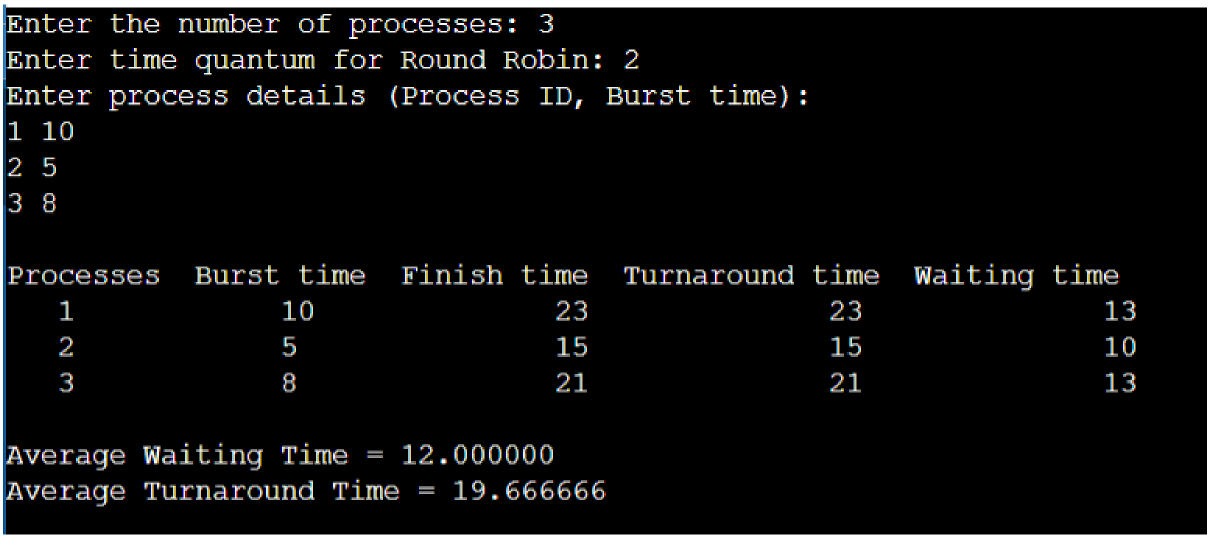
```
#include <stdio.h>
#include <stdlib.h>

struct Process
{
    int pid;      // process ID
    int bt;       // CPU burst time
    int remaining; // remaining burst time
    int finish;    // finish time
    int turnaround; // turnaround time
    int waiting;   // waiting time
};

void findWaitingTurnaroundTime(struct Process proc[], int n, int quantum)
{
    int time = 0;
    int remaining_processes = n;

    while (remaining_processes > 0)
    {
        for (int i = 0; i < n; i++)
        {
            if (proc[i].remaining > 0)
            {
                if (proc[i].remaining <= quantum)
                {
                    time += proc[i].remaining;
                    proc[i].remaining = 0;
                    proc[i].finish = time;
                    remaining_processes--;
                }
                else
                {
                    time += quantum;
                    proc[i].remaining -= quantum;
                }
            }
        }
    }

    // Calculate turnaround time and waiting time
    for (int i = 0; i < n; i++)
    {
        proc[i].turnaround = proc[i].finish;
        proc[i].waiting = proc[i].turnaround - proc[i].bt;
    }
}

void displayResults(struct Process proc[], int n)
{
    printf("\nProcesses  Burst time  Finish time  Turnaround time  Waiting time\n");

    for (int i = 0; i < n; i++)
    {
        printf("   %d\t\t%d\t\t%d\t\t%d\t\t%d\n", proc[i].pid, proc[i].bt, proc[i].finish,
proc[i].turnaround, proc[i].waiting);
    }
}

void calculateAverages(struct Process proc[], int n)
{
    int total_waiting = 0, total_turnaround = 0;
    for (int i = 0; i < n; i++)
    {
        total_waiting += proc[i].waiting;
        total_turnaround += proc[i].turnaround;
    }

    float avg_waiting = (float)total_waiting / (float)n;
    float avg_turnaround = (float)total_turnaround / (float)n;

    printf("\nAverage Waiting Time = %f", avg_waiting);
    printf("\nAverage Turnaround Time = %f", avg_turnaround);
}

int main()
{
    int n, quantum;
    printf("Enter the number of processes: ");
    scanf("%d", &n);

    printf("Enter time quantum for Round Robin: ");
    scanf("%d", &quantum);

    struct Process *proc = (struct Process *)malloc(n * sizeof(struct Process));

    printf("Enter process details (Process ID, Burst time):\n");
    for (int i = 0; i < n; i++)
    {
        scanf("%d %d", &proc[i].pid, &proc[i].bt);
        proc[i].remaining = proc[i].bt;
        proc[i].finish = 0;
        proc[i].turnaround = 0;
        proc[i].waiting = 0;
    }

    findWaitingTurnaroundTime(proc, n, quantum);
    displayResults(proc, n);
    calculateAverages(proc, n);

    free(proc);
    return 0;
}
```

Output-



```
Enter the number of processes: 3
Enter time quantum for Round Robin: 2
Enter process details (Process ID, Burst time):
1 10
2 5
3 8

Processes  Burst time  Finish time  Turnaround time  Waiting time
   1           10           23             23              13
   2           5            15             15              10
   3           8            21             21              13

Average Waiting Time = 12.000000
Average Turnaround Time = 19.666666
```