Ex. No.: 6b)
Date: 28.2.25.

SHORTEST JOB FIRST

Aim:

To implement the Shortest Job First (SJF) scheduling technique

Algorithm:

1. Declare the structure and its elements.
2. Get number of processes as input from the user.
3. Read the process name, arrival time and burst time
4. Initialize waiting time, turnaround time & flag of read processes to zero. 5. Sort based on burst time of all processes in ascending order 6. Calculate the waiting time and turnaround time for each process. 7. Calculate the average waiting time and average turnaround time. 8. Display the results.

Program Code:

```
#includ< stdio.h>.
int main
{
    int n;
    printf ("Enter the no. of processes: ");
    Scanf (" %d", &n);
    int pro [n], bt[n] , at [n] , wt [n] , tat [n] , ct[n];
    float total_tat = 0, tatal_wt = 0;
    for (int i = 0; i<n ; i++)
    {
        printf (" process %d burst time : ", i+1);
        Scanf (" %d", &bt [i]);
        printf (" process %d Arrival time: ", i+1);
        Scanf (" %d", &at [i]);
        pro[i] = i+1;
    }
    for (int i=0; i<n-1; i++)
    {
        for (int j=i; j<n-i-1; j++)
        {
            if (at[j] > at[j+1] || at [j] = = at [j+1] && bt[j] > bt
                                                            [j+1] )){
```

```
            int temp;
            temp = at [j];
            at [j] = at [j+1];
            at [j+1] = temp;

            temp = bt [j])
           bt [j] = bt[j+1];
           bt [j+1] = temp;
           temp = pro[j];
           pro[j] = pro[j+1];
           pro[j+1] = temp;
        }
     }
  }.
  ct [0] = at[0] + bt [0];
   for (int i=1; i<n; i++)
   {
        if (ct[i-1] < at[i])
            ct [i] = ct [i]+bt[i];
        else ct[i] = ct[i+1] + bt [i];
   }.
   for (int i=0; i<n; i++)
   {
      tat [i] = ct[i] - at [i];
      total_tat += tat[i];
   }
  for (int i=0; i<n; i++)
  {
      wt [i] = tat[i] - bt[i];
      total_wt += wt[i];
  }.
  float avg_wt = total_wt /n;
 float avg_tat = total_tat /n;
 printf ("process Arrival time Burst time  completion time turnaround time
                                                  waiting time \n");
 for (int i=0; i<n; i++){
     printf ("%d %d %d %d %d %d \n", pro [i],
                 at [i], bt[i], ct[i], tat[i], wt [i]);
    }.
 printf ("Average turnaround time = %.2f \n", avg_tat);
 printf ("Average waiting time = %.2f \n", avg_wt);
 return 0;
}.
```

Gantt chart:

P_2	P_4	P_3	P_5	P_1

tabulation:

Process	Bt(ms)	At(ms)	Ct(ms)	tAt = At - Ct (ms)	wt = TAT - Bt (ms).
1	10	0	19	19	9
2	1	0	1	1	0
3	2	0	4	4	2
4	1	0	2	2	1
5	5	0	9	9	4

Sample Output:

```
Enter the number of process:
4
Enter the burst time of the processes:
8 4 9 5

Process        Burst Time      Waiting Time   Turn Around Time

   2               4                0               4
   4               5                4               9
   1               8                9              17
   3               9               17              26

Average waiting time is: 7.5
Average Turn Around Time is: 13.0
```

```
Enter the no. of process : 5
Enter The Burst time of process1: 10
Enter the Burst time of process 2 : 1
Enter the Burst time of process 3: 2
Enter the Burst time of process 4: 1
Enter the Burst time of process 5: 5.

Process      Burst time      Waiting time      turn around time.
                                                  1
  2             1                0
  4             1                1                2
                                 2                4
  3             2
                                 4                9
  5             5
  1             10               9                19

              Average turn around time : 8.75.
              Average waiting time     : 4.00.
```

Result:

Thus the implementation of shortest Job first (SJF). cpu scheduling has been executed successfully.