# b. Round Robin

```c
#include <stdio.h>

#define MAX_PROC 10

int main() {
    int numProcs, timeQuantum, clock = 0, completed = 0;
    int arrivalTime[MAX_PROC], burstTime[MAX_PROC], remainingTime[MAX_PROC];
    int waitTime[MAX_PROC] = {0}, turnAroundTime[MAX_PROC] = {0};

    printf("Enter the number of processes: ");
    scanf("%d", &numProcs);

    printf("Enter the arrival and burst times for each process:\n");
    for (int i = 0; i < numProcs; i++) {
        scanf("%d %d", &arrivalTime[i], &burstTime[i]);
        remainingTime[i] = burstTime[i];
    }

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

    while (completed < numProcs) {
        completed = 0;
        for (int i = 0; i < numProcs; i++) {
            if (remainingTime[i] > 0 && arrivalTime[i] <= clock) {
                if (remainingTime[i] > timeQuantum) {
                    clock += timeQuantum;
                    remainingTime[i] -= timeQuantum;
                } else {
                    clock += remainingTime[i];
                    waitTime[i] = clock - arrivalTime[i] - burstTime[i];
                    turnAroundTime[i] = clock - arrivalTime[i];
                    remainingTime[i] = 0;
                }
            }
            if (remainingTime[i] == 0) completed++;
        }
        if (completed < numProcs) clock++;
    }

    printf("Process\tWaiting Time\tTurnaround Time\n");
```

```
    for (int i = 0; i < numProcs; i++) {
        printf("%d\t%d\t\t%d\n", i+1, waitTime[i], turnAroundTime[i]);
    }

    return 0;
}
```

```
Enter the number of processes: 4
Enter the arrival and burst times for each process:
0 5
1 3
2 4
3 2
Enter the time quantum: 2
Process  Waiting Time  Turnaround Time
1  0     5
2  3     6
3  4     6
4  5     7
```