

SHORTEST JOB FIRST(SJF) SCHEDULING.

AIM:

PROGRAM TO IMPLEMENT SHORTEST JOB FIRST SCHEDULING USING C LANGUAGE.

This is an approach which considers the next CPU burst. Each process possesses its next CPU burst.

When CPU is available, the process having the smallest next CPU burst is allocated CPU.

Now it may happen that two or more processes have the same next CPU burst. Then which process to allocate will be decided as per FCFS scheduling.

ALGORITHM:

1. Sort all the process according to the arrival time.
2. Then select that process which has minimum arrival time and minimum Burst time.
3. After completion of process make a pool of process which after till the completion of previous process and select that process among the pool which is having minimum Burst time.

Write a C program to implement Shortest job First(SJF) Scheduling?

Input 1: Total no. of Process(Ex: 4)

Input 2: Burst time of all four process(Ex: 6, 8,7,3)

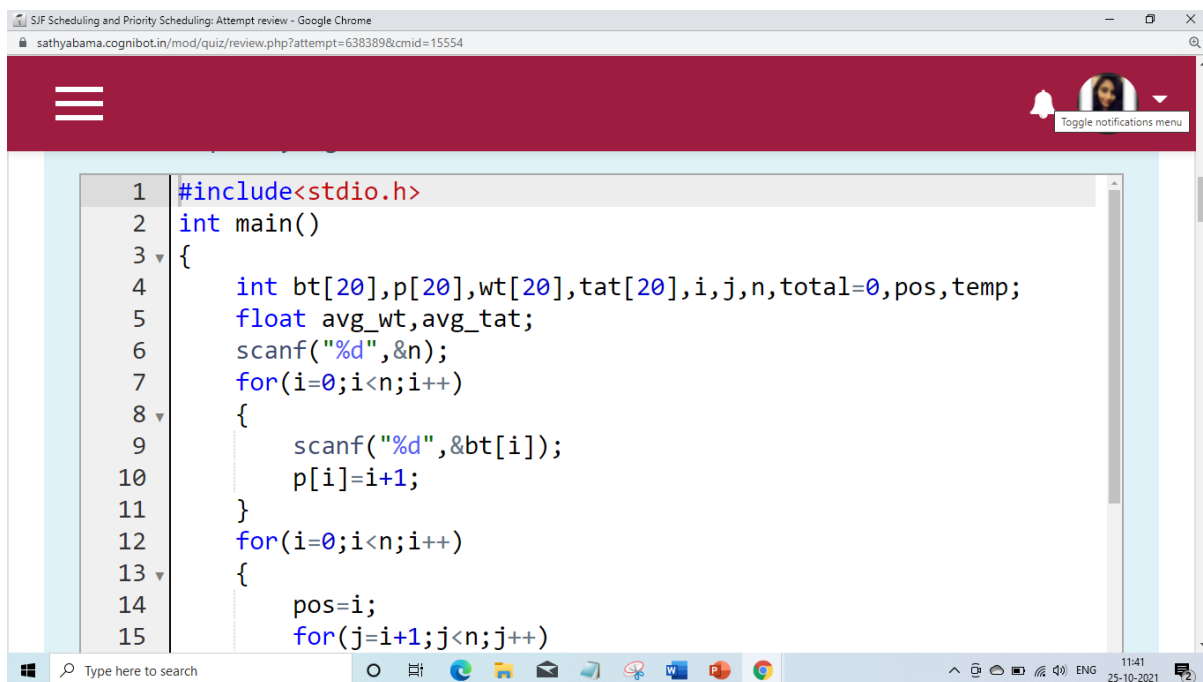
Output 1 : Average waiting time

Output 2: Average Turn around time

For example:

Test	Input	Result
T1	4	7.000000
	6	13.000000
	8	
	7	
	3	

PROGRAM:



The screenshot shows a web browser window with the title "SJF Scheduling and Priority Scheduling: Attempt review - Google Chrome". The address bar shows the URL "sathyabama.cognibot.in/mod/quiz/review.php?attempt=638389&cmid=15554". The browser interface includes a hamburger menu on the left, a user profile icon and "Toggle notifications menu" on the right, and a search bar at the bottom. The main content area displays a C program for SJF scheduling. The code is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;
5     float avg_wt,avg_tat;
6     scanf("%d",&n);
7     for(i=0;i<n;i++)
8     {
9         scanf("%d",&bt[i]);
10        p[i]=i+1;
11    }
12    for(i=0;i<n;i++)
13    {
14        pos=i;
15        for(j=i+1;j<n;j++)
```

The Windows taskbar at the bottom shows the search bar "Type here to search" and various application icons. The system tray on the right indicates the time "11:41" and date "25-10-2021".

SJF Scheduling and Priority Scheduling: Attempt review - Google Chrome
sathyabama.cognibot.in/mod/quiz/review.php?attempt=638389&cmid=15554

```
12  for(i=0;i<n;i++)
13  {
14      pos=i;
15      for(j=i+1;j<n;j++)
16      {
17          if(bt[j]<bt[pos])
18              pos=j;
19      }
20      temp=bt[i];
21      bt[i]=bt[pos];
22      bt[pos]=temp;
23      temp=p[i];
24      p[i]=p[pos];
25      p[pos]=temp;
26  }
```

Type here to search

11:42
25-10-2021

SJF Scheduling and Priority Scheduling: Attempt review - Google Chrome
sathyabama.cognibot.in/mod/quiz/review.php?attempt=638389&cmid=15554

```
21      bt[i]=bt[pos];
22      bt[pos]=temp;
23      temp=p[i];
24      p[i]=p[pos];
25      p[pos]=temp;
26  }
27  wt[0]=0;
28  for(i=1;i<n;i++)
29  {
30      wt[i]=0;
31      for(j=0;j<i;j++)
32          wt[i]+=bt[j];
33      total+=wt[i];
34  }
35  avg_wt=(float)total/n;
36  total=0;
```

Type here to search

11:42
25-10-2021

SJF Scheduling and Priority Scheduling: Attempt review - Google Chrome
sathyabama.cognibot.in/mod/quiz/review.php?attempt=638389&cmid=15554

```
33     total+=wt[i];  
34 }  
35 avg_wt=(float)total/n;  
36 total=0;  
37 for(i=0;i<n;i++)  
38 {  
39     tat[i]=bt[i]+wt[i];  
40     total+=tat[i];  
41 }  
42 avg_tat=(float)total/n;  
43 printf("%f",avg_wt);  
44 printf("\n%f\n",avg_tat);  
45 }
```

Type here to search

11:42 25-10-2021

RESULT:

SJF Scheduling and Priority Scheduling: Attempt review - Google Chrome
sathyabama.cognibot.in/mod/quiz/review.php?attempt=638389&cmid=15554

	Test	Input	Expected	Got	
✓	T1	4 6 8 7 3	7.000000 13.000000	7.000000 13.000000	✓
✓	T2	3 10 12 14	10.666667 22.666666	10.666667 22.666666	✓

Passed all tests! ✓

Type here to search

11:42 25-10-2021

RESULT:

SJF WAS SUCCESSFULLY IMPLEMENT USING C LANGUAGE.