main.c

Share Run

```c
#include <stdio.h>
struct Process {
    int id;
    int burst_time;
    int waiting_time;
    int turnaround_time;
};
void findWaitingTime(struct Process proc[], int n) {
    proc[0].waiting_time = 0;
    for (int i = 1; i < n; i++) {
        proc[i].waiting_time = proc[i - 1].burst_time + proc[i - 1]
            .waiting_time;
    }
}
void findTurnaroundTime(struct Process proc[], int n) {
    for (int i = 0; i < n; i++) {
        proc[i].turnaround_time = proc[i].burst_time + proc[i]
            .waiting_time;
    }
}
void findavgTime(struct Process proc[], int n) {
    findWaitingTime(proc, n);
    findTurnaroundTime(proc, n);
    float total_waiting_time = 0, total_turnaround_time = 0;
    for (int i = 0; i < n; i++) {
        total_waiting_time += proc[i].waiting_time;
        total_turnaround_time += proc[i].turnaround_time;
```

Output

```
Average waiting time: 7.25
Average turnaround time: 12.75


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

Programiz C Online Compiler

main.c

Share Run

```c
#include <stdio.h>
#include <stdlib.h>
#define MAX_PROCESSES 100
typedef struct {
    int id;
    int execution_time;
    int waiting_time;
} Process;
void sortProcesses(Process processes[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (processes[j].execution_time > processes[j + 1].execution_time) {
                Process temp = processes[j];
                processes[j] = processes[j + 1];
                processes[j + 1] = temp;
            }
        }
    }
}
void scheduleProcesses(Process processes[], int n) {
    sortProcesses(processes, n);
    int total_waiting_time = 0;
    printf("Process ID\tExecution Time\tWaiting Time\n");
    for (int i = 0; i < n; i++) {
        if (i > 0) {
            processes[i].waiting_time = processes[i - 1].waiting_time +
                processes[i - 1].execution_time;
```

Output

```
Enter number of processes: 3
Enter execution time for process 1: 4
Enter execution time for process 2: 5
Enter execution time for process 3: 6
Process ID  Execution Time  Waiting Time
1       4       0
2       5       4
3       6       9
Average Waiting Time: 4.33


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