**EXP NO:** 6d) **ROUND ROBIN SCHEDULING**

**DATE:16/2/25**

**PROGRAM:**

#include <stdio.h>

struct Process {

    int processID;

    int burstTime;

    int arrivalTime;

    int remainingTime;

    int waitingTime;

    int turnAroundTime;

};

int main() {

    int n, quantum;

    printf("Enter the number of processes: ");

    scanf("%d", &n);

    printf("Enter the time quantum: ");

    scanf("%d", &quantum);

    struct Process processes[n];

    // Reading process details

    for (int i = 0; i < n; i++) {

        processes[i].processID = i + 1;

        printf("Enter the burst time and arrival time of process %d:\n", i + 1);

        scanf("%d %d", &processes[i].burstTime, &processes[i].arrivalTime);

        processes[i].remainingTime = processes[i].burstTime;

        processes[i].waitingTime = 0;

        processes[i].turnAroundTime = 0;

    }

    int t = 0;  // Initialize time

    int completed = 0;

    // Round Robin Scheduling

    while (completed < n) {

        for (int i = 0; i < n; i++) {

            if (processes[i].remainingTime > 0) {

                if (processes[i].remainingTime > quantum) {

                    t += quantum;

                    processes[i].remainingTime -= quantum;

                } else {

                    t += processes[i].remainingTime;

                    processes[i].waitingTime = t - processes[i].burstTime - processes[i].arrivalTime;

                    processes[i].remainingTime = 0;

                    completed++;

                }

            }

        }

    }

// Calculate Turnaround Time

    for (int i = 0; i < n; i++) {

        processes[i].turnAroundTime = processes[i].waitingTime + processes[i].burstTime;

    }

    // Display results

    float totalWaitingTime = 0, totalTurnAroundTime = 0;

    printf("\nProcess Burst Time Arrival Time Waiting Time Turnaround Time\n");

    for (int i = 0; i < n; i++) {

        printf("%d\t\t%d\t\t%d\t\t%d\t\t%d\n", processes[i].processID, processes[i].burstTime,

               processes[i].arrivalTime, processes[i].waitingTime, processes[i].turnAroundTime);

        totalWaitingTime += processes[i].waitingTime;

        totalTurnAroundTime += processes[i].turnAroundTime;

    }

    printf("\nAverage Waiting Time: %.2f\n", totalWaitingTime / n);

    printf("Average Turnaround Time: %.2f\n", totalTurnAroundTime / n);

    return 0;

}

**OUTPUT:**

Enter the number of processes: 2

Enter the time quantum: 2

Enter the burst time and arrival time of process 1:

4

0

Enter the burst time and arrival time of process 2:

6

4

Process Burst Time Arrival Time Waiting Time Turnaround Time

1 4 0 2 6

2 6 4 0 6

Average Waiting Time: 1.00

Average Turnaround Time: 6.00