PRACTICAL-7

AIM: WRITE A PROGRAM TO IMPLEMENT ROUND ROBIN SCHEDULING ALGORITHM.

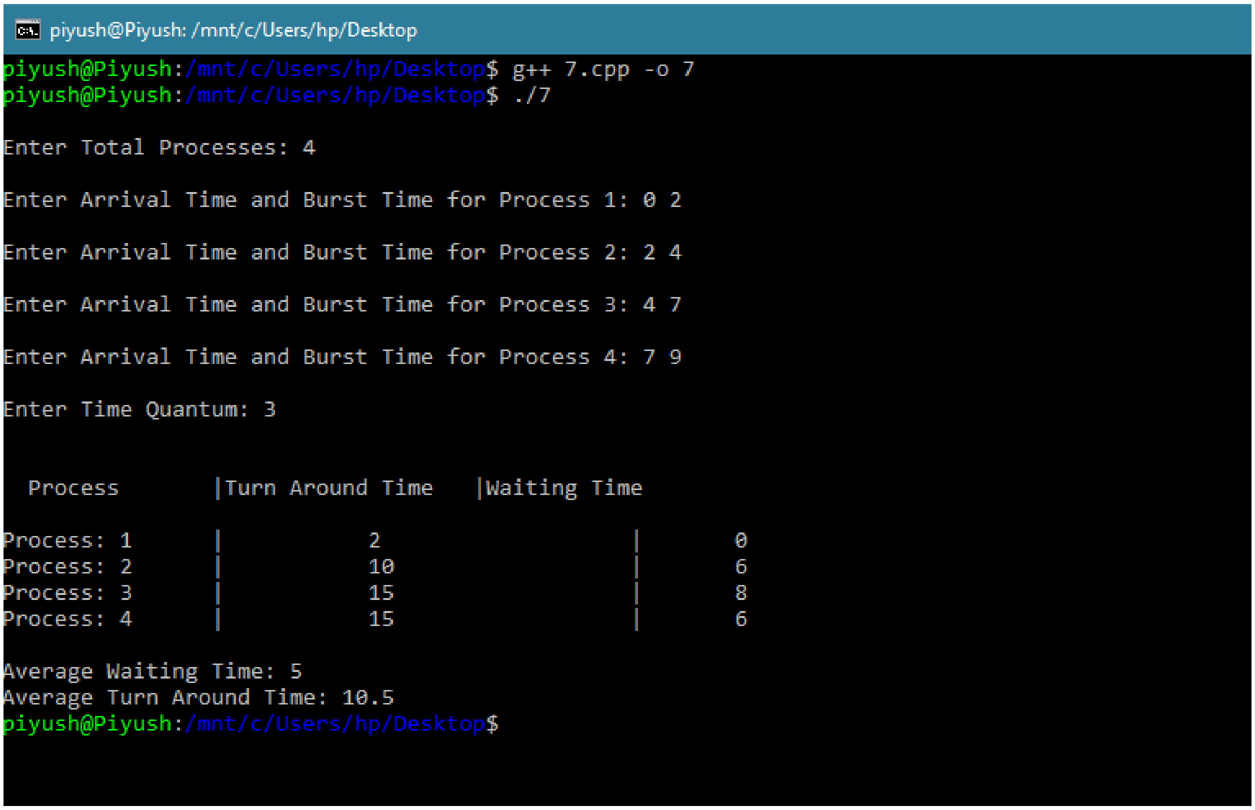
SOL:

```
#include<iostream>

#include<stdio.h>

using namespace std;

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];

cout<<"\nEnter Total Processes: ";

cin>>n;

remain=n;

for(count=0;count<n;count++)

{

cout<<"\nEnter Arrival Time and Burst Time for Process "<<count+1<<": ";

cin>>at[count];

cin>>bt[count];

rt[count]=bt[count];

}

cout<<"\nEnter Time Quantum: ";

cin>>time_quantum;

cout<<"\n\n  Process \t|Turn Around 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--;
cout<<"Process: "<<count+1<<" \t|\t    "<<time-at[count]<<"\t\t\t|\t"<<time-at[count]-bt[count]<<"\n";
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;

}

cout<<"\nAverage Waiting Time: "<<wait_time*1.0/n;

cout<<"\nAverage Turn Around Time: "<<turnaround_time*1.0/n<<endl;

return 0;

}
```

OUTPUT:



```
piyush@Piyush:/mnt/c/Users/hp/Desktop$ g++ 7.cpp -o 7
piyush@Piyush:/mnt/c/Users/hp/Desktop$ ./7

Enter Total Processes: 4

Enter Arrival Time and Burst Time for Process 1: 0 2

Enter Arrival Time and Burst Time for Process 2: 2 4

Enter Arrival Time and Burst Time for Process 3: 4 7

Enter Arrival Time and Burst Time for Process 4: 7 9

Enter Time Quantum: 3


  Process        |Turn Around Time    |Waiting Time

Process: 1       |           2                   |       0
Process: 2       |           10                  |       6
Process: 3       |           15                  |       8
Process: 4       |           15                  |       6

Average Waiting Time: 5
Average Turn Around Time: 10.5
piyush@Piyush:/mnt/c/Users/hp/Desktop$
```