

Cse-316 assignment



**Submitted .by .:- .**

**Name .: .Rishabh .kapoor**

**Reg.id .: .11814153**

**Section .: .K18PV**

**Submitted .to .:-**

**.**

**Name .: .Ms. Samreen Fayaz**

**Reg.id .: 25435**

Table of contents…

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Topics** | **Page no.** |
|  |  |  |
| 1. | Description. | 3 |
| 2. | Algorithm. | 4 |
| 3. | Time complexity. | 5 |
| 4. | Detailed code explanation. | 6 |
| 5. | Module explanation. | 7 |
| 6. | Boundary conditions. | 8 |
| 7. | Test cases. | 9 |
| 8. | Results for plagiarism. | 10 |
| 9. | Git-hub repository link. | 11 |

Description

The idea of this project was very interesting it basically was to make a schedular that first asks the number of tasks and then their details i.e. arrival time, burst time and priority and then inserts the process in one of the three queues based on the basis priority. After that it uses *first come - first serve algorithm* on the queue 3 and uses *round robin scheduling* on highest priority queue and again *priority scheduling* on the third queue. But that’s not it also implements round robin schedular on over all 3 queues on the basis of priority.

The priority of the queues is divided with a range of magnitude 5 i.e. the process of priority between 1-5 are assigned to queue 1, 6-10 are assigned to queue 2, and 11-15 are assigned to queue 3.

Algorithm

In total four algorithms are being used in this project.

* First come, first serve.
* Priority scheduling.
* Round robin.
* Process assignment with priority.

**First come, first serve**

The main idea of this algorithm is to execute the processes by solely taking arrival time into consideration no matter what may be the priority or burst time of the processes.

**Priority scheduling**

This algorithm executes processes according to the order of priority that is it executes the processes with higher priority regardless of their arrival or burst time.

**Round robin**

This algorithm is very similar to *fcfs* the only difference is that in this one every one of the processes is made to run for a specific amount of time allotted and then it is preempted(paused) and then same is done with the next process and so on till all the processes are executed. The duration of time allotted is called quantum.

**Process assignment with priority**

This algorithm compares the priorities of every processes and add them into respective queues according to its range.

Time complexity

This code is very complex and therefore its complexity is very hard to calculate but the fact that it is a non-recursive solution we do not have to worry about the fact that it may reach at the end of function calling stack.

It has 1 fcfs i.e. O(n)

It has 2 round robins i.e. O(n2)

It has further 1 priority schedular i.e. [sorting O(n2)+O(n) = O(n2)]

So adding all of the above because they are non-nested we get

O(n)+ O(n2)+ O(n2)= O(n2) [largest is O(n2)]

Therefore the complexity of the code is O(n2).

Detailed code explanation

The code is written in C++14 but the input and output are directed through “stdio.h” that makes it partly c and partly cpp program.

On running the code, it asks for a integer input which is for the number of processes and then it moves on to asking the details of the processes one by one, it asks arrival time, burst time and priority of the processes. Then the processes are added to their respective queues according to their priority ranges [1-15] then it schedules the processes in sequence as required.

As it is very conspicuous from the structure of the main function [driver method] that firstly Rrobin1, then Robin, after that fcfs and prior at the last are called now the functions of each method are described further.

* Rrobin1 is responsible for scheduling all three queues according to round robin with quantum 10.
* Robin is responsible to schedule all the processes of queue 1.
* Fcfs is responsible to manage and schedule the processes of queue 2. And,
* Prior is responsible to schedule the processes of queue 3.

Module explanation

There are 6 modules that are used here primarily.

* Main function [Driver code]
* Rrobin1
* Robin
* Fcfs
* Prior
* Queue assigner (in main)

The working of every module is conspicuous just by its very glance though it is a bit lengthy.

Main function is executed before anything else. Which further calls other methods Rrobin1 is supposed to handle scheduling of queues(1,2, and 3).

Robin schedules processes of queue 1.

fcfs schedules processes of queue 2.

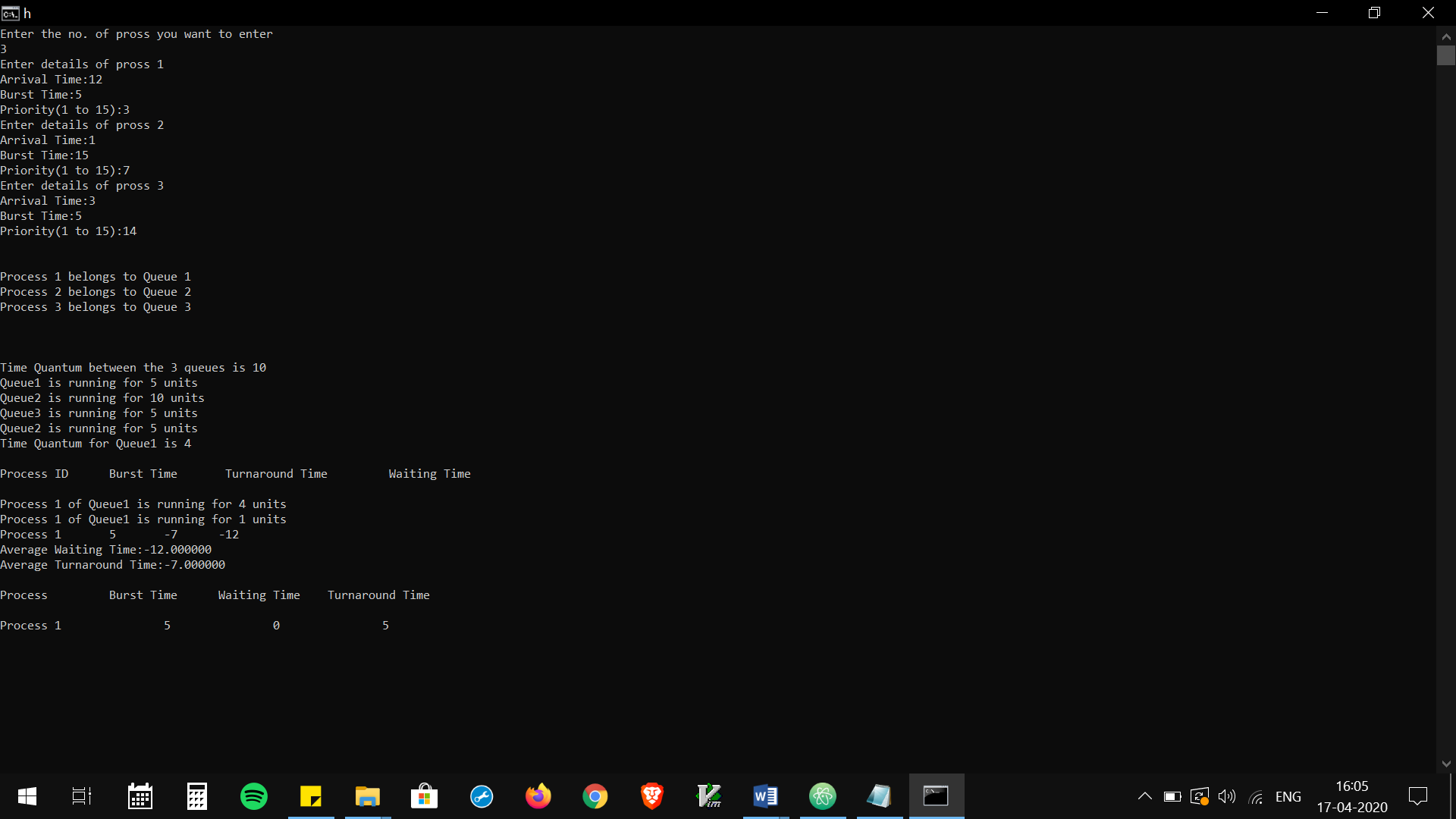
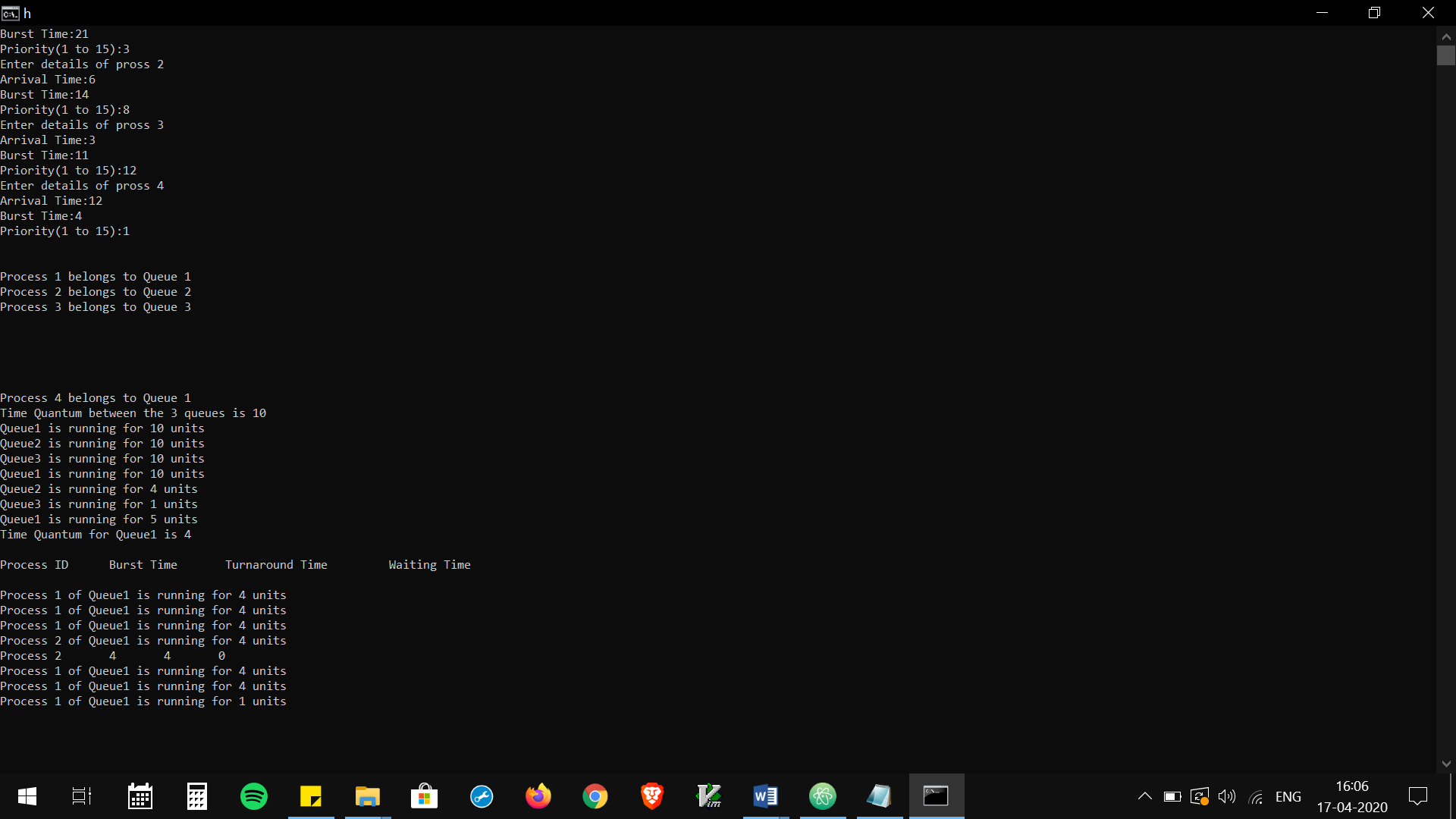
Prior schedules processes of queue 3.

Boundary conditions.

The only boundary of this code is that it can’t deal with floating input values or any results that needs to be type casted.

In which case it throws “FloatingPointException” but this happens once in a blue moon.

Test cases

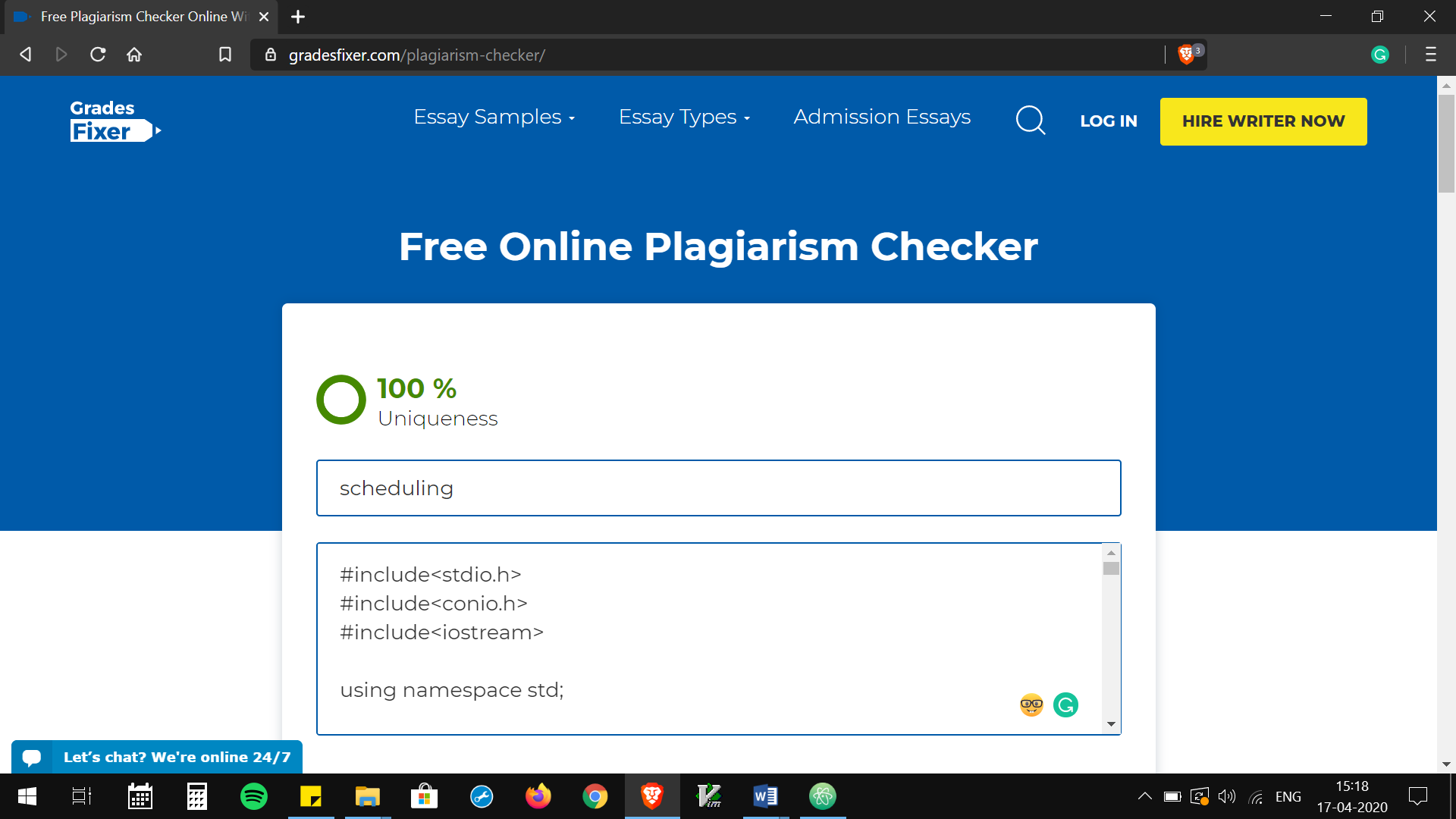


PASSED

PASSED

Results for plagiarism

The code was 99% genuine and the other one percent was the help taken from the internet.



From the test results evident that my solution to the problem is 100% unique.

Git- hub repository link

Link :- <https://github.com/rishabhkapoor101/osproject>