

University of Wollongong
School of CS & SE, Faculty of Informatics

CSCI203/MCS9203
Data Structures and Algorithms
Autumn 2012

Assignment 2

Due 5:00pm Thurs 5th April

1 Petrol Station

You are to use discrete simulation techniques to model a petrol station. The station has six pumps: two with 92 octane only and two with all of 92,95 and 98 octane, one with diesel and one with LPG.

Cars arrive wanting one of the four types of available fuel, and will start filling at any available pump which matches that fuel. If no such pump is available, they will queue at whichever pump has a matching fuel type fuel and the smallest queue. If a car arrives at the station and all pumps with a matching fuel type have 4 cars already queued, the car will leave the petrol station without filling up.

Once the tank is filled, the driver leaves the car at the pump to pay the petrol station attendant. There is currently only a single attendant for the entire petrol station, so if that attendant is busy they must wait in line to be served. After paying, the driver returns to the car and drives off, allowing another driver to access the pump.

2 Output

The petrol station owner wants to know whether extra pumps are needed for any particular type of fuel, and whether or not to hire another attendant. When your program is run, you should run two simulations, both using the same input file. The first simulation should use a single attendant. The second simulation should use two attendants serving from the same queue.

You should report:

- the total time simulated (from the arrival of the first customer to the departure of the last customer) and,
- for each different type of fuel:
 - the average queue length,
 - average time in queue and average time in system (time in queue + service time)
 - the number of cars served and
 - the number of cars that left before being served
- and for each pump

- total idle time.

This should be done for each of the two simulations.

3 Input

The simulation is to read the name of a file from standard input. This file will contain entries for each car that arrives, separated by newlines. The file is sorted by arrival time. Each line of the file contains the following, in order, separated by whitespace: arrival time, desired fuel, time required to fill tank, time required to pay attendant. Arrival is in the format HH:MM:SS, potentially ranging from 00:00:00 to 11:59:59. Desired fuel is represented by a single character (2,5,8,d or l). Time required to fill tank and time required to pay attendant are in seconds, so 130 represents two minutes and 10 seconds. For example:

```
00:02:37 5 261 20
00:04:12 2 349 18
00:04:13 d 382 107
```

The first car arrives at 00:02:37 wanting 95 octane fuel, requiring 261 seconds to fill the tank and 20 seconds to pay, and so on for subsequent cars.

4 A note on object orientation and the STL

You may use classes/modules for your code if you wish, but be careful that splitting the program into objects does not interfere with the readability of the algorithms. If the markers find that your algorithms are difficult to follow (due to being spread across multiple parts of the code or for any other reason), you will lose marks.

You are not to use the STL for this assignment. All data structures and related algorithms should be coded yourself.

5 Submission

You should submit both your source code (.c, .cpp, .h) and a makefile named **makefile** which makes your program using CC or cc. The resulting program should be named **petrolsim** (all lowercase). Submit using the command `submit -u <username> -c CSCI203 -a 2 <your files>`