**Project Topic :-**

**Faculty Query- Time Scheduler**

[USing RR Algorithm]

Submitted for:-

**Operating System (UCS303)**

Submitted To:-

**Dr. Shashank S. Singh Sir**

Submitted By:-

**Aryan Varshney – 101903605**

**Rohan Kumar -101903625**

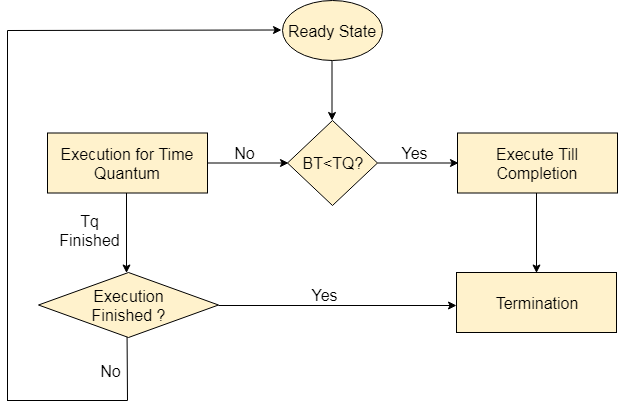


**THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY,**

**(Deemed To Be University), Patiala, Punjab, India**

**Introduction :-**

Round Robin scheduling algorithm is one of the most popular scheduling algorithm which can actually be implemented in most of the operating systems. This is the **pre-emptive version** of first come first serve scheduling. The Algorithm focuses on Time Sharing. In this algorithm, every process gets executed in a **cyclic way**. A certain time slice is defined in the system which is called time **quantum**. Each process present in the ready queue is assigned the CPU for that time quantum, if the execution of the process is completed during that time then the process will **terminate** else the process will go back to the **ready queue** and waits for the next turn to complete the execution.



**Problem Statement :-**

" Dr. Shashank Sir is a Linux expert who wants to have an online system where he can handle student queries. Since there can be multiple requests at any time he wishes to dedicate a fixed amount of time to every request so that everyone gets a fair share of his time. He will log into the system from 10am to 12am only. He wants to have separate requests queues for students and faculty. Implement a strategy for the same. The summary at the end of the session should include the total time he spent on handling queries and average query time. "

**Solution :-**

The given problem is scheduling problem. The problem can be solved by Round Robin algorithm.

Program execution sequence:

1. Taking inputs of queries from user

2. Sorting all queries according to Arrival-Time

3. Merging all queries (initial priority to Faculty's query)

4. Applying Round-Robin algorithm on merged queries

5. Print the result

**Steps to follow to execute the program:-**

1. Enter number of queries between 0 & 120

2. Make sure to keep value of Time-Quantum minimum

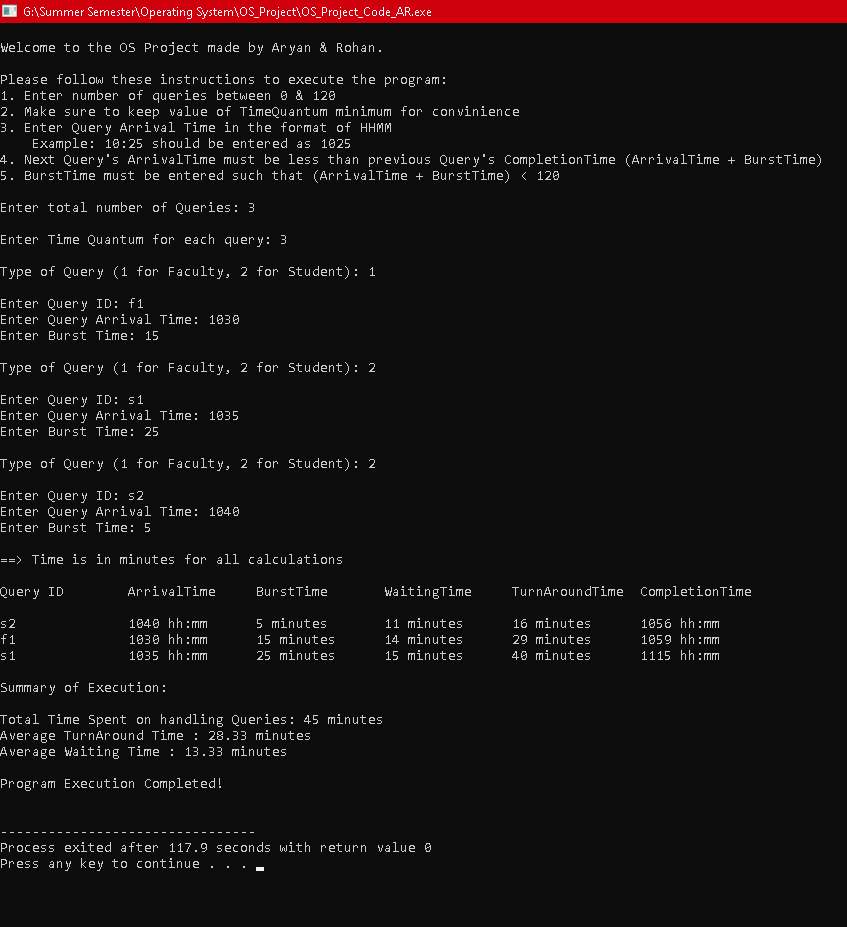
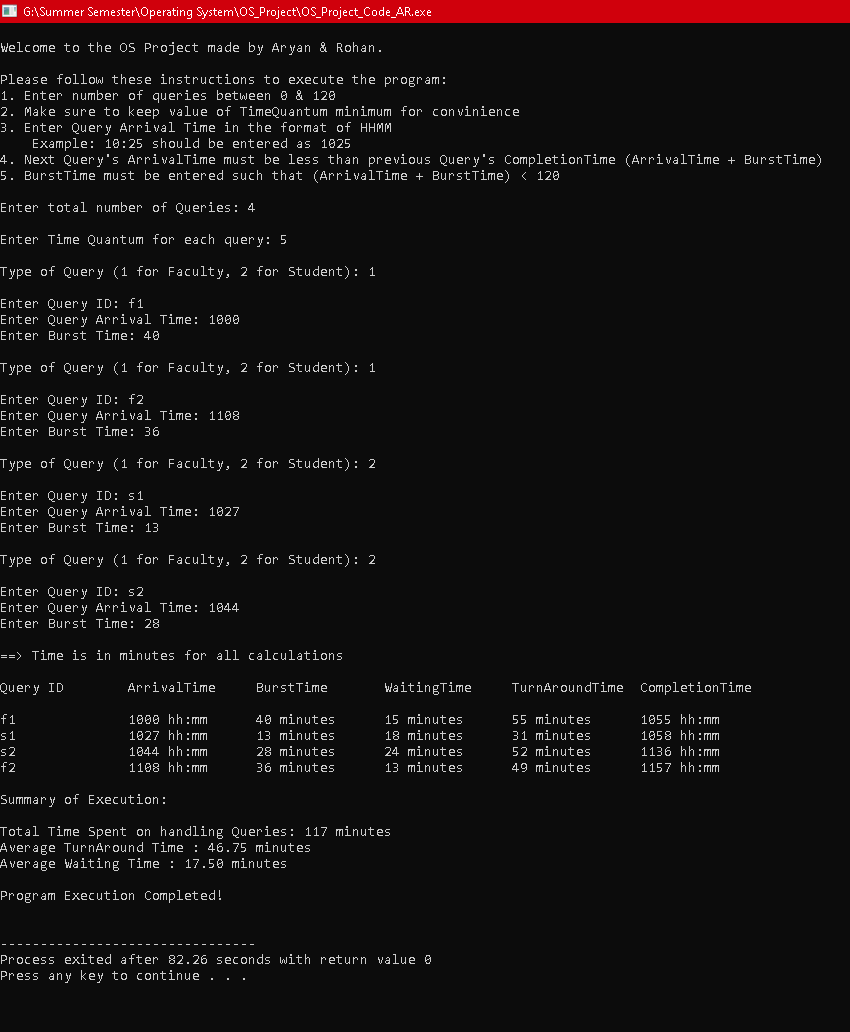
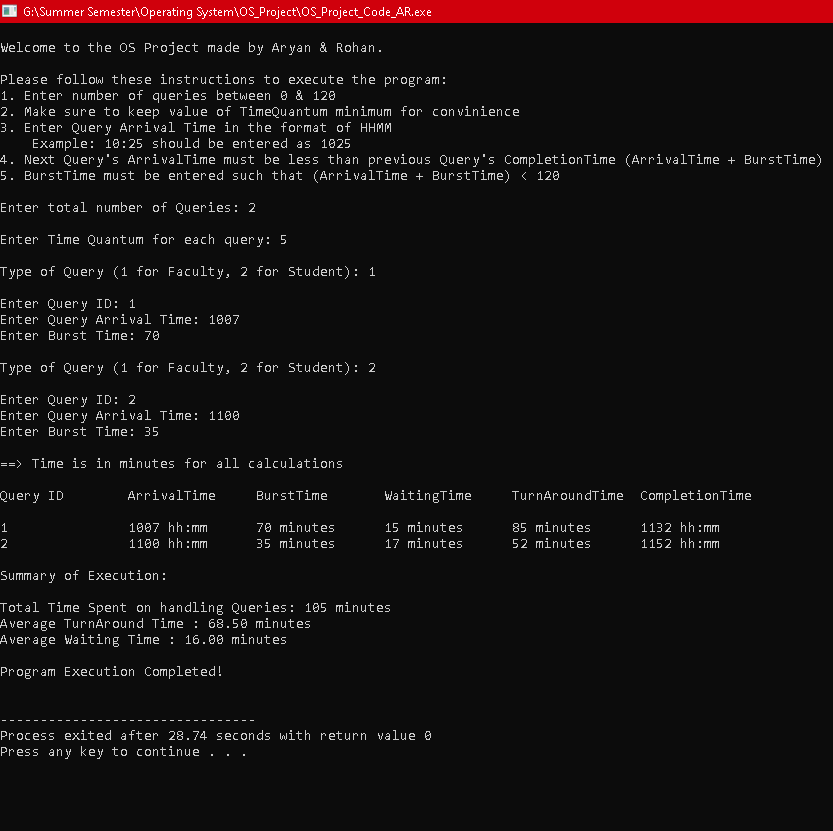
3. Enter Query Arrival Time in the format of HHMM

Example: 10:25 should be entered as 1025

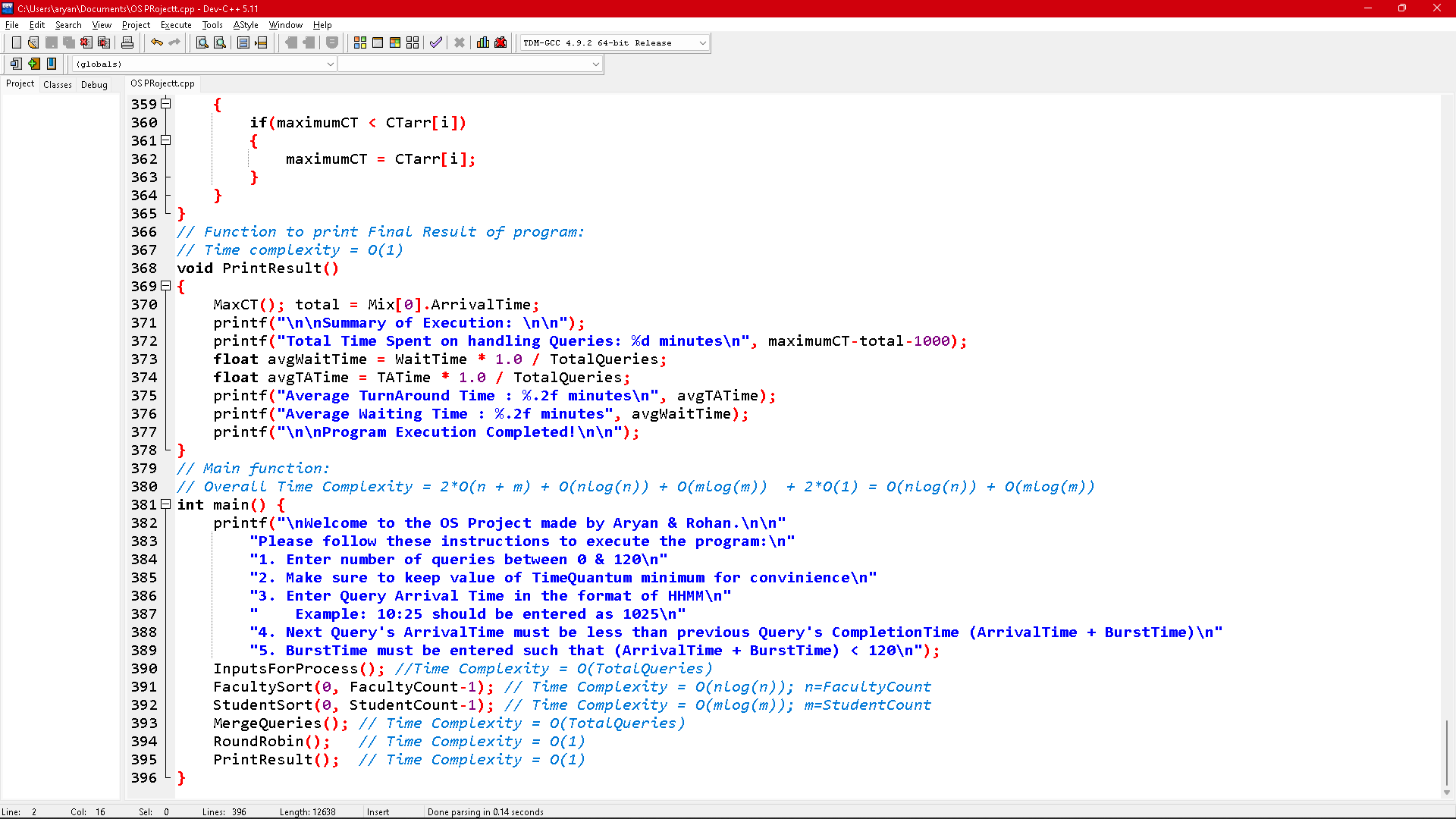
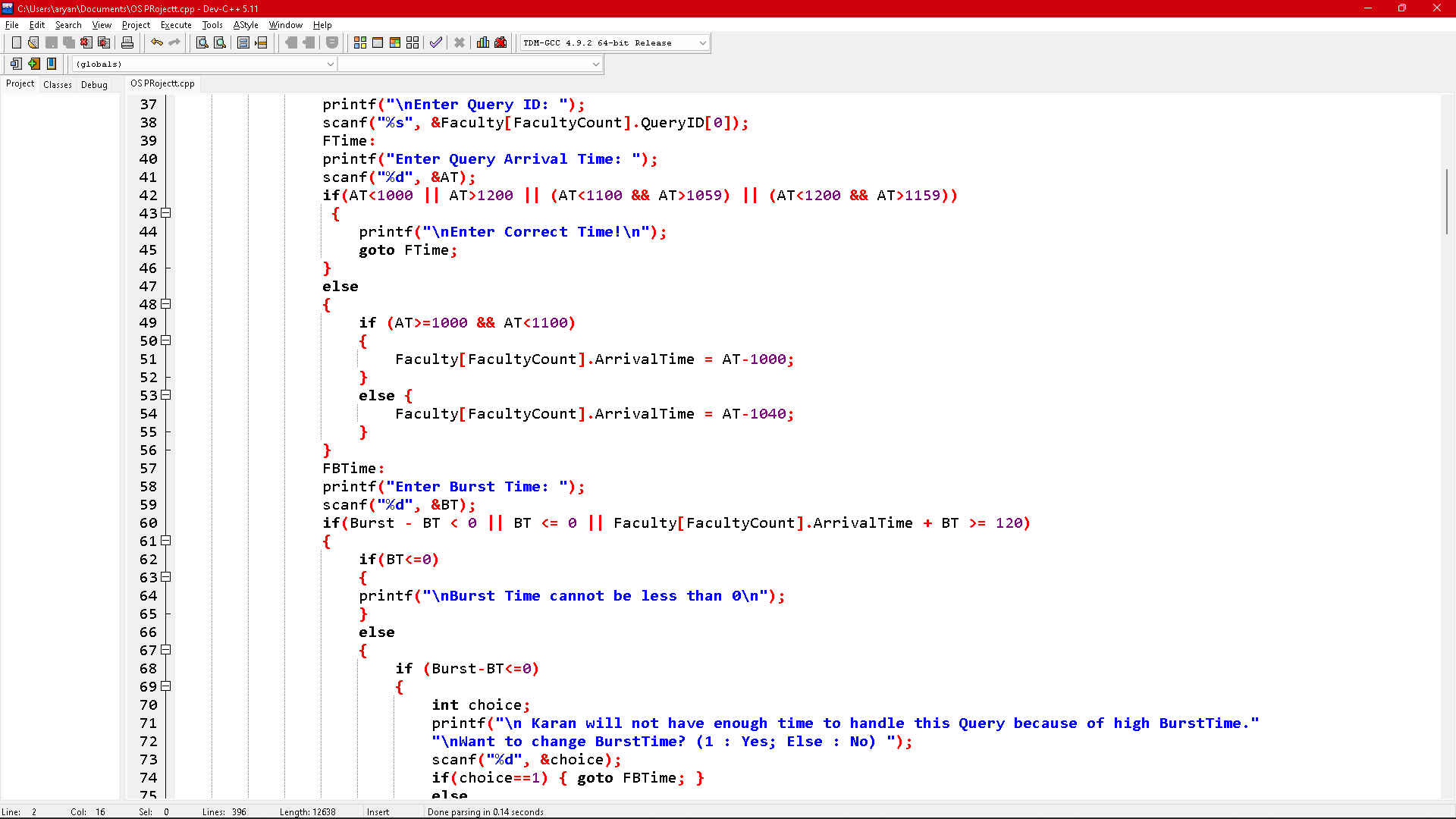
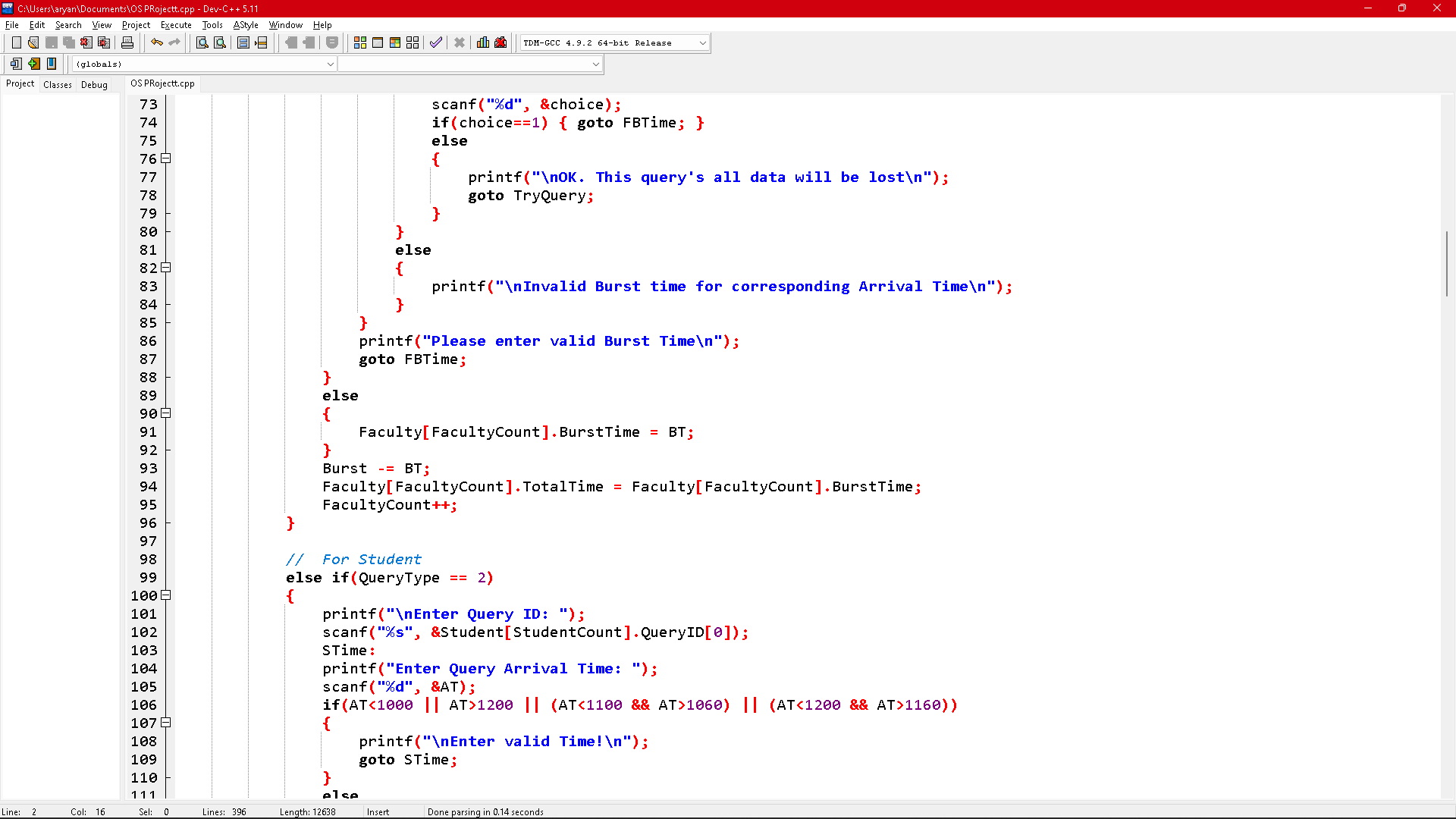
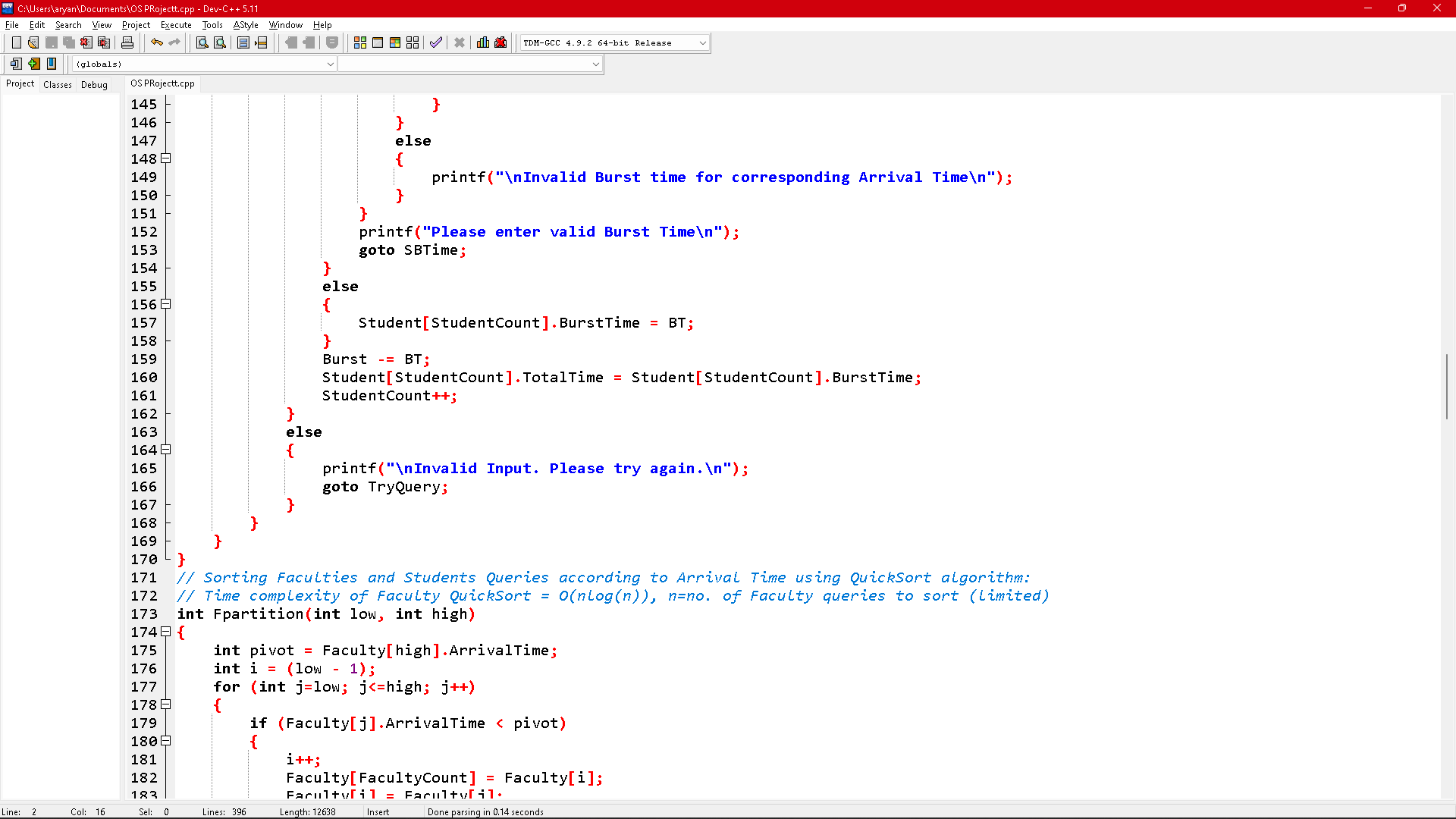
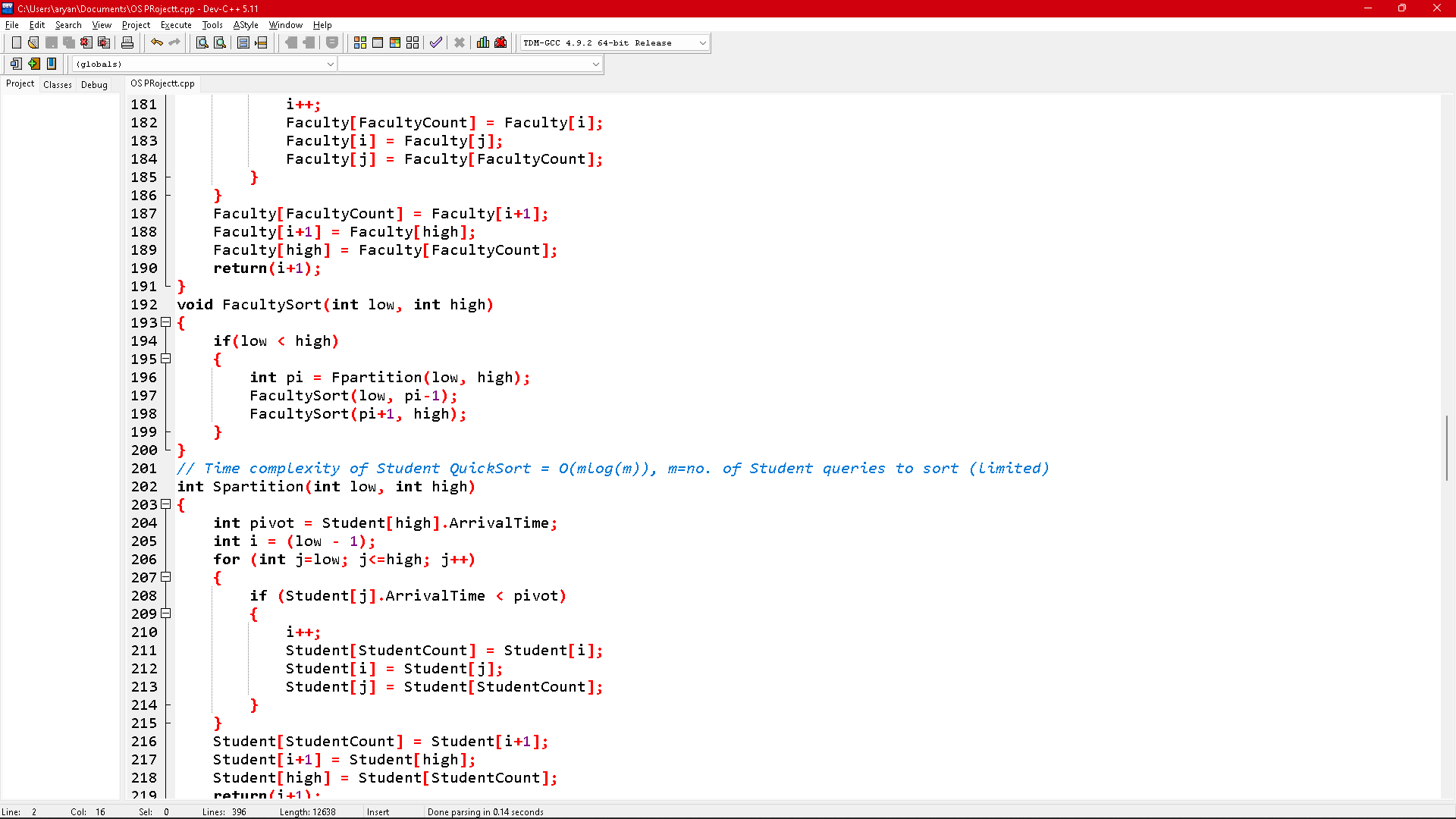
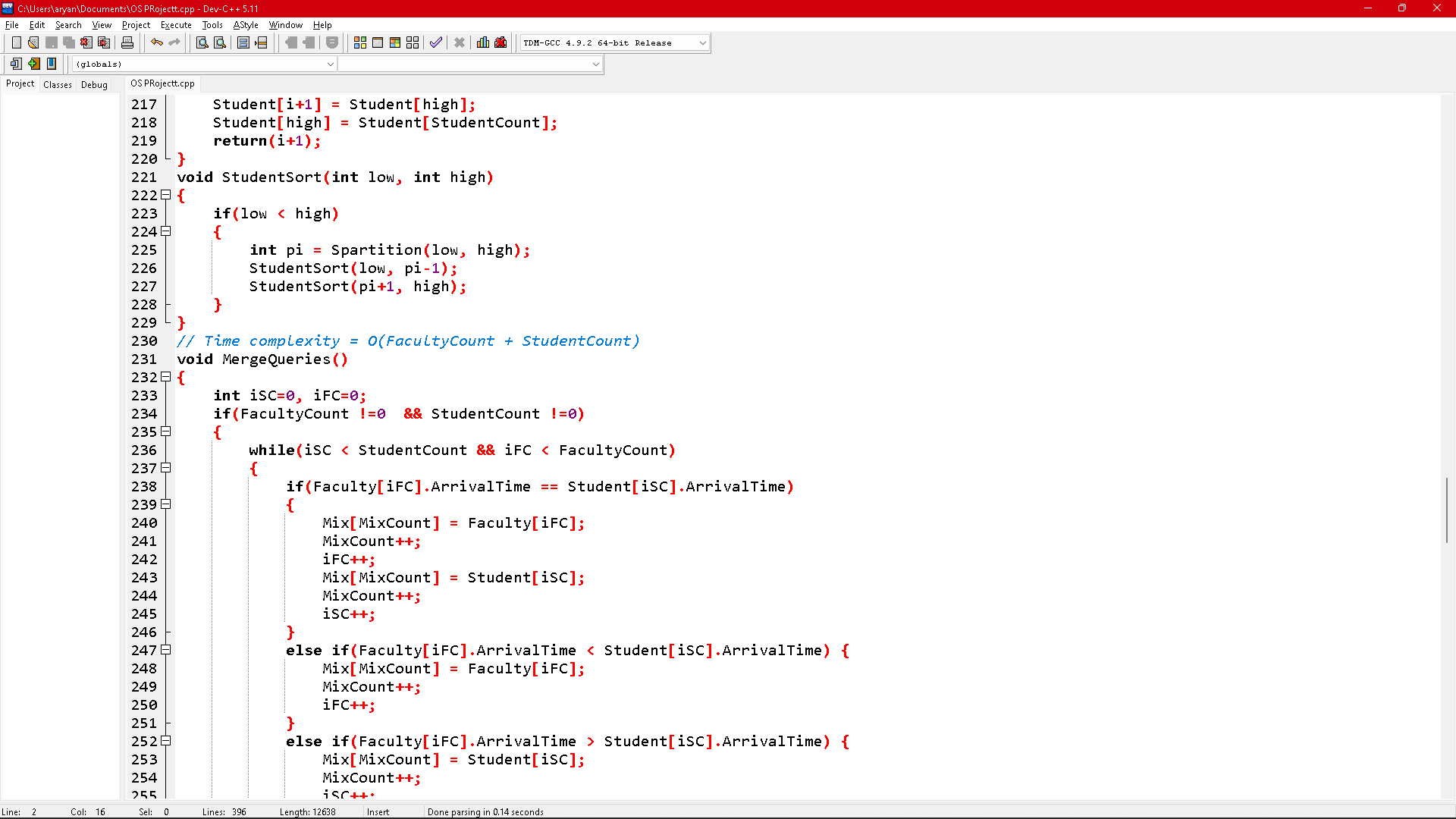
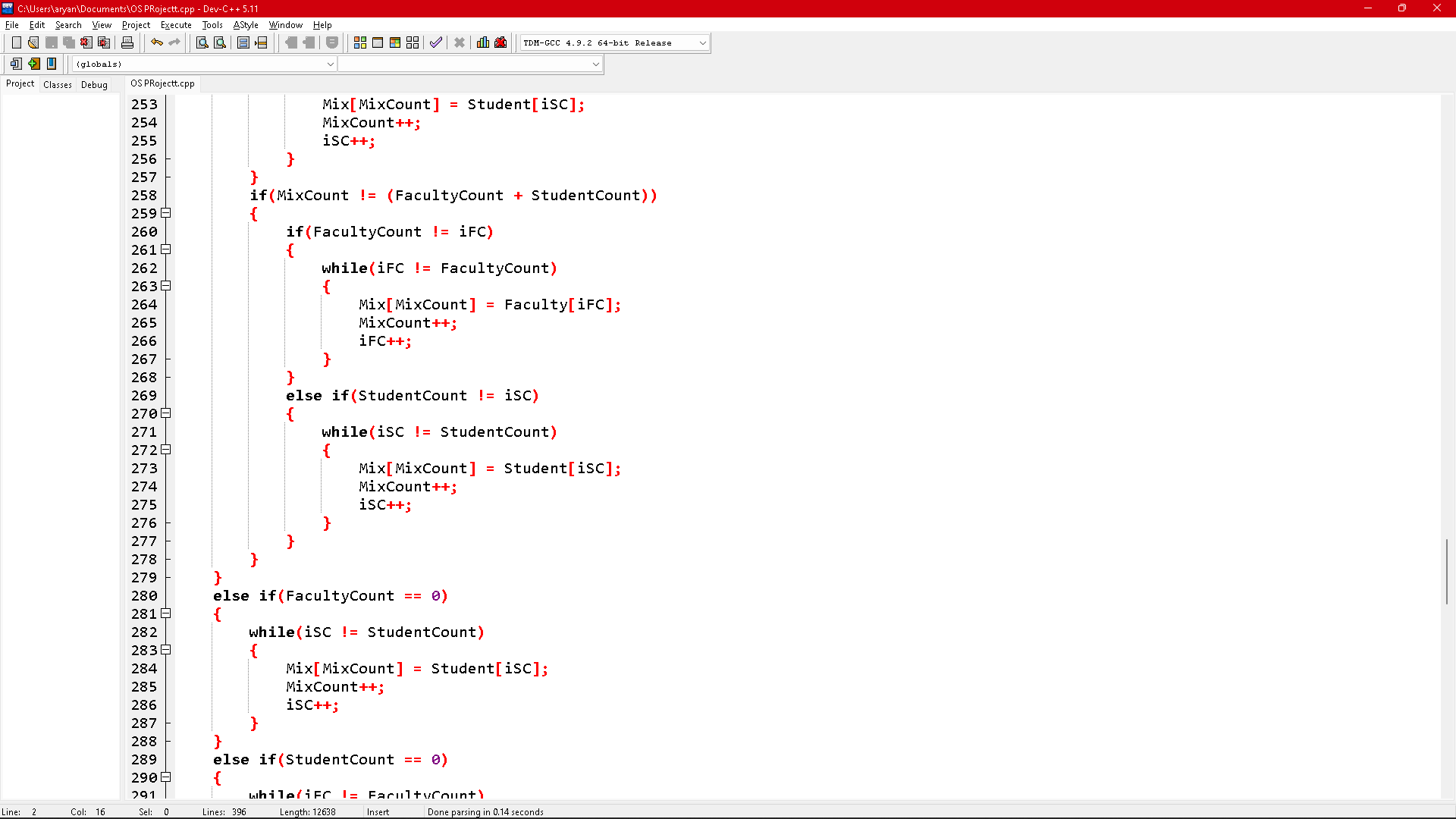
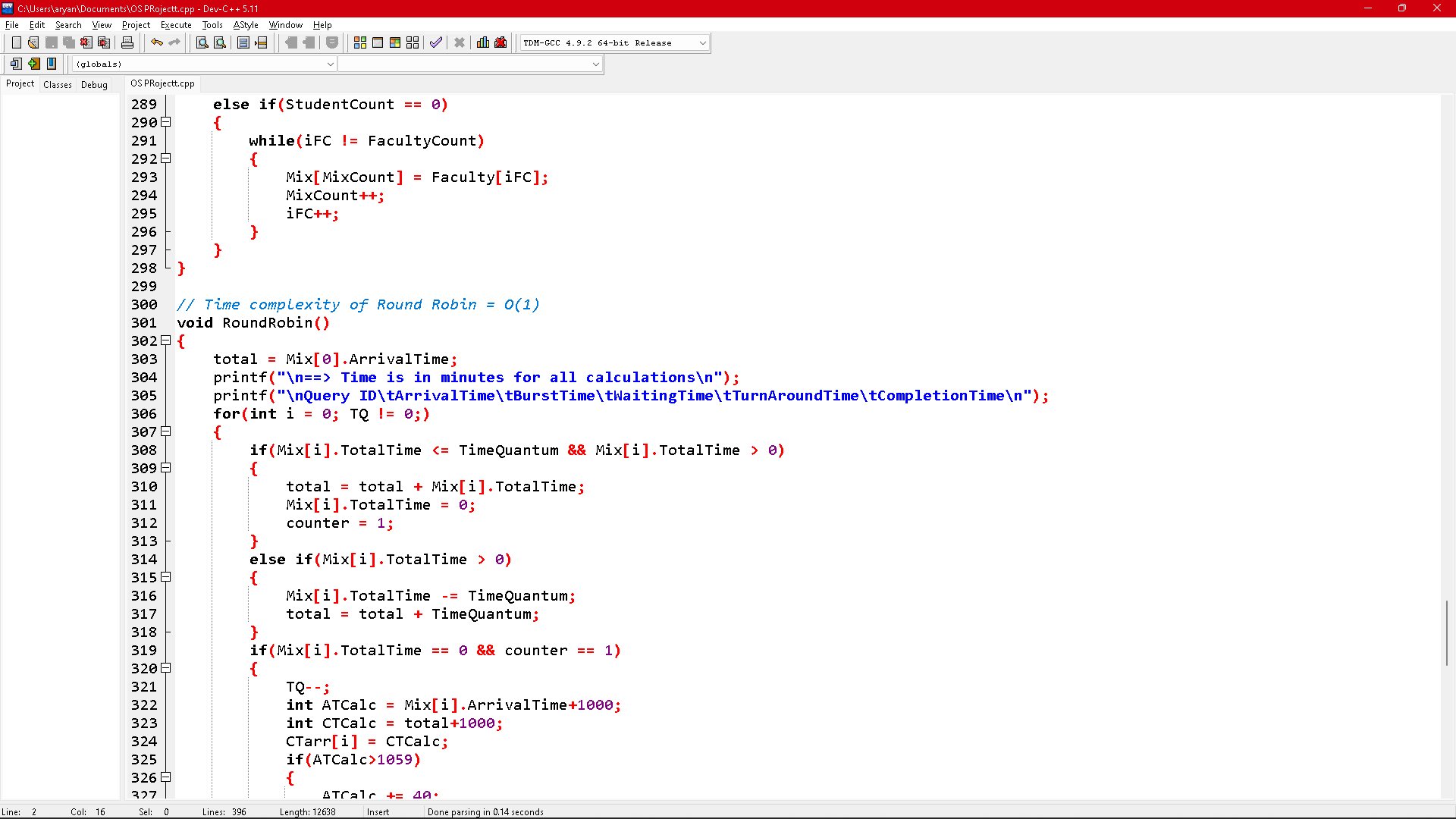
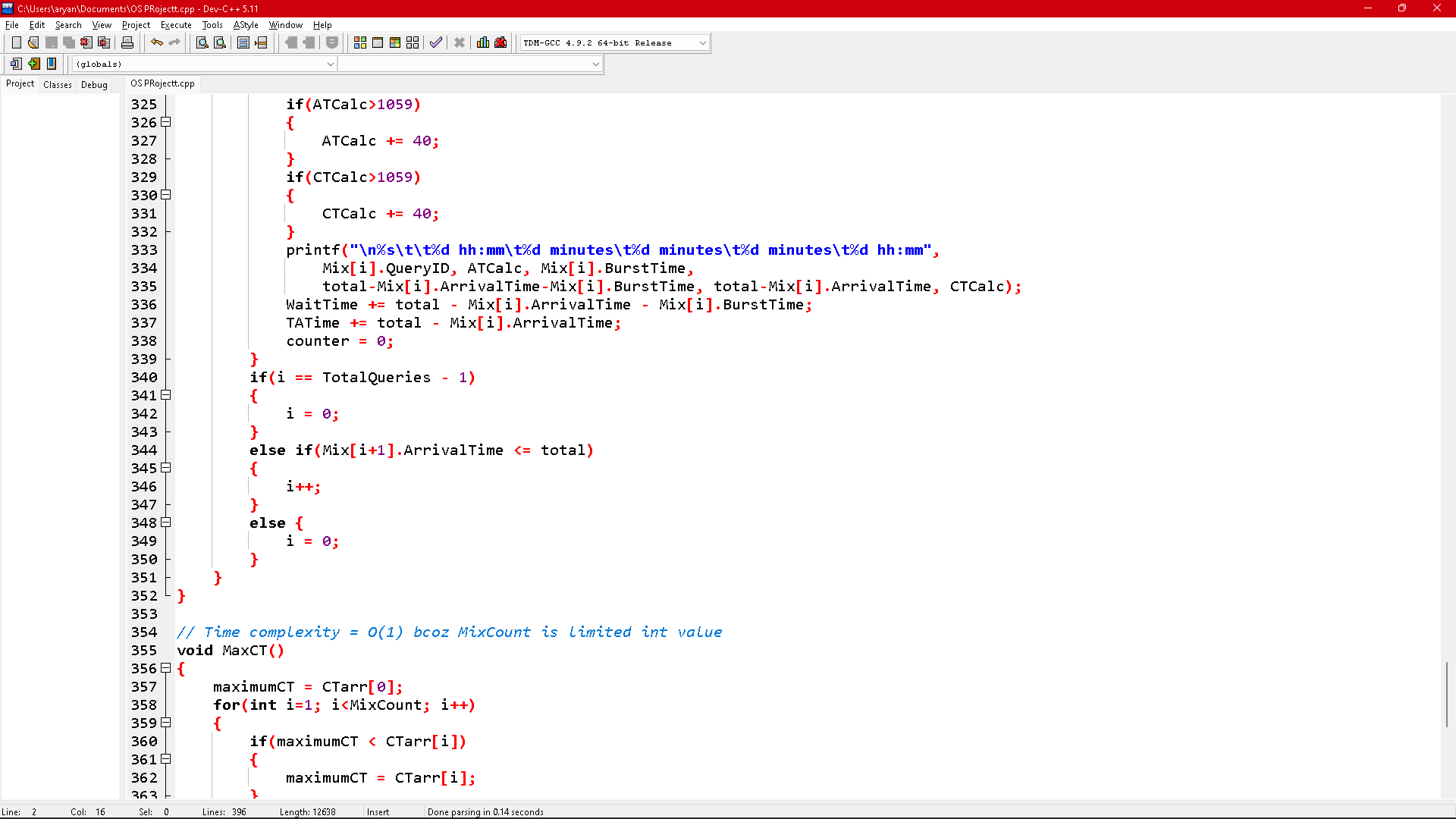
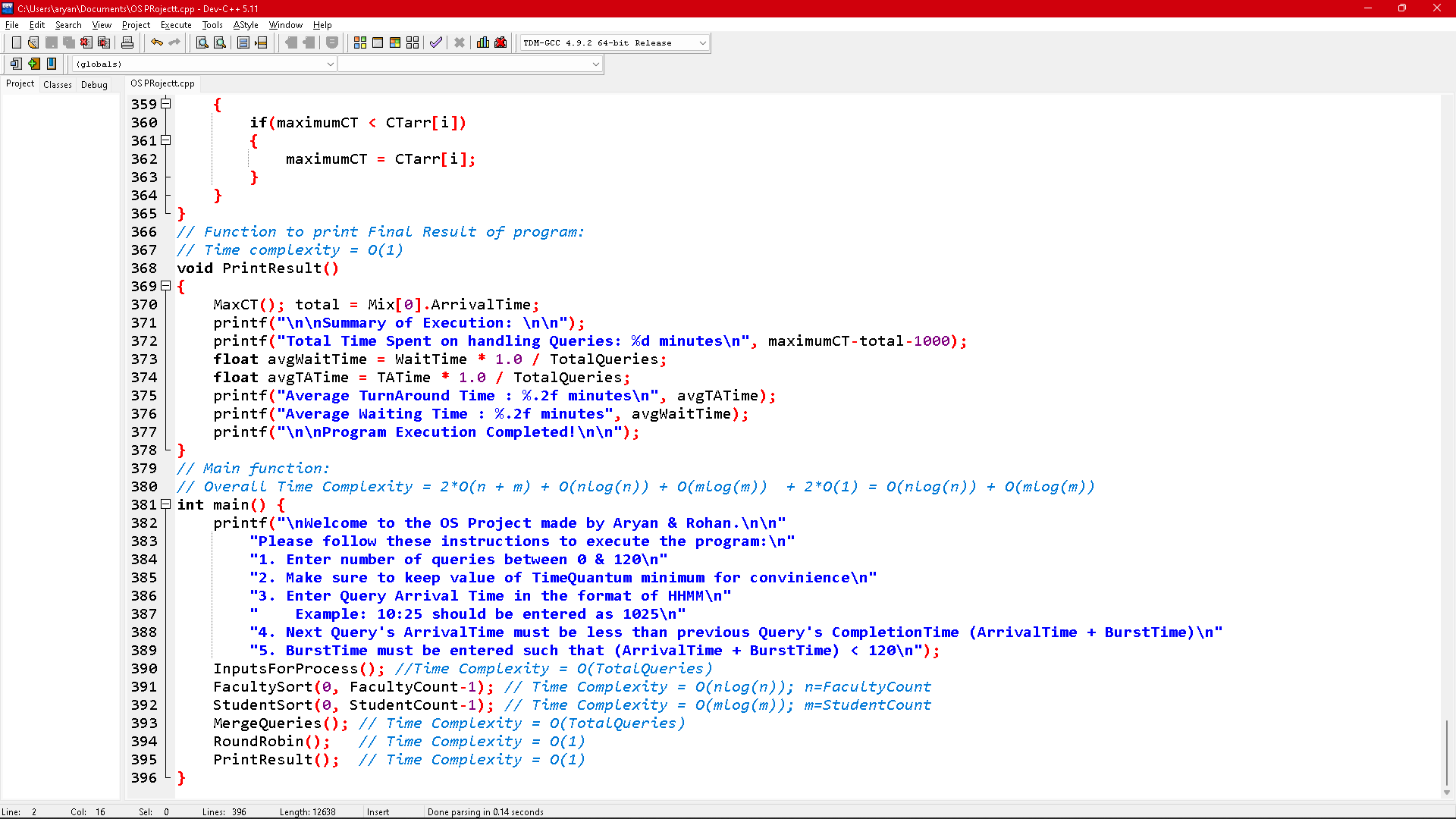
4. Next Query's Arrival-Time must be less than previous Query's Completion-Time (Arrival-Time + Burst-Time)

5. Burst-Time must be entered such that (Arrival-Time + Burst-Time) < 120

**Snap-Shots Of Output :-**



**Snap-Shots Of Code:-**



**Thank You 😊**