Erdinç Kandemir
150150110

10.12.2017

ITU COMPUTER ENGINEERING DEPARTMENT

BLG 233E DATA STRUCTURES

HOMEWORK -3

REPORT

///Used variable names in the program are showed in report like → *"variable" ///*

First of all, in program, all patients are held with *"patientData"* struct. And queues are based on enqueueing nodes which hold patientData in it. There are queue functions, such as
-create()
-enqueue()
-dequeue()
-isEmpty().

The program reads *"patientsInfo.txt"* file line by line(it discards first headline line) and puts elements appropriate data parts. It enqueues them in "patientList" queue struct.

After whole file is read. The program start treatments.

There are 5 different queues named;

-redNew
-yellowNew
-yellow
-greenNew
-green
(sorted based on priorities).

Patients whose arriving time is smaller then current time*("curTime")* are enqueued.

Whole patients are firstly enqueued in New queues in order to give priority to new patients. Patients who treated before are enqueued in normal queues.

After enqueued necessary new comers, the program check queues in order written above. If all queues are empty and there will be no any new comers(*"patientList"* is empty) the program ends("END" situation). If all queues are empty but there will be new comers, program prints "No Treatment" message on the console.

If any queue is not empty, program get one patient to treat(which is held *"curPatient"*). Based on patients code, current time, patients code and patients time to need is setting. Patients whose treat ends will leave hospital and others are enqueued appropriate queues based on their codes.

And program check newcomers, queues again until the end situation("END").

Program works with input style like this;

```
name            arriving time       treatment time      complaint
Patient1        0                   7                   RED
Patient2        2                   3                   RED
Patient3        3                   6                   GREEN
```

its output for input above;

```
1.time slice Patient1
2.time slice Patient1
3.time slice Patient1
4.time slice Patient2
5.time slice Patient2
6.time slice Patient2
7.time slice Patient3
8.time slice Patient3
9.time slice Patient3
10.time slice Patient1
11.time slice Patient1
12.time slice Patient1
13.time slice Patient3
14.time slice Patient3
15.time slice Patient3
16.time slice Patient1
Program ended with exit code: 0
```

All memory to keep patients are releasing during the dequeueing proccess. All other variables are static,