**Student Name**:

**Student ID**:

**Email Address**:

**GitHub Link**:

**Problem explanation:**

The problem assigned to me was based on Process Scheduling. We will solve this problem using Round Robin Scheduling. In this problem we deal with the queries of student and faculty in online system where there can be multiple requests at any time, we will dedicate a fixed amount of time to every process by using Quantum time. We will use two separate queues for student and faculty. We will assign number of process, burst time and arrival time as input. We will calculate the total time he spent on handling queries and average query time.

**Code**:

#include<stdio.h>

#include<stdbool.h>

**->Below is the waiting time function in which we are calculating the waiting time of each process.**

void WaitingTime(int process[], int n,int burst[], int wt[], int quan)

{ int i,t,rem[n]; t=0; for (i = 1 ; i <= n ; i++)

{ rem[i] = burst[i]; }

while(1)

{ bool flag= true; for(i = 1; i <= n; i++)

{ if (rem[i]>0)

{ flag=false;

if (rem[i]>quan)

{ t += quan; rem[i] -= quan;

} else

{

wt[i] = t - process[i];

t = t + rem[i]; rem[i] = 0;

}

}

}

if (flag == true) break;

}

}

**->Below is the function of TurnAroundTime in which we will calculate turnaround time of each process** void TurnAroundTime(int process[], int n,int burst[], int wt[], int temp[])

{

int i; for (i = 1; i <=n ; i++) temp[i] = burst[i] + wt[i];

}

**->Below is the AverageTime function, in which we will call WaitingTime(),**

**TurnAroundTime() functions, and displaycode of the burst time, processes, arrival time, waiting time, turn around time. We will also calculate the total time spent to handle student and faculty queries.**

void AvgTime(int process[], int n, int burst[],int q,char name)

{ int i,wt[n], temp[n], total\_wait = 0, total\_time= 0; int avg\_wait , avg\_turnaround;

WaitingTime(process, n,burst, wt, q);

TurnAroundTime(process, n, burst, wt, temp);

printf("Processes : BrustTime : ArivalTime: : TurnAroundTime : \n"); int count=0;

for (i=1; i<=n; i++)

{ total\_wait = total\_wait + wt[i]; total\_time = total\_time + temp[i];

count+=burst[i]; if(count>60)

{

break;

}

printf("%d \t\t %d \t\t %d \t\t %d \t\t\t%d\n",i,burst[i],process[i],wt[i],temp[i]);

}

avg\_wait=total\_wait/(i-1); avg\_turnaround=total\_time/(i-1); if(name=='s')

{

printf("\n Time he spend on handling of students query is : %d",avg\_turnaround); printf("\n Average query time is : %d",avg\_wait);

} else

{

printf("\n Time he spend on handling of faculty query is : %d",avg\_turnaround); printf("\n Average query time is : %d",avg\_wait);

}

}

**->Below is the main function, in which we will initialize the QuantumTime, ask the user to enter no.of processes, arriaval time, burst time. We call AvgTime function in this. We call all the functions in it and it prints the required output on desktop.**

int main() { int y;

int quantum =5; printf("#Enter no.of processes# : "); scanf("%d",&y); int p[y],p1[y]; int burst\_time[y],burst\_time1[y]; printf("\*\*\*\*\*\*\*\*Enter STUDENTS ArrivalTime:\*\*\*\*\*\*\*\* \n\n");

int i;

for(i=1;i<=y;i++)

{

printf("Enter student arival time: %d\t",i); scanf("%d",&p[i]);

printf("\n");

}

printf("\*\*\*\*\*\*\*Enter STUDENT BurstTime:\*\*\*\*\*\*\*\*\*\* \n\n"); for(i=1;i<=y;i++)

{ printf("Enter student burst time : %d\t",i);

scanf("%d",&burst\_time[i]); printf("\n");

}

printf("\*\*\*\*\*\*\*\*Enter Faculty ArrivalTime :\*\*\*\*\*\*\*\* \n\n"); for(i=1;i<=y;i++)

{ printf("Enter faculty arrival time : %d\t",i);

scanf("%d",&p1[i]);

printf("\n");

} int n = sizeof p / sizeof p[0]; printf("\*\*\*\*\*\*\*\*Enter FACULTY BurstTime :\*\*\*\*\*\*\*\* \n\n"); for(i=1;i<=y;i++)

{ printf("Enter faculty burst time : %d\t",i);

scanf("%d",&burst\_time1[i]);

printf("\n");

}

printf("sudesh sharma spends 1st one hour for handling student queries: \n\n"); AvgTime(p, n, burst\_time, quantum,'s'); n = sizeof p1 / sizeof p1[0]; printf("\n\n"); printf("sudesh sharma spends 2nd one hour for handling Faculty queries: \n\n"); AvgTime(p1, n ,burst\_time1,quantum,'f'); printf("\n\n\n");

printf("\nWe Used TIME QUANTUM = 5:\n\n");

return 0;

}

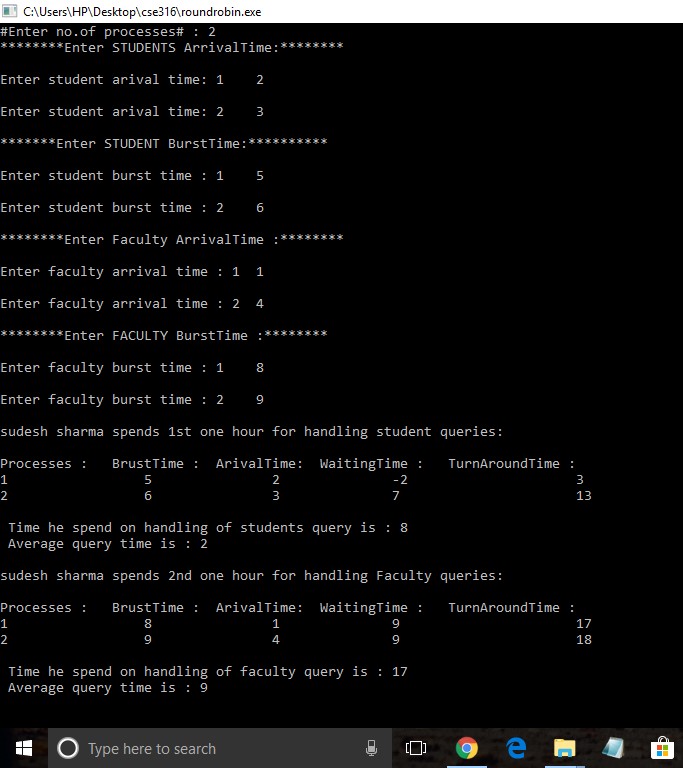
,

.

.\*

**Test case: 1**

**In this we assign 2 processes for student and faculty. We have calculated average waiting time , turnaround time. Total time Sudesh sharma has spent on handling faculty and student queries is 17 and 8.**



**Test case: 2**

**In this we assign 3 processes for student and faculty. We have calculated average waiting time, turnaround time. Total time Sudesh sharma has spent on handling faculty and student queries is 24 and 12.**

