CS1515
Foundations of Computing Science 2

Individual Programming Exercise (Practicals 10-12)

You are required to write a Java program which meets the specification set out below. It must be submitted using the normal submission mechanism by 12 noon on Friday May 13th.

Lateness

The standard Computing Science penalties for lateness will apply:

- Work submitted up to **one day late** attracts a **penalty of 10%**
- Work submitted up to **one week late** attracts a **penalty of 25%**
- Work handed in **more than one week late** is marked and returned but is counted as a **'No Paper'**

Copying

You should be aware that copying another student's work (plagiarism) can lead to disciplinary action being taken.

Read carefully

- the Departments general statement on Assessment
  http://www.csd.abdn.ac.uk/teaching/handbook/both/info.php?filename=assessment.txt
- the University's Code of Practice on Plagiarism
  http://www.abdn.ac.uk/registry/quality/appendix5x15.pdf
- the Department's Guidance on how to avoid Plagiarism
  http://www.csd.abdn.ac.uk/teaching/handbook/both/info.php?filename=cheating.txt

Road Pricing

Background

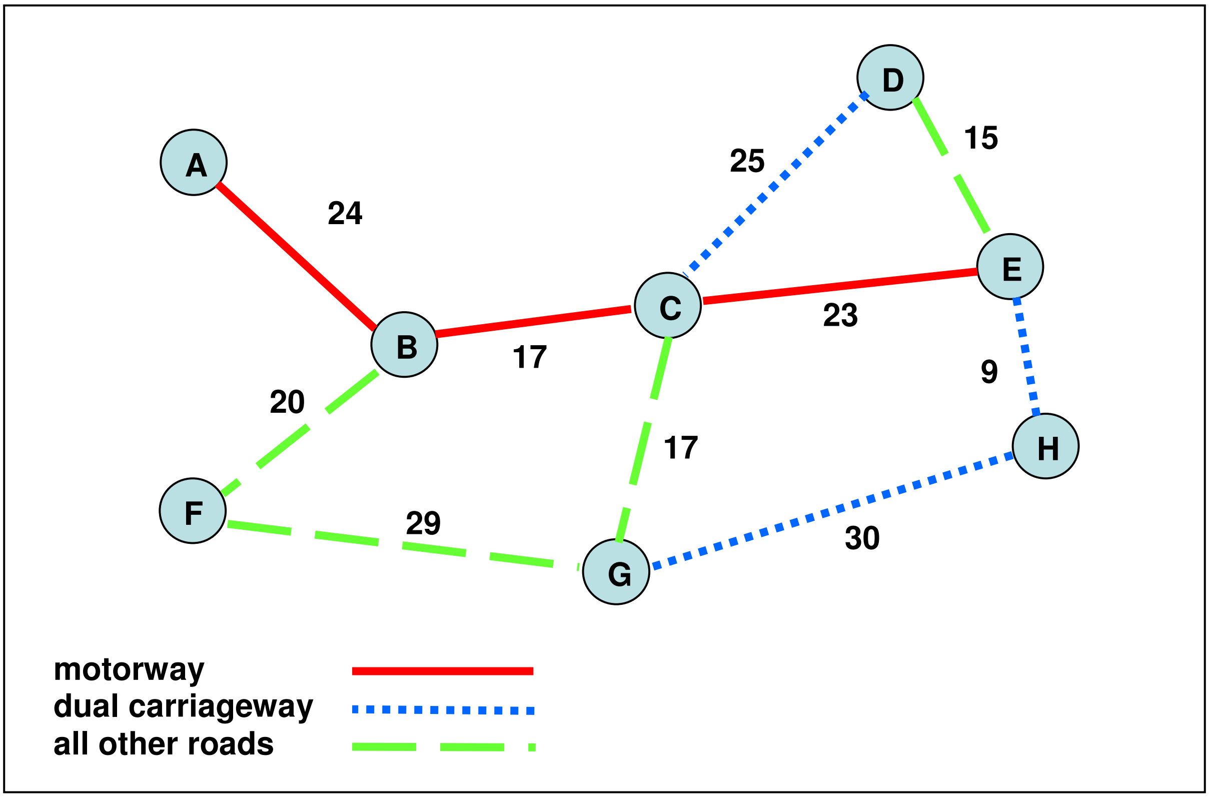
The government wishes to conduct an experiment in road pricing – i.e. charging motorists for the actual use they make of the roads rather than a flat rate tax.

Sensors will be set up at various points on a small network of roads; these sensors will be capable of extracting the registration numbers of vehicles which pass them as a string. This string and the time at which the vehicle passed the sensor are transmitted to a central monitoring station and recorded.

The government wants to influence the use of different types of road by different charging mechanisms. To this end it has divided roads into three types:

- motorways
- dual carriageways
- all other roads

Here is a diagram of the experimental network showing the location of the sensors; the name of a sensor is a single character. A sensor records a vehicle passing the point at which it is located irrespective of the direction in which the vehicle is travelling. Distances between sensors are in miles.



As you can see, the road network is effectively divided by the sensors into **segments**, with each segment being a specific type of road between two sensors and having a specific length. A segment is named by the two sensors at either end e.g. AB, CG. The order of the sensors is not important in naming the segment.

It is assumed that a vehicle:

- enters the network from outside at one of the sensor positions;
- once started along a road segment, continues along the segment to the end (i.e. doesn't do a U-turn in the middle);
- may continue from one segment to the next (e.g. A to B and then B to C)
- leaves the network at one of the sensor positions.

Vehicles are divided into two groups:

- private cars
- commercial (vans, lorries, busses, etc)

Vehicles are charged for travel along a particular segment according to the type of road, the length of the segment and the type of vehicle, as follows:

	private cars	**commercial**
motorways	nothing for the first 5 miles 2p per mile for the next 10 miles 1p per mile thereafter	2p per mile
dual carriageways	1p per mile	3p per mile for the first 10 miles 2p per mile thereafter
all other roads	free	5p per mile

The experiment will be run on a limited set of vehicles whose registration numbers and types are known in advance. All other vehicles using the system are ignored.

Task

To get CAS 15:

Your task is to write a prototype data processing system. It should handle one day's traffic movements. You will need to encode the information given in the diagram above; it is up to you how you do this. At the beginning of the next day the following text file is available for processing:

Vehicles

This contains one line for each known vehicle; each line contains the following information, separated by semi-colons:

- the registration number (as a string);
- the type of vehicle (as a string – either "private car" or "commercial");
- the current charges (in pounds) that have been accumulated up to the start of the previous day (a floating point number to two decimal places).

A line in this file might look like:

SP05 XDF;private;23.78

A sample file is provided at:

http://www.csd.abdn.ac.uk/~wvasconc/teaching/CS1515/practicals/RoadPricing/Vehicles.txt

The output of the system should consist of:

An updated version of the *Vehicles* file where the current charge has been updated to include the charge incurred the previous day.

At the end of the day, the following file is available:

SegmentData

This contains one line for each passage of a vehicle through a segment; each line contains the following information, separated by semi-colons:

- the registration number of the vehicle;
- the name of the segment (the names of the sensors at either end, e.g., AB, CG, etc.);

A line in this file might look like:

SP05 XDF;AB

A sample file is provided at:

http://www.csd.abdn.ac.uk/~wvasconc/teaching/CS1515/practicals/RoadPricing/SegmentData.txt

The outputs of the system should consist of:

An updated version of the *Vehicles* file where the current charges have been updated to include the charges incurred the previous day.

To get CAS 18:

Given that the system records the time that a vehicle enters and leaves a particular road segment, the average speed for the vehicle for that segment can be calculated. The government wants to use this information to enforce speed limits and to issue fines for vehicles breaking the limits. Again, the speed limits set vary depending on the type of road and vehicle as shown below:

	private cars	**commercial**
motorways	70 mph	60 mph
dual carriageways	60 mph	50 mph
all other roads	50 mph	50 mph

The information in the *SegmentData* file is extended to include timing information:

- the registration number of the vehicle;
- the name of the segment (two characters, each in the range A to H);
- the time of entry in hours and minutes using the 24 hour clock in the format hh:mm;
- the time of exit in hours and minutes using the 24 hour clock in the format hh:mm;

Extend the functionality of the system to output a *SpeedingTickets* file which contains one line for each offence committed; this should be a text file where each line should contain:

- the registration number of the vehicle;
- the time period over which the offence was committed;
- the road segment involved (as specified by the sensors at either end);
- the actual average speed;
- the type of road and the speed limit in force on that road.

To get CAS 20:

As usual – surprise me! – What about trying some animated simulation of cars moving through the network?