**Name: -** Mayank Bist

**ID: -** 11814526

**Github Address: -**

**Email Address: -** [mayankbist45@gmail.com](mailto:mayankbist45@gmail.com)

(As the lines of code is really large and wont fit in the expected size I putting the Github link to the code in C language.)

**Link to the code: -** <https://github.com/mayankbist45/Operating_System_Project/blob/master/main.c>

1. **Explain the problems in terms of operating system concept?**

The problem assigned is asking for using 2 queues to assign time to students as well as faculties in some order. And the time share should be fixed as mentioned in the problem. So, I will be applying Round Robin Algorithm in this scenario because we are dealing with fixed time quantum. But there is another catch we have two queues (one of faculties and other of students) so I decided to assign more priority to teacher’s queries and now if at certain point of time we have to decide between resolving teacher’s queries and student queries, we will proceed by resolving teacher’s query first.

1. **Write algorithm for proposed solution of assigned problem.**

The algorithm proposed above is as follows:

* Sort the faculties and students by their arrival time.
* Start with the minimum arrival time (10:00 AM) and push all the faculties and students in their respective queue.
* While all the faculties are not processed in the queue 1. Process them.
* While processing every query add all the new students and faculty arriving at updated time.
* Then process the student queue in same order.
* While processing the student or faculty queue if remaining burst time is greater than time quanta just subtract time quanta from the burst time and push it again to the back of the queue otherwise the process is completed and we have completion time.
* Now the Wait time = Completion – Arrival – Burst
* And Turnaround Time = Completion - Arrival
* Calculate Total time spent on handling queries and average query time and print the same.

1. **Calculate complexity of implemented algorithm. (Student must specify complexity of each line of code along with overall complexity)**

Since the program is quite large, I am going to specify complexity of the function which is processing the input in modified Round Robin Algorithm. Here are the steps: -

* Sort the array of faculties and students by their arrival time. Though we expect the input in sorted order in real life scenario because the arrival time will only increase with time (monotonic) but in order to make the program robust we are still sorting it with a comparator which is sorting it according to arrival time. ---- O( Flog(F) + S(log(S) )

Here F – No of Faculties

S – No of Students

* Start from minimum arrival time (10:00 AM) and push all the faculties and students with current arrival time to queue 1 and queue 2. O(F + S)
* Give the processes a certain time quantum and increase arrival time and again try to add all faculties and students with current arrival time to queue 1 and queue 2 respectively.
* The above two steps are just Round Robin algorithm in two queues with priority given to the queue 1 (faculty queue) -- O( F + S ).
* So the overall time complexity is O( Flog(F) + S(log(S) ) and the log factor is introduced in order to make the program robust.

1. **Explain all constraints given in the problem. Attach the code snippet of the implemented constraint.**

There is only one constraint given in the problem that is the admin use his system from 10AM to 12PM(14 hrs.) and I managed to validate the arrival time of the given input by sending it through a function which validated two things:-

* Arrival time is a valid time as asked.
* It falls in the range of 10:00 to 23:59

The code snippet of implemented validator is given below: -

A close up of a screen

Description automatically generated

1. **If you have implemented any additional algorithm to support the solution, explain need and usage of the same.**

The algorithm implemented needed to support two queues because we have to maintain two different queues for faculties and students. So, I added priority to the faculty queue and now we have two queue but we know which one to process first in case of collision.

Apart from that I used sorting as an additional measure to sort the inputs according to arrival time.

1. **Explain the boundary conditions of the implemented code**

The boundary conditions tested for this problem are as follows: -

* Using the arrival time as 10:00 [24 hr. format]
* The arrival time can vary up to 23:59 (maximum bounds)
* The burst time can vary and should be given in seconds There are no bounds in burst time. It should fit in integer [0 to 109]
* There can be several test cases. [1 to 105]

1. **Explain all test cases applied on the solution of assigned problem.**

Test Case 1: Here the test case is standard for the algorithm. Check second test to use only Round Robin Algorithm

1

2

54 10:10 6

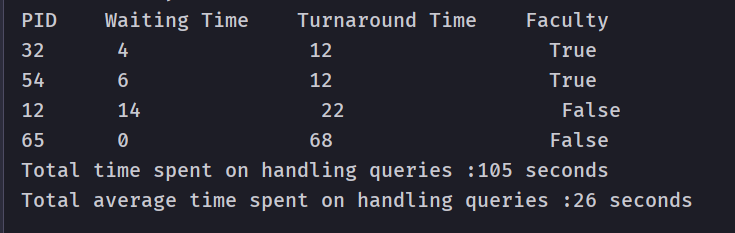
32 10:08 8

2

12 10:08 8

65 10:45 68

Output 1:



Test Case 2: This test case has only faculties that means only one queue will work hence the output should be Round Robin processed.

1

3

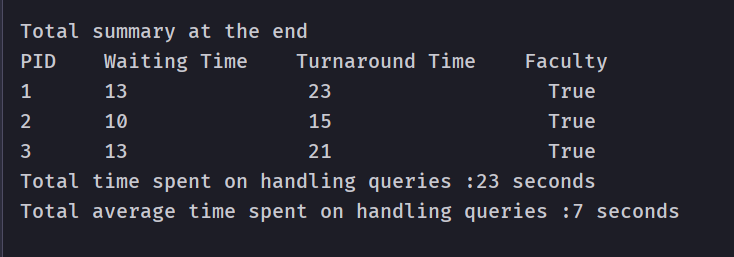
1 10:00 10

2 10:00 5

3 10:00 8

0

Output 2:



1. **Have you made minimum 5 revisions of solution on Github?**

Yes, I have made more than 5 revisions.

**Github Link: -** <https://github.com/mayankbist45/Operating_System_Project>