Task 11: Write a program to find the finish time, turnaround time, and waiting time using Round Robin Algorithm (input Takes by user).

```
//round robin
#include <stdio.h>


void findWaitingTime(int processes[], int n, int bt[], int wt[], int quantum) {
    int rem_bt[n];
    for (int i = 0; i < n; i++) {
        rem_bt[i] = bt[i];
    }
    int t = 0;
    while (1) {
        int done = 1;
        for (int i = 0; i < n; i++) {
            if (rem_bt[i] > 0) {
                done = 0;
                if (rem_bt[i] > quantum) {
                    t += quantum;
                    rem_bt[i] -= quantum;
                } else {
                    t = t + rem_bt[i];
                    wt[i] = t - bt[i];
                    rem_bt[i] = 0;
                }
            }
        }
        if (done == 1) {
            break;
        }
    }
```

```
}

void findTurnAroundTime(int processes[], int n, int bt[], int wt[], int tat[]) {
    for (int i = 0; i < n; i++) {
        tat[i] = bt[i] + wt[i];
    }
}

void findFinishTime(int processes[], int n, int at[], int bt[], int wt[], int ft[]) {
    for (int i = 0; i < n; i++) {
        ft[i] = at[i] + bt[i] + wt[i];
    }
}

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

    int processes[n], burst_time[n], arrival_time[n], waiting_time[n], turnaround_time[n],
finish_time[n];

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

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

```
    findWaitingTime(processes, n, burst_time, waiting_time, quantum);
    findTurnAroundTime(processes, n, burst_time, waiting_time, turnaround_time);
    findFinishTime(processes, n, arrival_time, burst_time, waiting_time, finish_time);

    printf("\nPID\tArrival Time\tBurst Time\tFinish Time\tTurnaround Time\tWaiting Time\n");
    for (int i = 0; i < n; i++) {
        printf("%d\t%d\t\t%d\t\t%d\t\t%d\t\t%d\n", i + 1, arrival_time[i], burst_time[i], finish_time[i],
turnaround_time[i], waiting_time[i]);
    }

    return 0;
}
```

Output-

```
Enter the number of processes: 2
Enter arrival time for process 1: 1
Enter burst time for process 1: 4
Enter arrival time for process 2: 0
Enter burst time for process 2: 5
Enter time quantum: 2

PID     Arrival Time    Burst Time      Finish Time     Turnaround Time Waiting Time
1       1               4               7               6                       2
2       0               5               9               9                       4
```