

EXP-8 :-

PROGRAM :-

```
#include <stdio.h>
```

```
#define MAX 10
```

```
struct Process {
```

```
    int id;
```

```
    int burst_time;
```

```
    int remaining_time;
```

```
};
```

```
void round_robin(struct Process processes[], int n, int quantum) {
```

```
    int time = 0;
```

```
    int completed = 0;
```

```
    while (completed < n) {
```

```
        for (int i = 0; i < n; i++) {
```

```
            if (processes[i].remaining_time > 0) {
```

```
                if (processes[i].remaining_time > quantum) {
```

```
                    time += quantum;
```

```
                    processes[i].remaining_time -= quantum;
```

```
                } else {
```

```
                    time += processes[i].remaining_time;
```

```
                    processes[i].remaining_time = 0;
```

```
                    completed++;
```

```
                    printf("Process %d completed at time %d\n", processes[i].id, time);
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
int main() {
```

```
    struct Process processes[MAX];
```

```
    int n, quantum;
```

```
printf("Enter number of processes: ");
scanf("%d", &n);
printf("Enter time quantum: ");
scanf("%d", &quantum);
for (int i = 0; i < n; i++) {
    processes[i].id = i + 1;
    printf("Enter burst time for process %d: ", processes[i].id);
    scanf("%d", &processes[i].burst_time);
    processes[i].remaining_time = processes[i].burst_time;
}
round_robin(processes, n, quantum);
return 0;
}
```

OUTPUT :-

```
Enter number of processes: 3
Enter time quantum: 2
Enter burst time for process 1: 8
Enter burst time for process 2: 15
Enter burst time for process 3: 20
Process 1 completed at time 20
Process 2 completed at time 37
Process 3 completed at time 43
```

```
=== Code Execution Successful ===
```