

main.c

Share

Run

```
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 }
```

Output

```
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 ===
```

main.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], process[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        process[i] = i + 1;
12    }
13    for (i = 0; i < n - 1; i++) {
14        for (j = 0; j < n - i - 1; j++) {
15            if (burst_time[j] > burst_time[j + 1]) {
16                int temp = burst_time[j];
17                burst_time[j] = burst_time[j + 1];
18                burst_time[j + 1] = temp;
19                temp = process[j];
20                process[j] = process[j + 1];
21                process[j + 1] = temp;
22            }
23        }
24    }
25    completion_time[0] = burst_time[0];
26    for (i = 1; i < n; i++) {
27        completion_time[i] = completion_time[i - 1] + burst_time[i];
28    }
29    for (i = 0; i < n; i++) {
30        turnaround_time[i] = completion_time[i];
31        waiting_time[i] = turnaround_time[i] - burst_time[i];
32    }
33    printf("\nProcess\tBurst Time\tCompletion Time\tTurnaround Time\tWaiting Time\n");
34    for (i = 0; i < n; i++) {
35        printf("%d\t%d\t\t%d\t\t%d\t\t\t%d\n", process[i], burst_time[i], completion_time[i],
36            turnaround_time[i], waiting_time[i]);
37    }
38    return 0;
39 }
```

Enter the number of processes: 2
Enter Burst Times for each process:
Process 1 Burst Time: 30
Process 2 Burst Time: 20

Process	Burst Time	Completion Time	Turnaround Time	Waiting Time
2	20	20	20	0
1	30	50	50	20

=== Code Execution Successful ===
=== Session Ended. Please Run the code again ===