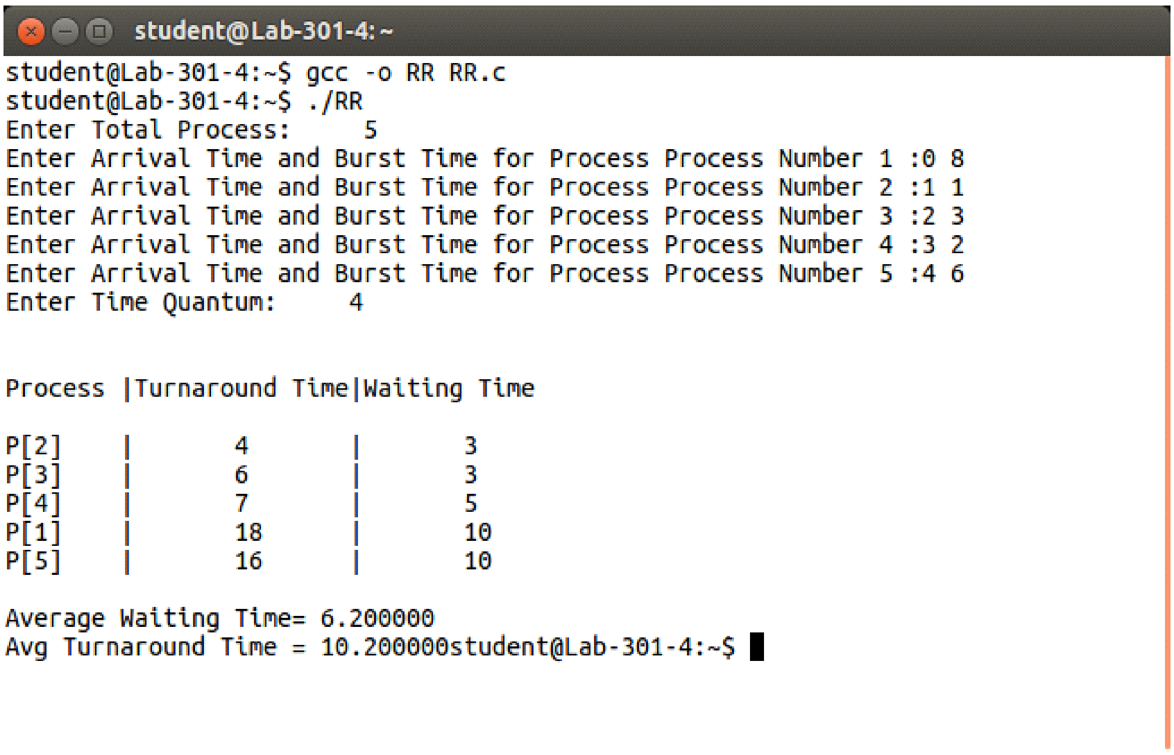
OS EXPERIMENT-7

PROGRAM : ROUND ROBIN SCHEDULING ALGORITHM

```
#include<stdio.h>
#include<conio.h>
int main()
{
int count,j,n,time,remain,flag=0,time_quantum;
int wait_time=0,turnaround_time=0,at[10],bt[10],rt[10];
printf("Enter Total Process:\t ");
scanf("%d",&n);
remain=n;
for(count=0;count<n;count++)
{
printf("Enter Arrival Time and Burst Time for Process Process Number %d :",count+1);
scanf("%d",&at[count]);
scanf("%d",&bt[count]);
rt[count]=bt[count];
}
printf("Enter Time Quantum:\t");
scanf("%d",&time_quantum);
printf("\n\nProcess\t|Turnaround Time|Waiting Time\n\n");
for(time=0,count=0;remain!=0;)
{
if(rt[count]<=time_quantum && rt[count]>0)
{
time+=rt[count];
rt[count]=0;
flag=1;
}
else if(rt[count]>0)
{
rt[count]-=time_quantum;
time+=time_quantum;
}
if(rt[count]==0 && flag==1)
{
remain--;
printf("P[%d]\t|\t%d\t|\t%d\n",count+1,time-at[count],time-at[count]-bt[count]);
wait_time+=time-at[count]-bt[count];
turnaround_time+=time-at[count];
flag=0;
}
if(count==n-1)
count=0;
else if(at[count+1]<=time)
count++;
else
count=0;
}
printf("\nAverage Waiting Time= %f\n",wait_time*1.0/n);
printf("Avg Turnaround Time = %f",turnaround_time*1.0/n);
return 0;
}
```

<u>OUTPUT:</u>



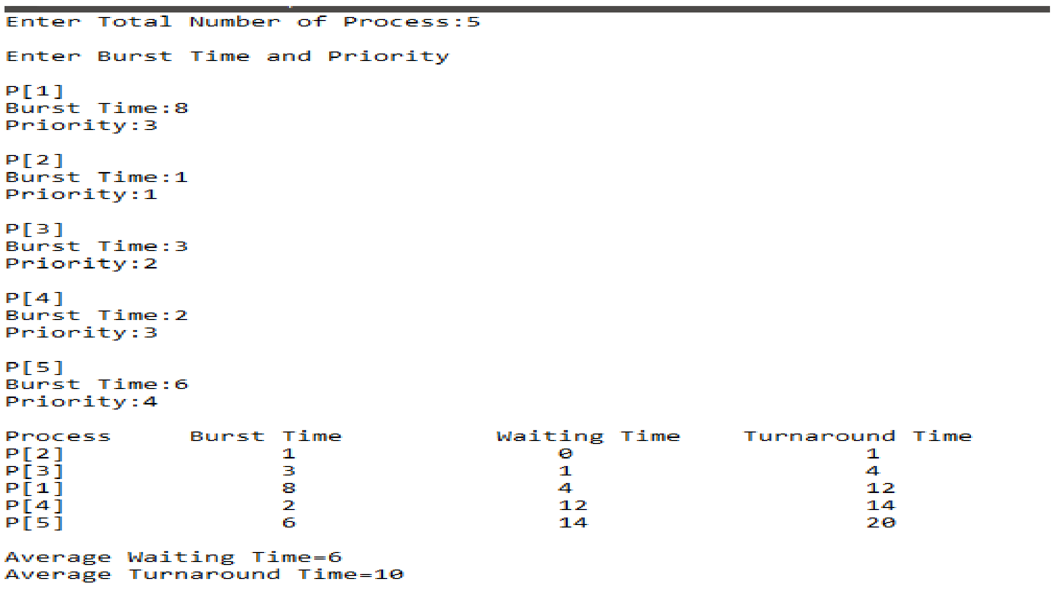
```
student@Lab-301-4:~$ gcc -o RR RR.c
student@Lab-301-4:~$ ./RR
Enter Total Process:     5
Enter Arrival Time and Burst Time for Process Process Number 1 :0 8
Enter Arrival Time and Burst Time for Process Process Number 2 :1 1
Enter Arrival Time and Burst Time for Process Process Number 3 :2 3
Enter Arrival Time and Burst Time for Process Process Number 4 :3 2
Enter Arrival Time and Burst Time for Process Process Number 5 :4 6
Enter Time Quantum:     4


Process |Turnaround Time|Waiting Time

P[2]    |       4       |       3
P[3]    |       6       |       3
P[4]    |       7       |       5
P[1]    |       18      |       10
P[5]    |       16      |       10

Average Waiting Time= 6.200000
Avg Turnaround Time = 10.200000student@Lab-301-4:~$
```

<u>PRAGRAM FOR PRIORITY(NON PREMPTIVE) SCHEDULING ALDORITHM</u>

```
#include<stdio.h>
int main()
{
   int bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,total=0,pos,temp,avg_wt,avg_tat;
   printf("Enter Total Number of Process:");
   scanf("%d",&n);

   printf("\nEnter Burst Time and Priority\n");
   for(i=0;i<n;i++)
   {
      printf("\nP[%d]\n",i+1);
      printf("Burst Time:");
      scanf("%d",&bt[i]);
      printf("Priority:");
      scanf("%d",&pr[i]);
      p[i]=i+1;
   }
```

```
    for(i=0;i<n;i++)
    {
        pos=i;
        for(j=i+1;j<n;j++)
        {
            if(pr[j]<pr[pos])
                pos=j;
        }
        temp=pr[i];
        pr[i]=pr[pos];
        pr[pos]=temp;

        temp=bt[i];
        bt[i]=bt[pos];
        bt[pos]=temp;

        temp=p[i];
        p[i]=p[pos];
        p[pos]=temp;
    }
    wt[0]=0;
    for(i=1;i<n;i++)
    {
        wt[i]=0;
        for(j=0;j<i;j++)
            wt[i]+=bt[j];

        total+=wt[i];
    }

    avg_wt=total/n;
    total=0;

    printf("\nProcess\t    Burst Time    \tWaiting Time\tTurnaround Time");
    for(i=0;i<n;i++)
    {
        tat[i]=bt[i]+wt[i];
        total+=tat[i];
        printf("\nP[%d]\t\t  %d\t\t    %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);
    }
    avg_tat=total/n;
    printf("\n\nAverage Waiting Time=%d",avg_wt);
    printf("\nAverage Turnaround Time=%d\n",avg_tat);
    return 0;
}
```

OUTPUT:-

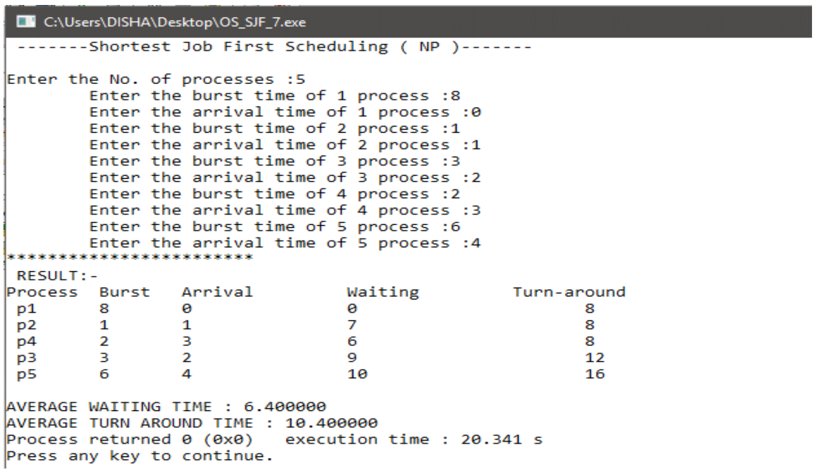


```
Enter Total Number of Process:5

Enter Burst Time and Priority

P[1]
Burst Time:8
Priority:3

P[2]
Burst Time:1
Priority:1

P[3]
Burst Time:3
Priority:2

P[4]
Burst Time:2
Priority:3

P[5]
Burst Time:6
Priority:4

Process      Burst Time           Waiting Time      Turnaround Time
P[2]               1                   0                   1
P[3]               3                   1                   4
P[1]               8                   4                   12
P[4]               2                   12                  14
P[5]               6                   14                  20

Average Waiting Time=6
Average Turnaround Time=10
```

<u>PRAGRAM FOR SJF(PREMPTIVE) SCHEDULING ALDORITHM</u>

```
#include <stdio.h>
int main()
{
    int arrival_time[10], burst_time[10], temp[10];
    int i, smallest, count = 0, time, limit;
    double wait_time = 0, turnaround_time = 0, end;
    float average_waiting_time, average_turnaround_time;
    printf("\nEnter the Total Number of Processes:\t");
    scanf("%d", &limit);
    printf("\nEnter Details of %d Processes\n", limit);
    for(i = 0; i < limit; i++)
    {
        printf("\nEnter Arrival Time:\t");
        scanf("%d", &arrival_time[i]);
        printf("Enter Burst Time:\t");
        scanf("%d", &burst_time[i]);
        temp[i] = burst_time[i];
    }
    burst_time[9] = 9999;
    for(time = 0; count != limit; time++)
    {
        smallest = 9;
        for(i = 0; i < limit; i++)
        {
            if(arrival_time[i] <= time && burst_time[i] < burst_time[smallest] && burst_time[i] > 0)
            {
                smallest = i;
            }
        }
        burst_time[smallest]--;
        if(burst_time[smallest] == 0)
        {
            count++;
            end = time + 1;
            wait_time = wait_time + end - arrival_time[smallest] - temp[smallest];
            turnaround_time = turnaround_time + end - arrival_time[smallest];
        }
    }
    average_waiting_time = wait_time / limit;
    average_turnaround_time = turnaround_time / limit;
    printf("\n\nAverage Waiting Time:\t%lf\n", average_waiting_time);
    printf("Average Turnaround Time:\t%lf\n", average_turnaround_time);
    return 0;
}
```

OUTPUT:

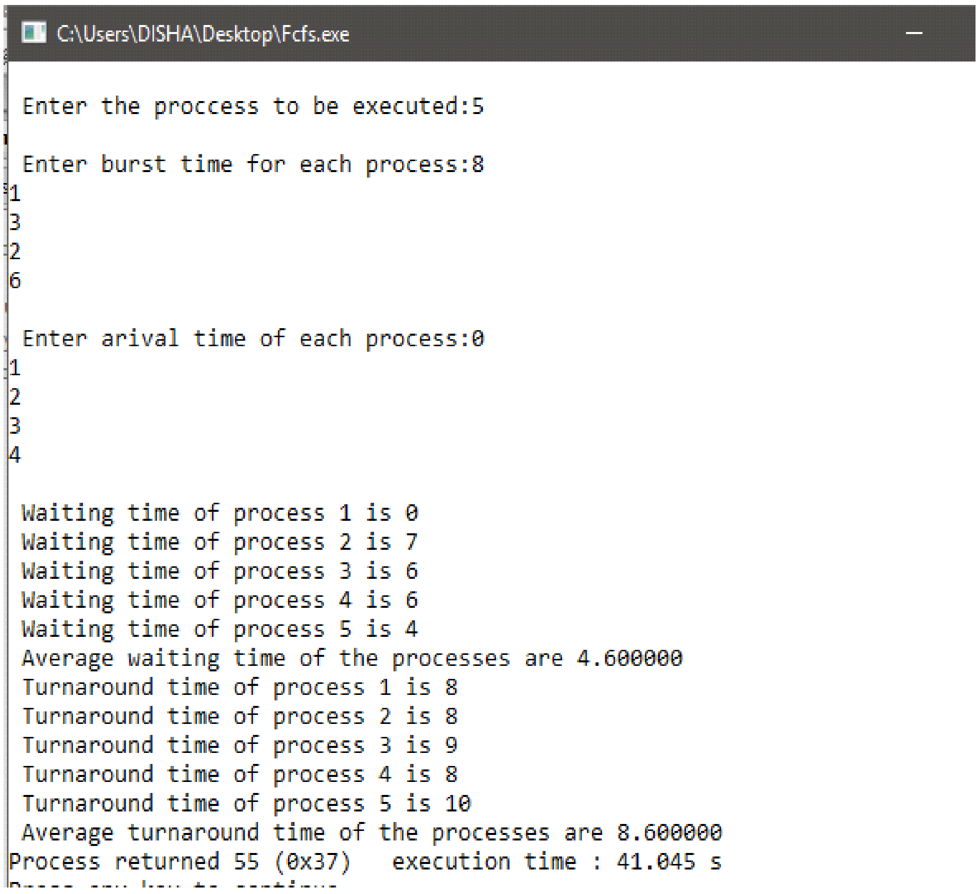


```
C:\Users\DISHA\Desktop\OS_SJF_7.exe
 -------Shortest Job First Scheduling ( NP )-------

Enter the No. of processes :5
        Enter the burst time of 1 process :8
        Enter the arrival time of 1 process :0
        Enter the burst time of 2 process :1
        Enter the arrival time of 2 process :1
        Enter the burst time of 3 process :3
        Enter the arrival time of 3 process :2
        Enter the burst time of 4 process :2
        Enter the arrival time of 4 process :3
        Enter the burst time of 5 process :6
        Enter the arrival time of 5 process :4
************************
 RESULT:-
Process  Burst   Arrival         Waiting         Turn-around
 p1      8       0               0                       8
 p2      1       1               7                       8
 p4      2       3               6                       8
 p3      3       2               9                       12
 p5      6       4               10                      16

AVERAGE WAITING TIME : 6.400000
AVERAGE TURN AROUND TIME : 10.400000
Process returned 0 (0x0)   execution time : 20.341 s
Press any key to continue.
```

PROGRAM FOR FCFS SCHEDULING ALDORITHM

```
#include<stdio.h>

#define MAX 20

void main()

{

        int i,n,bt[MAX],wt[MAX],tat[MAX],at[MAX],twt=0,ttat=0;

        float awt,atat;

        printf("\n Enter the proccess to be executed:");

        scanf("%d",&n);

        printf("\n Enter burst time for each process:");

        for(i=0;i<n;i++)
        {
                scanf("%d",&bt[i]);
                wt[i]=0;
        }
        printf("\n Enter arival time of each process:");
        for(i=0;i<n;i++)
                scanf("%d",&at[i]);
        printf("\n Waiting time of process 1 is 0");
        for(i=1;i<n;i++)
        {
                wt[i]=wt[i-1]+bt[i-1]-at[i];
                printf("\n Waiting time of process %d is %d ",i+1,wt[i]);
        }
        for(i=0;i<n;i++)
                twt=twt+wt[i];
        awt=(twt*1.0)/n;
        printf("\n Average waiting time of the processes are %f",awt);
        tat[0]=wt[0]+bt[0];
        printf("\n Turnaround time of process %d is %d",1,tat[0]);
        for(i=1;i<n;i++)
        {
                tat[i]=wt[i]+bt[i];
                printf("\n Turnaround time of process %d is %d",i+1,tat[i]);
        }
        for(i=0;i<n;i++)
                ttat=ttat+tat[i];
        atat=(ttat*1.0)/n;
        printf("\n Average turnaround time of the processes are %f",atat);
}
```

OUTPUT:


C:\Users\DISHA\Desktop\Fcfs.exe

```

 Enter the proccess to be executed:5

 Enter burst time for each process:8
1
3
2
6

 Enter arival time of each process:0
1
2
3
4

 Waiting time of process 1 is 0
 Waiting time of process 2 is 7
 Waiting time of process 3 is 6
 Waiting time of process 4 is 6
 Waiting time of process 5 is 4
 Average waiting time of the processes are 4.600000
 Turnaround time of process 1 is 8
 Turnaround time of process 2 is 8
 Turnaround time of process 3 is 9
 Turnaround time of process 4 is 8
 Turnaround time of process 5 is 10
 Average turnaround time of the processes are 8.600000
Process returned 55 (0x37)   execution time : 41.045 s
```