Zendesk Backend Exercise

Problem Statement

Your task is to design an automated valet car parking system where you manage a parking space for vehicles as they enter/exit it and manage its revenue. The data provided to you(refer to the Data Description section below) would include the number of parking lots available and the entry/exit details of each vehicle. For the purpose of this question you can assume that Cars and Motorcycles are the only two types of vehicles.

Each vehicle upon entry can only park in a lot available for that vehicle type. If there are no lots available for that vehicle type, it should be denied an entry into the space. As we're building a valet car park, all the lots in the parking space can be considered as being distinctly numbered eg: CarLot1, CarLot2,..., MotorcycleLot1, MotorcycleLot2,.... Each vehicle upon entering is allotted to the lot with the lowest number for that vehicular type eg: a car entering a parking space with the available lots CarLot2, CarLot4, CarLot5 would be assigned to CarLot2. When a vehicle wants to exit the car park, the system will return the parking lot that the vehicle will be removed from and charge them an appropriate parking fee(rounded up to the nearest hour, i.e., 1hr 1min is charged as 2hr): $1/hour for a motorcycle and $2/hour for a car.

You may use any language/framework for this exercise.

Data Description

Your program accepts an input file as an argument when running. The format of the file is described below:

```
3 4
Enter motorcycle SGX1234A 1613541902
Enter car SGF9283P 1613541902
Exit SGX1234A 1613545602
Enter car SGP2937F 1613546029
Enter car SDW2111W 1613549730
Enter car SSD9281L 1613549740
Exit SDW2111W 1613559745
```

The first line indicates the number of parking slots that are present in the space for Cars and Motorcycles respectively in the parking lot.

For each subsequent line, there would be two types of events:

- Vehicle entering the space: `Enter <motorcycle|car> <vehicle number> <timestamp>`. The program should print out either `accept` or `reject` based on the availability of lots in the parking space. If the vehicle is accepted, the program also returns the name of the lot being occupied by it.
- Vehicle exiting the space: `Exit <vehicle number> <timestamp>`. The program prints out the released lot and the parking fee.

Given the example above, the program output would look like:

```
Accept MotorcycleLot1
Accept CarLot1
MotorcycleLot1 2
Accept CarLot2
Accept CarLot3
Reject
CarLot3 6
```

Any erroneous cases should be handled with an appropriate error message.

Submission

Your submission should be well packaged and have clear instructions on how to run your code. You may use any language/framework for this exercise. You can assume that the code will be tested on a fresh installation of Ubuntu 16.04 and will not have any other editors/compilers/modules installed. Any additional packages needed need to be specified in your documentation.

Evaluation

Your submission will be judged on:

- code quality and architecture
- correctness of the solution
- extensibility of the design
- good programming practices such as clean code, clear comments, tests
- code packaging including project structure

Notes

1. The input provided in this example is just a sample representation of the possible inputs. It's up to you how you want to structure your implementation.

2. Other than the problem statement listed above, please make any and all design decisions needed for your implementation. Assumptions made should be listed in your readme or code.