This assignment will help you with simulation termination (if you use the relative-precision idea which is optional), confidence intervals, and variance reduction. You will reuse code you already have. You need to simulate a seven-node (open) tandem queue. Arrivals to the first queue are Poisson with rate $\lambda$ and every server has the same service rate $\mu$ (to keep things simple). From the analysis you saw in class where we used Burke’s Thm, you know the exact mean time $E[T]$ it takes a customer to exit the system (i.e., transit or sojourn time) from the moment the customer arrives. **Your result graphs will plot $\rho$ against $E[T]$.** This is because each queue becomes an M/M/1 queue.

Your task is to run a simulation and obtain enough samples to obtain a graph that shows your simulation results and the exact result. You must obtain, say, ten points, to plot each simulation-result graph against the exact graph.

In your report please do not forget to describe the setup, the parameters, the independent or dependent replications, and the way in which you try to measure only at steady-state to remove initialization bias, and the statistical results. The point of a report, as you already may know, is to describe your experiment briefly, but both accurately and completely. A reader should not have any questions about how you conducted the experiment and obtained your results. Specify the distributions you use for input, statistics, etc., along with other necessary information, for completeness. There should be no room for ambiguity.

Parameter description includes parameters you used in the whole experiment and modeling, not only the parameters of the
code. And you should also mention what values of those parameters you used to obtain specific results mentioned in your report. When describing the distributions you use for input/statistics, do not say things like, we used the random number generators supplied by the Professor. Be precise about the types of random variates you use. Additionally, for example, a reader might want to know that you used a linear congruential RNG for the uniform variates from which you got the exponential variates. This is what is mean by completeness.

1. Graph a 95%-CI for the unknown $E[T]$ based on the Normal distribution. Obtain at least 30 samples (and more if possible) of each data point in steady state before you build the CI. If you use the relative-precision idea, you should get this automatically for a small enough relative precision. Graph the vertical error bars on the graph in each case, to show that your CI’s contain the exact result.

   **Code Setup:**
   
   Input: $\lambda$, $\mu$
   
   Output: Mean($\bar{X}$), Sample variance ($S^2(n)$)), Confidence Interval, Theoretical $E[T]$

2. Repeat the experiment and build a 95%CI using the antithetic variate method. Follow the same code setup and graph as in Q1.
   
   In your report also explain how you used the inverse transform method to generate exponential random variates and how a pair of samples are generated with the random variates in your code. Do not mention generic things like following -
$U_1, \ldots, U_n$ was used to generate $I_1, \ldots, I_n$ for sample 1. Then $1 - U_1, \ldots, 1 - U_n$ was used to generate $I_1, \ldots, I_n$ for sample 2.

Explain briefly how you implement this in your code. All of these should be under the section **Antithetic Samples Generation**.

3. Repeat the experiment and build a 95% CI using the control variate method. Follow the same code setup and graph as Q1.
   In your code, also output the value of following - Control Constant ($\hat{c}^*$). As in the above part, include a brief descriptive section under the heading **Covariate Samples Generation**.

**Skeleton Files**
Download from [here](#)

**Turnin Instructions**
Your code will be executed in *data.cs.purdue.edu*, so make sure your code can be make and run in data.cs.purdue.edu. You can ssh to data.cs.purdue.edu. You should turnin the *project5* directory.

- Please just submit the source files necessary to compile with the Makefile.
- Maintain the directory and file names given to you in skeleton code.
- Do not include any additional output files.
- Do not change Makefile extension.
• After make, there should be three executables - normal, antithetic and control.

• No code submission will be allowed after April 30, 11:59 am

To turn in your project, make sure that you are in the directory that contains project5 directory. Then type the following:

```
turnin -c cs543 -p proj5 project5
```

To verify your turned-in work, do:

```
turnin -c cs543 -p proj5 -v
```

.................................................................

**Due: April 30, 11.59 am. [Turnin the code as usual and put the report in my mailbox; please remember that the mailroom closes at noon]**

Note that the code submission will enable the grader to make sure the code was written and the code runs and generates the right results. Your submitted code will not loop to get many replications etc. You will use your code and get the CIs requested in the three parts outlined above, and each of these require repeated runs — and this information will be contained in the three graphs and supporting explanation you submit in the report. You only need to insert a few lines of code to do some computation and do the repeated runs manually or, if you can use loops or scripts, automate the entire process. But please be careful to make sure you are doing it properly, because it’s easy to make mistakes in routine statistical calculations, and it’s even harder to recognize that a mistake was made.
This project should not take you much time given that you already have the code and will only insert some computation lines in the code. Please start on this without delay.