# ROUND ROBIN AND SRTF

```
#include<stdio.h>
#include<stdlib.h>
void roundrobin(int,int,int[],int[]);
void srtf();
main(){
int n,tq,choice;
int bt[10],st[10],i,j,k;
for(;;)
{
printf("enter the choice:\n");
printf("1.RoundRobin\n2.srtf\n3.exit\n");
scanf("%d",&choice);
switch(choice)
{
case 1: printf("RoundRobin scheduling algorithm");
        printf("Enter the number of process\n");
        scanf("%d",&n);
        printf("Enter the burst time of processes\n");
        for(i=0;i<n;i++)
        {
        scanf("%d",&bt[i]);
        st[i]=bt[i];
        }
        printf("Enter the time quantum\n");
        scanf("%d",&tq);
        roundrobin(n,tq,st,bt);
        break;
case 2: printf("\n\n.......SRTF.......\n\n");
        srtf();
```

```
            break;
case 3: exit(0);
            break;
}
}
}
void roundrobin(int n,int tq,int rt[],int bt[])
{
int time=0;
int tat[10],wt[10],i,count=0,swt=0,temp1,sq=0,j,k,stat=0;
float awt=0.0,atat=0.0;
while(1)
{
for(i=0,count=0;i<n;i++)
{
temp1=tq;
if(rt[i]==0)
{
count++;
continue;
}
if(rt[i]>tq)
rt[i]=rt[i]-tq;
else if(rt[i]>=0)
{
temp1=rt[i];
rt[i]=0;
}
sq=sq+temp1;
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 no burst time waiting time turnaround time\n");
for(i=0;i<n;i++)
printf("%d\t\t %d\t\t %d\t\t %d\t\t\n",i+1,bt[i],wt[i],tat[i]);
printf("average waiting time is %f\n average turnaround time is %f\n",awt,atat);
}
void srtf()
{
int n,j=0,at[10],bt[10],rt[10],finished=0,smallest,time=0,i,endtime,swt=0,stat=0;
printf("enter no of processes:\n");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("enter the arrival time dor p[%d]:",i+1);
scanf("%d",&at[i]);
printf("enter burst time for p[%d]:",i+1);
scanf("%d",&bt[i]);
rt[i]=bt[i];
}
rt[100]=999;
printf("process\t|waiting time\t|turnaround time\n");
```

```
for(time=0;finished!=n;time++)
{
smallest=100;
for(i=0;i<n;i++
)
{
if(at[i]<=time && rt[i] <rt[smallest] && rt[i]>0)
{
smallest=i;
}
}
rt[smallest]--;
if(rt[smallest]==0
)
{
finished++;
endtime =
time+1;
j=smallest;
printf("p[%d]\t\t %d\t\t %d\n",j+1,endtime-bt[j]-at[j],endtime-
at[j]);swt += endtime-bt[j]-at[j];
stat += endtime-at[j];
}
}
float
awt=0.0,atat=0.0;
awt=(float)swt/n;
atat=(float)stat/n;
printf("average waiting time %f\n",awt);
printf("average turnaround time
%f\n",atat);
```

```
}
```