Name: Kushal Kishor Shankhapal **Date:** 30/07/2023
Roll No: 56 **Subject:** OS Lab

RR: Round Robin

Code:

```
#include <stdio.h>

#define MAX_PROCESSES 10  // Define a constant for the maximum number of
processes

int main() {
    int i, limit, total_time = 0, time_quantum;
    int wait_time = 0, turnaround_time = 0;
    int arrival_time[MAX_PROCESSES], burst_time[MAX_PROCESSES],
temp_burst_time[MAX_PROCESSES];
    int completion_time[MAX_PROCESSES];
    int completed[MAX_PROCESSES] = {0}; // To keep track of completed
processes
    int remaining_processes, counter;
    float average_wait_time, average_turnaround_time;

    // Input number of processes
    printf("Enter Total Number of Processes (max %d):\n\t",
MAX_PROCESSES);
    scanf("%d", &limit);

    // Input arrival and burst times for each process
    for (i = 0; i < limit; i++) {
        printf("Enter Details of Process[%d]\n", i + 1);
        printf("Arrival Time:\t");
        scanf("%d", &arrival_time[i]);
        printf("Burst Time:\t");
        scanf("%d", &burst_time[i]);
        temp_burst_time[i] = burst_time[i];  // Copy burst times for
processing
    }

    // Input time quantum
    printf("Enter Time Quantum:\n\t");
    scanf("%d", &time_quantum);

    printf("\nProcess ID\tAT\tBT\tCT\tTAT\tWT\n");

    remaining_processes = limit;
    counter = 0;

    // Main Round Robin scheduling loop
    while (remaining_processes > 0) {
        int process_found = 0;

        for (i = 0; i < limit; i++) {
            if (arrival_time[i] <= total_time && !completed[i]) {
                process_found = 1;

                if (temp_burst_time[i] <= time_quantum &&
temp_burst_time[i] > 0) {
                    total_time += temp_burst_time[i];
```

```
                    temp_burst_time[i] = 0;
                    completion_time[i] = total_time; // Record completion
time
                    counter = 1;
                } else if (temp_burst_time[i] > 0) {
                    temp_burst_time[i] -= time_quantum;
                    total_time += time_quantum;
                }

                // Check if the process is complete
                if (temp_burst_time[i] == 0 && counter == 1) {
                    remaining_processes--;

                    // Calculate turnaround time and waiting time
                    int turnaround = completion_time[i] - arrival_time[i];
                    int wait = turnaround - burst_time[i];
                    wait_time += wait;
                    turnaround_time += turnaround;

                    completed[i] = 1; // Mark process as completed
                    counter = 0;
                }
            }
        }

        // If no process was found that could be executed, advance time
        if (!process_found) {
            total_time++;
        }
    }

    // Print process details in the original order
    for (i = 0; i < limit; i++) {
        if (completed[i]) { // Only print completed processes
            int turnaround = completion_time[i] - arrival_time[i];
            int wait = turnaround - burst_time[i];
            printf("Process[%d]\t%d\t%d\t%d\t%d\t%d\n", i + 1,
arrival_time[i], burst_time[i], completion_time[i], turnaround, wait);
        }
    }

    // Calculate average waiting time and turnaround time
    average_wait_time = (float)wait_time / limit;
    average_turnaround_time = (float)turnaround_time / limit;

    // Print average times
    printf("\nAverage Waiting Time:\t%f", average_wait_time);
    printf("\nAverage Turnaround Time:\t%f\n", average_turnaround_time);

    return 0;
}
```

Output:

```
pl-13@pl13-OptiPlex-3020:~/Kushal_Assignments-main/OS_Lab/RR$ gcc RR_1.c
pl-13@pl13-OptiPlex-3020:~/Kushal_Assignments-main/OS_Lab/RR$ ./a.out
Enter Total Number of Processes (max 10):
      4
```

Enter Details of Process[1]
Arrival Time: 0
Burst Time: 8
Enter Details of Process[2]
Arrival Time: 1
Burst Time: 4
Enter Details of Process[3]
Arrival Time: 2
Burst Time: 9
Enter Details of Process[4]
Arrival Time: 3
Burst Time: 5
Enter Time Quantum:
4

Process ID	AT	BT	CT	TAT	WT
Process[1]	0	8	20	20	12
Process[2]	1	4	8	7	3
Process[3]	2	9	26	24	15
Process[4]	3	5	25	22	17

Average Waiting Time: 11.750000
Average Turnaround Time: 18.250000