/\* 2. SJF(Non-Preemptive) \*/

import java.util.Scanner;

class SJF1{

public static void main(String args[]){

int burst\_time[],process[],waiting\_time[],tat[],i,j,n,total=0,pos,temp;

float wait\_avg,TAT\_avg;

Scanner s = new Scanner(System.in);

System.out.print("Enter number of process: ");

n = s.nextInt();

process = new int[n];

burst\_time = new int[n];

waiting\_time = new int[n];

tat = new int[n];

System.out.println("\nEnter Burst time:");

for(i=0;i<n;i++)

{

System.out.print("\nProcess["+(i+1)+"]: ");

burst\_time[i] = s.nextInt();;

process[i]=i+1; //Process Number

}

//Sorting

for(i=0;i<n;i++)

{

pos=i;

for(j=i+1;j<n;j++)

{

if(burst\_time[j]<burst\_time[pos])

pos=j;

}

temp=burst\_time[i];

burst\_time[i]=burst\_time[pos];

burst\_time[pos]=temp;

temp=process[i];

process[i]=process[pos];

process[pos]=temp;

}

//First process has 0 waiting time

waiting\_time[0]=0;

//calculate waiting time

for(i=1;i<n;i++)

{

waiting\_time[i]=0;

for(j=0;j<i;j++)

waiting\_time[i]+=burst\_time[j];

total+=waiting\_time[i];

}

//Calculating Average waiting time

wait\_avg=(float)total/n;

total=0;

System.out.println("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");

for(i=0;i<n;i++)

{

tat[i]=burst\_time[i]+waiting\_time[i]; //Calculating Turnaround Time

total+=tat[i];

System.out.println("\n p"+process[i]+"\t\t "+burst\_time[i]+"\t\t "+waiting\_time[i]+"\t\t "+tat[i]);

}

//Calculation of Average Turnaround Time

TAT\_avg=(float)total/n;

System.out.println("\n\nAverage Waiting Time: "+wait\_avg);

System.out.println("\nAverage Turnaround Time: "+TAT\_avg);

}

}

/\* 2. SJF(Preemptive)\*/

import java.util.Scanner;

class sjf\_swap1{

public static void main(String args[])

{

int burst\_time[],process[],waiting\_time[],tat[],arr\_time[],completion\_time[],i,j,n,total=0,total\_comp=0,pos,temp;

float wait\_avg,TAT\_avg;

Scanner s = new Scanner(System.in);

System.out.print("Enter number of process: ");

n = s.nextInt();

process = new int[n];

burst\_time = new int[n];

waiting\_time = new int[n];

arr\_time=new int[n];

tat = new int[n];

completion\_time=new int[n];

//burst time

System.out.println("\nEnter Burst time:");

for(i=0;i<n;i++)

{

System.out.print("\nProcess["+(i+1)+"]: ");

burst\_time[i] = s.nextInt();;

process[i]=i+1; //Process Number

}

//arrival time

System.out.println("\nEnter arrival time:");

for(i=0;i<n;i++)

{

System.out.print("\nProcess["+(i+1)+"]: ");

arr\_time[i] = s.nextInt();;

process[i]=i+1; //Process Number

}

//Sorting

for(i=0;i<n;i++)

{

pos=i;

for(j=i+1;j<n;j++)

{

if(burst\_time[j]<burst\_time[pos])

pos=j;

}

temp=burst\_time[i];

burst\_time[i]=burst\_time[pos];

burst\_time[pos]=temp;

temp=process[i];

process[i]=process[pos];

process[pos]=temp;

System.out.println("process"+process[i]);

}

//completion time new

for(i=1;i<n;i++)

{

completion\_time[i]=0;

for(j=0;j<i;j++)

completion\_time[i]+=burst\_time[j];

total\_comp+=completion\_time[i];

}

//First process has 0 waiting time

waiting\_time[0]=0;

//calculate

waiting time

for(i=1;i<n;i++)

{

waiting\_time[i]=0;

for(j=0;j<i;j++)

waiting\_time[i]+=burst\_time[j];

total+=waiting\_time[i];

}

//Calculating Average waiting time

wait\_avg=(float)total/n;

total=0;

System.out.println("\nPro\_number\t Burst Time \tcompletion\_time\tWaiting Time\tTurnaround Time");

for(i=0;i<n;i++)

{

tat[i]=burst\_time[i]+waiting\_time[i];

//Calculating Turnaround Time

total+=tat[i];

System.out.println("\n"+process[i]+"\t\t "+burst\_time[i]+"\t\t "+completion\_time[i]+"\t\t"+waiting\_time[i]+"\t\t "+tat[i]);

}

//Calculation of Average Turnaround Time

TAT\_avg=(float)total/n;

System.out.println("\n\nAWT: "+wait\_avg);

System.out.println("\nATAT: "+TAT\_avg);

}

}