

Ex. No.: 6d)

## ROUND ROBIN SCHEDULING

Aim: To implement the Round Robin (RR) scheduling technique

## Program code:

```
#include <stdio.h>

int main() {
    int i, j, n, time = 0, tq, remaining;
    int bt[10], rt[10], wt[10], tat[10];
    float avg_wt = 0, avg_tat = 0;

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

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

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

    while (remaining != 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];
                    rt[i] = 0;
                    remaining--;
                }
            }
        }
    }

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

    avg_wt /= n;
    avg_tat /= n;

    printf("\nAverage Waiting Time: %.2f", avg_wt);
    printf("\nAverage Turnaround Time: %.2f\n", avg_tat);

    return 0;
}
```

Output:

```
Enter total number of processes: 3
Enter burst time for P[1]: 10
Enter burst time for P[2]: 5
Enter burst time for P[3]: 8
Enter time quantum: 3

Process Burst Time      Waiting Time      Turnaround Time
P[1]    10              13              23
P[2]    5               9              14
P[3]    8              14              22

Average Waiting Time: 12.00
Average Turnaround Time: 19.67
```