2. Shortest Job First (SJF) Scheduling

c

#**include**<stdio.h>

**int** main() {

**int** n, bt[20], wt[20], tat[20], p[20], i,j,temp;

**float** avg\_wt=0, avg\_tat=0;

printf("Enter number of processes: ");

scanf("%d",&n);

printf("Enter Burst Time:\n");

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

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

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

p[i] = i+1;

}

*// Sorting burst time in ascending order*

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

**for**(j=i+1;j<n;j++) {

**if**(bt[i]>bt[j]) {

temp=bt[i];

bt[i]=bt[j];

bt[j]=temp;

temp=p[i];

p[i]=p[j];

p[j]=temp;

}

}

}

wt[0]=0;

**for**(i=1;i<n;i++) {

wt[i]=wt[i-1]+bt[i-1];

}

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

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

avg\_wt+=wt[i];

avg\_tat+=tat[i];

}

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

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

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

}

printf("\n\nAverage Waiting Time: %.2f",avg\_wt/n);

printf("\nAverage Turnaround Time: %.2f",avg\_tat/n);

**return** 0;

}

**Output:**

text

Enter number of processes: 4

Enter Burst Time:

P[1]: 6

P[2]: 8

P[3]: 7

P[4]: 3

Process Burst Time Waiting Time Turnaround Time

P[4] 3 0 3

P[1] 6 3 9

P[3] 7 9 16

P[2] 8 16 24

Average Waiting Time: 7.00

Average Turnaround Time: 13.00

*Based on SJF algorithm implementation described in Scaler Topics*[*2*](https://www.scaler.com/topics/sjf-scheduling-in-c/)