To run the program, type the following commands:

make
./netsim <seed> <graphFileName>

If a file with the given file name does EXIST in the folder, the program reads the graph from it.
If a file with the given file name dose NOT exist in the folder, the program CREATS a graph and stores it in a file with this file name, then reads the graph from it.
There is a graph file named *graph.txt* in the folder, here is an example output of the program:

```
ramin@LAPTOP:~/Network Simulator$ ./netsim 3 graph.txt
Graph was read from graph.txt
Graph is completely connected.
Shortest Paths found.
Routing Statistics:
Total number of packets generated: 20000
Total number of packets that reached destination: 10326
Time for completion for the transmissions:
Max: 105 seconds
Min: 0 seconds
Avg: 8 seconds
Number for packets dropped at a router:
Max: 4175
Min: 0
Avg: 1777
```

An example output of the program with a file name which doesn't exist would be:

```
ramin@LAPTOP:~/Network Simulator$ ./netsim 3 newGraph.txt
Created graph
Generated edges
Graph is saved to newGraph.txt
Graph was read from newGraph.txt
Graph is completely connected.
Shortest Paths found.
Routing Statistics:
Total number of packets generated: 20000
Total number of packets that reached destination: 11147
Time for completion for the transmissions:
Max: 174 seconds
Min: 0 seconds
Avg: 5 seconds
Number for packets dropped at a router:
Max: 4001
Min: 0
Avg: 2713
```

When the program initiates, first it collects the seed and graphFileName arguments that will be used to create the graph. The graph will then be generated. Once this is done, the graph will be saved into a text file and immediately read. Then the shortest paths are calculated, packets are going to be sent throughout the network. A global virtual clock is used to simulate the progress of time. This clock is represented by a for loop. For each instance of time (some value of the for loop index), the program performs all the events that need to be performed in that time instance. The program also will keep track of the total packets transmitted, total packets lost, and the delay.