

F^o First Come first Serve

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    int totalp, i, tWT=0, tTAT=0;
    float awt, atat, tat;
    int BT[4];
    int WT[4];
    int TAT[i];
    i=0;
    WT[i]=0;
    totalp = 4;
    printf (" enter the number of processes \n");
    Scanf ("%d", &totalp);
    printf (" enter the burst time of max. 4 processes \n");
    for (i=0; i<totalp; i++)
    {
        Scanf ("%d", &BT[i]);
    }
    i=0;
    printf (" Waiting time for all processes are \n");
    printf (" p[%d] = %d \n", i, WT[i]);
    for (i=0; i<totalp; i++)
    {
        WT[i] = WT[i-1] + BT[i-1];
        printf (" p[%d] = %d \n", i, WT[i]);
    }
    for (i=0; i<totalp; i++) {
```

```
twt = twt + WT[i];  
}  
awt = (float) twt / totalp;  
printf ("total waiting time = %d\n", twt);  
printf ("average waiting time = %f\n", awt);  
printf ("turnaround time for all processes are 1h");  
for (i=0; i<totalp; i++)  
{  
    TAT[i] = BT[i] + WT[i];  
    printf ("p[%d] = %d\n", i, TAT[i]);  
}  
for (i=0; i<totalp; i++)  
{  
    ttat = ttat + TAT[i];  
}  
atot = ttat / Totalp;  
printf ("total turnaround time = %d\n", ttat);  
printf ("average turnaround time = %f\n", atot);  
getch();  
}
```

Output:- Enter the number of processes

4

Enter the burst time of Max. 4 processes

3

4

2

7

Waiting time for all processes are

p[0]=0

p[1]=0

$p[0] = 3$

$p[1] = 7$

$p[2] = 9$

total waiting time = 19

Average waiting time = 4.750000

turnaround time for all processes are

$p[0] = 3$

$p[1] = 7$

$p[2] = 9$

$p[3] = 16$

total turnaround time = 35

average turnaround time = 8.000000

II Short Job first

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

```
Struct PCB {
```

```
    int burst, pid, wait, turnaround;
```

```
};
```

```
Void pline (int);
```

```
Void main ()
```

```
{
```

```
Struct PCB p[10], temp;
```

```
int i, n, j=1, sum=0, w_total=0, t_total=0;
```

```
float w_avg=0.0, t_avg=0.0;
```

```
printf ("SJF Algorithm \n");
```

```
printf ("Enter the total number of processes : ");
```

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

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

```
{
```

```
printf ("Enter the burst time of process %d : %h", i+1);
```

```
Scanf ("%d", &p[i].burst);
```

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

```
}
```

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

```
{
```

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

```
{
```

```
if (p[j].burst > p[j+1].burst)
```

```
{
```

```
temp = p[j];
```

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

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

```
}
```

```
}
```

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

```
{
```

```
p[i].wait = sum;
```

```
sum = sum + p[i].burst;
```

```
p[i].turnaround = sum;
```

```
}
```

```
pline (35);
```

```
printf ("PID \t Burst \t wait \t Turnaround");
```

```
pline (35);
```

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

```
{
```

```
printf ("%d \t %d \t %d \t %d \t %d", p[i].pid, p[i].burst,
```

```
p[i].wait, p[i].turnaround);
```

```
total += p[i].wait;
```

```
total += p[i].turnaround;
```

{

$$w_avg = w_total / (\text{float})n;$$

$$t_avg = t_total / (\text{float})n;$$

pointf ("In Total waiting Time : %d", w-total);

pointf ("In Average Waiting Time : %.3f", w-avg);

pointf ("In Total Turnaround Time : %d", t-total);

pointf ("In Average Turnaround Time : %.3f", t-avg);

{

void pline (int x)

{

int i;

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

{

pointf (" - ");

{

pointf ("\n");

Output: Enter the total number of processes: 3

Enter the burst time of process 1:

8

Enter the burst time of process 2:

13

Enter the burst time of process 3:

3

PID	Burst	Wait	Turnaround
3	3	0	3
1	8	3	11
2	13	11	24

Total Waiting Time: 14

Average Waiting Time: 4.667

Total Turnaround Time: 38
Average Turnaround Time: 12.667

Priority Scheduling

```
#include <iostream>
#include <float.h>
#include <stdlib.h>
```

```
using namespace std;
typedef struct
```

```
{ int pno;
    int pri;
    int btime;
    int wtime;
}
```

```
Sp;
int main()
{
```

```
    int i, j, n;
    int tbm = 0, totWTime = 0, totTime = 0;
    Sp* p, t;
```

```
Cout << "In PRIORITY SCHEDULING. In";
```

```
Cout << "In enter the no of process-- In";
```

```
Cin >> n;
```

```
p = (Sp*) malloc(sizeof(SP));
```

```
Cout << " enter the burst time and priority : In";
```

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

{ cout << "process" << i+1 << ":";

(in >> p[i].btime >> p[i].pri;

p[i].pno = i+1;

p[i].wtime = 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;

}

}

(out << "In process \t burst time \t waiting time \t turnaround time \n";

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

{

totwtime += p[i].wtime = tbm;

tbm += p[i].btime;

(out << "\n" << p[i].pno << " \t \t " << p[i].btime;

(out << " \t \t " << p[i].wtime << " \t \t " << p[i].wtime + p[i].btime;

}

tottime = tbm + totwtime;

(out << "In total waiting time :" << totwtime;

(out << "In average waiting time :" << (float) totwtime / n;

(out << "In total turnaround time :" << tottime;

(out << "In avg turnaround time :" << (float) tottime / n;

}

Output:

Enter the number of process

2 enter the burst time and priority:
process 1:3

4

process	burstime	waiting time	turnaround time
1	3	0	$\frac{3}{4}$
total waiting time: 3		3	
average waiting time: 1.5			
total turnaround time: 3			
avg. turnaround time: 1.5			

Round Robin Scheduling

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    int n, i, qt, Count = 0, temp, Sq = 0, bt[10], wt[10],
        tat[10], sum = bt[10];
    float awt = 0, atat = 0;
    clrscr();
    printf ("Enter number of process");
    Scanf ("%d", &n);
    printf ("Enter burst time of process");
    for (i=0; i < n; i++)
    {
        Scanf ("%d", &bt[i]);
    }
}
```

grem - bt[i] = bt[i];

}

printf ("Enter Quantum time");

scanf ("%d", &qt);

while (1)

{

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

{

temp = qt;

if (grem - bt[i] == 0)

{

Count ++;

Continue;

}

if (grem - bt[i] > qt)

grem - bt[i]

else

{ if (grem - bt[i] >= 0)

temp = grem - bt[i];

grem - bt[i] = 0;

}

Sq = Sq + temp;

tat[i] = Sq;

}

if (n == Count)

break;

}

printf ("In process/t burst time/t turnaround time

/t waiting time /n");

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

$$wt[i] = tat[i] - bt[i];$$

$$awt = awt + wt[i];$$

$$atat = atat + tat[i];$$

```
printf (" %d %d %d %d ", i+1, bt[i], tat[i],  
       wt[i]);
```

{

$$awt = awt/n;$$

$$atat = atat/n;$$

```
printf (" Average Waiting Time = %.2f \n", awt);
```

```
printf (" Average Turnaround Time = %.2f ", atat);
```

```
getch();
```

{}

Output:

Enter number of process: 4

Enter burst time of process 5 3 1 4

Enter quantum time 2

process	bust time	turnaround time	Waiting time
1	5	13	8
2	3	10	7
3	1	5	4
4	4	12	8

Average Waiting Time = 6.750000

Average Turnaround Time = 10.000000