DOCUMENTATION

ASSIGNMENT *No2*

STUDENT NAME: Terente Ioana-Emilia
GROUP:30422

CONTENTS

1. Assignment Objective

The main objective of the assignment is to design and implement an application aiming to analyze queuing-based systems by (1) simulating a series of N clients arriving for service, entering Q queues, waiting, being served and finally leaving the queues, and (2) computing the average waiting time, average service time and peak hour.

The sub-objectives of the project are to:

- Analyze the problem and identify the requirements
- Design the simulation application
- Implement the simulation application
- Test the simulation application

2. Problem Analysis, Modeling, Scenarios, Use Cases

Problem Analysis:

This application should simulate customers waiting to receive a service (e.g. supermarket, bank, etc.) just like in the real world, they have to wait in queues, each queue processing clients simultaneously. The idea is to analyse how many clients can be served in a certain simulation interval, by entering parameters in an intuitive, user-friendly, application graphical interface.

The customers are generated randomly, each having its own service time and arrival time, the number of clients depends on the input values.

The user can set:

- Number of queues available to process clients
- Number of clients
- Maximum simulation time: how many seconds the simulation hold
- Minimum /maximum arrival time: the moment in time when the client gets in the queue
- Minimum/maximum service time: period of time that the client need to be served

The user can see from graphical user interface at each second:

- Current time
- Clients queue
- Available queues with correspondent clients that have entered the queue

Primary Actor: user

Modelling:

Main success scenario:

1. The user inserts the values for the: number of clients, number of queues, maximum simulation time, minimum and maximum arrival time, minimum and maximum service time.
2. The user clicks on the “Start simulation” button that validates the input data.
3. The application validates the data and displays as each second passes by: the current simulation time and the available queues with the correspondent clients

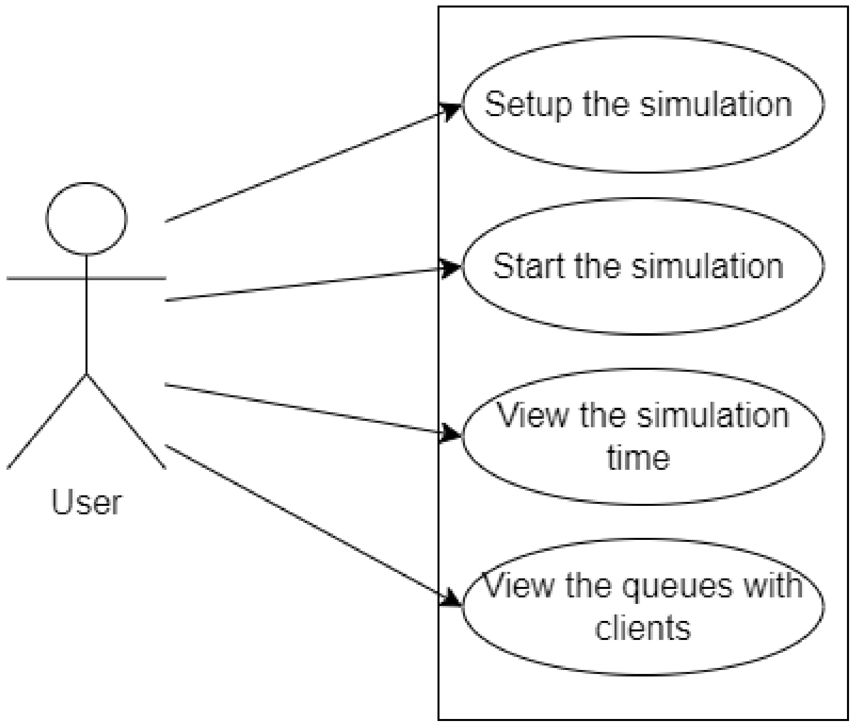
Alternative Sequence: Invalid values for the setup parameters

- The user inserts invalid values for the application’s setup parameters
- The application displays an error message and requests the user to insert valid values
- The scenario returns to step 1

Scenarios:

- The user has to introduce all the data, it assumes that all the introduced data is correct and it contains only digits. Anyway, if the data is not correct or the text fields are left empty, the application will not work properly.
- The user has introduced perfectly all the data inputs and then presses the “Start simulation” button, after this the application is displaying the logs in real time simulation and the evolutions of the queues.

Use Case: setup simulation



Functional Requirements:

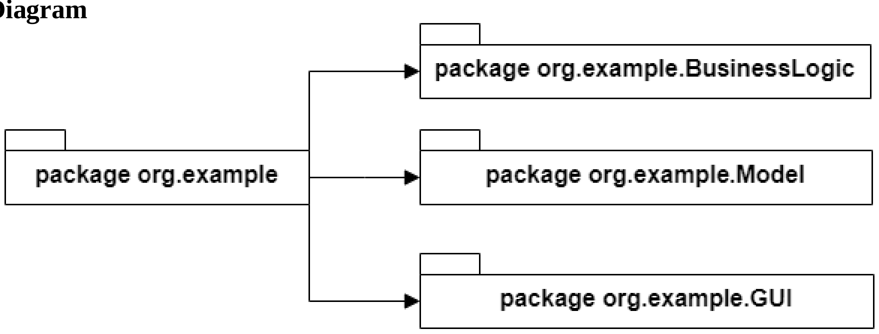
- The simulation application should allow users to setup the simulation.
- The simulation application should allow users to start the simulation.
- The simulation application should display the real-time queues evolution.
- The user should be able to see the result of the inserted values.

Non-Functional Requirements:

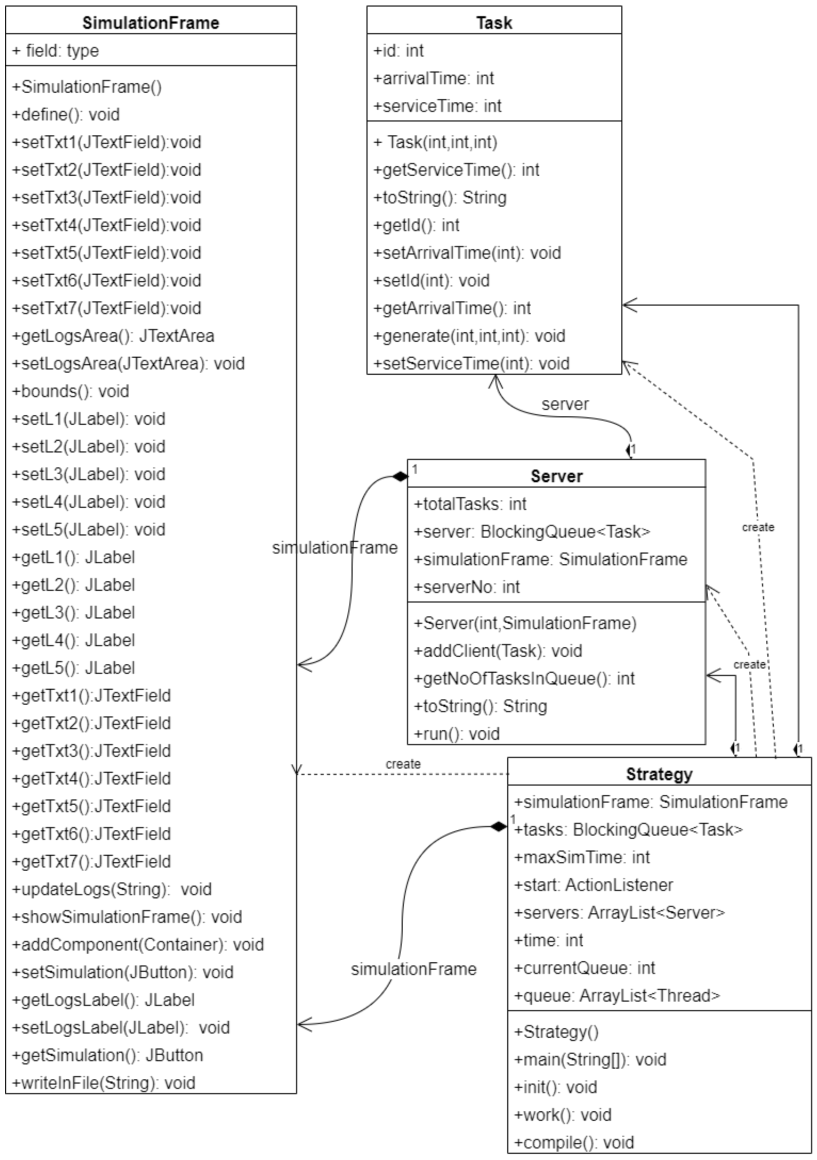
- Usability: The simulation application should be intuitive and easy to use by the user, the user can easily insert values for the number of queues, number of clients, simulation time, min/max arrival time and min/max service time.
- Performance: The simulation application should be able to handle a large number of clients and queues and distribute them properly.
- Security: The simulation application should be designed with security in mind, with appropriate measures taken to prevent unauthorized access or data loss
- Liability
- Scalability

3. Design

UML Package Diagram



UML Class Diagram



4. Implementation

Task class represents each client that needs to be served. Each task has the following variables*: id*, *arrivalTime* (moment is time that the client enters the sever/queues) and *serviceTime* (amount of time that the client needs to stay first in queue). This class has only getters and setters except the two methods: *generate:* random generates using Math random arrivalTime and serviceTime that was taken from the GUI and *toString:* that returns (id,arrivalTime,serviceTime) used to display clients.

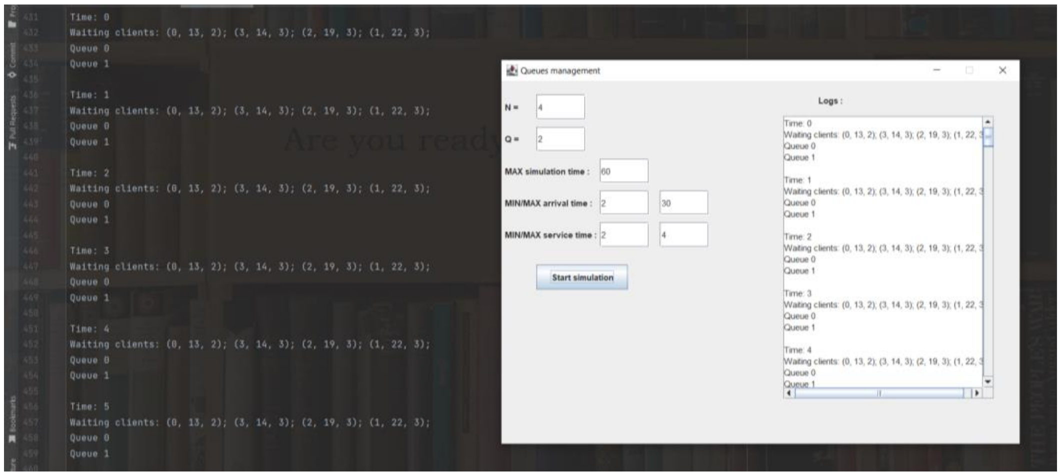
Server class represents each queue. I have declared the tasks queue as BlockingQueue<Tasks> to ensure synchronization while using threads. Here, one important method which through: implements Runnable interface can access Threads is the one that *@0verrides the run* and while there are still clients in queue using synchronized access ensures that only one client can have access at a time. If there are clients the thread is put to sleep the amount of time equal to 1s*serviceTime and the client is removed from the waiting clients queue (BlockingQueue<Task>). If there are no clients the thread is put to sleep for one second. Another method is *toString* that displays the queue number followed by the (id, arrivalTime, serviceTime).

Strategy class is where the simulation starts and the button" Start simulation" is put to work. *init()* method is used to generate the clients, add then to an auxiliar ArrayList<Task> and after sorting them by the arrivalTime put them in the BlockingQueue<Task>. In the same method generate the threads for each queue and get the maxSimTime from the GUI. The other method is *work().* This is the method that starts the simulation. It creates a new thread that runs until the current time is equal to the maximum simulation time and while there are tasks in the queue and their arrivalTime is equal to the current time then the task is added to the server for processing. Variable string is used to display the current time followed on the next line by the queue of remained waiting clients. The logs are updated each second using Thread.sleep(1000); and the current time is increased. Thread is started using .start().

SimulationFrame class is the one containing the graphical user interface, the part that the user sees. Except the getters and setter and methods where the text fields, label, button and frame where defined there are two methods; one that writes in the output.txt file the current simulation, but keeps the previous simulations as well, and the other methos that updates the queues in the GUI.

5. Results

Here are some examples of application testing using the input data sets from the support presentation:
For Test 1 we have:



```
Time: 0
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1

Time: 1
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1

Time: 2
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1

Time: 3
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1

Time: 4
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1

Time: 5
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1
```

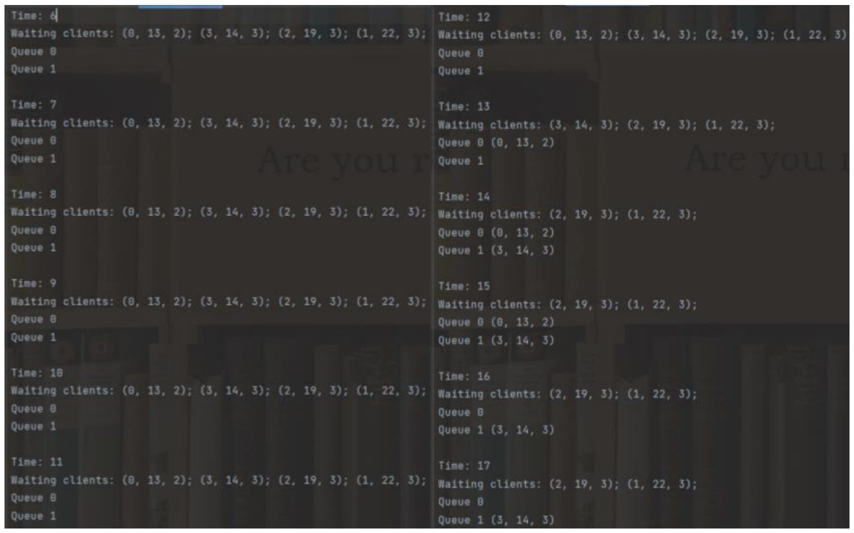
Queues management

N = 4
Q = 2
MAX simulation time : 60
MIN/MAX arrival time : 2 30
MIN/MAX service time : 2 4

Start simulation

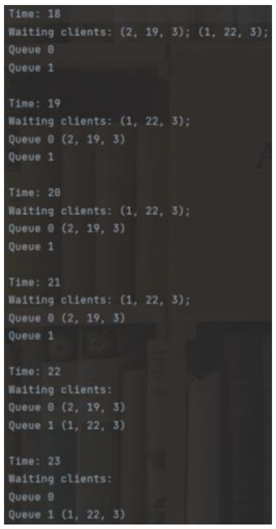
Logs :

```
Time: 0
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3
Queue 0
Queue 1

Time: 1
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3
Queue 0
Queue 1

Time: 2
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3
Queue 0
Queue 1

Time: 3
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3
Queue 0
Queue 1

Time: 4
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3
Queue 0
Queue 1
```



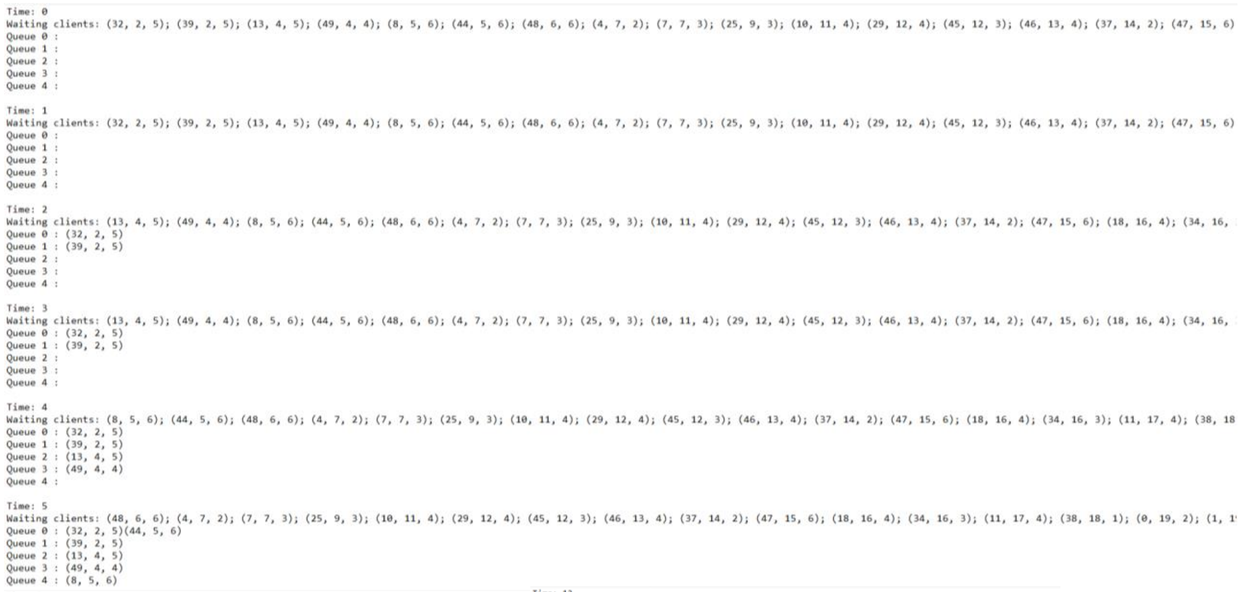
```
Time: 6
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1

Time: 7
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1

Time: 8
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1

Time: 9
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1

Time: 10
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1

Time: 11
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1
```

```
Time: 12
Waiting clients: (0, 13, 2); (3, 14, 3); (2, 19, 3); (1, 22, 3)
Queue 0
Queue 1

Time: 13
Waiting clients: (3, 14, 3); (2, 19, 3); (1, 22, 3);
Queue 0 (0, 13, 2)
Queue 1

Time: 14
Waiting clients: (2, 19, 3); (1, 22, 3);
Queue 0 (0, 13, 2)
Queue 1 (3, 14, 3)

Time: 15
Waiting clients: (2, 19, 3); (1, 22, 3);
Queue 0 (0, 13, 2)
Queue 1 (3, 14, 3)

Time: 16
Waiting clients: (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1 (3, 14, 3)

Time: 17
Waiting clients: (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1 (3, 14, 3)
```

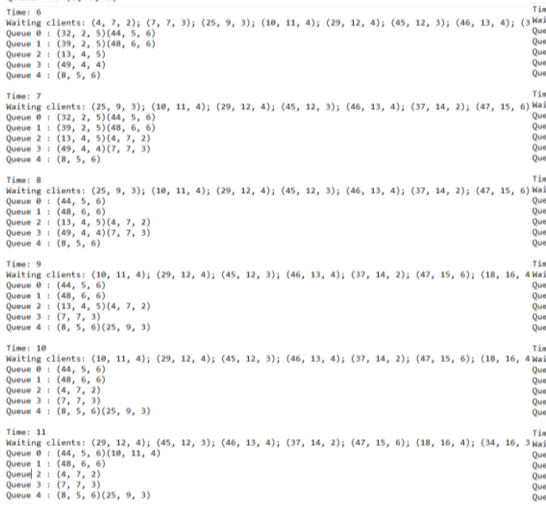


```
Time: 18
Waiting clients: (2, 19, 3); (1, 22, 3);
Queue 0
Queue 1

Time: 19
Waiting clients: (1, 22, 3);
Queue 0 (2, 19, 3)
Queue 1

Time: 20
Waiting clients: (1, 22, 3);
Queue 0 (2, 19, 3)
Queue 1

Time: 21
Waiting clients: (1, 22, 3);
Queue 0 (2, 19, 3)
Queue 1

Time: 22
Waiting clients:
Queue 0 (2, 19, 3)
Queue 1 (1, 22, 3)

Time: 23
Waiting clients:
Queue 0
Queue 1 (1, 22, 3)
```

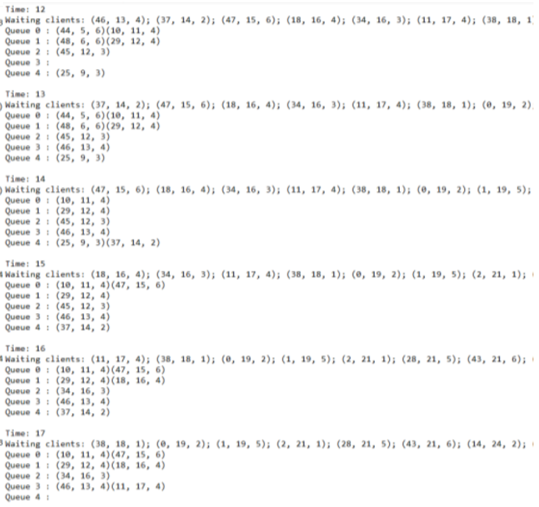
For Test2 we have:



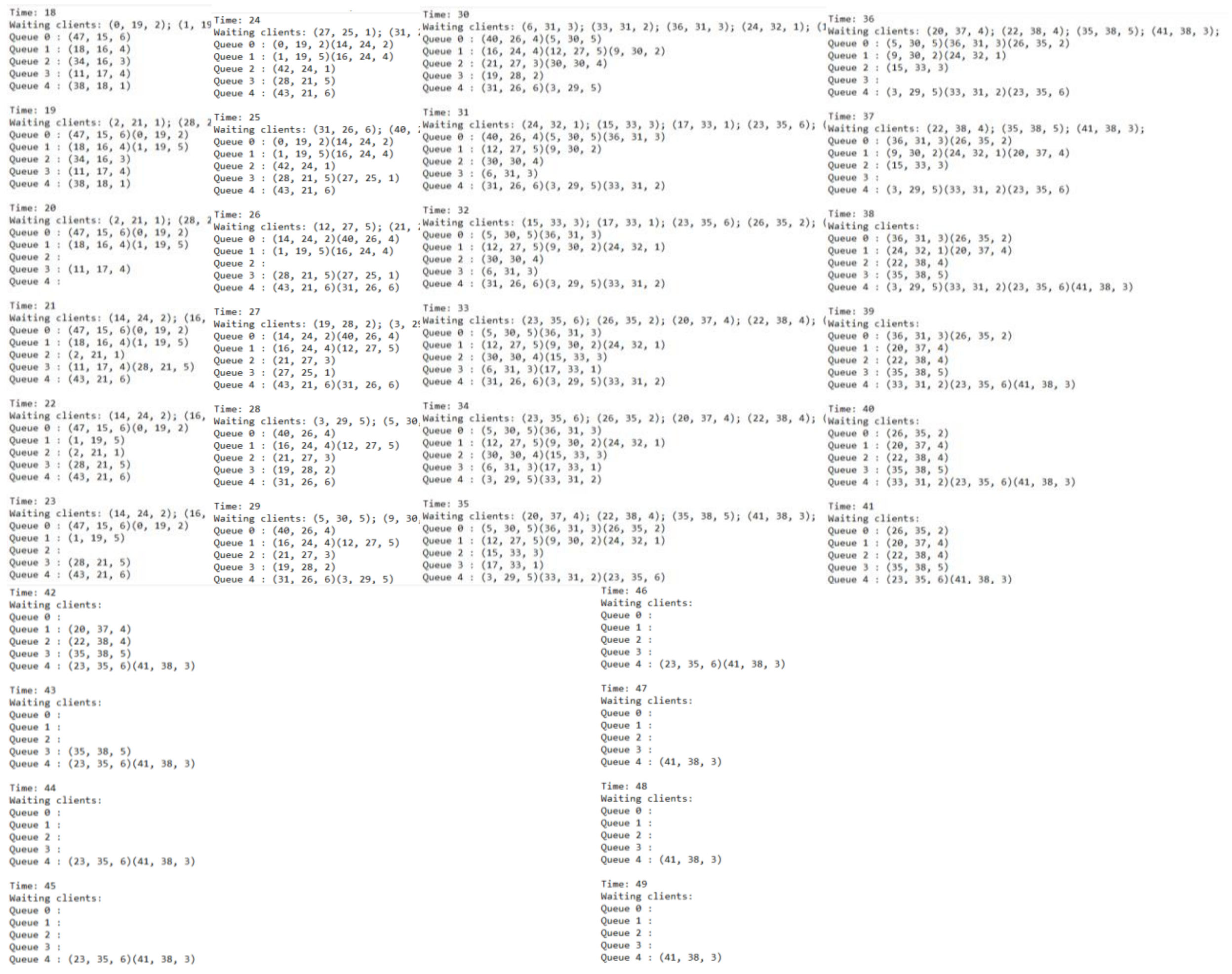
```
Time: 0
Waiting clients: (32, 2, 5); (39, 2, 5); (13, 4, 5); (49, 4, 4); (8, 5, 6); (44, 5, 6); (48, 6, 6); (4, 7, 2); (7, 7, 3); (25, 9, 3); (10, 11, 4); (29, 12, 4); (45, 12, 3); (46, 13, 4); (37, 14, 2); (47, 15, 6)
Queue 0 :
Queue 1 :
Queue 2 :
Queue 3 :
Queue 4 :

Time: 1
Waiting clients: (32, 2, 5); (39, 2, 5); (13, 4, 5); (49, 4, 4); (8, 5, 6); (44, 5, 6); (48, 6, 6); (4, 7, 2); (7, 7, 3); (25, 9, 3); (10, 11, 4); (29, 12, 4); (45, 12, 3); (46, 13, 4); (37, 14, 2); (47, 15, 6)
Queue 0 :
Queue 1 :
Queue 2 :
Queue 3 :
Queue 4 :

Time: 2
Waiting clients: (13, 4, 5); (49, 4, 4); (8, 5, 6); (44, 5, 6); (48, 6, 6); (4, 7, 2); (7, 7, 3); (25, 9, 3); (10, 11, 4); (29, 12, 4); (45, 12, 3); (46, 13, 4); (37, 14, 2); (47, 15, 6); (18, 16, 4); (34, 16,
Queue 0 : (32, 2, 5)
Queue 1 : (39, 2, 5)
Queue 2 :
Queue 3 :
Queue 4 :

Time: 3
Waiting clients: (13, 4, 5); (49, 4, 4); (8, 5, 6); (44, 5, 6); (48, 6, 6); (4, 7, 2); (7, 7, 3); (25, 9, 3); (10, 11, 4); (29, 12, 4); (45, 12, 3); (46, 13, 4); (37, 14, 2); (47, 15, 6); (18, 16, 4); (34, 16,
Queue 0 : (32, 2, 5)
Queue 1 : (39, 2, 5)
Queue 2 :
Queue 3 :
Queue 4 :

Time: 4
Waiting clients: (8, 5, 6); (44, 5, 6); (48, 6, 6); (4, 7, 2); (7, 7, 3); (25, 9, 3); (10, 11, 4); (29, 12, 4); (45, 12, 3); (46, 13, 4); (37, 14, 2); (47, 15, 6); (18, 16, 4); (34, 16, 3); (11, 17, 4); (38, 18
Queue 0 : (32, 2, 5)
Queue 1 : (39, 2, 5)
Queue 2 : (13, 4, 5)
Queue 3 : (49, 4, 4)
Queue 4 :

Time: 5
Waiting clients: (48, 6, 6); (4, 7, 2); (7, 7, 3); (25, 9, 3); (10, 11, 4); (29, 12, 4); (45, 12, 3); (46, 13, 4); (37, 14, 2); (47, 15, 6); (18, 16, 4); (34, 16, 3); (11, 17, 4); (38, 18, 1); (0, 19, 2); (1, 1
Queue 0 : (32, 2, 5)(44, 5, 6)
Queue 1 : (39, 2, 5)
Queue 2 : (13, 4, 5)
Queue 3 : (49, 4, 4)
Queue 4 : (8, 5, 6)
```



```
Time: 6
Waiting clients: (4, 7, 2); (7, 7, 3); (25, 9, 3); (10, 11, 4); (29, 12, 4); (45, 12, 3); (46, 13, 4); (3
Queue 0 : (32, 2, 5)(44, 5, 6)
Queue 1 : (39, 2, 5)(48, 6, 6)
Queue 2 : (13, 4, 5)
Queue 3 : (49, 4, 4)
Queue 4 : (8, 5, 6)

Time: 7
Waiting clients: (25, 9, 3); (10, 11, 4); (29, 12, 4); (45, 12, 3); (46, 13, 4); (37, 14, 2); (47, 15, 6)
Queue 0 : (32, 2, 5)(44, 5, 6)
Queue 1 : (39, 2, 5)(48, 6, 6)
Queue 2 : (13, 4, 5)(4, 7, 2)
Queue 3 : (49, 4, 4)(7, 7, 3)
Queue 4 : (8, 5, 6)

Time: 8
Waiting clients: (25, 9, 3); (10, 11, 4); (29, 12, 4); (45, 12, 3); (46, 13, 4); (37, 14, 2); (47, 15, 6)
Queue 0 : (44, 5, 6)
Queue 1 : (48, 6, 6)
Queue 2 : (13, 4, 5)(4, 7, 2)
Queue 3 : (49, 4, 4)(7, 7, 3)
Queue 4 : (8, 5, 6)

Time: 9
Waiting clients: (10, 11, 4); (29, 12, 4); (45, 12, 3); (46, 13, 4); (37, 14, 2); (47, 15, 6); (18, 16, 4
Queue 0 : (44, 5, 6)
Queue 1 : (48, 6, 6)
Queue 2 : (13, 4, 5)(4, 7, 2)
Queue 3 : (7, 7, 3)
Queue 4 : (8, 5, 6)(25, 9, 3)

Time: 10
Waiting clients: (10, 11, 4); (29, 12, 4); (45, 12, 3); (46, 13, 4); (37, 14, 2); (47, 15, 6); (18, 16, 4
Queue 0 : (44, 5, 6)
Queue 1 : (48, 6, 6)
Queue 2 : (4, 7, 2)
Queue 3 : (7, 7, 3)
Queue 4 : (8, 5, 6)(25, 9, 3)

Time: 11
Waiting clients: (29, 12, 4); (45, 12, 3); (46, 13, 4); (37, 14, 2); (47, 15, 6); (18, 16, 4); (34, 16, 3
Queue 0 : (44, 5, 6)(10, 11, 4)
Queue 1 : (48, 6, 6)
Queue 2 : (4, 7, 2)
Queue 3 : (7, 7, 3)
Queue 4 : (8, 5, 6)(25, 9, 3)
```

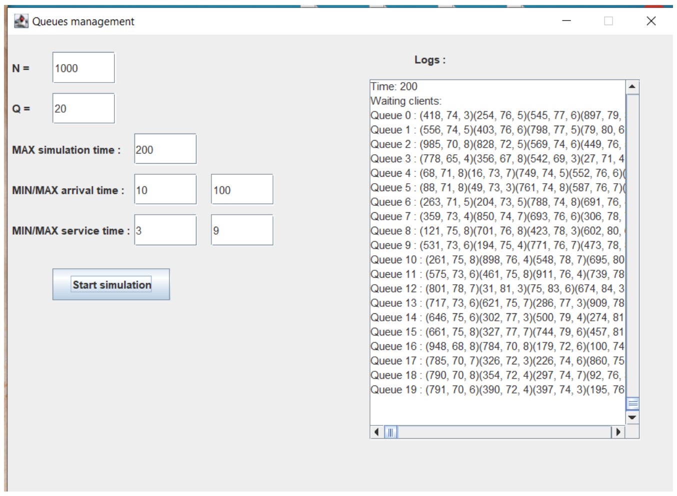


```
Time: 12
Waiting clients: (46, 13, 4); (37, 14, 2); (47, 15, 6); (18, 16, 4); (34, 16, 3); (11, 17, 4); (38, 18, 1
Queue 0 : (44, 5, 6)(10, 11, 4)
Queue 1 : (48, 6, 6)(29, 12, 4)
Queue 2 : (45, 12, 3)
Queue 3 :
Queue 4 : (25, 9, 3)

Time: 13
Waiting clients: (37, 14, 2); (47, 15, 6); (18, 16, 4); (34, 16, 3); (11, 17, 4); (38, 18, 1); (0, 19, 2)
Queue 0 : (44, 5, 6)(10, 11, 4)
Queue 1 : (48, 6, 6)(29, 12, 4)
Queue 2 : (45, 12, 3)
Queue 3 : (46, 13, 4)
Queue 4 : (25, 9, 3)

Time: 14
Waiting clients: (47, 15, 6); (18, 16, 4); (34, 16, 3); (11, 17, 4); (38, 18, 1); (0, 19, 2); (1, 19, 5);
Queue 0 : (10, 11, 4)
Queue 1 : (29, 12, 4)
Queue 2 : (45, 12, 3)
Queue 3 : (46, 13, 4)
Queue 4 : (25, 9, 3)(37, 14, 2)

Time: 15
Waiting clients: (18, 16, 4); (34, 16, 3); (11, 17, 4); (38, 18, 1); (0, 19, 2); (1, 19, 5); (2, 21, 1);
Queue 0 : (10, 11, 4)(47, 15, 6)
Queue 1 : (29, 12, 4)
Queue 2 : (45, 12, 3)
Queue 3 : (46, 13, 4)
Queue 4 : (37, 14, 2)

Time: 16
Waiting clients: (11, 17, 4); (38, 18, 1); (0, 19, 2); (1, 19, 5); (2, 21, 1); (28, 21, 5); (43, 21, 6);
Queue 0 : (10, 11, 4)(47, 15, 6)
Queue 1 : (29, 12, 4)(18, 16, 4)
Queue 2 : (34, 16, 3)
Queue 3 : (46, 13, 4)
Queue 4 : (37, 14, 2)

Time: 17
Waiting clients: (38, 18, 1); (0, 19, 2); (1, 19, 5); (2, 21, 1); (28, 21, 5); (43, 21, 6); (14, 24, 2);
Queue 0 : (10, 11, 4)(47, 15, 6)
Queue 1 : (29, 12, 4)(18, 16, 4)
Queue 2 : (34, 16, 3)
Queue 3 : (46, 13, 4)(11, 17, 4)
Queue 4 :
```



```
Time: 18
Waiting clients: (0, 19, 2); (1, 19
Queue 0 : (47, 15, 6)
Queue 1 : (18, 16, 4)
Queue 2 : (34, 16, 3)
Queue 3 : (11, 17, 4)
Queue 4 : (38, 18, 1)

Time: 19
Waiting clients: (2, 21, 1); (28, 2
Queue 0 : (47, 15, 6)(0, 19, 2)
Queue 1 : (18, 16, 4)(1, 19, 5)
Queue 2 : (34, 16, 3)
Queue 3 : (11, 17, 4)
Queue 4 : (38, 18, 1)

Time: 20
Waiting clients: (2, 21, 1); (28, 2
Queue 0 : (47, 15, 6)(0, 19, 2)
Queue 1 : (18, 16, 4)(1, 19, 5)
Queue 2 :
Queue 3 : (11, 17, 4)
Queue 4 :

Time: 21
Waiting clients: (14, 24, 2); (16,
Queue 0 : (47, 15, 6)(0, 19, 2)
Queue 1 : (18, 16, 4)(1, 19, 5)
Queue 2 : (2, 21, 1)
Queue 3 : (11, 17, 4)(28, 21, 5)
Queue 4 : (43, 21, 6)

Time: 22
Waiting clients: (14, 24, 2); (16,
Queue 0 : (47, 15, 6)(0, 19, 2)
Queue 1 : (1, 19, 5)
Queue 2 : (2, 21, 1)
Queue 3 : (28, 21, 5)
Queue 4 : (43, 21, 6)

Time: 23
Waiting clients: (14, 24, 2); (16,
Queue 0 : (47, 15, 6)(0, 19, 2)
Queue 1 : (1, 19, 5)
Queue 2 :
Queue 3 : (28, 21, 5)
Queue 4 : (43, 21, 6)

Time: 24
Waiting clients: (27, 25, 1); (31,
Queue 0 : (0, 19, 2)(14, 24, 2)
Queue 1 : (1, 19, 5)(16, 24, 4)
Queue 2 : (42, 24, 1)
Queue 3 : (28, 21, 5)
Queue 4 : (43, 21, 6)

Time: 25
Waiting clients: (31, 26, 6); (40,
Queue 0 : (0, 19, 2)(14, 24, 2)
Queue 1 : (1, 19, 5)(16, 24, 4)
Queue 2 : (42, 24, 1)
Queue 3 : (28, 21, 5)(27, 25, 1)
Queue 4 : (43, 21, 6)

Time: 26
Waiting clients: (12, 27, 5); (21,
Queue 0 : (14, 24, 2)(40, 26, 4)
Queue 1 : (1, 19, 5)(16, 24, 4)
Queue 2 :
Queue 3 : (28, 21, 5)(27, 25, 1)
Queue 4 : (43, 21, 6)(31, 26, 6)

Time: 27
Waiting clients: (19, 28, 2); (3, 29
Queue 0 : (14, 24, 2)(40, 26, 4)
Queue 1 : (16, 24, 4)(12, 27, 5)
Queue 2 : (21, 27, 3)
Queue 3 : (27, 25, 1)
Queue 4 : (43, 21, 6)(31, 26, 6)

Time: 28
Waiting clients: (3, 29, 5); (5, 30,
Queue 0 : (40, 26, 4)
Queue 1 : (16, 24, 4)(12, 27, 5)
Queue 2 : (21, 27, 3)
Queue 3 : (19, 28, 2)
Queue 4 : (31, 26, 6)

Time: 29
Waiting clients: (5, 30, 5); (9, 30,
Queue 0 : (40, 26, 4)
Queue 1 : (16, 24, 4)(12, 27, 5)
Queue 2 : (21, 27, 3)
Queue 3 : (19, 28, 2)
Queue 4 : (31, 26, 6)(3, 29, 5)

Time: 30
Waiting clients: (6, 31, 3); (33, 31, 2); (36, 31, 3); (24, 32, 1); (1
Queue 0 : (40, 26, 4)(5, 30, 5)
Queue 1 : (16, 24, 4)(12, 27, 5)(9, 30, 2)
Queue 2 : (21, 27, 3)(30, 30, 4)
Queue 3 : (19, 28, 2)
Queue 4 : (31, 26, 6)(3, 29, 5)

Time: 31
Waiting clients: (24, 32, 1); (15, 33, 3); (17, 33, 1); (23, 35, 6); (
Queue 0 : (40, 26, 4)(5, 30, 5)(36, 31, 3)
Queue 1 : (12, 27, 5)(9, 30, 2)
Queue 2 : (30, 30, 4)
Queue 3 : (6, 31, 3)
Queue 4 : (31, 26, 6)(3, 29, 5)(33, 31, 2)

Time: 32
Waiting clients: (15, 33, 3); (17, 33, 1); (23, 35, 6); (26, 35, 2); (
Queue 0 : (5, 30, 5)(36, 31, 3)
Queue 1 : (12, 27, 5)(9, 30, 2)(24, 32, 1)
Queue 2 : (30, 30, 4)
Queue 3 : (6, 31, 3)
Queue 4 : (31, 26, 6)(3, 29, 5)(33, 31, 2)

Time: 33
Waiting clients: (23, 35, 6); (26, 35, 2); (20, 37, 4); (22, 38, 4); (
Queue 0 : (5, 30, 5)(36, 31, 3)
Queue 1 : (12, 27, 5)(9, 30, 2)(24, 32, 1)
Queue 2 : (30, 30, 4)(15, 33, 3)
Queue 3 : (6, 31, 3)(17, 33, 1)
Queue 4 : (31, 26, 6)(3, 29, 5)(33, 31, 2)

Time: 34
Waiting clients: (23, 35, 6); (26, 35, 2); (20, 37, 4); (22, 38, 4); (
Queue 0 : (5, 30, 5)(36, 31, 3)
Queue 1 : (12, 27, 5)(9, 30, 2)(24, 32, 1)
Queue 2 : (30, 30, 4)(15, 33, 3)
Queue 3 : (6, 31, 3)(17, 33, 1)
Queue 4 : (3, 29, 5)(33, 31, 2)

Time: 35
Waiting clients: (20, 37, 4); (22, 38, 4); (35, 38, 5); (41, 38, 3);
Queue 0 : (5, 30, 5)(36, 31, 3)(26, 35, 2)
Queue 1 : (12, 27, 5)(9, 30, 2)(24, 32, 1)
Queue 2 : (15, 33, 3)
Queue 3 : (17, 33, 1)
Queue 4 : (3, 29, 5)(33, 31, 2)(23, 35, 6)

Time: 36
Waiting clients: (20, 37, 4); (22, 38, 4); (35, 38, 5); (41, 38, 3);
Queue 0 : (5, 30, 5)(36, 31, 3)(26, 35, 2)
Queue 1 : (9, 30, 2)(24, 32, 1)
Queue 2 : (15, 33, 3)
Queue 3 :
Queue 4 : (3, 29, 5)(33, 31, 2)(23, 35, 6)

Time: 37
Waiting clients: (22, 38, 4); (35, 38, 5); (41, 38, 3);
Queue 0 : (36, 31, 3)(26, 35, 2)
Queue 1 : (9, 30, 2)(24, 32, 1)(20, 37, 4)
Queue 2 : (15, 33, 3)
Queue 3 :
Queue 4 : (3, 29, 5)(33, 31, 2)(23, 35, 6)

Time: 38
Waiting clients:
Queue 0 : (36, 31, 3)(26, 35, 2)
Queue 1 : (24, 32, 1)(20, 37, 4)
Queue 2 : (22, 38, 4)
Queue 3 : (35, 38, 5)
Queue 4 : (3, 29, 5)(33, 31, 2)(23, 35, 6)(41, 38, 3)

Time: 39
Waiting clients:
Queue 0 : (36, 31, 3)(26, 35, 2)
Queue 1 : (20, 37, 4)
Queue 2 : (22, 38, 4)
Queue 3 : (35, 38, 5)
Queue 4 : (33, 31, 2)(23, 35, 6)(41, 38, 3)

Time: 40
Waiting clients:
Queue 0 : (26, 35, 2)
Queue 1 : (20, 37, 4)
Queue 2 : (22, 38, 4)
Queue 3 : (35, 38, 5)
Queue 4 : (33, 31, 2)(23, 35, 6)(41, 38, 3)

Time: 41
Waiting clients:
Queue 0 : (26, 35, 2)
Queue 1 : (20, 37, 4)
Queue 2 : (22, 38, 4)
Queue 3 : (35, 38, 5)
Queue 4 : (23, 35, 6)(41, 38, 3)

Time: 42
Waiting clients:
Queue 0 :
Queue 1 : (20, 37, 4)
Queue 2 : (22, 38, 4)
Queue 3 : (35, 38, 5)
Queue 4 : (23, 35, 6)(41, 38, 3)

Time: 43
Waiting clients:
Queue 0 :
Queue 1 :
Queue 2 :
Queue 3 : (35, 38, 5)
Queue 4 : (23, 35, 6)(41, 38, 3)

Time: 44
Waiting clients:
Queue 0 :
Queue 1 :
Queue 2 :
Queue 3 :
Queue 4 : (23, 35, 6)(41, 38, 3)

Time: 45
Waiting clients:
Queue 0 :
Queue 1 :
Queue 2 :
Queue 3 :
Queue 4 : (23, 35, 6)(41, 38, 3)

Time: 46
Waiting clients:
Queue 0 :
Queue 1 :
Queue 2 :
Queue 3 :
Queue 4 : (23, 35, 6)(41, 38, 3)

Time: 47
Waiting clients:
Queue 0 :
Queue 1 :
Queue 2 :
Queue 3 :
Queue 4 : (41, 38, 3)

Time: 48
Waiting clients:
Queue 0 :
Queue 1 :
Queue 2 :
Queue 3 :
Queue 4 : (41, 38, 3)

Time: 49
Waiting clients:
Queue 0 :
Queue 1 :
Queue 2 :
Queue 3 :
Queue 4 : (41, 38, 3)
```

For Test3 here is the output result:



N = 1000

Q = 20

MAX simulation time : 200

MIN/MAX arrival time : 10 100

MIN/MAX service time : 3 9

Logs :

```
Time: 200
Waiting clients:
Queue 0 : (418, 74, 3)(254, 76, 5)(545, 77, 6)(897, 79,
Queue 1 : (556, 74, 5)(403, 76, 6)(798, 77, 5)(79, 80, 6
Queue 2 : (985, 70, 8)(828, 72, 5)(569, 74, 6)(449, 76,
Queue 3 : (778, 65, 4)(356, 67, 8)(542, 69, 3)(27, 71, 4
Queue 4 : (68, 71, 8)(16, 73, 7)(749, 74, 5)(552, 76, 6)(
Queue 5 : (88, 71, 8)(49, 73, 3)(761, 74, 8)(587, 76, 7)(
Queue 6 : (263, 71, 5)(204, 73, 5)(788, 74, 8)(691, 76,
Queue 7 : (359, 73, 4)(850, 74, 7)(693, 76, 6)(306, 78,
Queue 8 : (121, 75, 8)(701, 76, 8)(423, 78, 3)(602, 80,
Queue 9 : (531, 73, 6)(194, 75, 4)(771, 76, 7)(473, 78,
Queue 10 : (261, 75, 8)(898, 76, 4)(548, 78, 7)(695, 80
Queue 11 : (575, 73, 6)(461, 75, 8)(911, 76, 4)(739, 78
Queue 12 : (801, 78, 7)(31, 81, 3)(75, 83, 6)(674, 84, 3
Queue 13 : (717, 73, 6)(621, 75, 7)(286, 77, 3)(909, 78
Queue 14 : (646, 75, 6)(302, 77, 3)(500, 79, 4)(274, 81
Queue 15 : (661, 75, 8)(327, 77, 7)(744, 79, 6)(457, 81
Queue 16 : (948, 68, 8)(784, 70, 8)(179, 72, 6)(100, 74
Queue 17 : (785, 70, 7)(326, 72, 3)(226, 74, 6)(860, 75
Queue 18 : (790, 70, 8)(354, 72, 4)(297, 74, 7)(92, 76,
Queue 19 : (791, 70, 6)(390, 72, 4)(397, 74, 3)(195, 76
```

6. Conclusions

This assignment was a good practice in remembering the OOP concepts learned in the first semester, but also learning the new ones. At first, I found it challenging and hard to understand, but after putting things on paper it was a lot easier.

First of all, this project helped me understand the notion of Threads in Java. I have understood how to create a class that runs Threads using the "implement Runnable" and how to manage multiple queues at the same time. As well as for the BlockingQueue<> interface and how to put Threads to sleep

On the other hand, searching for new concepts online and figuring them by myself, helped me getting a better understanding of the idea of the subject.

By the means of this project I managed to improve my knowledge about queues processing and how they are implemented on a system, I learned about Threads and about synchronizing them, how to put a Thread to sleep and how to create

7. Bibliography

The references that I consulted during the implementation of assignment:

1. Courses: https://dsrl.eu/courses/pt/
2. Thread Synchronization in Java: https://dotnettutorials.net/lesson/thread-synchronization-in-java/
3. BlockingQueue<> methods: https://jenkov.com/tutorials/java-util-concurrent/blockingqueue.html
4. Java Thread and Runnable Tutorial: https://www.callicoder.com/java-multithreading-thread-and-runnable-tutorial/
5. How to Use Scroll Panes: https://docs.oracle.com/javase/tutorial/uiswing/components/scrollpane.html
6. Java Runnable Interface: https://www.javatpoint.com/runnable-interface-in-java