Name: Hitendra Sisodia
Sap id: 500091910

Lab 4

Ques: Implement both the preemtive and non-preemtive version of the SJF Algorithm.

Non-Preemptive SJF

In Preemptive SJF Scheduling, jobs are put into the ready queue as they come. A process with shortest burst time begins execution.

Non-Preemptive SJF

In non-preemptive scheduling, once the CPU cycle is allocated to process, the process holds it till it reaches a waiting state or terminated.

1.) Preemtive Scheduling.

Source Code

```
#include<bits/stdc++.h>
int main()
{
    int arrival_time[10], burst_time[10], temp[10];
    int i, smallest, count = 0, time, limit;
    double wait_time = 0, turnaround_time = 0, end;
    float average_waiting_time, average_turnaround_time;
    printf("Enter the Total Number of Processes:");
    scanf("%d", &limit);
    printf("Enter Details of %d Processesn", limit);
    for(i = 0; i < limit; i++)
    {
        printf("nEnter Arrival Time:");
        scanf("%d", &arrival_time[i]);
        printf("Enter Burst Time:");
        scanf("%d", &burst_time[i]);
        temp[i] = burst_time[i];
    }
    burst_time[9] = 9999;
    for(time = 0; count != limit; time++)
    {
        smallest = 9;
        for(i = 0; i < limit; i++){
            if(arrival_time[i] <= time && burst_time[i] < burst_time[smallest] && burst_time[i] > 0){
                smallest = i;
            }
            burst_time[smallest]--;
            if(burst_time[smallest] == 0){
                count++;
                end = time + 1;
                wait_time = wait_time + end - arrival_time[smallest] - temp[smallest];
                turnaround_time = turnaround_time + end - arrival_time[smallest];
            }
        }
    }
    average_waiting_time = wait_time / limit;
    average_turnaround_time = turnaround_time / limit;
    printf("Average Waiting Time:%lf", average_waiting_time);
    printf("Average Turnaround Time:%lf", average_turnaround_time);
    return 0;
```

Output

```
Enter the Total Number of Processes:2
Enter Details of 2 Processes
Enter Arrival Time:0
Enter Burst Time:7
Enter Arrival Time:2
Enter Burst Time:3
Average Waiting Time:-0.500000
Average Turnaround Time:4.500000
```

Name: Hitendra Sisodia
Sap id: 500091910

2.) Non-Preemtive Scheduling.

Source Code

```
#include<bits/stdc++.h>
int main()
{
    int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;
    float avg_wt,avg_tat;
    printf("Enter number of process:");
    scanf("%d",&n);
    printf("Enter Burst Time:");
    for(i=0;i<n;i++)
    {
        printf("%d:",i+1);
        scanf("%d",&bt[i]);
        p[i]=i+1;
    }
    for(i=0;i<n;i++){
        pos=i;
        for(j=i+1;j<n;j++){
            if(bt[j]<bt[pos]){
                pos=j;
            }
        }
        temp=bt[i];
        bt[i]=bt[pos];
        bt[pos]=temp;
        temp=p[i];
        p[i]=p[pos];
        p[pos]=temp;
    }
    wt[0]=0;
    for(i=1;i<n;i++){
        wt[i]=0;
        for(j=0;j<i;j++){
            wt[i]+=bt[j];
        }
        total+=wt[i];
    }
    avg_wt=(float)total/n;
    total=0;
    printf("Process Burst Time Waiting Time Turnaround Time");
    for(i=0;i<n;i++){
        tat[i]=bt[i]+wt[i];
        total+=tat[i];
        printf("np%dtt  %dtt    %dttt%d",p[i],bt[i],wt[i],tat[i]);
    }
    avg_tat=(float)total/n;
    printf("Average Waiting Time=%f",avg_wt);
    printf("Average Turnaround Time=%f",avg_tat);
}
```

Output

```
Enter number of process:2
Enter Burst Time
1:4
2:3
PB Time  W Time TA Time
np2tt  3tt    0ttt3
np1tt  4tt    3ttt7
Average Waiting Time=1.500000Average Turnaround Time=5.000000
```