**Write a program simulate the working of the Round Robin CPU scheduling algorithms:**

1. **Round Robin :**

**Assume all the processes arrive at the same time.**

**ALGORITHM:**

1. **Start**
2. **Declarethearraysize**
3. **Readthenumberof processes to be inserted**
4. **Readthe burst timesofthe processes**
5. **Read the Time Quantum**
6. **If the burst time of a process is greater than time Quantum then subtract time quantum from the burst time**

**Else**

**Assign the burst time to time quantum.**

1. **calculatetheaveragewaitingtimeand turnaround time of theprocesses.**
2. **Displaythe values**
3. **Stop**

**#include<stdio.h>**

**void main()**

**{**

**int st[10],bt[10],wt[10],tat[10],n,tq;**

**int i,count=0,swt=0,stat=0,temp,sq=0;**

**float awt=0.0,atat=0.0;**

**printf("Enternumberofprocesses:");**

**scanf("%d",&n);**

**printf("Enterbursttimeforsequences:");**

**for(i=0;i<n;i++)**

**{**

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

**st[i]=bt[i];**

**}**

**printf("Entertimequantum:");**

**scanf("%d",&tq);**

**while(1)**

**{**

**for(i=0,count=0;i<n;i++)**

**{**

**temp=tq;**

**if(st[i]==0)**

**{**

**count++; continue;**

**}**

**if(st[i]>tq) st[i]=st[i]-tq; else if(st[i]>=0)**

**{**

**temp=st[i]; st[i]=0;**

**}**

**sq=sq+temp;**

**tat[i]=sq;**

**}**

**if(n==count) break;**

**}**

**for(i=0;i<n;i++)**

**{**

**wt[i]=tat[i]-bt[i];**

**swt=swt+wt[i];**

**stat=stat+tat[i];**

**}**

**awt=(float)swt/n;**

**atat=(float)stat/n;**

**printf("Process\_noBursttimeWaittimeTurnaroundtime");**

**for(i=0;i<n;i++)**

**printf("\n%d\t %d\t%d\t%d",i+1,bt[i],wt[i],tat[i]);**

**printf("\nAvgwaittimeis%fAvgturnaroundtimeis %f",awt,atat);**

**}**