Aim : To write C program to implement SJF CPU scheduling algorithm

sushmit@SushmitEnvy: ~/SIT

```
sushmit@SushmitEnvy:~/SIT$ cat sjf.c
#include <stdio.h>

int main() {
    int bt[20], p[20], wt[20], tat[20], ct[20];
    int i, j, temp, n;
    int totalwt = 0, totaltat = 0, totalct = 0;
    float avgwt, avgtat, avgct;

    // Input number of processes
    printf("Enter Number of processes: ");
    scanf("%d", &n);

    // Input burst times
    printf("Enter Burst Time:\n");
    for (i = 0; i < n; i++) {
        printf("Process %d: ", i + 1);
        scanf("%d", &bt[i]);
        p[i] = i + 1;  // Process ID
    }

    // Sorting processes by Burst Time using Bubble Sort (Shortest Job First Scheduling)
    for (i = 0; i < n; i++) {
        for (j = 0; j < n - 1; j++) {
            if (bt[j] > bt[j + 1]) {
                // Swap burst times
                temp = bt[j];
                bt[j] = bt[j + 1];
                bt[j + 1] = temp;

                // Swap process IDs
                temp = p[j];
                p[j] = p[j + 1];
                p[j + 1] = temp;
            }
        }
    }
```

sushmit@SushmitEnvy: ~/SIT

```
    // Calculating Waiting Time, Turnaround Time, and Completion Time
    printf("Process \tBurst Time\tWaiting Time\tTurnaround Time\tCompletion Time\n");
    for (i = 0; i < n; i++) {
        wt[i] = 0;
        tat[i] = 0;

        // Waiting Time calculation
        for (j = 0; j < i; j++) {
            wt[i] += bt[j];
        }

        // Turnaround Time = Waiting Time + Burst Time
        tat[i] = wt[i] + bt[i];

        // Completion Time = Turnaround Time (as processes execute sequentially)
        ct[i] = tat[i];

        // Accumulate totals for averages
        totalwt += wt[i];
        totaltat += tat[i];
        totalct += ct[i];

        // Print process details
        printf("%d \t\t%d \t\t%d \t\t%d \t\t%d\n", p[i], bt[i], wt[i], tat[i], ct[i]);
    }

    // Compute averages
    avgwt = (float)totalwt / n;
    avgtat = (float)totaltat / n;
    avgct = (float)totalct / n;

    // Display averages
    printf("Average Waiting Time: %.2f\n", avgwt);
    printf("Average Turnaround Time: %.2f\n", avgtat);
    printf("Average Completion Time: %.2f\n", avgct);

    return 0;
```

```
sushmit@SushmitEnvy:~/SIT$ ./sjf.out
Enter Number of processes
4
Enter Burst Time:
Process 1:
11
Process 2:
22
Process 3:
12
Process 4:
13
Process          burst time      Waiting time    Turnaround Time
1                11              0               11
3                12              11              23
4                13              23              36
2                22              36              58
Averge Waiting time : 17.500000
Average time around time : 32.000000sushmit@SushmitEnvy:~/SIT$
```