

## Data Structure and Advanced Programming

(Programming) Homework #9

Due: 2020/5/19 08:00am (CST)

**NOTE: Please upload your C++ source codes (by copy-paste) to PDOGS (<https://pdogs.ntu.im/judge/>) before the due date and time.**

1. Implement the event-driven simulation of a bank that we talked about in the classroom (actually in the video; you can also refer the Chapter 13 in the textbook<sup>1</sup>). A queue of arrival events will represent the line of customers in the bank. Maintain the arrival events and departure events in a priority queue, sorted by the time of the event. Use a link-based implementation for the event list.

The input is a text file of arrival and transaction times. Each line of the file contains the arrival time and required transaction time for a customer. The arrival times are ordered by increasing time.

Your program must count customers and keep track of their cumulative waiting time. These statistics are sufficient to compute the average waiting time after the last event has been processed. Display a trace of the events executed and a summary of the computed statistics (the total number of arrivals and average time spent waiting in line).

If the input file is as the following table:

1	5
---	---

The output will be shown as the follows:

Simulation Begins

Processing an arrival event at time: 1

Processing a departure event at time: 6

Simulation Ends

Final Statistics:

Total number of people processed: 1

Average amount of time spent waiting: 0.0

If the input file is as the following table:

1	5
2	5
4	5
20	5
22	5

---

<sup>1</sup> *Data Abstraction and Problem Solving with C++: Walls and Mirrors* by Carrano and Henry, sixth edition, Pearson, 2012.

24 5  
26 5  
28 5  
30 5  
88 3

The output will be shown as the follows:

Simulation Begins

Processing an arrival event at time: 1  
Processing an arrival event at time: 2  
Processing an arrival event at time: 4  
Processing a departure event at time: 6  
Processing a departure event at time: 11  
Processing a departure event at time: 16  
Processing an arrival event at time: 20  
Processing an arrival event at time: 22  
Processing an arrival event at time: 24  
Processing a departure event at time: 25  
Processing an arrival event at time: 26  
Processing an arrival event at time: 28  
Processing an arrival event at time: 30  
Processing a departure event at time: 30  
Processing a departure event at time: 35  
Processing a departure event at time: 40  
Processing a departure event at time: 45  
Processing a departure event at time: 50  
Processing an arrival event at time: 88  
Processing a departure event at time: 91  
Simulation Ends

Final Statistics:

Total number of people processed: 10  
Average amount of time spent waiting: 5.6

2. Modify the simulation so that it accounts for three tellers, each with a distinct line. You should keep in mind that there should be
  - a. Three queues (named A, B, C), one for each teller (also named A, B, C).
  - b. The rule for choosing a line when processing an arrival event is to enter the shortest line. If there are two or more shortest lines, choose them by alphabet order.
3. The bank is considering the following change: Instead of having three distinct lines (one for each teller), there will be a single line for the three tellers. The person at the front of the line will go to the first available teller. Modify the simulation to account for this variation.