

Output:

enter the no of process : 4

enter the bt of process 1 : 3

enter the bt of process 2 : 6

enter the bt of process 3 : 4

enter the bt of process 4 : 2

total waiting time : 16.000000

total tt time : 31.000000

average waiting time : 4.000000

Average tt time : 7.750000

Process	Bt	wt	Tt
P(1)	2	0	2
P(2)	3	2	5
P(3)	4	5	9
P(4)	6	9	15

write the program to implement SJF Scheduling algorithm  
to determine average wait time & average turnaround time  
#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

struct process {

int bt;

int wt;

int ct;

int pno;

} p[10];

int main()

{

struct process temp;

int n, i, j;

float tot\_wt = 0, tot\_ct = 0, avg1, avg2;

printf("enter the no of process");

scanf("%d", &n);

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

{

printf("enter the bt of process %d:", i+1);

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

p[i].pno = i+1;

}

for(i=0; i<n-1; i++)

{

Teacher's Signature.....



```

for (j = 0; j < n - i - 1; j++)
{
    if (p[j].bt > p[j+1].bt)
    {
        temp = p[j];
        p[j] = p[j+1];
        p[j+1] = temp;
    }
}

```

```

p[0].wt = 0;
p[0].ct = p[0].bt + p[0].wt;
for (j = 1; j < n; j++)
{
    p[j].wt = p[j-1].ct;
    p[j].ct = p[j].bt + p[j].wt;
}
for (i = 0; i < n; i++)
{

```

```

    tot_wt = tot_wt + p[i].wt;
    tot_ct = tot_ct + p[i].ct;
}
printf("total waiting time: %f\n", tot_wt);
printf("total ct time: %f\n", tot_ct);
avg1 = (tot_wt / n);
printf("average waiting time: %f\n", avg1);
avg2 = (tot_ct / n);
printf("Average ct time: %f\n", avg2);

```

Teacher's Signature.....

```
printf("Process | E | B | E | W | E | T | E | \n");  
for (i=0; i<n; i++)  
{
```

```
    printf("p(i-d) | t-i-d | t-i-d | \n", ci+1, ci) of  
    , p(i).w, p(i).eg;
```

```
    printf("\n");
```

```
}
```

```
return 0;
```

```
}
```

Teacher's Signature.....