

Homework #1

RELEASE DATE: 03/01/2013

DUE DATE: 03/15/2013, 12:00

As directed below, you need to submit your code to the designated place on the course website.

Any form of cheating, lying or plagiarism will not be tolerated. Students can get zero scores and/or get negative scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts.

Discussions on course materials and homework solutions are encouraged. But you should write the final solutions alone and understand them fully. Books, notes, and Internet resources can be consulted, but not copied from.

Since everyone needs to write the final solutions alone, there is absolutely no need to lend your homework solutions and/or source codes to your classmates at any time. In order to maximize the level of fairness in this class, lending and borrowing homework solutions are both regarded as dishonest behaviors and will be punished according to the honesty policy.

You need to write your homework report in English. For programming, we only allow Java.

This homework set comes with 200 points and 25 bonus points. In general, every homework set of ours would come with a full credit of 200 points.

1 Description

In this homework, you are asked to implement a highway simulator to simulate the traffic jam that happens in every lunar-new-year vacation. The idea of this homework comes from this interesting description.

<http://disp.cc/b/27-3ple>

Don't worry. We are not here to torture you (yet :-P). We ask you to run a simplified task, as described below.

- (1) We will assume the simulation to be run on discrete time and space measures. That is, you only need to consider integer-valued units of time/distance.
- (2) We will assume the highway to be a single-lane one with a finite-sized interval (well, is there an infinitely long highway?). The length of the interval would be an integer value in [100, 1000], and will be given to your program as an input.
- (3) We will assume the speed of car to be a non-negative integer.

In this homework, you need to implement a program for the following three tasks.

- (1) Simulate a single-lane highway, where the number of waiting cars at the beginning (distance mark 1) of the highway at time step 1 will given as a parameter. All the cars on the highway determines its speed at the next time step by the following constraints:
 - highest speed: The driver should obey the highway speed limit, which is 4.
 - safe distance: The driver should keep a distance of at least twice the value of the current car speed to the car in the immediate front.
 - highway respect: The driver should drive as fast as possible under the constraints above.
 - reaction time for speeding down: 0 time step (if a driver decides to speed down, the change happens immediately at the 1st-next time step)
 - reaction time for speeding up: 1 time step (if a driver decides to speed up, she/he holds the same speed at the 1st-next time step, and then the changes happens at the 2nd-next time step)

At most one waiting car at the beginning can join the highway at any time step immediately (without being restricted by the reaction time for speeding up) if and only if the car can take a speed of at least 1 while keeping the safe distance to the car in the immediate front. At the end of the highway, the car would just go down safely and disappear.

- (2) Simulate with the same scenario above, but with the first driver getting a sudden headache and slowing her/his car down to half of the initial speed at time step 5. The driver then recovers immediately at time step 6.
- (3) Simulate with the same scenario as the first task, while adding an interchange at distance mark 50. Your program will be given the number of cars waiting at the interchange. A waiting car at the interchange can join the main lane at any time step immediately (without being restricted by the reaction time for speeding up) if and only if the car will not bump into any existing cars (i.e. there is no car “occupying” distance mark 50) and the car can take a speed of at least 1 while keeping the safe distance to the car in the immediate front.

Your whole program shall be executed as

```
java CarSimulation task_number highway_len num_car_at_beginning [num_car_at_intersection]
```

such as

```
java CarSimulation 2 1000 200
```

or

```
java CarSimulation 3 1000 200 100
```

You are asked to output a snapshot of the lane for 200 time steps. Each line of the snapshot looks like

```
...x.....x.....
```

which represents the lane with `highway_length` characters. The character `.` means no car and `x` means one car. If anytime a “traffic accident” happens, the simulation stops.

You are suggested to implement classes of `Car` and `Highway`. But we also welcome you to think about other designs.

2 Requirements

- (135 points) Finish the tasks described above.
- (25 points) Write some comments for your code, and generate the document with `javadoc`.
- (40 points) Write a short report with at **most** two A4 pages in English that contains the following items:
 - (1) your name and school ID
 - (2) how you tested the correctness of your program
- (Bonus 25 points) Other than the basic tasks above, you can use your creativity to design a better program. We reserve 25 points for anything “additional” to the tasks described above, such as simulating a multiple-lane highway. If you write anything worth getting the bonus, you are allowed to explain it with at **one** additional A4 page in English.

You should submit your report in **PDF** format. See <http://jsc.cc.ntu.edu.tw/ntucc/pcroom/manual/Word2Pdf.htm> for some possible instructions for converting from Word to PDF. You may get no points for submitting a non-PDF report.

3 Submission File

Please upload a single ZIP encrypted file to CEIBA. The zip file should be like `b86506054.zip`, where the file name should be changed to your own school ID. The ZIP file should contain the following items:

- `src/`, your source code
- `class/`, the class files compiled from your source code
- `doc/`, which is the file generated from `javadoc -d doc/ src/*.java`
- a PDF file `report.pdf`, which is your report file written in English
- a text file `README`, which contains directions on how to compile your file
- an optional text file `MEDAL`, in case you want to use the gold medals, with the number of medals listed as a single number in the first line of the file. Use your medals wisely—usage cannot be retracted.