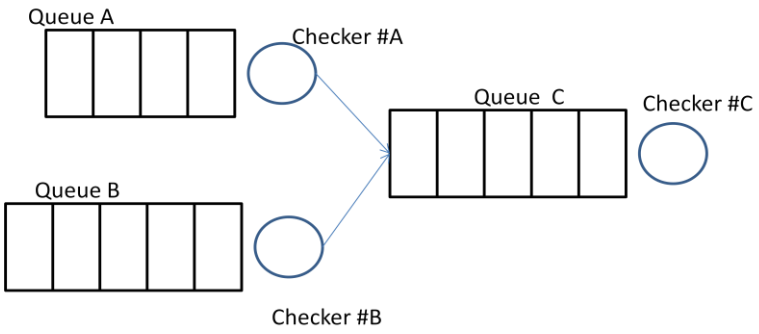
Programming Assignment – B

Submit your assignment as a report. You need to do the following for each of the problems to get full points.

1) **Describe the workings of your code and challenges that you encountered during your program development.**
2) **Program listing with line numbers.**
3) **Create test cases and report their output.**

---

1. Simulate 3-Queue System [40 points]



Assume that we have a hypothetical queuing system that has 3 queues: Queue-A, Queue-B, Queue-C, in which travelers are waiting to be cleared for their flights. Each queue has an inspector and we call him/her Checker. Queue-A has an inspector named Checker-A. Similarly, Queue-B and Queue-C has inspectors named Checker-B and Checker-C respectively. Each checker is responsible for processing his/her queue. A checker will be busy for a certain amount of time somewhere between 1 min to 15 mins in order to process a person's credentials. You can use a random generator to simulate the checker's time.

Effectively, a person has to go though two queues, either through Queues A, C or Queues B, C.

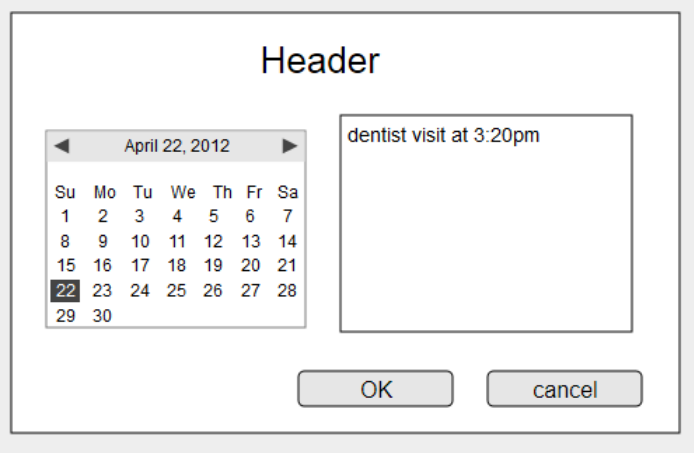
From time to time the checkers themselves peek in their respective queues to find the length of their queues. If they find that their queue length is increasing steadily then they will expedite their processing. You need to come up with a policy (or a set of policies) that these TSA checkers will use to manage their queues effectively.

Initially you start the simulation with a population of say ***N*** (e.g., 50) people who want to be checked in so that they can board their flights. Make sure that ***N*** people are assigned to different queues at different times somewhere between 1 min to 10 minutes. For example, say a person P1 arrives at Queue-A at time t[1]=0. The next person, say P2, arrives at Queue-B at t[2]=2.1 seconds (i.e., 2.1 secs later). The third person, say P3, arrives at Queue-B at t[3]=3.8 minutes later after P2. People arrive at different times but never together.

How you distribute 50 people across different queues is up to you. For example, Queue-A can have max capacity of 30, and Queue-B has 20. It is up to you how you set up the initial distribution of people. Afterwards, you can alter the input capacities to say 11 persons going to Queue A and 39 people going to the Queue B. Different initial distributions become your different test cases. The program ends when all 50 people have been processed. Compute the total time when all persons have been processed through both the queues. You should use all the methods that you find in Java collection's queue interface. You can also use *System.currentMilliSecs()* to the get current time from system to simulate time, and other collection objects in your program in necessary. The goal of program is to simulate this queuing system as best as you can.

2. Write a Swing Appointment Reminder Application [30 points]

Write a GUI program by creating a JFrame that has a visual month calendar and an editor. When you click on the date (say 22 as shown in figure) you should see some appointment notes in the editor that you added earlier. You should be able to add and retrieve notes through the editor. If there is a Java calendar widget in Java libraries you can use it in your program. Add an appropriate header for your application.



3. UML Diagram

Construct UML Diagrams for a Library Management System. Think how library functions. It has books, magazine, journals, newspapers, periodicals, etc. It also has borrowers, card, membership policies, and renew policies, library access and so on. [20 points]