

Name → Akansha Katuri
 Course → BSC.IT
 Roll No. → 2023109

```

1 #include <stdio.h>
  int waiting_time (int proc[], int n, int burst_time[],
                    int wait_time[])
{
    wait_time[0] = 0;
    for (int i = 1; i < n; i++)
        wait_time[i] = burst_time[i-1] + wait_time[i-1];
    return 0;
}

  int turn_around_time (int proc[], int n, int burst_time[],
                        int wait_time[], int tat[])
{
    int i;
    for (i = 0; i < n; i++)
        tat[i] = burst_time[i] + wait_time[i];
    return 0;
}

  int avg_time (int proc[], int n, burst_time[])
{
    int wait_time[n], tat[n], total_wt = 0,
        total_tat = 0;
    int i;
    waiting_time(proc, n, burst_time, wait_time);
    turn_around_time(proc, n, burst_time, wait_time, tat);
    printf("Processes burst waiting Turn around\n");
    for (i = 0; i < n; i++)
        total_wt = total_wt + wait_time[i];

```



```
total_tat = total_tat + tat[i];  
printf ("%d\t.%d\t\t %d\t %d\n", it  
burst_time[i], wait_time[i], tat[i]);
```

```
printf("Average waiting time = %f\n", (float)
//float)n);
```

```
printf("Average turn around time = %f\n",  
(float) total_tat / (float) n);
```

2. Settlement 00

ent main ()

ent proc $[\]_0 = \{1, 2, 3\}$

$$ent + n = \text{size of proc} / \text{size of proc}[0]$$

ent burst Dtime[] = {5, 8, 12, 4}

avg time (proc, n, burst time)

return 0;

2

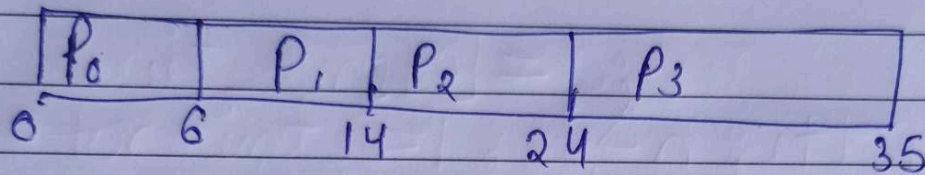
Compile Result

Processes around	Burst	Waiting	Turn
1	5	0	5
2	8	5	1
3			
3	12	13	2
5			

Average waiting time = 6.000000
Average turn around time = 14.33333

[Process completed - press Enter]

Process	A.T	CPU T
P ₀	0	6
P ₁	1	8
P ₂	2	10
P ₃	3	11



Turn Around time — $CA - AT$
 W.T

$$W.T = TAT - B.T$$

Process	A.T	CPU.T	TAT	W.T
P ₀	0	6	6	0
P ₁	1	8	7	1
P ₂	2	10	8	2
P ₃	3	11	8	3