

main.c



Share

Run

Output

```
1 #include <stdio.h>
2 int main() {
3     int n, i;
4     printf("Enter the number of processes: ");
5     scanf("%d", &n);
6     int burst_time[n], completion_time[n], turnaround_time[n], waiting_time[n];
7     printf("Enter Burst Times for each process:\n");
8     for(i = 0; i < n; i++) {
9         printf("Process %d Burst Time: ", i + 1);
10        scanf("%d", &burst_time[i]);
11    }
12    completion_time[0] = burst_time[0];
13    for(i = 1; i < n; i++) {
14        completion_time[i] = completion_time[i - 1] + burst_time[i];
15    }
16    for(i = 0; i < n; i++) {
17        turnaround_time[i] = completion_time[i];
18        waiting_time[i] = turnaround_time[i] - burst_time[i];
19    }
20    printf("\nProcess\tBurst Time\tCompletion Time\tTurnaround Time\tWaiting Time\n");
21    for(i = 0; i < n; i++) {
22        printf("%d\t%d\t\t%d\t\t%d\t\t%d\n", i + 1, burst_time[i], completion_time[i],
23            turnaround_time[i], waiting_time[i]);
24    }
25    return 0;
26 }
```

Enter the number of processes: 2

Enter Burst Times for each process:

Process 1 Burst Time: 20

Process 2 Burst Time: 10

Process	Burst Time	Completion Time	Turnaround Time	Waiting Time
1	20	20	20	0
2	10	30	30	20

=== Code Execution Successful ===

mcin.c



Share

Run

Output

```
1 #include <stdio.h>
2 int main() {
3     int n, i, j;
4     printf("Enter the number of processes: ");
5     scanf("%d", &n);
6     int burst_time[n], priority[n], process[n], completion_time[n], turnaround_time[n], waiting_time[n];
7     printf("Enter Burst Times and Priorities for each process:\n");
8     for (i = 0; i < n; i++) {
9         printf("Process %d Burst Time: ", i + 1);
10        scanf("%d", &burst_time[i]);
11        printf("Process %d Priority: ", i + 1);
12        scanf("%d", &priority[i]);
13        process[i] = i + 1;
14    }
15    for (i = 0; i < n - 1; i++) {
16        for (j = 0; j < n - i - 1; j++) {
17            if (priority[j] > priority[j + 1]) {
18                int temp = priority[j];
19                priority[j] = priority[j + 1];
20                priority[j + 1] = temp;
21                temp = burst_time[j];
22                burst_time[j] = burst_time[j + 1];
23                burst_time[j + 1] = temp;
24                temp = process[j];
25                process[j] = process[j + 1];
26                process[j + 1] = temp;
27            }
28        }
29    }
30    completion_time[0] = burst_time[0];
31    for (i = 1; i < n; i++) {
32        completion_time[i] = completion_time[i - 1] + burst_time[i];
33    }
34    for (i = 0; i < n; i++) {
35        turnaround_time[i] = completion_time[i];
36        waiting_time[i] = turnaround_time[i] - burst_time[i];
37    }
38    printf("\nProcess\tPriority\tBurst Time\tCompletion Time\tTurnaround Time\tWaiting Time\n");
39    for (i = 0; i < n; i++) {
40        printf("%d\t%d\t%d\t%d\t%d\t%d\n", process[i], priority[i], burst_time[i],
41            completion_time[i], turnaround_time[i], waiting_time[i]);
42    }
43    return 0;
44 }
```

```
Enter the number of processes: 2
Enter Burst Times and Priorities for each process:
Process 1 Burst Time: 5
Process 1 Priority: 2
Process 2 Burst Time: 3
Process 2 Priority: 1

Process Priority    Burst Time    Completion Time    Turnaround Time    Waiting Time
2 1 3 3 3 0
1 2 5 8 8 3

=== Code Execution Successful ===
```