EXP 8: Construct a C program to simulate Round Robin scheduling algorithm with C.

```
#include <stdio.h>

int main() {
    int n, i, time = 0, tq, remain;
    int bt[20], rt[20], wt[20], tat[20];
    float total_wt = 0, total_tat = 0;

    printf("Enter number of processes: ");
    scanf("%d", &n);
    remain = n;

    for (i = 0; i < n; i++) {
        printf("Enter burst time for process %d: ", i + 1);
        scanf("%d", &bt[i]);
        rt[i] = bt[i]; // Remaining time
    }

    printf("Enter time quantum: ");
    scanf("%d", &tq);

    while (remain > 0) {
        for (i = 0; i < n; i++) {
            if (rt[i] > 0) {
                if (rt[i] > tq) {
                    time += tq;
                    rt[i] -= tq;
                } else {
                    time += rt[i];
                    wt[i] = time - bt[i]; // Final waiting time
                    rt[i] = 0;
                    remain--;
                }
            }
        }
    }

    // Turnaround time = waiting time + burst time
    for (i = 0; i < n; i++) {
        tat[i] = bt[i] + wt[i];
        total_wt += wt[i];
        total_tat += tat[i];
    }

    // Print results
    printf("\nProcess\tBurst Time\tWaiting Time\tTurnaround Time\n");
    for (i = 0; i < n; i++) {
        printf("P%d\t%d\t\t%d\t\t%d\n", i + 1, bt[i], wt[i], tat[i]);
    }

    printf("\nAverage Waiting Time = %.2f", total_wt / n);
    printf("Average Turnaround Time = %.2f\n", total_tat / n);

    return 0;
}
```

Sample Input

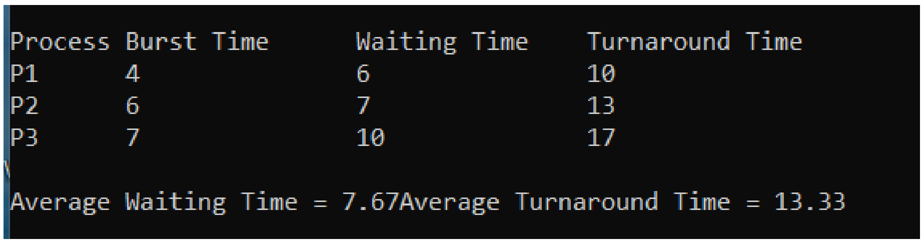
```
Enter number of processes: 3
Enter burst time for process 1: 4
```

Enter burst time for process 2: 6

Enter burst time for process 3: 7

Enter time quantum: 3

Sample Output



```
Process Burst Time      Waiting Time    Turnaround Time
P1      4               6               10
P2      6               7               13
P3      7               10              17

Average Waiting Time = 7.67Average Turnaround Time = 13.33
```