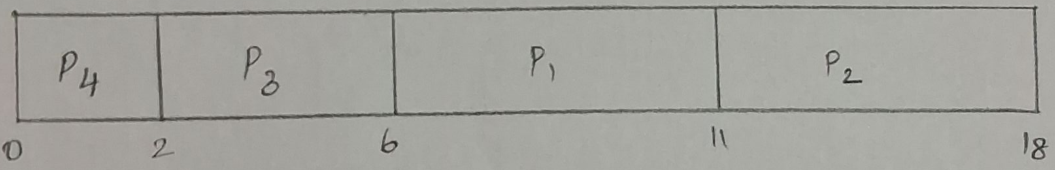
SJF

Process	AT (ms)	BT (ms)	CT (ms)	TAT (ms)	WT (ms)
P_1	0	5	11	11	6
P_2	0	7	18	18	11
P_3	0	4	6	6	2
P_4	0	2	2	2	0

Average TAT = 9.25 ms

Average WT = 4.75 ms.



Ex. No.: 6b)
Date: 26.02.25

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>
#include < stdlib.h>
int main ()
{
  int n;
  printf("\nEnter the number of processes:\n");
  scanf("%d", &n);
  int bt[n], at=0, ct[n], tat[n], wt[n];
  printf("\nEnter the burst time\n");
  for(int i=0 ; i<n; i++)
  {
     scanf("%d", &bt[i]);
  }
  int sbt[n];
  for(int i=0; i<n; i++)
  {
     sbt[i]= bt[i];
  }
```

```
for( int i =0; i<n-1; i++)
{
    for(int j =0; j<n-1-i; j++)
    {
        if(sbt[j+1] < sbt[j])
        {
            int temp = sbt[j+1];
            sbt[j+1]= sbt[j];
            sbt[j] = temp;
        }
    }
}
int c=0;
for(int i=0; i<n; i++)
{
    for( int j =0; j<n; j++)
    {
        if (sbt[i]==bt[j])
        {
            ct[j]= c+bt[j];
            c = ct[j];
            tat[j]= ct[j] - at;
            wt[j]= tat[j]- bt[j];
        }
    }
}
printf("\n The completion time: \n");
for(int i =0; i<n; i++)
    printf("%d\n", ct[i]);
printf("\n Turn Around Time : \n");
for(int i =0; i<n; i++)
    printf("%d", tat[i]);
printf("\n Wait Time : \n");
for(int i=0; i<n; i++)
    printf("%d\n", wt[i]);
```

```
int atat=0, awt=0;
for(int i=0; i<n; i++)
{
   atat = atat + tat[i];
   awt = awt + wt[i];
}
printf("Average Turn around time : ~~%d.2f"~~ %.2f"(float) atat/n);
printf("Average wait time : ~~%d.2f"~~ %.2f%."(float) awt/n);
return 0;
}
```

output:

```
Enter the no. of processes :
   4
Enter the burst time:
  5
  7
  4
  2
The completion time:
  11
  18
  6
  2
The turn around time:
  11
  18
  6
  2
The wait time:
 6
 11
 2
 0
Average TAT : 9.25 ms
Average WT: 4.75 ms.
```

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
```

Result:

The shortest job algorithm is executed using C.

[illegible]