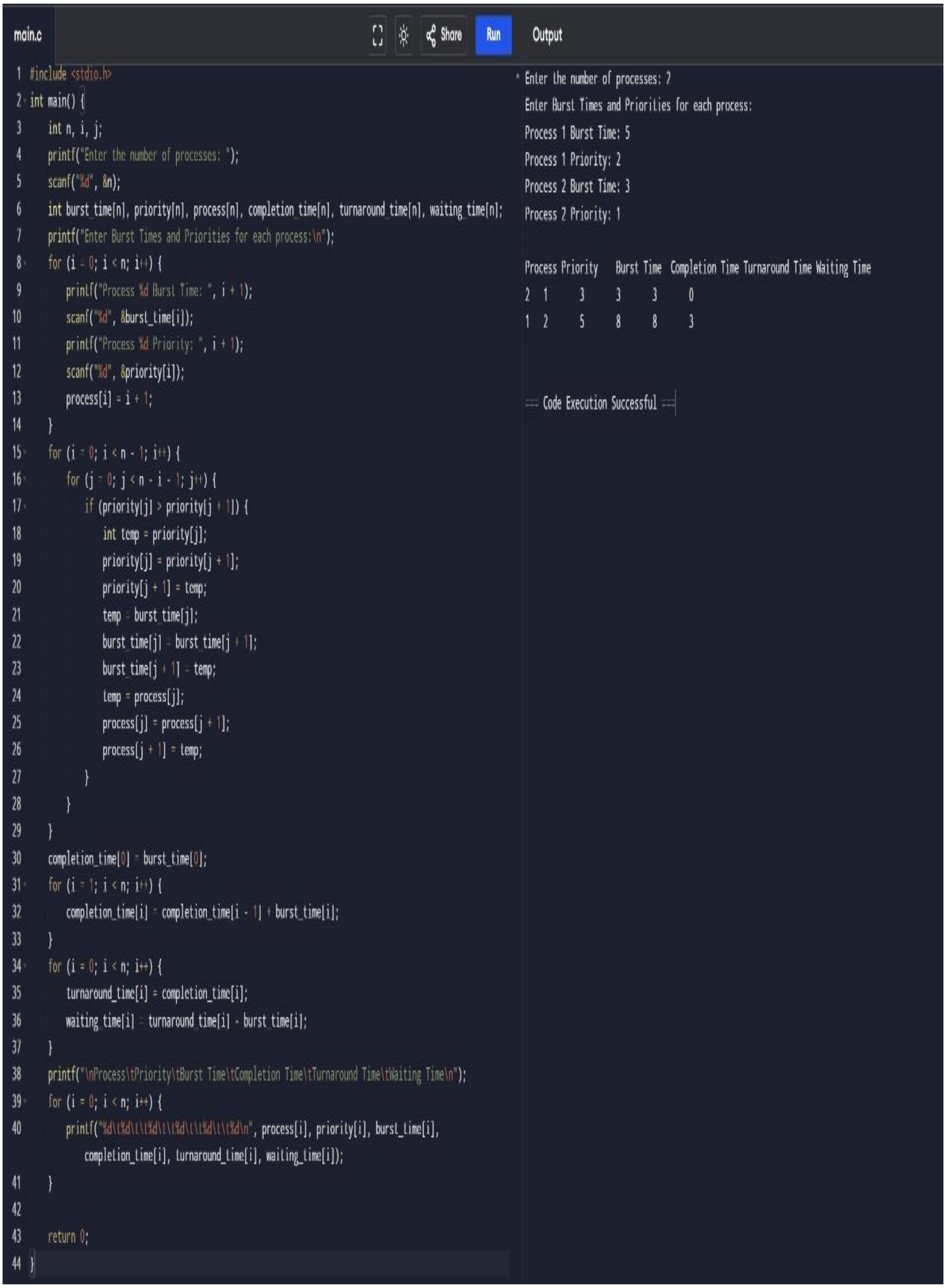


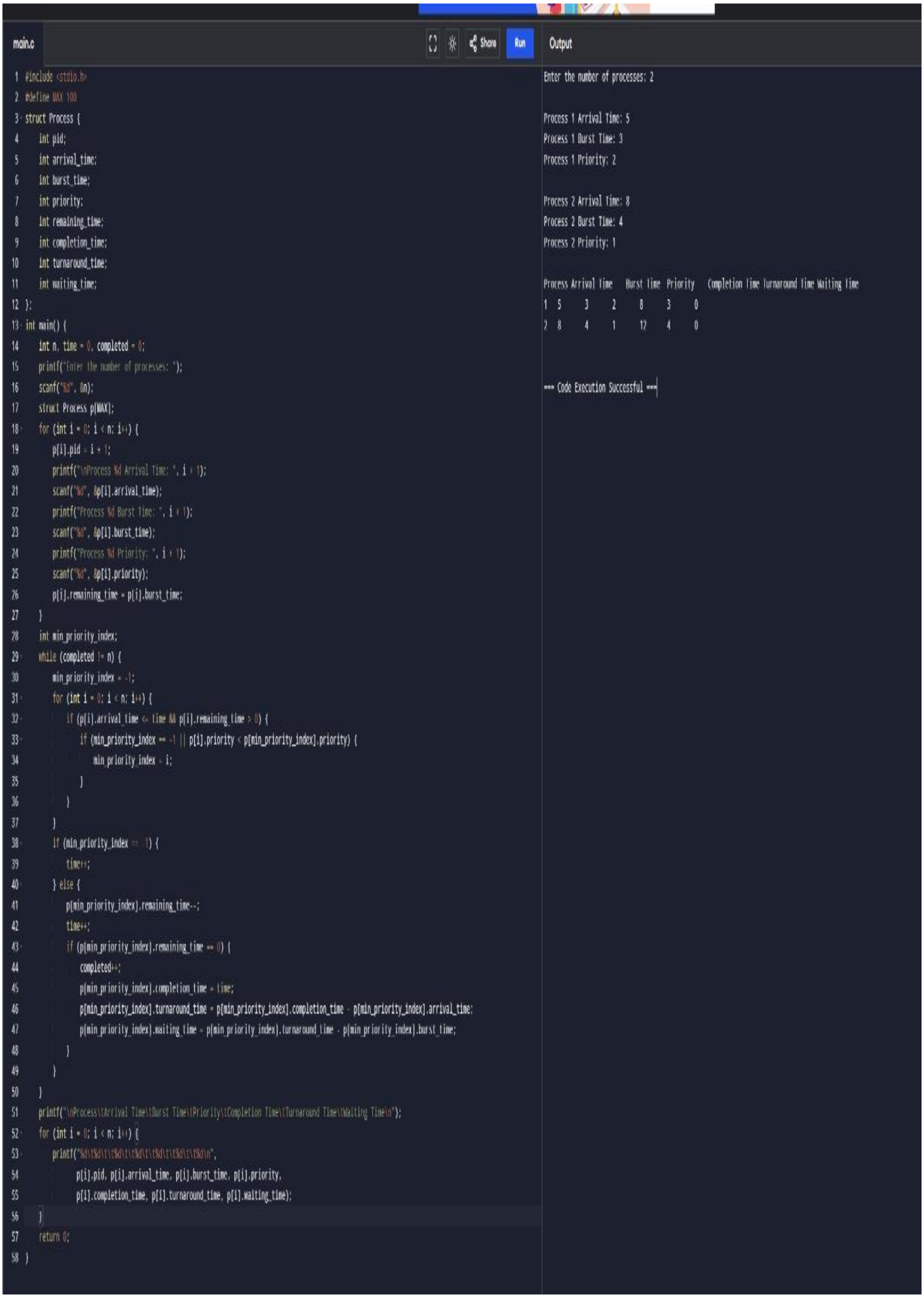
```c
#include <stdio.h>
int main() {
    int n, i, j;
    printf("Enter the number of processes: ");
    scanf("%d", &n);
    int burst_time[n], priority[n], process[n], completion_time[n], turnaround_time[n], waiting_time[n];
    printf("Enter Burst Times and Priorities for each process:\n");
    for (i = 0; i < n; i++) {
        printf("Process %d Burst Time: ", i + 1);
        scanf("%d", &burst_time[i]);
        printf("Process %d Priority: ", i + 1);
        scanf("%d", &priority[i]);
        process[i] = i + 1;
    }
    for (i = 0; i < n - 1; i++) {
        for (j = 0; j < n - i - 1; j++) {
            if (priority[j] > priority[j + 1]) {
                int temp = priority[j];
                priority[j] = priority[j + 1];
                priority[j + 1] = temp;
                temp = burst_time[j];
                burst_time[j] = burst_time[j + 1];
                burst_time[j + 1] = temp;
                temp = process[j];
                process[j] = process[j + 1];
                process[j + 1] = temp;
            }
        }
    }
    completion_time[0] = burst_time[0];
    for (i = 1; i < n; i++) {
        completion_time[i] = completion_time[i - 1] + burst_time[i];
    }
    for (i = 0; i < n; i++) {
        turnaround_time[i] = completion_time[i];
        waiting_time[i] = turnaround_time[i] - burst_time[i];
    }
    printf("\nProcess\tPriority\tBurst Time\tCompletion Time\tTurnaround Time\tWaiting Time\n");
    for (i = 0; i < n; i++) {
        printf("%d\t%d\t\t%d\t\t%d\t\t%d\t\t%d\n", process[i], priority[i], burst_time[i],
            completion_time[i], turnaround_time[i], waiting_time[i]);
    }

    return 0;
}
```

```
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 ===
```



```c
#include <stdio.h>
#define MAX 100
struct Process {
    int pid;
    int arrival_time;
    int burst_time;
    int priority;
    int remaining_time;
    int completion_time;
    int turnaround_time;
    int waiting_time;
};
int main() {
    int n, time = 0, completed = 0;
    printf("Enter the number of processes: ");
    scanf("%d", &n);
    struct Process p[MAX];
    for (int i = 0; i < n; i++) {
        p[i].pid = i + 1;
        printf("\nProcess %d Arrival Time: ", i + 1);
        scanf("%d", &p[i].arrival_time);
        printf("Process %d Burst Time: ", i + 1);
        scanf("%d", &p[i].burst_time);
        printf("Process %d Priority: ", i + 1);
        scanf("%d", &p[i].priority);
        p[i].remaining_time = p[i].burst_time;
    }
    int min_priority_index;
    while (completed != n) {
        min_priority_index = -1;
        for (int i = 0; i < n; i++) {
            if (p[i].arrival_time <= time && p[i].remaining_time > 0) {
                if (min_priority_index == -1 || p[i].priority < p[min_priority_index].priority) {
                    min_priority_index = i;
                }
            }
        }
        if (min_priority_index == -1) {
            time++;
        } else {
            p[min_priority_index].remaining_time--;
            time++;
            if (p[min_priority_index].remaining_time == 0) {
                completed++;
                p[min_priority_index].completion_time = time;
                p[min_priority_index].turnaround_time = p[min_priority_index].completion_time - p[min_priority_index].arrival_time;
                p[min_priority_index].waiting_time = p[min_priority_index].turnaround_time - p[min_priority_index].burst_time;
            }
        }
    }
    printf("\nProcess\tArrival Time\tBurst Time\tPriority\tCompletion Time\tTurnaround Time\tWaiting Time\n");
    for (int i = 0; i < n; i++) {
        printf("%d\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\n",
               p[i].pid, p[i].arrival_time, p[i].burst_time, p[i].priority,
               p[i].completion_time, p[i].turnaround_time, p[i].waiting_time);
    }
    return 0;
}
```

```
Enter the number of processes: 2

Process 1 Arrival Time: 5
Process 1 Burst Time: 3
Process 1 Priority: 2

Process 2 Arrival Time: 8
Process 2 Burst Time: 4
Process 2 Priority: 1

Process Arrival Time    Burst Time  Priority    Completion Time Turnaround Time Waiting Time
1   5       3       2       8       3       0
2   8       4       1       12      4       0


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