1. Simulation for the newspaper seller's problem

In class, we discussed the optimization problem for the newspaper seller by simulation. Complete the computer program for the newspaper sellers to find the optimal number for maximum profit. Run your program to simulate the system for 1000, 10000, and 100000 days.

Number of Newspapers	40	50	60	70	80	90	100	Optimal Number
1000 days	-206.80	1,935.15	3,799.59	5,171.09	5,768.50	2,316.99	586.24	80
10000 days	139.75	18,918.39	37,732.29	45,778.20	46,848.44	29,873.14	6,614.45	80
100000 days	-9,365.34	194,371.04	374,652.94	468,859.35	478,260.39	303,179.69	64,185.65	80

2. Single server queue simulation

(a) Modify the program SingleServerQueue.java by adding the capability to compute the maximum delay, the proportion of jobs delayed and the number of jobs in the system at a specified time (known at compile time).

At top of while loop:

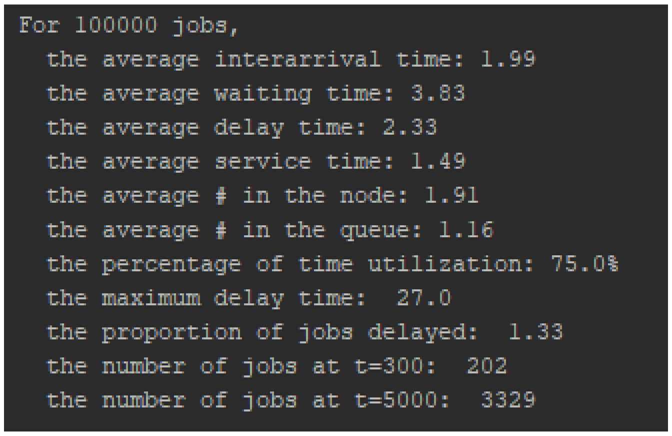
```
if (295 < time && time < 305)
    t = index;
if (4995 < time && time < 5005)
    y = index;

arrival=getArrival(arrival);
if(arrival<departure) {
    delay = departure - arrival;            // Delay in queue
    if (curr < delay) {                     // If delay is larger than largest delay
        curr = Math.round(delay);           // Set largest delay to delay
    }
}
```

At print statements

```
System.out.println("  the maximum delay time:  "+curr);
System.out.println("  the proportion of jobs delayed:  "
        + (Math.round((quotient(quotient(sum.interarrival, index),
quotient(sum.service, index))) * 100.0) / 100.0));
System.out.println("  the number of jobs at t=300:  "+Math.round(t));
System.out.println("  the number of jobs at t=5000:  "+Math.round(y));
```

(b) Run your program for 100000 jobs. What was the maximum delay experienced? What proportion of jobs was delayed? How many jobs were in the system at time t=300 and t=5000 respectively?



```
For 100000 jobs,
  the average interarrival time: 1.99
  the average waiting time: 3.83
  the average delay time: 2.33
  the average service time: 1.49
  the average # in the node: 1.91
  the average # in the queue: 1.16
  the percentage of time utilization: 75.0%
  the maximum delay time:  27.0
  the proportion of jobs delayed:  1.33
  the number of jobs at t=300:  202
  the number of jobs at t=5000:  3329
```

(c) Discuss the logic for your program modification.

Maximum delay

- At top of while loop check if job is delayed
- If current tasks' wait time is larger than stored maximum delay time
  - Replace stored wait time with current tasks' wait time

Proportion of jobs delayed

- λ: mean rate of arrival = sum of interarrival / index
- μ: mean service rate = sum of service time / index
- ρ = λ/μ utilization of the server
  - Print ρ

Number of jobs at t=300 and t=5000

- Create variable to store the current time of the system
- For every iteration of the while loop add the duration of the job to the current time
- If the current time is equal to a t=300 or t=5000
  - Return/print index (number of jobs)

3. A single-server machine shop simulation

(a) Modify the program SingleServerMachineShop1.java to output the number of failures for each machine and the number of machines at the Service Station at a specified time (known at compile time).

Change nextFailure(…)

```
public static double nextFailure(double[] failure, myInt m)
{
    double t= failure[0];
    int f = 0;
    m.setInt(0);
    for(int i=1; i<M; i++)
        if(failure [i]<t)
        {
            f++;
            t=failure[i];
            m.setInt(i);
        }
    System.out.println("Machine failures: "+f);
    return t;
}
```

Top of while loop

```
if(intarrive < 0.1){
    t++;
}

if (995 < spec && spec < 1005)
    sum.f = String.valueOf(t);
if (4995 < spec && spec < 5005)
    sum.fff = String.valueOf(t);

arrival  =nextFailure(failure, m);
if(arrival<departure) {
    delay = departure - arrival;
    t--;
}
```

(b) Run your program for 100000 machine failures. Output the number of failures for each machine and the number of machines at the Service Station at time t=1000 and t=5000 respectively.

(c) Further modify your program from (a) to stop the simulation at a specified time (defined in the program as a constant, for example, the simulation will stop at time t=100000, instead of a number of machine failures).

(d) Run your program for 100000 time unit. Output the number of failures for each machine and the number of machines at the Service Station at time t=2000 and t=10000 respectively.

(e)Discuss the logic for your program modifications.