

```

#include <stdio.h>
void calculateCompletionTime(int burst_time[], int arrival_time[], int n, int
quantum) {
    int remaining_time[n];
    int completion_time[n];
    int waiting_time[n];
    int turnaround_time[n];
    int total_waiting_time = 0;
    int total_turnaround_time = 0;
    for (int i = 0; i < n; i++) {
        remaining_time[i] = burst_time[i];
        waiting_time[i] = 0;
        turnaround_time[i] = 0;
        completion_time[i] = -1; // Initialize to -1 indicating process not
completed yet
    }

    int currentTime = 0;
    int completed = 0;
    int quantum_time = quantum;

    printf("Process List\tArrival Time\tBurst Time\tCompletion Time\tTurnaround
Time\tWaiting Time\n");

    while (completed != n) {
        for (int i = 0; i < n; i++) {
            if (arrival_time[i] <= currentTime && remaining_time[i] > 0) {
                if (remaining_time[i] <= quantum_time) {
                    currentTime += remaining_time[i];
                    remaining_time[i] = 0;
                    completion_time[i] = currentTime;
                    completed++;
                    turnaround_time[i] = completion_time[i] - arrival_time[i];
                    waiting_time[i] = turnaround_time[i] - burst_time[i];
                    total_waiting_time += waiting_time[i];
                    total_turnaround_time += turnaround_time[i];
                    printf("P%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\n", i + 1,
arrival_time[i], burst_time[i], completion_time[i], turnaround_time[i],
waiting_time[i]);
                } else {
                    currentTime += quantum_time;
                    remaining_time[i] -= quantum_time;
                }
            }
        }
    }

    float average_waiting_time = (float)total_waiting_time / n;
    float average_turnaround_time = (float)total_turnaround_time / n;
    printf("\nAverage Waiting Time: %.2f\n", average_waiting_time);
    printf("Average Turnaround Time: %.2f\n", average_turnaround_time);
}

int main() {

```

```

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

int burst_time[n], arrival_time[n];
printf("Enter burst time and arrival time for each process:\n");
for (int i = 0; i < n; i++) {
    printf("Burst time for process %d: ", i + 1);
    scanf("%d", &burst_time[i]);
    printf("Arrival time for process %d: ", i + 1);
    scanf("%d", &arrival_time[i]);
}

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

calculateCompletionTime(burst_time, arrival_time, n, quantum);

return 0;
}

```

```

Enter the number of processes: 3
Enter burst time and arrival time for each process:
Burst time for process 1: 8
Arrival time for process 1: 2
Burst time for process 2: 7
Arrival time for process 2: 0
Burst time for process 3: 9
Arrival time for process 3: 1
Enter the quantum time: 4

```

Process	AT	BT	CT	TAT	WT
P2	0	7	15	15	8
P1	2	8	23	21	13
P3	1	9	24	23	14

```

Average Waiting Time: 11.67
Average Turnaround Time: 19.67

```