Student name : Anuj Pundir
Student ID: 11811882
Email-Id: pundiranuj28@gmail.com
Github link: https://github.com/AnujPundir29/Scheduling_Algo-OS

Name : Anuj Pundir

Reg. number : 11811882

Course Code: P132-H

Course title: B.tech CSE(Hons.)

Student name : Anuj Pundir
Student ID: 11811882
Email-Id: pundiranuj28@gmail.com
Github link: https://github.com/AnujPundir29/Scheduling_Algo-OS

1. Problem :

Sudesh Sharma 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.

2. Description :

The given problem is based upon solving queries of persons of different classes i.e. Faculty and Students. Thus, these queries can be compared to different processes in terms of operating system where each process has its demands and needs resources and time for its execution. And this demands of processes are handled by the CPU. In the given scenario, Mr. Sudesh Sharma, Linux expert, can be considered as a CPU, who solves the queries of either Faculty or Student by allocating proper resources to their individual demands and processing them by allocating them time accordingly. Now, Mr. Sharma, wants to provide priority for each query based upon its class, as well as, he wants to dedicate a fixed amount of time to every request. Thus in Operating System, if we divide the requests into two separate queues i.e. Faculty and Student such that the first queue contains faculty queries has higher priority and the second contains student queries which has lower priority, then we can resolve the problem, by allocating them required resources based upon their priorities as done in the scheduling algorithm in operating systems.

3. Algorithm :

This program contains the implementation of modified Round Robin algorithm and a custom Job Merger algorithm.

4. Description(Purpose of use) :

Complexity of input and output : O(n) (each).

Complexity of scheduling the process: $O(n^3)$

Student name : Anuj Pundir
Student ID: 11811882
Email-Id: pundiranuj28@gmail.com
Github link: https://github.com/AnujPundir29/Scheduling_Algo-OS

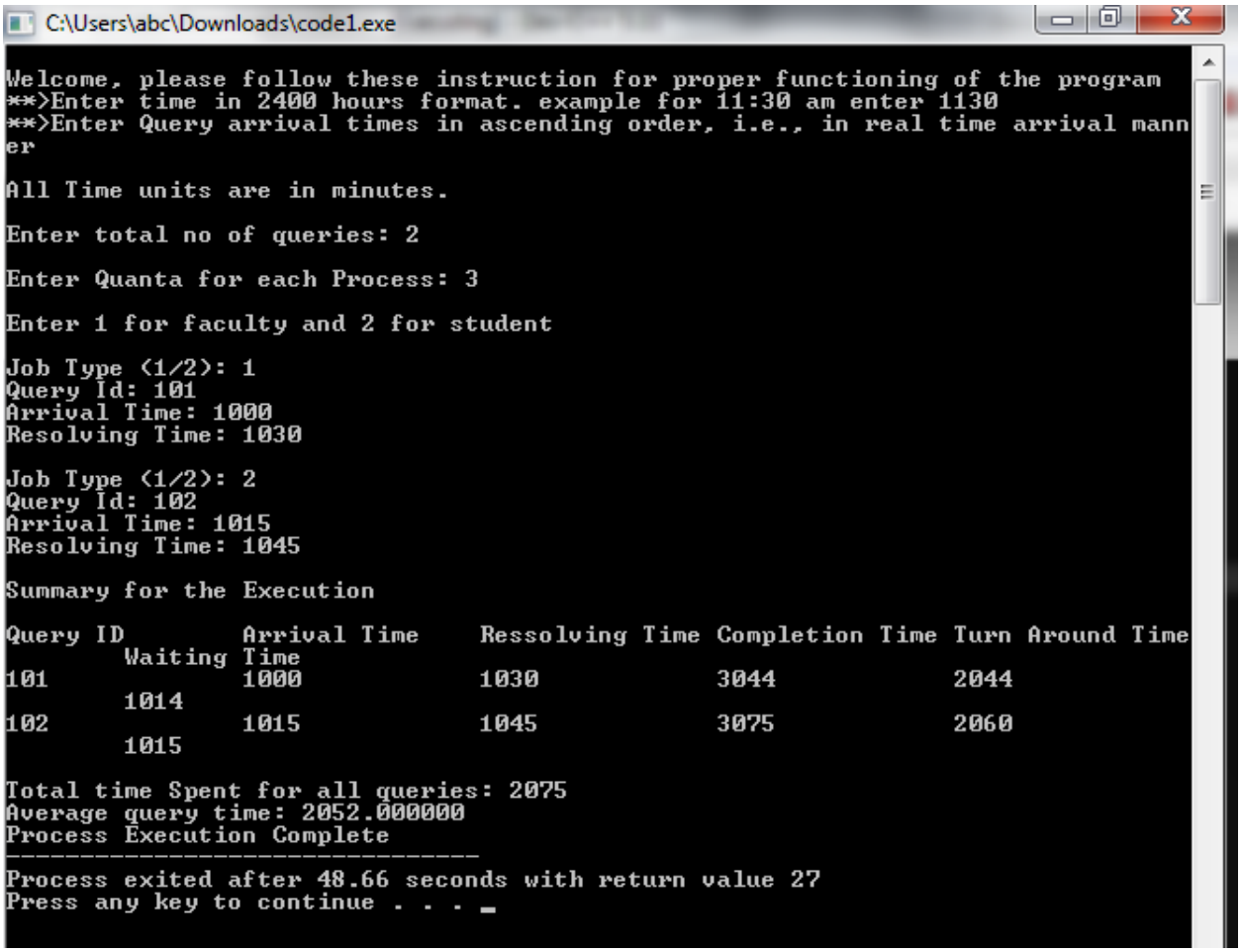
Overall Complexity : $0(n^3)$

5. Boundary Conditions :

Arrival time and Burst time should not be negative and should be written as suggested in program.

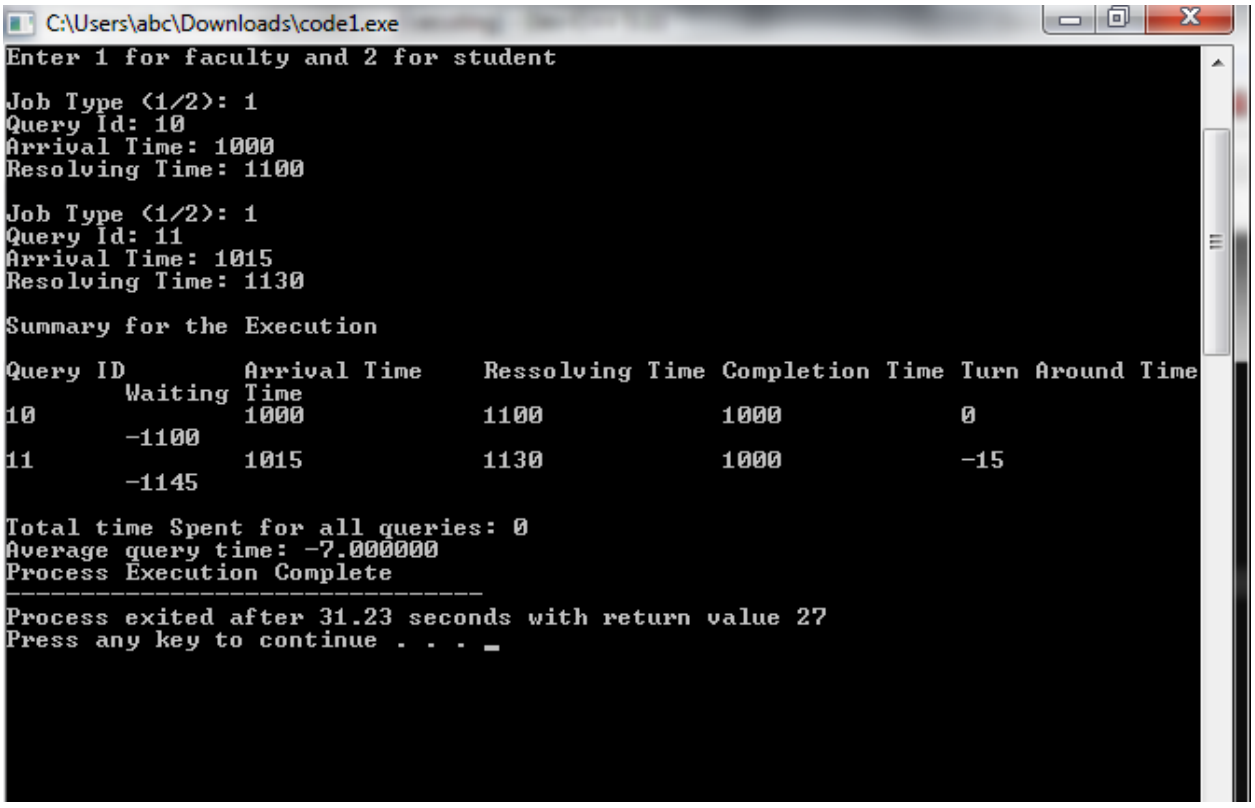
6. Test Cases :

1. This is the sample test case in which one query is of faculty and one is of student.



Student name : Anuj Pundir
Student ID: 11811882
Email-Id: pundiranuj28@gmail.com
Github link: https://github.com/AnujPundir29/Scheduling_Algo-OS

2. This is the sample test case in which both query is for faculty.



Github Link: https://github.com/AnujPundir29/Scheduling_Algo-OS