# 1) Simulate the following CPU scheduling algorithms

a) FCFS
b) SJF
c) Priority
d) Round Robin

## a) FCFS:

**AIM: A program to simulate the FCFS CPU scheduling algorithm**

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
char pn[10][10];
int arr[10],bur[10],star[10],finish[10],tat[10],wt[10],i,n;
int totwt=0,tottat=0;
clrscr();
printf("Enter the number of processes:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Enter the Process Name, Arrival Time & Burst Time:");
scanf("%s%d%d",&pn[i],&arr[i],&bur[i]);
}
for(i=0;i<n;i++)
{
if(i==0)
{
star[i]=arr[i];
wt[i]=star[i]-arr[i];
finish[i]=star[i]+bur[i];
tat[i]=finish[i]-arr[i];
}
else
{
star[i]=finish[i-1];
wt[i]=star[i]-arr[i];
finish[i]=star[i]+bur[i];
tat[i]=finish[i]-arr[i];
}
}
printf("\nPName     Arrtime     Burtime    Start     TAT   Finish");
for(i=0;i<n;i++)
{
printf("\n%s\t%6d\t\t%6d\t%6d\t%6d\t%6d",pn[i],arr[i],bur[i],star[i],tat[i],finish[i]);
totwt+=wt[i];
tottat+=tat[i];
}
printf("\nAverage Waiting time:%f",(float)totwt/n);
printf("\nAverage Turn Around Time:%f",(float)tottat/n);
getch();
}
```

## OUTPUT:

**Input:**

```
Enter the number of processes: 3
Enter the Process Name, Arrival Time & Burst Time: 1 2 3
Enter the Process Name, Arrival Time & Burst Time: 2 5 6
Enter the Process Name, Arrival Time & Burst Time: 3 6 7
```

**Output:**

```
PName Arrtime   Burtime   Srart  TAT  Finish
1        2          3        2      3     5
2        5          6        5      6     4
3        6          7        6      7     10

Average Waiting Time: 3.333
Average Turn Around Time: 7.000
```

# b) SJF:

## AIM: A program to simulate the SJF CPU scheduling algorithm

## PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
int et[20],at[10],n,i,j,temp,st[10],ft[10],wt[10],ta[10];
int totwt=0,totta=0;
float awt,ata;
char pn[10][10],t[10];
clrscr();
printf("Enter the number of process:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Enter process name, arrival time & execution time:");
flushall();
scanf("%s%d%d",pn[i],&at[i],&et[i]);
}
for(i=0;i<n;i++)
for(j=0;j<n;j++)
{
if(et[i]<et[j])
{
temp=at[i];
at[i]=at[j];
at[j]=temp;
temp=et[i];
et[i]=et[j];
et[j]=temp;
strcpy(t,pn[i]);
strcpy(pn[i],pn[j]);
strcpy(pn[j],t);
}
}
for(i=0;i<n;i++)
{
if(i==0)
st[i]=at[i];
else
st[i]=ft[i-1];
wt[i]=st[i]-at[i];
ft[i]=st[i]+et[i];
ta[i]=ft[i]-at[i];
totwt+=wt[i];
totta+=ta[i];
}
awt=(float)totwt/n;
ata=(float)totta/n;
printf("\nPname\tarrivaltime\texecutiontime\twaitingtime\ttatime");
for(i=0;i<n;i++)
printf("\n%s\t%5d\t\t%5d\t\t%5d\t\t%5d",pn[i],at[i],et[i],wt[i],ta[i]);
printf("\nAverage waiting time is:%f",awt);
printf("\nAverage turnaroundtime is:%f",ata);
getch();
}
```

## OUTPUT:

**Input:**

```
Enter the number of processes: 3
Enter the Process Name, Arrival Time & Burst Time: 1 4 6
Enter the Process Name, Arrival Time & Burst Time: 2 5 15
Enter the Process Name, Arrival Time & Burst Time: 3 6 11
```

**Output:**

```
Pname   arrivaltime   executiontime   waitingtime  tatime
1             4              6              0         6
3             6              11             4         15
2             5              15             16        31

Average Waiting Time: 6.6667
Average Turn Around Time: 17.3333
```

## c) Priority:

**AIM: A program to simulate the priority CPU scheduling algorithm**

## PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
int et[20],at[10],n,i,j,temp,p[10],st[10],ft[10],wt[10],ta[10];
int totwt=0,totta=0;
float awt,ata;
char pn[10][10],t[10];
clrscr();
printf("Enter the number of process:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Enter process name,arrivaltime,execution time & priority:");
flushall();
scanf("%s%d%d%d",pn[i],&at[i],&et[i],&p[i]);
}
for(i=0;i<n;i++)
for(j=0;j<n;j++)
{
if(p[i]<p[j])
{
temp=p[i];
p[i]=p[j];
p[j]=temp;
temp=at[i];
at[i]=at[j];
at[j]=temp;
temp=et[i];
et[i]=et[j];
et[j]=temp;
strcpy(t,pn[i]);
strcpy(pn[i],pn[j]);
strcpy(pn[j],t);
}
}
for(i=0;i<n;i++)
{
```

```
if(i==0)
{
st[i]=at[i];
wt[i]=st[i]-at[i];
ft[i]=st[i]+et[i];
ta[i]=ft[i]-at[i];
}
else
{
st[i]=ft[i-1];
wt[i]=st[i]-at[i];
ft[i]=st[i]+et[i];
ta[i]=ft[i]-at[i];
}
totwt+=wt[i];
totta+=ta[i];
}
awt=(float)totwt/n;
ata=(float)totta/n;
printf("\nPname\tarrivaltime\texecutiontime\tpriority\twaitingtime\ttatime");
for(i=0;i<n;i++)
printf("\n%s\t%5d\t\t%5d\t\t%5d\t\t%5d\t\t%5d",pn[i],at[i],et[i],p[i],wt[i],ta[i]);
printf("\nAverage waiting time is:%f",awt);
printf("\nAverage turnaroundtime is:%f",ata);
getch();
}
```

## OUTPUT:

**Input:**

```
Enter the number of processes: 3
Enter the Process Name, Arrival Time, execution time & priority: 1 2 3 1
Enter the Process Name, Arrival Time, execution time & priority: 2 4 5 2
Enter the Process Name, Arrival Time, execution time & priority: 3 5 6 3
```

**Output:**

```
Pname    arrivaltime   executiontime   priority   waitingtime     tatime
1              2              3            1            0              3
2              4              5            2            1              6
3              5              6             3           5              11


Average Waiting Time: 2.0000
Average Turn Around Time: 6.6667
```

## d) Round Robin:

**AIM: A program to simulate the Round Robin CPU scheduling algorithm**

## PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int et[30],ts,n,i,x=0,tot=0;
char pn[10][10];
clrscr();
printf("Enter the no of processes:");
scanf("%d",&n);
printf("Enter the time quantum:");
scanf("%d",&ts);
for(i=0;i<n;i++)
{
printf("enter process name  & estimated time:");
scanf("%s %d",pn[i],&et[i]);
}
printf("The processes are:");
for(i=0;i<n;i++)
printf("process %d: %s\n",i+1,pn[i]);
for(i=0;i<n;i++)
tot=tot+et[i];
while(x!=tot)
{
for(i=0;i<n;i++)
{
if(et[i]>ts)
{
x=x+ts;
printf("\n %s -> %d",pn[i],ts);

et[i]=et[i]-ts;
}

else
if((et[i]<=ts)&&et[i]!=0)
{
x=x+et[i];
printf("\n %s  ->  %d",pn[i],et[i]);
et[i]=0;}
}
}
printf("\n Total Estimated Time:%d",x);
getch();
}
```

## OUTPUT:

**Input:**

```
Enter the no of processes: 2
Enter the time quantum: 3

Enter the process name & estimated time: p1 12
Enter the process name & estimated time: p2 15
```

**Output:**

```
p1 -> 3
p2 -> 3
p1 -> 3
p2 -> 3
p1 -> 3
p2 -> 3
p1 -> 3
p2 -> 3
p2 -> 3

Total Estimated Time: 27
```