Introduction

Ahmad is managing a car parking garage. Ahmad is responsible for allowing cars to enter the parking and collecting payments when cars leave. Write a C++ program that helps Ahmad manage the parking more efficiently. Once the user (Ahmad) runs the program, the program shall ask the user to enter the price of parking per hour. This price will be used to calculate the amount that shall be collected by Ahmad when a car leaves the parking garage. Then, the program shall provide the user with the following options by showing a menu. The user can choose one of these options by entering its number (1, 2, 3…). Once the user is done with one operation, the program shall show the menu again to allow the user to execute another operation. Ahmad shall be able to execute as many operations as he wants.

Tips

- Cars are charged for whole hours. For example, if a car stays in the garage for 15 minutes, the car will be charged the price of one hour. And if a car stays in the garage for 1 hour and 5 minutes, the car will be charged the price of two hours.
- Parking slots are numbered using a combination of characters and numbers. An example of parking slot number is A-123C.
- You do not have to maintain the list of parking slots in your application. The user can enter any number (E.g. A-123C) when asked to enter a parking slot number.
- You shall use hash tables for storing reservations. The key to be used is parking slot number.

Available Options

1. Check-In Car: This option is executed whenever a car enters the parking garage. When the user chooses this option, the program shall ask the user to enter the car plate number and the parking slot number. The program shall determine the current time by calling the function getCurrentTime() function that is already implemented inside the template file. This function will return the number of minutes since 12:00 AM. Once the user enters this data, the program shall check if the parking slot is available. If it is available, the program shall reserve the parking slot, and show a message that slot has been reserved. Otherwise, the program shall show an error message that parking slot is already busy.

2. Print Parking Slot Info: When the user chooses this option, the program shall ask the user to enter the parking slot number. Then, the program shall show whether the parking slot is available. If it is available, the program shall also show how long the car has been in the parking slot, as well as the amount due for the parking slot.

3. Checkout Car: This option should be executed when a car leaves the parking. When the user chooses this option, the program shall ask the user to enter the parking slot number. Then, the program shall show how much money shall be collected. Once executed, this operation shall make the parking slot available for other cars.

4. Print Total Income: When the user chooses this option, the program shall show how much money was collected so far. It aims at checking whether Ahmad has the right amount of money in his pocket.

5. Print Cars in Parking: When the user chooses this option, the program shall print the plate number of each car that is already in the parking garage. This operation shall also print the parking slot number, and how long the car has been in the parking garage.

Implementation

In order to develop the proposed program, you will need to create the following classes and structures.

1. **reservationType struct:** An instance of this structure will contain data about a single parking slot reservation. This class will contain the following member variables: parkingSlotNumber (String), carPlateNumber (String), and arrivalTime (int). These member variables must be passed to the constructor and they cannot be changed after that. You can get the arrivalTime by calling the getCurrentTime function.

2. **nodeType struct:** An instance of this structure will represent a single node in the linkedList that is referenced by each element of the hash table (The hash table will use separate chaining to handle collisions). This node will contain two member variables: reservation (of type reservationType*), and link (of type nodeType*).

3. **parkingGarageHashTable:** This class shall be implemented as a hash table. It will contain the business logic of the parking garage. This class shall expose the following methods, which will be called from the main function:

//The constructor will accept the price per hour so that the checkout method calculates the amount that shall be collected. The passed parameter shall be stored in a class member variable (E.g. double pricePerHour)

public parkingGarageHashTable(double price);

//This method shall reserve the parking slot for the car if it is available. If the parking slot is available, the method shall return true. If the parking slot is not available (Already reserved by another car), the method shall return false.

public bool checkIn(String carPlateNumber, String parkingSlotNumber);

//This method shall return the reservation that already exists on a specific parking slot. If no car is parking in the specified parking slot, this method shall return null.

public reservation* getReservation(String parkingSlotNumber);

//This method shall remove the car from a parking slot, so that parking slot becomes available for other cars. This method shall return the amount that shall be collected from the customer. In order to calculate the amount of money, you have to calculate how long the car has been in the garage. To do that, you call the function getCurrentTime to get the current time. Then you calculate (current – reservation.arrivalTime). This formula will return the number of minutes. Finally, you convert the minutes into whole hours. For example, 65 minutes shall become 2 hours. This method shall add the calculated amount to the totalAmount member variable.

public double checkout(int parkingSlotNumber);

//This method shall return the total amount of money that has been collected so far. This amount is supposed to be stored in the member variable (E.g. totalAmount)

public double getTotalAmount();

//This method shall return all active reservations

public reservation[] getAllReservations();

Submission

Please use the attached template to develop the program, and submit one c++ code file that contains all classes along with the main function.