

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

```
#include <stdio.h>

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

    int at[n], bt[n], rem[n], wt[n], tat[n];
    for (int i = 0; i < n; i++) {
        printf("Enter arrival time of P%d: ", i + 1);
        scanf("%d", &at[i]);
        printf("Enter burst time of P%d: ", i + 1);
        scanf("%d", &bt[i]);
        rem[i] = bt[i]; // remaining burst time
        wt[i] = 0;
        tat[i] = 0;
    }

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

    int time = 0, done = 0;
    printf("\nGantt Chart:\n");

    while (done < n) {
        int progressed = 0; // to check if any process executed
        for (int i = 0; i < n; i++) {
            if (at[i] <= time && rem[i] > 0) {
                progressed = 1;
                int exec = rem[i] < tq ? rem[i] : tq; // execute min(rem, tq)
                printf("| P%d ", i + 1);
                rem[i] -= exec;
                time += exec;
                if (rem[i] == 0) {
                    done++;
                    tat[i] = time - at[i];
                    wt[i] = tat[i] - bt[i];
                }
            }
        }
        if (!progressed) time++; // idle time
    }
    printf("\n");
```

```

printf("\nProcess\tArrival\tBurst\tWaiting\tTurnaround\n");
double total_wt = 0, total_tat = 0;
for (int i = 0; i < n; i++) {
    printf("P%d\t%d\t%d\t%d\t%d\n", i + 1, at[i], bt[i], wt[i], tat[i]);
    total_wt += wt[i];
    total_tat += tat[i];
}

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

return 0;
}

```

## OUTPUT:

```

[[{"id": 1, "process": "P1", "arrival": 0, "burst": 2, "waiting": 0, "turnaround": 2}, {"id": 2, "process": "P2", "arrival": 3, "burst": 2, "waiting": 2, "turnaround": 2}, {"id": 3, "process": "P3", "arrival": 5, "burst": 1, "waiting": 0, "turnaround": 1}], [{"time": 0, "process": "P1"}, {"time": 2, "process": "P1"}, {"time": 2, "process": "P2"}, {"time": 4, "process": "P2"}, {"time": 4, "process": "P3"}, {"time": 5, "process": "P3"}], [{"text": "Enter number of processes: 3", "x": 10, "y": 10}, {"text": "Enter arrival time of P1: 0", "x": 10, "y": 12}, {"text": "Enter burst time of P1: 2", "x": 10, "y": 14}, {"text": "Enter arrival time of P2: 3", "x": 10, "y": 16}, {"text": "Enter burst time of P2: 2", "x": 10, "y": 18}, {"text": "Enter arrival time of P3: 5", "x": 10, "y": 20}, {"text": "Enter burst time of P3: 1", "x": 10, "y": 22}, {"text": "Enter time quantum: 4", "x": 10, "y": 24}, {"text": "Gantt Chart:", "x": 10, "y": 26}, {"text": "| P1 | P2 | P3 |", "x": 10, "y": 28}, {"text": "Process Arrival Burst Waiting Turnaround", "x": 10, "y": 30}, {"text": "P1 0 2 0 2", "x": 10, "y": 32}, {"text": "P2 3 2 0 2", "x": 10, "y": 34}, {"text": "P3 5 1 0 1", "x": 10, "y": 36}, {"text": "Average Waiting Time = 0.00", "x": 10, "y": 38}, {"text": "Average Turnaround Time = 1.67", "x": 10, "y": 40}, {"text": "-----", "x": 10, "y": 42}, {"text": "Process exited after 38.5 seconds with return value 0", "x": 10, "y": 44}, {"text": "Press any key to continue . . .", "x": 10, "y": 46}]]
```