SOFTWARE ENGINEERING ACADEMY
COMPFEST X - GO-JEK
SECOND STAGE FINAL ASSIGNMENT

A. Problem Description

You are going to make a command line app called 'go-cli'.

1. Your go-cli app can be executed in three ways:

1.1. By running go-cli without any parameters

In this case, your app will generate a 20 * 20 map, put 5 drivers at random coordinates, and put your user at a random coordinate.

1.2. By running go-cli and passing three parameters

In this case, the first parameter (let say n) would be the size of the map (n * n), the second and the third parameter (let say x and y) will be the coordinate of your user (x, y). Your app still need to generate 5 drivers and put them randomly on the map.

1.3. By passing a filename as go-cli parameter

In this case, your app will generate a map with size as specified in the file, put your user in the coordinate as specified in the file, and put the number of drivers and their positions as specified in the file as well. You are free to decide the format of data in the file, but you have to be able to argue your choice of design of this data format.

2. Once the app is running, user will be prompted choices of action.

2.1. Show Map

When user chooses this action, your app will display the map and show where the user is and where the drivers are.

2.2. Order Go Ride

When user chooses this action, your app will ask user's destination. User should input the coordinate that he/she wants to go to. After that, your app will allocate the nearest driver to user and display the name of the driver, the route to destination, as well as the price estimate. You should note that from user's position to user's destination, driver can't go diagonally. So, for example, if user starts from (1,1) and wants to go to (2,2), the route would be:

- start at (1,1)
- go to (1,2)
- turn right
- go to (2,2)
- finish at (2,2)

You should also design your app so that the price can be modified easily. The formula for the price is: unit cost * unit of distance. Going from

(1,1) to (1,2) counts as 1 unit of distance. If unit cost is set as 300, for instance, price estimate for our route above is 600 (300 * 2 units of distance).

After displaying all this information, user will be asked to confirm the trip. If user confirms, trip is marked as completed.

Once the trip is completed, the information of driver's name, route, and price is stored in a text file. Again, you are free to choose the data format to store this file as long as you can rationalise your design decision.

2.3. View History

If user chooses this action, your app will display all trips history of the users. If no history is recorded yet, your app will display nothing.

B. Additional Information

Create a local git repository containing your code submission. You have to include a README file in your git repository submission. Write down any assumptions that you make when you create your app in this file. Also, write down any design decisions that you make.

We suggest you to commit your progress to your local repository often. When you think you are ready, zip your local repository and submit it to the submission slot in COMPFEST X Website. Make sure the hidden .git folder is included in the zip file that you send.

Your submission will be judged by the following items:

1. **Completion**

   The more complete the features that you write is the better.

2. **Design Decision**

   The more you apply the SOLID principle (you may want to Google that) in your app is the better. Also the clearer you explain any design decisions you make is the better.

3. **Edge cases**

   The more edge cases that your app can handle is the better.