Ex. No.: 6b)
Date:

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:

```
bt=[] #bt stands for burst time
print("Enter the number of process: ")
n=int(input())
processes=[]
for i in range(0,n):
 processes.insert(i,i+1)
print("Enter the burst time of the processes: \n")
bt=list(map(int, raw_input().split()))
for i in range(0,len(bt)-1): #applying 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]
 processes[j+1]=temp
wt=[] #wt stands for waiting time
avgwt=0 #average of waiting time
tat=[] #tat stands for turnaround time
avgtat=0 #average of total turnaround time
wt.insert(0,0)
tat.insert(0,bt[0])
for i in range(1,len(bt)):
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 Burst Time\t Waiting Time\t Turn Around
Time")
for i in range(0,n):
 print(str(processes[i])+"\t\t"+str(bt[i])+"\t\t"+str(wt[i])
+"\t\t"+str(tat[i]))
print("Average Waiting time is: "+str(avgwt))
print("Average Turn Around Time is: "+str(avgtat))
```

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
```