**Student Name:** Vikash Choudhary

**Student ID:** 11808743 (K18LCB53)

**Email Address:** [cvikash0404@gmail.com](mailto:cvikash0404@gmail.com)

Problem:

**Ques. 3. Consider a scheduler which schedules the job by considering the arrival time of the processes where arrival time if given as 0 is discarded or displayed as error. The scheduler implements the shortest job first scheduling policy, but checks the queue of the processes after the every process terminates and time taken for checking and arranging the process according to the shortest job is 2 time unit.**

**Compute the waiting time, turnaround time and average waiting time and turnaround time of the processes. Also compute the total time taken by the processor to compute all the jobs. The inputs for the number of requirements, arrival time and burst time should be provided by the user.**

**Consider the following units for reference. Process Arrival time Burst Time 1 0 6 2 3 2 3 5 1 4 9 7 5 10 5 6 12 3 7 14 4 8 16 5 9 17 7 10 19 2**

**Develop a scheduler which submits the processes to the processor in the defined scenario, and compute the scheduler performance by providing the waiting time for process, turnaround time for process and average waiting time and turnaround time.**

Algorithm:

Step 1: Start the process

Step 2:

take input and assign number of processes and burst time and arrival time to each.

Step 3:

use for loop to take input from user

Step4:

Use another for loop for comparison in this for we take two for loop such that one no. can be compared with all

Step 4.1:

In comparison, if previous value is smaller to any value than swap the student.

Step 5:

After comparison, print the waiting time, turnover time, average waiting time, average turnover time

Step 6:

finish

Code:

1. **#include<stdio.h>**
2. **#include<conio.h>**
3. **int main()**
4. **{**
5. **int bt[10],p[10],n,temp,i,j,wt[10],sum=0;**
6. **float avg;**
7. **printf("Enter total no of proces:");**
8. **scanf("%d",&n);**
9. **printf("\n Enter burst time for each process:-");**
10. **for(i=0;i<n;i++)**
11. **{**
12. **printf("\nBurst time of process P%d:",i);**
13. **scanf("%d",&bt[i]);**
14. **p[i]=i;**
15. **}**
16. **for(i=0;i<n-1;i++)**
17. **{**
18. **for(j=i+1;j<n;j++)**
19. **{**
20. **if(bt[i]>bt[j])**
21. **{**
22. **temp=bt[i];**
23. **bt[i]=bt[j];**
24. **bt[j]=temp;**
25. **temp=p[i];**
26. **p[i]=p[j];**
27. **p[j]=temp;**
28. **}**
29. **}**
30. **}**
31. **wt[0]=0;**
32. **for(i=1;i<n;i++)**
33. **{**
34. **wt[i]=wt[i-1]+bt[i-1];**
35. **}**
36. **for(i=0;i<n;i++)**
37. **{**
38. **sum+=wt[i];**
39. **}**
40. **avg=(float)sum/n;**
41. **printf("\n Waiting time for each process:-");**
42. **for(i=0;i<n;i++)**
43. **{**
44. **printf("\n Waiting time for process P%d is %d**
45. **sec.",p[i],wt[i]);**
47. **}**
48. **printf("\n Average waiting time is %f sec.",avg);**
49. **getch();**
50. **return 0;**
51. **}**

**Output :**

