**FCFS:-**

**import** java.util.Scanner;

**class** process

{

**int** id,at=0,wt=0,tat=0,bt=0;

process(**int** a)

{

**this**.id=a+1;

}

}

**public** **class** FCFS

{Scanner sc=**new** Scanner(System.***in***);

process p[];

**int** n;

**void** input()

{

System.***out***.println("enter no of processes:");

n=sc.nextInt();

p=**new** process[n];

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

{

p[i]=**new** process(i);

System.***out***.println("enter arrival time for process"+p[i].id+":");

p[i].at=sc.nextInt();

System.***out***.println("enter burst time for process"+p[i].id+":");

p[i].bt=sc.nextInt();

}

}

**void** sort()

{

**for**(**int** i=0;i<n-1;i++)

{

**for**(**int** j=0;j<n-1;j++)

{

**if**(p[j].at>p[j+1].at)

{

process temp=p[j];

p[j]=p[j+1];

p[j+1]=temp;

}

}

}

}

**void** calc()

{

**int** ip=0,count=0,i=0;

**float** wtavg=0,tatavg=0;

**while**(count<n)

{

**if**(p[i].at>ip)

{

ip=p[i].at;

}

**else**

{

System.***out***.println("Current running process p"+p[i].id+"\nfrom "+ip+ " to "+(ip+p[i].bt) );

p[i].wt=ip-p[i].at;

p[i].tat=p[i].wt+p[i].bt;

wtavg+=p[i].wt;

tatavg+=p[i].tat;

System.***out***.println("Waiting time of process p"+p[i].id+"="+p[i].wt);

System.***out***.println("TAT time of process p"+p[i].id+"="+p[i].tat);

ip+=p[i].bt;

i++;

count++;

}

}

wtavg/=n;

tatavg/=n;

System.***out***.println("Average wt="+wtavg);

System.***out***.println("Average tat="+tatavg);

}

**Output:-**

enter no of processes:

5

enter arrival time for process1:

3

enter burst time for process1:

4

enter arrival time for process2:

5

enter burst time for process2:

3

enter arrival time for process3:

0

enter burst time for process3:

2

enter arrival time for process4:

5

enter burst time for process4:

1

enter arrival time for process5:

4

enter burst time for process5:

3

Current running process p3

from 0 to 2

Waiting time of process p3=0

TAT time of process p3=2

Current running process p1

from 3 to 7

Waiting time of process p1=0

TAT time of process p1=4

Current running process p5

from 7 to 10

Waiting time of process p5=3

TAT time of process p5=6

Current running process p2

from 10 to 13

Waiting time of process p2=5

TAT time of process p2=8

Current running process p4

from 13 to 14

Waiting time of process p4=8

TAT time of process p4=9

Average wt=3.2

Average tat=5.8

**SJF:-**

**import** java.util.Scanner;

**class** process2

{

**int** id,at=0,wt=0,tat=0,bt=0,ex=0;

process2(**int** a)

{

**this**.id=a+1;

}

}

**public** **class** SJF

{Scanner sc=**new** Scanner(System.***in***);

process2[] p;

**int** n;

**void** input()

{

System.***out***.println("enter no of processes:");

n=sc.nextInt();

p=**new** process2[n];

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

{

p[i]=**new** process2(i);

System.***out***.println("enter arrival time for process"+p[i].id+":");

p[i].at=sc.nextInt();

System.***out***.println("enter burst time for process"+p[i].id+":");

p[i].bt=sc.nextInt();

p[i].ex=p[i].bt;

}

}

**int** select(**int** ip)

{**int** min=9999;

**int** i=0,sel=-1;

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

{

**if**(p[i].at<=ip)

{

**if**(p[i].ex<min && p[i].ex!=0)

{

min=p[i].ex;

sel=i;

}

}

}

**return** sel;

}

**void** calc()

{

**int** ip=0,count=0,i=0;

**float** wtavg=0,tatavg=0;

**while**(count<n)

{

i=select(ip);

**if**(i==-1)

{

**continue**;

}

p[i].ex-=1;

**if**(p[i].ex==0)

{

count++;

System.***out***.println("p"+p[i].id+"->"+ip+" to "+(ip+1));

ip++;

p[i].tat=ip-p[i].at;

tatavg+=p[i].tat;

p[i].wt-=p[i].at;

wtavg+=p[i].wt;

System.***out***.println("Waiting time for p"+p[i].id+"="+p[i].wt);

System.***out***.println("TAT time for p"+p[i].id+"="+p[i].tat);

}

**else**

{

System.***out***.println("p"+p[i].id+"->"+ip+" to "+(ip+1));

ip++;

}

**for**(**int** k=0;k<n;k++)

{

**if**((k+1)!=p[i].id && p[k].ex!=0)

{

p[k].wt++;

}

}

}

wtavg/=n;

tatavg/=n;

System.***out***.println("Average wt="+wtavg);

System.***out***.println("Average tat="+tatavg);

}

**public** **static** **void** main(String[] args)

{

SJF f=**new** SJF();

f.input();

f.calc();

}

}

**Output:-**

enter no of processes:

5

enter arrival time for process1:

3

enter burst time for process1:

1

enter arrival time for process2:

1

enter burst time for process2:

4

enter arrival time for process3:

4

enter burst time for process3:

2

enter arrival time for process4:

0

enter burst time for process4:

6

enter arrival time for process5:

2

enter burst time for process5:

3

p4->0 to 1

p2->1 to 2

p2->2 to 3

p1->3 to 4

Waiting time for p1=0

TAT time for p1=1

p2->4 to 5

p2->5 to 6

Waiting time for p2=1

TAT time for p2=5

p3->6 to 7

p3->7 to 8

Waiting time for p3=2

TAT time for p3=4

p5->8 to 9

p5->9 to 10

p5->10 to 11

Waiting time for p5=6

TAT time for p5=9

p4->11 to 12

p4->12 to 13

p4->13 to 14

p4->14 to 15

p4->15 to 16

Waiting time for p4=10

TAT time for p4=16

Average wt=3.8

Average tat=7.0

**Round Robin:-**

**import** java.util.Scanner;

**class** process3

{

**int** id,at=0,wt=0,tat=0,bt=0,ex=0;

process3(**int** a)

{

**this**.id=a+1;

}

}

**public** **class** Round

{Scanner sc=**new** Scanner(System.***in***);

process3 p[];

**int** order[];

**int** n,q;

**void** input()

{

System.***out***.println("enter no of processes:");

n=sc.nextInt();

p=**new** process3[n];

order=**new** **int**[n];

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

{

p[i]=**new** process3(i);

System.***out***.println("enter arrival time for process"+p[i].id+":");

p[i].at=sc.nextInt();

System.***out***.println("enter burst time for process"+p[i].id+":");

p[i].bt=sc.nextInt();

p[i].ex=p[i].bt;

}

System.***out***.println("enter order in which you want to execute process:");

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

{

order[i]=sc.nextInt();

}

System.***out***.println("Enter time quantum:");

q=sc.nextInt();

}

**void** calc()

{

**int** ip=0,count=0,i=0,z=0;

**float** wtavg=0,tatavg=0;

**while**(count<n)

{

**boolean** flag=**false**;

**for**(**int** k=0;k<n;k++)

{

**if**(p[k].ex==0)

{

**continue**;

}

**if**(z==n)

{

z=0;

}

**if**(order[z]==p[k].id && p[k].ex!=0)

{

flag=**true**;

i=k;

**if**(p[i].at>ip && p[i].ex!=0)

{

ip=p[i].at;

}

**if**(p[i].ex<q)

{

System.***out***.println("Current running process p"+p[i].id+"\nfrom "+ip+ " to "+(ip+p[i].ex) );

ip+=p[i].ex;

p[i].ex=0;

}

**else**{

System.***out***.println("Current running process p"+p[i].id+"\nfrom "+ip+ " to "+(ip+q) );

ip+=q;

p[i].ex-=q;

}

**if**(p[i].ex==0)

{

p[i].tat=ip-p[i].at;

tatavg+=p[i].tat;

p[i].wt=p[i].tat-p[i].bt;

wtavg+=p[i].wt;

System.***out***.println("Waiting time for p"+p[i].id+"="+p[i].wt);

System.***out***.println("TAT time for p"+p[i].id+"="+p[i].tat);

count++;

}

z++;

**break**;

}

**if**(k==(n-1)&& flag==**false**)

{

z++;

**break**;

}

}

}

wtavg/=n;

tatavg/=n;

System.***out***.println("Average wt="+wtavg);

System.***out***.println("Average tat="+tatavg);

}

**public** **static** **void** main(String[] args)

{

Round f=**new** Round();

f.input();

f.calc();

}

}

**Output:-**

enter no of processes:

5

enter arrival time for process1:

0

enter burst time for process1:

5

enter arrival time for process2:

1

enter burst time for process2:

3

enter arrival time for process3:

2

enter burst time for process3:

1

enter arrival time for process4:

3

enter burst time for process4:

2

enter arrival time for process5:

4

enter burst time for process5:

3

enter order in which you want to execute process:

2

3

1

0

4

Enter time quantum:

2

Current running process p2

from 1 to 3

Current running process p3

from 3 to 4

Waiting time for p3=1

TAT time for p3=2

Current running process p1

from 4 to 6

Current running process p4

from 6 to 8

Waiting time for p4=3

TAT time for p4=5

Current running process p2

from 8 to 9

Waiting time for p2=5

TAT time for p2=8

Current running process p1

from 9 to 11

Current running process p1

from 11 to 12

Waiting time for p1=7

TAT time for p1=12

**Priority:-**

**import** java.util.Scanner;

**class** process4

{

**int** id,at=0,wt=0,tat=0,bt=0,prio=0;

**boolean** done=**false**;

process4(**int** a)

{

**this**.id=a+1;

}

}

**public** **class** Priority

{Scanner sc=**new** Scanner(System.***in***);

process4 p[];

**int** n;

**void** input()

{

System.***out***.println("enter no of processes:");

n=sc.nextInt();

p=**new** process4[n];

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

{

p[i]=**new** process4(i);

System.***out***.println("enter arrival time for process"+p[i].id+":");

p[i].at=sc.nextInt();

System.***out***.println("enter burst time for process"+p[i].id+":");

p[i].bt=sc.nextInt();

System.***out***.println("enter priority for process"+p[i].id+":");

p[i].prio=sc.nextInt();

}

}

**void** sort()

{

**for**(**int** i=0;i<n-1;i++)

{

**for**(**int** j=0;j<n-1;j++)

{

**if**(p[j].prio>p[j+1].prio)

{

process4 temp=p[j];

p[j]=p[j+1];

p[j+1]=temp;

}

}

}

}

**int** select(**int** ip,**int** i)

{

**if**(p[i].at<=ip && p[i].done==**false**)

{

**return** i;

}

**else**{

**for**(**int** k=0;k<n;k++)

{

**if**(p[k].at<=ip && p[i].done==**false**)

{

**return** k;

}

}

**return** -1;

}

}

**void** calc()

{

**int** ip=0,count=0,i=0;

**float** wtavg=0,tatavg=0;

**while**(count<n)

{

**int** z=select(ip,i);

**if**(z==-1)

{

ip=p[i].at;

}

**else**

{

i=z;

System.***out***.println("p"+p[i].id+"->"+ip+" to "+(ip+p[i].bt));

ip=ip+p[i].bt;

p[i].done=**true**;

p[i].tat=ip-p[i].at;

tatavg+=p[i].tat;

p[i].wt=p[i].tat-p[i].bt;

wtavg+=p[i].wt;

System.***out***.println("Waiting time for p"+p[i].id+"="+p[i].wt);

System.***out***.println("TAT time for p"+p[i].id+"="+p[i].tat);

count++;

i++;

}

}

wtavg/=n;

tatavg/=n;

System.***out***.println("Average wt="+wtavg);

System.***out***.println("Average tat="+tatavg);

}

**public** **static** **void** main(String[] args)

{

Priority f=**new** Priority();

f.input();

f.sort();

f.calc();

}

}

**Output:-**

enter no of processes:

5

enter arrival time for process1:

0

enter burst time for process1:

4

enter priority for process1:

2

enter arrival time for process2:

1

enter burst time for process2:

3

enter priority for process2:

3

enter arrival time for process3:

2

enter burst time for process3:

1

enter priority for process3:

4

enter arrival time for process4:

3

enter burst time for process4:

5

enter priority for process4:

5

enter arrival time for process5:

4

enter burst time for process5:

2

enter priority for process5:

5

p1->0 to 4

Waiting time for p1=0

TAT time for p1=4

p2->4 to 7

Waiting time for p2=3

TAT time for p2=6

p3->7 to 8

Waiting time for p3=5

TAT time for p3=6

p4->8 to 13

Waiting time for p4=5

TAT time for p4=10

p5->13 to 15

Waiting time for p5=9

TAT time for p5=11

Average wt=4.4

Average tat=7.4