

Program #3 - Due Tuesday, Sept. 22

At a bank, when a typical teller waits on a customer, the length of time it takes to complete the customer's transaction varies, depending on the complexity of the transaction. Let's suppose that for a given customer, this length of time (measured in whole number of seconds) is random with a minimum length of 30 seconds and a maximum length to be entered by the user. Over the course of a whole number of hours (to be entered by the user), customers arrive like clockwork, one customer every n seconds, where n is a whole number entered by the user. If a teller is available when a new customer arrives, then that customer is waited on right away. However, if all tellers are busy, the customer goes to the end of an available line, always to the end of the one that has the fewest number of customers in it when he/she arrives. When one customer completes a transaction, the next customer in that line is helped.

Your program should

1. allow the user to input the values specified above,
2. enqueue new customers as appropriate,
3. dequeue customers when a teller is available,
4. compute the wait time in line for each customer,
5. compute average wait time per customer for each simulation, and
6. compute the average of these average wait times per customer for a certain number of simulations (again, to be entered by the user).

Carry out this simulation if there is one teller available, then repeat for two, three, ..., nine, ten tellers available.

In addition to your program, I would like sample output for 5 runs of your program.