Programiz C Online Compiler

main.c

Share Run

```c
#include <stdio.h>
#include <stdlib.h>
#define MAX_PROCESSES 10
typedef struct {
    int id;
    int priority;
    int burst_time;
    int waiting_time;
} Process;

void calculateWaitingTime(Process processes[], int n) {
    int total_waiting_time = 0;
    for (int i = 0; i < n; i++) {
        total_waiting_time += processes[i].waiting_time;
    }
    printf("Total Waiting Time: %d\n", total_waiting_time);
}

void scheduleProcesses(Process processes[], int n) {
    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            if (processes[i].priority < processes[j].priority) {
                Process temp = processes[i];
                processes[i] = processes[j];
                processes[j] = temp;
            }
        }
```

Output

```
Enter number of processes: 1
Enter Process ID, Priority, and Burst Time for Process 1: 2,6,7
Scheduled Processes:
Process ID: 2, Priority: 832, Burst Time: 832
Total Waiting Time: 0


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

Programiz C Online Compiler

main.c

Share Run

```c
#include <stdio.h>
#include <stdlib.h>
struct Process {
    int id;
    int burst_time;
    int priority;
};
void sortProcesses(struct Process proc[], int n) {
    struct Process temp;
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (proc[j].priority > proc[j + 1].priority) {
                temp = proc[j];
                proc[j] = proc[j + 1];
                proc[j + 1] = temp;
            }
        }
    }
}
void findWaitingTime(struct Process proc[], int n, int waiting_time[]) {
    waiting_time[0] = 0;
    for (int i = 1; i < n; i++) {
        waiting_time[i] = proc[i - 1].burst_time + waiting_time[i - 1];
    }
}
void findTurnAroundTime(struct Process proc[], int n, int waiting_time[], int
    turn_around_time[]) {
```

Output

```
Process ID  Burst Time  Priority    Waiting Time    Turnaround Time
2       5       1       0       5
1       10      2       5       15
3       8       3       15      23


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