CSCI203
ASSIGNMENT 2
Due 5pm 04/05/2015

(Individual Work – 10% of subject marks)

This assignment covers the following topics:

- Discrete Event Simulation
- Priority Queue

Print Servers use priority queues to allow certain (urgent) print jobs to be printed ahead of less important print jobs. Priority queues are also used so that certain users' print jobs (managers, directors) are printed more quickly than other users' print jobs. In this assignment, you will write a simulator for a Print Server that utilizes priority queues.

The print jobs coming into our Print Server range in priority from 1 (lowest priority) to 4 (highest priority). Each print job is placed in a queue and is served in order from highest priority to lowest priority. Within each priority level, print jobs are served based on first-in, first out (FIFO) order.

Besides having a priority level, we will also associate an *arrival time* and a *printing time* with each print job. The *arrival time* is the time where the print job arrives at the Print Server and the *printing time* is the amount of time needed to finish the print job.

Your program should consist of

- An appropriate data structure to store the print jobs. A print job would have the following data:
  - a job ID (a positive integer)
  - a priority level (integer between 1 and 4 inclusive)
  - an arrival time (in minute, floating point numbers)
  - a printing time (between 0.50 to 5.00 minutes)

- For the purpose of this simulation, the print jobs will be supplied from a text file (`print_jobs.txt`). So you may want to have a function that retrieves print jobs from this file.

- A `Priority Queue` data structure (max-heap)

- A function to add a print job to the priority queue while still maintaining the max-heap structure.

- A function to remove a print job from the priority queue while still maintain the max-heap structure.

- A function to process the print jobs, i.e. pull the appropriate print job out of the queue, and then printing out the current time (start with 0), job's ID,

priority level and printing time once its "processing" completes. Repeat this for all print jobs. Keep track of the waiting time (the amount of time spent waiting in the queue) for each print job.

You can also have additional functions that you deem necessary.

At the end of the simulation, print the following information out to the screen:

- the average waiting time (in minutes) for each print job,
  Average waiting time=(Total waiting time for all print jobs)/(Total number of print jobs)

- The maximum waiting time of the print jobs

- The probability that a print job has to wait in a queue for 5 minutes or more
  Probability that a print job has to wait in queue for 5 min of more = (Numbers of Print jobs that wait for 5 min or more) / (Total number of print jobs)

Bonus Task (optional):
Since our Print Server always pick and process the print job with the highest priority, print jobs with low priority may have to wait for very long (or never being picked at all if higher priority print jobs keep arriving). One solution is to increase the priority of the print jobs gradually over time while they are waiting in the queue. This solution, if implemented properly, should see a drop in maximum waiting time and reduces the number of print jobs that have to wait for too long.

Submission:

Use appropriate comments in your source code. Zip and upload the entire project folder to *moodle* on or before the due date. An extension of time may be granted in certain circumstances. A request for an extension must be made to the Lecturer before the due date. Late submission without granted extension will be marked but the mark awarded will be reduced by 10% for each day late. Submissions will not be accepted if more than three days late.

Assessment:

Marks will be awarded based on the quality of the implemented algorithm and data structures. Note that marks may be deducted for other reasons, e.g. if your code is too messy or inefficient, is not well commented, etc.

For code that does not compile, does not work or for programs that crash, the most you can get is half the marks (i.e. 5 marks or less). Note that you may be contacted to explain your code.