7) Write a program to implement Round Robin scheduling algorithm.

```
#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:\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\t\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;
}
```

OUTPUT :

```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
g++ -o main round.cpp &&./main
Enter Total Process:     4
Enter Arrival Time and Burst Time for Process Process Number 1
:0 4
Enter Arrival Time and Burst Time for Process Process Number 2
:2 5
Enter Arrival Time and Burst Time for Process Process Number 3
:5 7
Enter Arrival Time and Burst Time for Process Process Number 4
:6 6
Enter Time Quantum: 2


Process |Turnaround Time|Waiting Time

P[1]     |   6   |               2
P[2]     |   11   |               6
P[4]     |   15   |               9
P[3]     |   17   |               10

Average Waiting Time= 6.750000
Avg Turnaround Time = 12.250000
```