

OS LAB Exp:4

```
//4.a)
#include<stdio.h>
int main(){
    int n, bt[20], wt[20], tat[20], i, j; float avwt = 0, avtat = 0;
    printf("Enter Total Number of processes (max 20) : ");
    scanf("%d",&n);
    printf("\n Enter Process Burst Time \n");
    for( i = 0; i < n; i++){
        printf("P[%d] : ",i+1);
        scanf("%d" , &bt[i]);
    }
    wt[0] = 0;
    for( i = 1; i < n ; i++){
        wt[i] = 0;
        for ( j = 0; j < i; j++)
            wt[i] += bt[j];
    }
    printf("Gantt chart:\n");
    for(int k=1;k<=n;k++)
    {
        printf("p[%d]___",k);
    }
    printf("\n Process \tBurst Time\tcomp time\tWaiting Time\tTurnaround Time");
    for( i = 0; i < n; i++){
        tat[i] = bt[i] + wt[i];
        avwt += wt[i];
        avtat += tat[i];
        printf("\n P[%d] \t\t\t %d \t\t\t %d\t\t\t %d \t\t\t %d",i+1,bt[i],tat[i],wt[i],tat[i]);
    }
    avwt /= i;
    avtat /= i;
    printf("\n\n Average Waiting Time:%f",avwt); printf("\n Average Turnaround Time:%f \n",avtat);
    return 0;
}
```

Output

Enter Total Number of processes (max 20) : 3

Enter Process Burst Time

P[1] : 3

P[2] : 1

P[3] : 2

Gantt chart:

p[1]___p[2]___p[3]___

Process	Burst Time	comp time	Waiting Time	Turnaround Time
P[1]	3	3	0	3
P[2]	1	4	3	4
P[3]	2	6	4	6

Average Waiting Time:2.333333

Average Turnaround Time:4.333333

```
//4.b)
#include<stdio.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("\n Enter Burst Time:\n");
    for( i = 0; i < n; i++){
        printf("p%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("Gantt Chart:\n");
    for(int k=0;k<=n;k++)
    {
        printf("p[%d]___",p[k]);
    }
    printf("\nProcess\tBurst Time\tcompletion time\tWaiting Time\tTurnaround Time");
    for( i = 0; i < n; i++){
        tat[i] = bt[i] + wt[i];
        total += tat[i];
        printf("\n p%d\t\t\t %d\t\t\t %d\t\t\t %d \t\t\t %d",p[i],bt[i],tat[i],wt[i],tat[i]);
    }
    avg_tat = (float)total / n;
    printf("\n\nAverage Waiting Time=%f",avg_wt); printf("\nAverage TurnaroundTime=%f\n",avg_tat);
}
```

Output

Enter number of process:3

Enter Burst Time:

p1:3

p2:1

p3:2

Gantt Chart:

p[2]___p[3]___p[1]___

Process	Burst Time	completion time	Waiting Time	Turnaround Time
p2	1	1	0	1
p3	2	3	1	3
p1	3	6	3	6

Average Waiting Time=1.333333

Average TurnaroundTime=3.333333

```

int n,i,qt,count=0,temp,sq=0,bt[10],wt[10],tat[10],rem_bt[10];
float awt=0,atat=0;
printf("Enter number of process: ");
scanf("%d",&n);
printf("Enter burst time of process: ");
for(i=0;i<n;i++)
{
    scanf("%d",&bt[i]);
    rem_bt[i]=bt[i];
}
printf("Enter quantum time: ");
scanf("%d",&qt);
printf("Gantt chart\n");
while(1)
{
    for(i=0,count=0;i<n;i++)
    {
        temp=qt;
        if(rem_bt[i]==0)
        {
            count++;
            continue;
        }
        if(rem_bt[i]>qt)
        {
            rem_bt[i]=rem_bt[i]-qt; printf("p[%d] ",i+1);
        }
        else
        if(rem_bt[i]>=0)
        {
            printf("p[%d] ",i+1);
            temp=rem_bt[i];
            rem_bt[i]=0;
        }
        sq=sq+temp;
        tat[i]=sq;
    }
    if(n==count)
        break;
}
printf("\nprocess\tburst time\tcompletion time\tturnaround time\n");
for(i=0;i<n;i++)
{
    wt[i]=tat[i]-bt[i];
    awt=awt+wt[i];
    atat=atat+tat[i];
    printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t",i+1,bt[i],tat[i],wt[i],rem_bt[i],sq,awt,atat);
}
awt=awt/n;
atat=atat/n;
printf("\nAverage waiting time=%f\n",awt);
printf("Average turnaround time=%f",atat);

```

Output

```

Enter number of process: 3
Enter burst time of process: 3
1
2
Enter quantum time: 1
Gantt chart
p[1] p[2] p[3] p[1] p[3] p[1]
process burst time      completion time turnaround time waiting time
1          3            6          6          3
2          1            2          2          1
3          2            5          5          3
Average waiting time=2.33333
Average turnaround time=4.33333

```

// 4.d PROGRAM TO STIMULATE NON-PREEMPTIVE Priority

```
#include<stdio.h>
#include<stdio.h>
#include<stdlib.h>
```

```
typedef struct{
```

```
    int pno;
    int pri;
    int bt;
    int wt;
} sp;
```

```
int main(){
```

```
    int i,j,n;
```

```
    int tbm = 0, totwt = 0, tottat = 0;
```

```
    sp *p,t;
```

```
    printf("\n Enter the Number of Process : ");
```

```
    scanf("%d", &n);
```

```
    p = (sp*) malloc (sizeof(sp));
```

```
    printf("\nEnter the Burst time and priority : \n");
```

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

```
        printf("Process %d : ", i+1);
```

```
        scanf("%d%d", &p[i].bt,&p[i].pri);
```

```
        p[i].pno = i+1;
```

```
        p[i].wt=0;
```

```
    }
```

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

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

```
            if(p[i].pri>p[j].pri){
```

```
                t = p[i];
```

```
                p[i] = p[j];
```

```
                p[j] = t;
```

```
            }
```

```
        }
```

```
        printf("\nProcess\tBurst Time\tWaiting Time\tcompletion time\tTurnaround Time\n");
```

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

```
            totwt += p[i].wt=tbm;
```

```
            tbm += p[i].bt;
```

```
            printf("\n%d\t\t\t\t\t",p[i].pno,p[i].bt);
```

```
            printf("\t\t\t\t\t",p[i].wt,p[i].wt+p[i].bt,p[i].wt+p[i].bt);
```

```
        }
```

```
        tottat = tbm + totwt;
```

```
        printf("\nGantt chart:\n");
```

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

```
            {
```

```
                printf("p[%d]_",p[i].pno);
```

```
            }
```

```
            printf("\n Average Waiting time:%f", (float)totwt/n);
```

```
            printf("\n Avg Turnaround time:%f", (float)tottat/n);
```

```
        return 0;
```

```
    }
```

Output

```
Enter the Number of Process : 3
```

```
Enter the Burst time and priority :
```

```
Process 1 : 3
```

```
2
```

```
Process 2 : 1
```

```
1
```

```
Process 3 : 2
```

```
3
```

Process	Burst Time	Waiting Time	completion time	Turnaround Time
---------	------------	--------------	-----------------	-----------------

2	1	0	1	1
---	---	---	---	---

1	3	1	4	4
---	---	---	---	---

3	2	4	6	6
---	---	---	---	---

```
Gantt chart:
```

```
p[2]_p[1]_p[3]_
```

```
Average Waiting time:1.666667
```

```
Avg Turnaround time:3.666667
```