**New Information Since Handing Out the Project**

May 5, 2015 - As was announced when handing out the project in class, for Problems 1-3, the only performance measure that needs to be reported is server utilization. In other words, the programs do not need to also output the average number of machines waiting for service, effective arrival rate, etc. However, you can output all of these values for a 10% extra credit bonus.

**Purpose**

This project will give you practice with the following:

* Programming in C++11.
* Building and running a multiple-server queueing model.
* Analyzing data that is produced by queueing models that you have created.

**Problems from the book (Banks et. al, 5th edition)**

Use C++11 in XCode for all of these problems.

**Problem 1**

Build a queuing network simulation of Example 6.17 on p. 261 which is as follows: "Example 6.17 — There are two workers who are responsible for 10 milling machines. The machines run on the average for 20 minutes, then require an average 5-minute service period, both times exponentially distributed. Therefore, λ = 1/20 and μ = 1/5. Compute the various measures of performance for this system." Note that each machine require the dedicated focus of a worker for the service time but not the milling time.

Your program should should output the various measures shown in the example, but by implementing a queueing network that simulates each machine running for 100 run/service cycles. Each time a machine starts a run or service period, the program should output a line that reads "<time>: Machine <x> starts running." or "<time>: Machine <x> starts service." Time should be displayed in minutes, to two decimal places. <x> is the number of the machine.

The program should end by outputing the following values: server utilization, average number of machines waiting for service, effective arrival rate, average wait time in the queue, expected number of machines being serviced or waiting to be serviced, average number of machines being serviced, and average number of running machines. One value should be displayed per line, to three significant digits. Each value should be labeled, as in:  
     server utilization: 56.1%  
     average number of machines waiting for service: 3.42  
     ...

**Problem 2**

Complete Chapter 6, Exercise 17, on p. 271 which is as follows: "In Example 6.17, compare the systems with c = 1, c = 2, and c = 3 servers on the basis of server utilization ρ (the proportion of time a typical server is busy)." Implement this as specified for Problem 1 but with one input modification and one output modification, as follows: For input, prompt the user on the command line to specify 1 to 3 servers, such as "Enter the number of servers (from 1 to 3):". For output, add an extra line at the start of the output stating the number of servers.

Discuss, in what becomes a "Project\_3\_Summary.pdf" document, how the output values change with c = 1, c = 2, and c = 3.

For 10 points extra credit (both grads and undergrads): Expand the permissible number of servers from "1 to 3" to "0 to 12" and have the program dynamically allocate servers for use in the simulation. c = 0 may be handled as a special case. Your code must be well-organized and well-commented.

**Problem 3**

Complete Chapter 6, Exercise 18, on p. 271 which is as follows: "In Example 6.17, increase the number of machines by 2, then compare the systems with c = 1, c = 2, and c = 3 servers on the basis of server utilization ρ (the proportion of time a typical server is busy)." Implement this as specified for Problem 2 but prompting for another input value ("10 or 12") and by outputting the new value at the start of the output. Again, discuss all output changes in your Project\_3\_Summary document.

Across Problems 1, 2, and 3, you may end up with a single body of code that, except for a few minor changes, accomplishes all three problems. Even if this is the case, please submit three different XCode projects.

**Problem 4** (Required for graduate students; 20 points extra credit for undergraduates)

Complete Chapter 9, Exercise 19, on p. 383 which is as follows: "The Crosstowner was a bus that cut a diagonal path from northeast Atlanta to southwest Atlanta. .The time required to complete the route was recorded by the bus operator. The bus runs from Monday through Friday. The times of the last fifty 8:00 A.M. runs, in minutes, are as follows: [[Text file containing the data.](https://www.cs.uoregon.edu/Classes/15S/cis410human/course_materials/P3p4-input.txt)] How are these run times distributed? Develop and test a suitable model."

In your Project\_3\_Summary, plot a histogram of the data, and state which distribution you believe fits the data.

Your program should read in the text file, and output (a) the calculated value of chi-squared and (b) a statement of whether the data follows your assumed distribution.