

1. FCFS

INPUT :

```
import java.util.*;

public class FCFS {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

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

        int n = sc.nextInt();

        int pid[] = new int[n];

        int ar[] = new int[n];

        int bt[] = new int[n];

        int ct[] = new int[n];

        int ta[] = new int[n];

        int wt[] = new int[n];

        int temp;

        float avgwt = 0, avgta = 0;

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

            System.out.println("enter process " + (i + 1) + " arrival time: ");

            ar[i] = sc.nextInt();

            System.out.println("enter process " + (i + 1) + " burst time: ");

            bt[i] = sc.nextInt();

            pid[i] = i + 1;

        }

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

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

                if (ar[j] > ar[j + 1]) {

                    temp = ar[j];

                    ar[j] = ar[j + 1];

                    ar[j + 1] = temp;

                    temp = bt[j];
```

```

        bt[j] = bt[j + 1];
        bt[j + 1] = temp;
        temp = pid[j];
        pid[j] = pid[j + 1];
        pid[j + 1] = temp;
    }
}

}

for (int i = 0; i < n; i++) {
    if (i == 0) {
        ct[i] = ar[i] + bt[i];
    } else {
        if (ar[i] > ct[i - 1]) {
            ct[i] = ar[i] + bt[i];
        } else
            ct[i] = ct[i - 1] + bt[i];
    }

    ta[i] = ct[i] - ar[i];
    wt[i] = ta[i] - bt[i];
    avgwt += wt[i];
    avgta += ta[i];
}

System.out.println("\npid arrival burst complete turn waiting");
for (int i = 0; i < n; i++) {
    System.out.println(pid[i] + " \t " + ar[i] + "\t" + bt[i] + "\t" + ct[i] + "\t" + ta[i]
+ "\t" + wt[i]);
}

sc.close();

System.out.println("\naverage waiting time: " + (avgwt / n)); // printing average
waiting time.

```

```
        System.out.println("average turnaround time:" + (avgta / n)); // printing average  
turnaround time.
```

```
    }
```

```
}
```

OUTPUT:

enter no of process: 5

enter process 1 arrival time:2

enter process 1 burst time:6

enter process 2 arrival time:5

enter process 2 burst time:2

enter process 3 arrival time:1

enter process 3 burst time:8

enter process 4 arrival time:0

enter process 4 burst time:3

enter process 5 arrival time:4

enter process 5 burst time:4

pid arrival burst complete turn waiting

4	0	3	3	3	0
---	---	---	---	---	---

3	1	8	11	10	2
---	---	---	----	----	---

1	2	6	17	15	9
---	---	---	----	----	---

5	4	4	21	17	13
---	---	---	----	----	----

2	5	2	23	18	16
---	---	---	----	----	----

average waiting time: 8.0

average turnaround time:12.6

2.SJF (Preemptive)

INPUT :

```
import java.util.Scanner;

public class PreemptiveSJF {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

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

        int n = sc.nextInt();

        int pid[] = new int[n];

        int ar[] = new int[n];

        int bt[] = new int[n];

        int ct[] = new int[n];

        int ta[] = new int[n];

        int wt[] = new int[n];

        int f[] = new int[n];

        int k[] = new int[n];

        int temp, tot=0,st=0;

        float avgwt=0,avgta=0;

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

        {

            System.out.println("enter process " + (i+1) + " arrival time: ");

            ar[i] = sc.nextInt();

            System.out.println("enter process " + (i+1) + " burst time: ");

            bt[i] = sc.nextInt();

            k[i] = bt[i];

            pid[i] = i+1;

        }

        while(true){

            int min = 99 , c =n;

            if(tot == n)
```

```

        break;
    for(int i=0;i<n;i++){
        if(ar[i]<=st && f[i] == 0 && bt[i]<min ){
            min = bt[i];
            c = i;
        }
    }
    if (c == n) st++;
    else{
        bt[c]--;
        st++;
        if(bt[c] == 0){
            ct[c] = st;
            f[c] = 1;
            tot++;
        }
    }
}

for(int i=0;i<n;i++){
    ta[i] = ct[i] - ar[i];
    wt[i] = ta[i] - k[i];
    avgwt += wt[i];
    avgta += ta[i];
}

System.out.println("pid arrival burst complete turn waiting");

for(int i=0;i<n;i++)
{
    System.out.println(pid[i] +"\t"+ ar[i]+" \t"+ k[i] +"\t"+ ct[i] +"\t"+ ta[i] +"\t"+ wt[i]);
}

```

```

        System.out.println("\naverage tat is "+ (float)(avgta/n));
        System.out.println("average wt is "+ (float)(avgwt/n));
        sc.close();
    }
}

```

OUTPUT:

```

enter no of process: 5
enter process 1 arrival time: 2
enter process 1 burst time: 6
enter process 2 arrival time: 5
enter process 2 burst time: 2
enter process 3 arrival time: 1
enter process 3 burst time: 8
enter process 4 arrival time:0
enter process 4 burst time:3
enter process 5 arrival time:4
enter process 5 burst time:4
pid  arrival  burst  complete turn waiting
1    2    6    15    13    7
2    5    2    7    2    0
3    1    8    23    22    14
4    0    3    3    3    0
5    4    4    10    6    2
average tat is 9.2
average wt is 4.6

```

3.PRIORITY (Non Preemptive)

INPUT :

```
import java.util.Scanner;

public class Priority {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the number of processes:");

        int n = sc.nextInt();

        int i, pos = 0, temp;

        System.out.println("Enter the burst times:");

        int p[] = new int[n];

        int bt[] = new int[n];

        int pt[] = new int[n];

        int wt[] = new int[n];

        int tat[] = new int[n];

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

            p[i] = i + 1;

            bt[i] = sc.nextInt();

        }

        System.out.println("Enter priority time:");

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

            pt[i] = sc.nextInt();

        }

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

            pos = i;

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

                if (pt[j] < pt[pos]) {

                    pos = j;

                }

            }

        }

    }

}
```

```

        temp = pt[pos];
        pt[pos] = pt[i];
        pt[i] = temp;
        temp = p[pos];
        p[pos] = p[i];
        p[i] = temp;
        temp = bt[pos];
        bt[pos] = bt[i];
        bt[i] = temp;
    }
    wt[0] = 0;
    for (i = 1; i < n; i++) {
        wt[i] = wt[i - 1] + bt[i - 1];
    }
    System.out.println("Process\tBT\tpriority\twaiting time\tTAT");
    for (i = 0; i < n; i++) {
        tat[i] = bt[i] + wt[i];
        System.out.println(p[i] + "\t" + bt[i] + "\t" + pt[i] + "\t\t" + wt[i] + "\t\t" + tat[i]);
    }
}
}

```

OTUPUT :

Enter the number of processes:

7

Enter the burst times:

3

5

4

2

9

4

10

Enter priority time:

2

6

3

5

7

4

10

Process BT			priority	waiting time	TAT
1	3	2	0	3	
3	4	3	3	7	
6	4	4	7	11	
4	2	5	11	13	
2	5	6	13	18	
5	9	7	18	27	
7	10	10	27	37	

4.ROUND (Non Preemptive)

INPUT:

```
import java.util.Scanner;

public class Round {
    int burst[],run[],np,quantom=0,wait[],time=0,rp=0,ta[];
    public Round(){
        Scanner sc=new Scanner(System.in);
        Scanner sc1=new Scanner(System.in);
        System.out.println("Enter the number of pro: ");
        np=sc.nextInt();
        rp=np;
        burst=new int[np];
        run=new int[np];
        wait=new int[np];
        ta=new int[np];
        System.out.println("Enter their burst times: ");
        for(int i=0;i<np;i++){
            burst[i]=sc1.nextInt();
            run[i]=burst[i];
            wait[i]=0;
        }
        System.out.println("Enter Quantum: ");
        quantom=sc.nextInt();
        logic();
    }
    public void logic(){
        int i=0;
        while(rp!=0){
            if(run[i]>quantom){
                run[i]=run[i]-quantom;
```

```

        time=time+quantom;

        System.out.println("Process: "+i+"time: "+time);

    }

    else if(run[i]<=quantom && run[i]>0){

        time=time+run[i];

        run[i]=run[i]-run[i];

        ta[i]=time;

        rp--;

        System.out.println("Process: "+i+"time: "+time);

    }

    i++;

    if(i==np){

        i=0;

    }

}

for(int j=0;j<np;j++){

    wait[j]=ta[j]-burst[j];

    System.out.println(ta[j]);

    System.out.println(" ");

    System.out.println(wait[j]);

}

}

public static void main(String[] args) {

    new Round();

}

}

```

OUTPUT :

Enter the number of pro:

4

Enter their burst times:

5

4

2

1

Enter Quantum:

1

Process: 0time: 1

Process: 1time: 2

Process: 2time: 3

Process: 3time: 4

Process: 0time: 5

Process: 1time: 6

Process: 2time: 7

Process: 0time: 8

Process: 1time: 9

Process: 0time: 10

Process: 1time: 11

Process: 0time: 12

12

7

11

7

7

5

4

3