Ex. No.: 6b)
Date: 26/2/25

SHORTEST JOB FIRST

Aim:

To implement the Shortest Job First (SJF) scheduling technique

Algorithm:

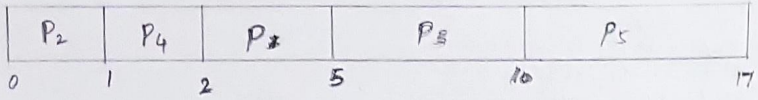
1. Declare the structure and its elements.
2. Get number of processes as input from the user.
3. Read the process name, arrival time and burst time
4. Initialize waiting time, turnaround time & flag of read processes to zero. 5. Sort based on burst time of all processes in ascending order 6. Calculate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

```
#include <stdio.h>
int main()
{
  int n;
  printf("Enter the number of processes: ");
  scanf("%d", &n);
  int process[n], bt[n], at[n], wt[n], tat[n], ct[n];
  float total_wt = 0, total_tat = 0;
  printf("Enter the Burst time and Arrival time :\n");
  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]);
    process[i] = i+1;
  }
  for(int i=0; i<n-1; i++){
    for(int j=0; j<n-i-1; j++){
      if(at[j] > at[j+1] || (at[j] == at[j+1] && bt[j] > bt[j+1]))
```

```
for(int i=0; i<n-1; i++){
  for(int j=0; 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 = process[j];
    process[j] = process[j+1];
    process[j+1] = temp;
}}}
ct[0] = at[0]+bt[0];
for(int i=1; i<n; i++)
  if(ct[i-1] < at[i])
    ct[i] = at[i]+bt[i];
  else
    ct[i] = ct[i-1]+bt[i];
for(int i=0; i<n; i++)
  tat[i] = ct[i-1]-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: Turn around time
                                                    Waiting Time: \n");
for(int i=0; i<n; i++)
printf("%d %d %d %d %d %d\n", pno[i],
        at[i], bt[i], ct[i], tat[i], wt[i]);
}
printf("Average waiting time = %.2f\n", avg_wt);
printf("Average turn around time = %.2f\n", avg_tat);
return 0;
}
```

Gantt Chart:



Calculation:

Process	AT (ms)	BT (ms)	CT (ms)	TAT = CT − AT (ms)	WT = TAT − BT (ms)
1	0	3	5	5	2
2	0	1	1	1	0
3	0	5	10	10	5
4	0	1	2	2	1
5	0	7	17	17	10

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:
3
1
5
1
7

Process	Burst Time	Waiting Time	Turn Around Time
2	1	0	1
4	1	1	2
1	3	2	5
3	5	5	10
5	7	10	17

Avg. waiting time is: 3.6 ms
Avg. turn around time is: 7 ms

Result:

Thus the implementation of Shortest Job First (SJF) CPU scheduling has been successfully implemented