**Homework 3:**

**Simple Discrete Even Simulation Using Queues**

Due Date: March 12, 2021.

**Points: 80 points**: 2 bonus points if submitted by March 10, 10 bonus points for those who volunteer to tutor other customers seeking help. 3 bonus points for each student being tutored.

Submit **source code files** and **sample text I/O** and capture I/O console covering all possible scenarios as a word document.

Please make sure that all screen captures are captured in a large enough font to see. And the text appears in black with white background. Small or hard to view screen captures are not accepted.

**The assignment:**

Customers arrive at the bank to make deposits or withdrawals and are served by a teller.

The following information is kept in a structure about each customer.

1. A structure named customer contains at least the following information:

struct Customer {

int ID; customerId;

string fName; // // customer first name as either an array of char or a string

string lName; // customer last name as either an array of char or a string

double amount of deposit or withdrawal

char Transaction Type: Deposit or Withdrawal

int arrivalTime. ;

int serviceTime ;

double Balance: Account balance

int TellerNo; // a value between 1 and 3

};

2. Create an input file containing the information of at least 35 customers.

3. Create an array of struct: read the customers’ data from the input file into the array.

customer custList[n]; // n= 35

4. Create a queue: you can use the standard library template for the queue, or you can write your own code for the different queue operations. The queue stores data of type customer.

5. Also submit an output file of all output that is being sent to the console.

What to do:

You need to code at least the following functions:

I. **Random number generator**

II. **Initialization function:**

* initialize an int named timer= 0 ;
* initialize an int named clock = 0 ;
* initialize the number of transactions served by each teller to zero.

1. Read customers’ data from the input file into an array of structures.

2. For each customer in the array,

a.       Generate a random number between 1 and 3 representing the teller station number.

b. Generate a number between 1 and 10 and add it to the timer, then store it as the arrival time value for the customer. 5 5 8

**III. . Create the queue function.**

enqueue the customer structure into the queue

**IV. Teller Function**

4. Dequeue the customers one at a time from the queue and do the following:

a. Add the amount of deposit or subtract the amount of the withdrawal to/from the balance.

b. Increment the number of transactions served by the teller specified for that teller.

c. Generate a random number waitTime between 0 and 5 and

d. Update the clock where clock = clock+waitTime

store the update clock time in the service time in the corresponding structure for that customer.

**Reporting function**

The reporting function place a call to the following functions:

1. **Customer summary**: display for each customer: name, balance, amount, and type of transactions (deposit, withdrawal), arrival time, and waiting time: write the output to both the screen and file.

2. Display for each Teller, the total number of transactions served by each teller with a subtotal for each transaction type in addition to the a total of the number of all transactions severed by that teller. Writer the output to both the screen and file.

3. For the system: Display the total number of wait time, and average number of transactions served by each teller.

Write the output to both the screen and file.