```c
#include <stdio.h>
#include <stdlib.h>
#define MAX_TASKS 5
#define SIMULATION_TIME 20
typedef struct {
    int id;
    int period;
    int execution;
    int remaining;
    int next_start;
} Task;
Task tasks[MAX_TASKS];
int task_count;
int compare_RMS(const void *a, const void *b) {
    return ((Task *)a)->period - ((Task *)b)->period;
}
void rate_monotonic_scheduling() {
    printf("\nRate-Monotonic Scheduling (RMS) Simulation:\n");
    qsort(tasks, task_count, sizeof(Task), compare_RMS);

    for (int time = 0; time < SIMULATION_TIME; time++) {
        int selected_task = -1;

        for (int i = 0; i < task_count; i++) {
            if (time >= tasks[i].next_start && tasks[i].remaining > 0) {
                selected_task = i;
                break;
            }
        }
        if (selected_task != -1) {
            tasks[selected_task].remaining--;
            printf("Time %d: Task %d executing\n", time, tasks[selected_task].id);

            if (tasks[selected_task].remaining == 0) {
                tasks[selected_task].next_start += tasks[selected_task].period;
                tasks[selected_task].remaining = tasks[selected_task].execution;
            }
        } else {
            printf("Time %d: CPU idle\n", time);
        }
    }
}
int main() {
    printf("Enter number of tasks (max %d): ", MAX_TASKS);
    scanf("%d", &task_count);
    if (task_count > MAX_TASKS || task_count <= 0) {
        printf("Invalid number of tasks!\n");
        return 1;
    }
    for (int i = 0; i < task_count; i++) {
        printf("Enter Task %d execution time and period: ", i + 1);
        scanf("%d %d", &tasks[i].execution, &tasks[i].period);
        tasks[i].id = i + 1;
        tasks[i].remaining = tasks[i].execution;
        tasks[i].next_start = 0;
    }
rate_monotonic_scheduling();
return 0;
}
```

```
PS C:\Users\STUDENT\Desktop\Archita OS lab> cd "c:\Users\STUDENT\Desktop\Archita OS lab\" ; if ($?) { gcc rate.c -o rate } ; if ($?) { .\rate }
Enter number of tasks (max 5): 3
Enter Task 1 execution time and period: 2 5
Enter Task 2 execution time and period: 1 7
Enter Task 3 execution time and period: 2 10

Rate-Monotonic Scheduling (RMS) Simulation:
Time 0: Task 1 executing
Time 1: Task 1 executing
Time 2: Task 2 executing
Time 3: Task 3 executing
Time 4: Task 3 executing
Time 5: Task 1 executing
Time 6: Task 1 executing
Time 7: Task 2 executing
Time 8: CPU idle
Time 9: CPU idle
Time 10: Task 1 executing
Time 11: Task 1 executing
Time 12: Task 3 executing
Time 13: Task 3 executing
Time 14: Task 2 executing
Time 15: Task 1 executing
Time 16: Task 1 executing
Time 17: CPU idle
Time 18: CPU idle
Time 19: CPU idle
PS C:\Users\STUDENT\Desktop\Archita OS lab>
```