**ASSIGNMENT-8**

**Q1. Write a program to implement Round Robin scheduling algorithms for the following two cases:**

**P# AT BT**

**P1 0 3**

**P2 1 5**

**P3 3 2**

**Implement the Shortest remaining time algorithm and perform a dry run of the program for the above values manually and verify your output.**

#include<stdio.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:");

scanf("%d",&n);

remain=n;

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

{

printf("Enter Arrival Time and Burst Time for Process Number %d:",count+1);

scanf("%d",&at[count]);

scanf("%d",&bt[count]);

rt[count]=bt[count];

}

printf("Enter Time Quantum:");

scanf("%d",&time\_quantum);

printf("\n\nProcess\t|\tAT\t|\tBT\t|\tTAT\t|\tWT\t\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\t|\t%d\t|\t%d\n",count+1,at[count],bt[count],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;

}

**OUTPUT-**









