

NAME- SANMAY DAS

DEPARTMENT- CSE (4TH SEM , SEC - A)

ENROLLMENT NO- 2011200001045

REGISTRATION NO- 200010455106

OS LAB ASSIGNMENT-5

Q1.write a C program to implement FCFS scheduling algorithm.

CODE:

```
#include<stdio.h>

int main()

{
    // #ifndef ONLINE_JUDGE
    // freopen("input.txt", "r", stdin);
    // freopen("output.txt", "w", stdout);
    // #endif
    int n,bt[20],wt[20],tat[20],avwt=0,avtat=0,i,j;
    printf("Enter total number of processes:");
    scanf("%d",&n);
```

```
printf("\nEnter Process Burst Time\n");
```

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

```
{
```

```
    printf("P[%d]:",i+1);
```

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

```
}
```

```
wt[0]=0;
```

```
for(i=1;i<n;i++)
```

```
{
```

```
    wt[i]=0;
```

```
    for(j=0;j<i;j++)
```

```
        wt[i]+=bt[j];
```

```
}
```

```
printf("\nProcess\t\tBurst Time\tWaiting Time\tTurnaround Time");
```

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

```
{
```

```
    tat[i]=bt[i]+wt[i];
```

```
    avwt+=wt[i];

    avtat+=tat[i];

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

}

avwt/=i;

avtat/=i;

printf("\n\nAverage Waiting Time: %d",avwt);

printf("\n\nAverage Turnaround Time: %d\n",avtat);


return 0;

}
```

OUTPUT:

```

1 #include<stdio.h>
2
3 int main()
4 {
5     #ifndef ONLINE_JUDGE
6     freopen("input.txt", "r", stdin);
7     freopen("output.txt", "w", stdout);
8     #endif
9     int n,bt[20],wt[20],tat[20],avwt=0,avtat=0,i,j;
10    printf("Enter total number of processes:");
11    scanf("%d",&n);
12
13    printf("\nEnter Process Burst Time\n");
14    for(i=0;i<n;i++)
15    {
16        printf("P[%d]:",i+1);
17        scanf("%d",&bt[i]);
18    }
19
20    wt[0]=0;
21
22    for(i=1;i<n;i++)
23    {
24        wt[i]=0;
25        for(j=0;j<i;j++)
26            wt[i]+=bt[j];
27    }
28
29    printf("\nProcess\t\tBurst Time\tWaiting Time\tTurnaround Time");
30
31    for(i=0;i<n;i++)
32    {
33        tat[i]=bt[i]+wt[i];
34        avwt+=wt[i];
35        avtat+=tat[i];
36        printf("\nP[%d]\t\t%d\t\t%d\t\t%d",i+1,bt[i],wt[i],tat[i]);
37    }
38
39    avwt/=i;
40

```

input.txt

```

3
2 33 2 1

```

output.txt

```

1 Enter total number of processes:
2 Enter Process Burst Time
3 P[1]:P[2]:P[3]:
4 Process Burst Time Waiting Time
5 Turnaround Time
6 P[1] 33 0 33
7 P[2] 2 33 35
8 P[3] 1 35 36
9
10 Average Waiting Time: 22
11 Average Turnaround Time: 34

```

Q2. write a C program to implement SJF scheduling algorithm.

CODE:

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
// #ifndef ONLINE_JUDGE
```

```
// freopen("input.txt", "r", stdin);
```

```
// freopen("output.txt", "w", stdout);
```

```
// #endif
```

```
int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;
```

```
float avg_wt,avg_tat;  
  
printf("Enter number of process:");  
  
scanf("%d",&n);
```

```
printf("\nEnter Burst Time:\n");
```

```
for(i=0;i<n;i++)  
{  
    printf("p%d:",i+1);  
    scanf("%d",&bt[i]);  
    p[i]=i+1;  
}
```

```
//sorting of burst times
```

```
for(i=0;i<n;i++)  
{  
    pos=i;  
    for(j=i+1;j<n;j++)  
    {  
        if(bt[j]<bt[pos])  
            pos=j;  
    }  
}
```

```
temp=bt[i];  
bt[i]=bt[pos];  
bt[pos]=temp;
```

```
temp=p[i];  
p[i]=p[pos];  
p[pos]=temp;  
}
```

```
wt[0]=0;
```

```
for(i=1;i<n;i++)  
{  
    wt[i]=0;  
    for(j=0;j<i;j++)  
        wt[i]+=bt[j];  
  
    total+=wt[i];  
}
```

```
avg_wt=(float)total/n;
```

```
total=0;
```

```
printf("\nProcesst\tBurst Time\t\tWaiting Timet\tTurnaround Time");
```

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

{

```
tat[i]=bt[i]+wt[i];
```

```
total+=tat[i];
```

```
printf("\np%d\t\t\t %d\t\t\t %d\t\t\t\t\t %d",p[i],bt[i],wt[i],tat[i]);
```

}

```
avg_tat=(float)total/n;
```

```
printf("\nnAverage Waiting Time=%f",avg_wt);
```

```
printf("\nAverage Turnaround Time=%f",avg_tat);
```

```
return 0;
```

}

OUTPUT:

```
sjf.c x FCFS.c x try.cpp x cp2.cpp x cp1.cpp x cp.cpp x list.cpp x \_cpp x + input.txt x
1 #include<stdio.h>
2 int main()
3 {
4     #ifndef ONLINE_JUDGE
5     freopen("input.txt", "r", stdin);
6     freopen("output.txt", "w", stdout);
7     #endif
8     int bt[20], p[20], wt[20], tat[20], i, j, n, total=0, pos, temp;
9     float avg_wt, avg_tat;
10    printf("Enter number of process:");
11    scanf("%d", &n);
12
13    printf("\nEnter Burst Time:\n");
14    for(i=0; i<n; i++)
15    {
16        printf("p%d:", i+1);
17        scanf("%d", &bt[i]);
18        p[i]=i+1;
19    }
20
21    //sorting of burst times
22    for(i=0; i<n; i++)
23    {
24        pos=i;
25        for(j=i+1; j<n; j++)
26        {
27            if(bt[j]<bt[pos])
28                pos=j;
29        }
30
31        temp=bt[i];
32        bt[i]=bt[pos];
33        bt[pos]=temp;
34
35        temp=p[i];
36        p[i]=p[pos];
37        p[pos]=temp;
38    }
39
40    wt[0]=0;
41
42    [Finished in 476ms]
Line 55, Column 58 Spaces: 4 C
```

```
input.txt
5
2 4 3 7 1 2
```

```
output.txt
1 Enter number of process:
2 Enter Burst Time:
3 p1:p2:p3:p4:p5:
4 Processt Burst Time Waiting Timet Turnaround Time
5 p4 1 0 1
6 p5 2 1 3
7 p2 3 3 6
8 p1 4 6 10
9 p3 7 10 17
10 nAverage Waiting Time=4.000000
11 Average Turnaround Time=7.400000
```