

write a C program  
CPU scheduling algorithm to find the  
turnaround time and waiting time.  
→ Priority (non-preemptive)  
→ Round Robin

① #include <stdio.h>  
#include <conio.h>

```
int main ()  
{  
    int n, i, bt[20], priority[20], at[20], j,  
    temp, wt[20], tat[20], sum = 0, avgwt = 0,  
    avgtat = 0;  
    printf ("Enter number of processes: ");  
    scanf ("%d", &n);  
  
    for (i = 0; i < n; i++)  
    {  
        printf ("Enter arrival time for P[%d]: ",  
                i + 1);  
        scanf ("%d", &at[i]);  
        printf ("Enter burst time for P[%d]: ",  
                i + 1);  
        scanf ("%d", &bt[i]);  
        printf ("Enter priority of P[%d]: ", i + 1);  
        scanf ("%d", &priority[i]);  
    }  
}
```

```

for (i=0; i<n-1; i++)
{
    for (j=0; j<n-1-i; j++)
    {
        if (priority[j] > priority[j+1])
        {
            temp = priority[j];
            priority[j] = priority[j+1];
            priority[j+1] = temp;
            temp = bt[j];
            bt[j] = bt[j+1];
            bt[j+1] = temp;
            temp = at[j];
            at[j] = at[j+1];
            at[j+1] = temp;
        }
    }
}

for (i=0; i<n; i++)
{
    wt[i] = sum + at[i];
    tat[i] = wt[i] + bt[i];
    printf ("%d/n", wt[i]);
    avgwt += wt[i];
    sum += bt[i];
}

```

~~float avgwtf = (float) avgwt / n;~~  
~~float avgtatf = (float) avgtat / n;~~  
 printf ("In Total average waiting  
 time : %f", avgwtf);  
 printf ("In Total average turnaround  
 time : %f", avgtatf);

getch ()

return 0;

}

### Output

Enter number of processes : 4

Enter arrival time for P[1] : 0

Enter burst time for P[1] : 4

Enter priority of P[1] : 3

Enter arrival time for P[2] : 1

Enter burst time for P[2] : 3

Enter priority of P[2] : 4

Enter arrival time for P[3] : 2

Enter burst time for P[3] : 3

Enter priority of P[3] : 6

Enter arrival time for P[4] : 3

Enter burst time of P[4] : 5

Enter priority of P[4] : 5

0

3

4

10

Total average waiting time : 4.25

Total average turnaround time : 8.0

Program At CPU-T Priority

A	0	4	3
B	1	3	4
C	2	3	6
D	3	5	5

A	A	A	A	C	D	B
0	1	2	3	4	7	12
A(3)	A(2)	A(1)	B(4)	B(6)	B(5)	
B(4)	B(3)	B(2)	(6)	D(5)		
C(6)	(16)	D(5)				
D(5)						

$$TAT = 4, 14, 5, 9$$

$$\Rightarrow ATAT = \underline{8}$$

$$WT = 0, 11, 2, 4$$

$$AWT = \frac{17}{4} = 4.25$$

```

② #include <stdio.h>
#include <stdlib.h>
int main ()
{
    int n, i, bt[10], qbt[10], qvt, stz[10],
        tat[10], wt[10], at[10];
    float atatzo, awt = 0;
    printf ("Enter the number of processes");
    scanf ("%d", &n);
    printf ("Enter burst time of the processes");
    for (i=0; i<n; i++)
    {
        scanf ("%d", &bt[i]);
        qbt[i] = bt[i];
    }
    printf ("Enter the arrival time of the processes");
    for (i=0; i<n; i++)
    {
        scanf ("%d", &at[i]);
    }
    printf ("Enter the time quantum");
    scanf ("%d", &qvt);
    while (1)
    {
        int donezo;
        for (i=0; i<n; i++)
        {
            if (qbt[i]>0)
            {
                donezo;
            }
        }
    }
}

```

if ( $gbt[i] > qt$ )

{  $gbt[i] = qt;$

$st += qt;$

y

else

{  $st += gbt[i];$

$wt[i] = st - bt[i] - at[i];$

$gbt[i] \geq 0;$

$tat[i] = st - at[i];$

}

}

} if (done)

break;

y printing ("In Process Burst Time + turnaround  
time (" + n + " processes)");  
time if waiting-time(");

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

{ printing ("Burst Time + Turnaround Time = ", i+1, "

{ printing ("Waiting Time + Turnaround Time = ", i+1, "

$bt[i], tat[i], wt[i]);$

$awt += wt[i];$

$atat += tat[i];$

}

$awt / n = w$

$atat / n = a$

printing ("Average Turnaround time :  
", awt / n, ", average waiting time :  
", atat / n, ");

printing ("Average waiting time :  
", awt / n, ");

Output:

Enter the number of processes: 5

Enter the burst time of processes: 5 3 1 2 3

Enter the arrival time of the process:

0  
1  
2  
3  
4

Enter the time quantum: 2

Process      Burst time      Turnaround time      Waiting time

1	5	14	9
2	3	11	8
3	1	3	2
4	2	4	2
5	3	9	6

Average turnaround time: 8.2

Average waiting time: 3.84

TQ 22

	AT	BT
P1	0	5
P2	1	3
P3	2	1
P4	3	2
P5	4	3

P1	P2	P3	P1	P4	P5	P2	P1	P5
0	2	4	5	7	9	11	12	13
P2	P3	P1	P1	P4	P5	P2	P1	P5
P1	P1	P2	P4	P5	P2	P1	P1	P5
P3	P4	P5	P5	P2	P1	P5	P1	P5
P1	P5	P2	P2	P1	P1	P1	P1	P5
P2								

$\text{ATAT}_2 = \frac{43}{5} = 8.6$   
 $\text{ATAT}_2 = 11, 3, 4, 9 \Rightarrow$   
 $\text{AWT} = \frac{29}{5} = 5.8$   
 $\text{WT} = 1, 8, 2, 6 \Rightarrow$

```
Enter the number of processes: 5
Enter the burst time of the processes: 5
3
1
2
3
Enter the arrival time of the processes: 0
1
2
3
4
Enter the time quantum: 2

Process Burst Time      Turnaround Time Waiting Time
1      5                14                  9
2      3                11                  8
3      1                3                   2
4      2                4                   2
5      3                9                   6

Average Turnaround Time: 8.20
Average Waiting Time: 5.40

Process returned 0 (0x0)  execution time : 20.260 s
Press any key to continue.
```

```
Enter number of processors: 4
Enter arrival time for p[1]: 0
Enter burst time for p[1]: 4
Enter priority of p[1]: 3
Enter arrival time for p[2]: 1
Enter burst time for p[2]: 3
Enter priority of p[2]: 4
Enter arrival time for p[3]: 2
Enter burst time for p[3]: 3
Enter priority of p[3]: 6
Enter arrival time for p[4]: 3
Enter burst time for p[4]: 5
Enter priority of p[4]: 5
0
3
4
10

Total average waiting time: 4.250000
Total average turnaround time: 8.000000
```