**Simulation project**

**Task 2 - Write the basic simulation code**

*Remember that any exchange of code with another student is forbidden as it constitutes cheating*. *Do not give and do not copy code! We will run Moss to verify that there has not been any code sharing. Offenders will receive zero for the entire project. Take this warning very seriously.*

**Objectives**

In this task, you will write the simulation code to reproduce the hand simulation. Also, you will introduce randomness in your simulation by assuming that the inter-arrival and service times are exponentially distributed.

You can use any high-level language of your choice. You can develop your program on your personal computer, but it has to run and compile on eos, so that the TA can run your program. Programming elegance is not required! What is important is that your program runs correctly. For this, you are *strongly* advised to follow the simulation logic described in the previous assignment.

**Problem description**

*Task 2.1: Write the simulation code for the hand simulation described in task 1.*

* The simulation code should prompt the user for the input values:
  + mean inter-arrival time of RT messages
  + mean inter-arrival time of nonRT messages
  + mean service time of an RT message
  + mean service time of a nonRT message
* Use the same initial conditions as in task 1.1.
* Print out the same table you produced in the hand simulation task 1.1 and make sure that it is the same.
* Print out the same table you produced in the hand simulation task 1.2 and make sure that it is the same.

*Task 2.2: Introduce randomness in your simulation*

* Use the program you developed in task 2.1, but assume that the inter-arrival times and service times are exponentially distributed. Make sure you define the clocks as real variables. The mean values of these two exponential distributions are the same as input values used in task 1.1, i.e: mean inter-arrival time of RT messages = 10, mean inter-arrival time of nonRT messages = 5, mean service time of an RT message = 2, mean service time of a nonRT message = 4.
* Use the same initial conditions as in task 1.
* Each time you want to generate an inter-arrival time or a service time, draw a pseudo-random number *r* and then obtain the exponential variate using the expression: -(mean)loge(*r*).
* Print a line of output each time you handle an event, i.e., you advance the master clock, with the same information as in the tables in task 1. Run your simulation until *MC* exceeds 200.

Go over the results by hand to make sure that the simulation advances properly from event to event. This is your last chance to make sure that your implementation is correct !!

**What to submit**

Submit two different files, one for the task 2.1 and a separate one for task 2.2. In each file, include the code and the output table(s).

**Grading**

* 30 points for each version of the code if it runs and compiles on eos correctly
* 20 points for each version of the code if it gives the correct output on test data provided by the TA.