问题二：计算车辆停车费用

请实现 Vehicle 作为抽象基类。并实现 Car、Bicycle 两个派生类，分别对应汽车和自 行车两种类型。并满足以下要求：

1. 车辆对象的构造需依次提供车牌号，（拥有者）姓名。
2. 汽车对象的构造需额外提供座位数（int 类型）。
3. 实现 **get_charge(int arrive_time, int depart_time)函数**，返回类型为 double，在不 考虑折扣的情况下，计算该车辆的停车费用（此处为简化计算，所有时间已表示为从某 时刻开始的小时数，记为整数类型）。

调用接口示例：

```
    Vehicle* car1 = new Car("suA8888", "xiaoming", 5);
    Vehicle* car2 = new Car("suA9999", "xiaohong", 7);
    Vehicle* bicycle1 = new Bicycle("ofo001", "xiaoming");
    Vehicle* bicycle2 = new Bicycle("ofo002", "xiaohong");
    Vehicle* bicycle3 = new Bicycle("ofo003", "xiaogang");
    cout << car1->get_charge(8, 10) << endl;
    cout << car2->get_charge(8, 10) << endl;
    cout << bicycle1->get_charge(8, 9) << endl;
    cout << bicycle2->get_charge(8, 10) << endl;
    cout << bicycle3->get_charge(8, 11) << endl;
    delete car1;
    delete car2;
    delete bicycle1;
    delete bicycle2;
    delete bicycle3;
```



问题三：停车场计费

第三问可以修改前两问的文件，加上你需要的接口

建议使用STL相关容器。

设计停车场系统 ParkSystem 类，需要完成以下功能：

1. 向停车场系统中加入卡：**add_card(Card *card)**

2. 车辆 vehicle 在 time 时间驶入: **vehicle_arrive(Vehicle *vehicle, int time)** 安排车进入车位，记录车辆进入时间，**返回"Successfully Arranged\n"**的string。
3. 车辆 vehicle 在 time 时间离开: **vehicle_depart(Vehicle *vehicle, int time)** 完成结账（从卡的余额中扣除相应金额，需要考虑相应的折扣），**返回"车牌号:金 额\n"**的string，最后出车。金额用double即可。
4. 打印停车场当前状态：**print_status()** 用一个变量string str存储这个函数中所有需要打印的内容。（下面所说的打印都是将字符串加到str后面）

   首先先**打印"ParkSystem:\n"**，对于拥有Card的所有用户进行遍历，打印**姓名+"\n"**，**"card:"+卡号+"\n"**。

   然后打印遍历到的**当前客户在停车场中所停的车辆的车牌号**+"\n"。最后打印当前停车场的card数量，停车数量，以及停车场的总收入，分别用空格分隔，最后加上"\n"。

   这里的打印顺序：

   1. **打印Card的相关信息按照new的顺序。**（即遍历的时候按照new的顺序遍历card用户）
   2. **打印vehicle的顺序按照arrive的顺序。**

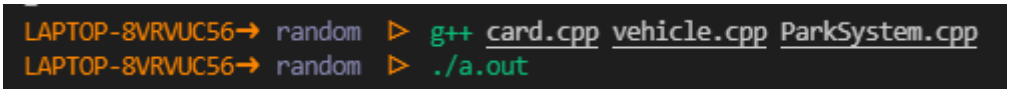
调用接口示例：

```
ParkSystem* park_system = new ParkSystem();
Card* monthcard1 = new MonthCard("monthcard001", "xiaohong", 50);
Card* yearcard1 = new YearCard("yearcard001", "xiaoming", 80);
Card* card1 = new Card("card1", "xiaoli", 60);

Vehicle* car1 = new Car("suA8888", "xiaohong", 5);
Vehicle* bicycle2 = new Bicycle("ofo002", "xiaoming");
Vehicle* car2 = new Car("suA9999", "xiaoming", 6);
string str = "";
park_system->add_card(monthcard1);
park_system->add_card(yearcard1);
park_system->add_card(card1);
str+=park_system->print_status();
str+=park_system->vehicle_arrive(car1, 8);
str += park_system->print_status();
str += park_system->vehicle_arrive(bicycle2, 14);
str += park_system->print_status();
str += park_system->vehicle_depart(car1, 10);
str += park_system->vehicle_depart(bicycle2, 19);
str += park_system->vehicle_arrive(car2, 6);
str += park_system->vehicle_arrive(bicycle2, 7);
str += park_system->print_status();
cout << str;
delete park_system;
delete monthcard1;
delete yearcard1;
delete card1;
delete car1;
delete bicycle2;
delete car2;
//system("pause");
```

如果用VS需要加上system("pause");

输出如下：




```
ParkSystem:
xiaohong
card:monthcard001
xiaoming
card:yearcard001
xiaoli
card:card1
3 0 0.000000
Successfully Arranged
ParkSystem:
xiaohong
card:monthcard001
suA8888
xiaoming
card:yearcard001
xiaoli
card:card1
3 1 0.000000
Successfully Arranged
ParkSystem:
xiaohong
card:monthcard001
suA8888
xiaoming
card:yearcard001
ofo002
xiaoli
card:card1
3 2 0.000000
suA8888:8.000000
ofo002:2.100000
Successfully Arranged
Successfully Arranged
ParkSystem:
xiaohong
card:monthcard001
xiaoming
card:yearcard001
suA9999
ofo002
xiaoli
card:card1
3 2 10.100000
```

Tips:

如果你需要快速比较文件是否相同，可以在shell下使用

```
diff <(cat filename1) <(cat filename2)
```

其中hello.txt中内容为hello，helloworld.txt中的内容为hello world。