**BLG 233E HOMEWORK 3 REPORT**

**SOME EXPLANATIONS**

1. In the patients file, after the patient’s name, the first integer is the arrival time and the second integer is the treatment time. The third variable is the patient’s code.
2. The patients are given a minimum treatment time based on their codes. (RED: 1, YELLOW: 2, GREEN: 3)

**QUEUE FUNCTIONS**

I implemented my queue structure with create, enqueue, dequeue, peek, isempty, clear and print functions.

**Create:** Sets the front and back pointers to NULL.

**Enqueue:** Puts the patient to the back of the queue.

**Dequeue:** Returns the front element in the queue by popping it. If the queue is empty, returns NULL.

**Peek:** Returns the front element in the queue without popping it. (Front pointer is not changed.)

**isEmpty:** Returns whether the queue is empty (front == NULL) or not.

**Clear:** I wrote this function when I started the project but never used it. It simply clears a queue.

**Print:** Prints a queue. I used it for debugging purposes.

**PROGRAM WORKFLOW**

My program first reads all of the “patientsInfo.txt” file, then puts it in a queue. After reading all elements it checks the front element to see if it’s the arrival time of front, in a while loop. If it’s the arrival time the front element is dequeued and placed in the redQueue, yellowQueue or greenQueue according to it’s code.

After that the program checks if we just finished treating a patient. If we did, the program updates that patient’s code and puts the patient in the respective queue.

After that if we’re not treating any patients at the moment, the program looks at first the redQueue then the yellowQueue and then the greenQueue. If we find an element in a queue, we stop looking and continue. (for example we don’t check the greenQueue if we found a yellow patient.)

Then the program checks if we have a current patient and are we treating him/her. If not, it starts the treatment and calculates the treatment time.

Next, since we decided what we’re doing in the current time slice, we print it.

If we’re treating a patient it checks if the treatment time is finished or not. If the treatment time is finished and the patient is cured, it deletes the patient. If the treatment time is finished and the patient is not cured, we update the code of the patient.

The program works until all queues are empty and all patients have been cured (and deleted). In the end I wrote a getchar() functions for users to see the output.

**OTHER FUNCTIONS**

**createPatient:** Creates a new patient node.

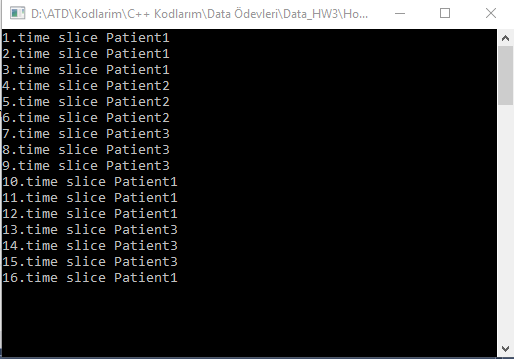
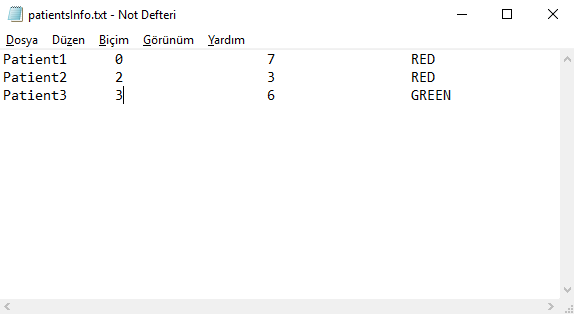
**readFromFile:** It reads the file “patientsInfo.txt” via filestream.

**deletePatient:** Deletes a patient node with all of it’s contents.

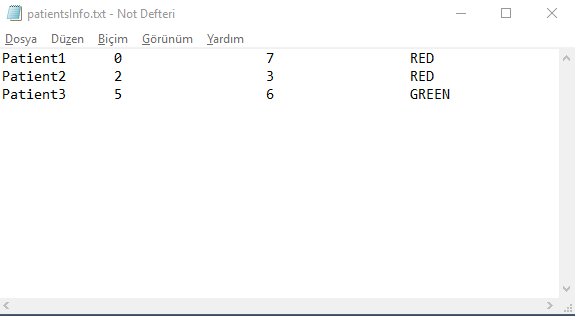
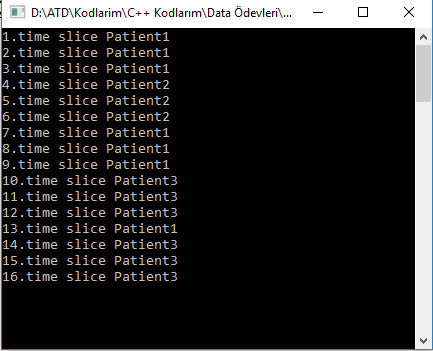
**minimumTreat:** Returns the minimum treat time of a patient (RED = 1, YELLOW = 2, GREEN = 3). If the patient’s remaining treatment time is shorter, it returns the remaining time.

**codeUpdate:** Updates the code at the end of each treatment. (RED to YELLOW, YELLOW to GREEN, doesn’t do anything for other inputs).

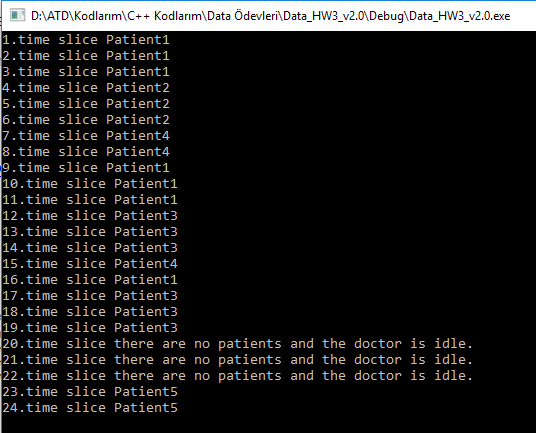
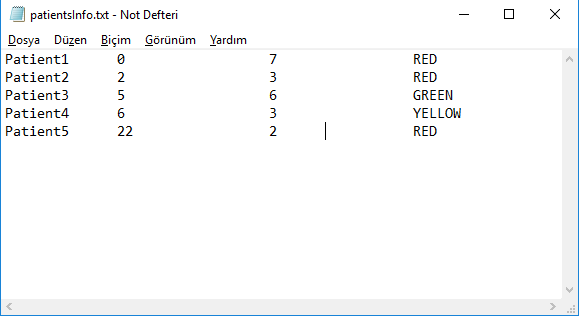
**MY INPUT OUTPUT**



Input – Output 1



Input – Output 2



Input – Output 3

Ali Tolga Dinçer

Student No: 150150730

Mail: [dincer15@itu.edu.tr](mailto:dincer15@itu.edu.tr)