Ex. No.: 7b
Date: 30.3.24

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:

```
//sjf.py
bt=[ ]
print("Enter the number of processes: ");
n=int(input())
processes=[]
for i in range(0,n):
        processes.insert(i, i+1)
print("Enter BT of the processes: ");
bt = list(map(int, input().split()))
for i in range(0, len(bt)-1):                    # bubble sort on bt
      for j in range(0, len(bt)-i-1):
            if (bt[j]>bt[j+1]):
                  temp=bt[j]
                  bt[j]= bt[j+1]
                  bt[j+1]=temp
                  temp=processes[j]
                  processes[j]=processes[j+1]
                  process[j+1]=temp
```

```
wt=[]
avgwt=0
tat=[]
avgtat=0
wt.insert(0,0)
tat.insert(0,bt[0])
for i in range(1,n):
        wt.insert(i, wt[i-1]+bt[i-1])
        tat.insert(i, wt[i]+bt[i])
        avgwt += wt[i]
        avgtat += tat[i]
avgwt = float(avgwt)/n
avgtat = float(avgtat)/n
print("\n")
print("Process \t\tBT\t\t WT\t\t TAT ")
for i in range(0,n):
     print(str(processes[i])+"\t\t"+str(bt[i])+"\t\t" + wt[i]+"\t\t"+
       tat[i])
print(" Average WT is : " + avgwt)
print (" Averge TAT is : " + avgtat)
```

Output:

```
Enter the no. of processes:
 3
Enter the burst time of the process:
 3    5   2
Process      BT      WT       TAT
                      0         2
  3          2
  1          3        2         5
             5        5        10
  2

Average waiting time is: 2.33
Average ~~to~~Turn Around Time is: 5.0
```

10/10

23/4/24

RESULT: The program has been compiled & executed successfully.