OS LAB ROUND ROBIN

```
#include <stdio.h>
void roundRobin(int n, int bt[], int qt) {
   int wt[n], tat[n], rem_bt[n];
   for (int i = 0; i < n; i++) rem_bt[i] = bt[i];
   int t = 0, done;
   while (1) {
      done = 1;
      for (int i = 0; i < n; i++) {
         if (rem_bt[i] > 0) {
            done = 0;
            if (rem_bt[i] > qt) {
               t += qt;
               rem_bt[i] -= qt;
            } else {
               t += rem_bt[i];
               wt[i] = t - bt[i];
               rem_bt[i] = 0;
            }
         }
      }
      if (done) break;
   }

   int total_wt = 0, total_tat = 0;
   printf("PID  Burst Time  Waiting Time  Turnaround Time\n");
   for (int i = 0; i < n; i++) {
      tat[i] = bt[i] + wt[i];
      total_wt += wt[i];
      total_tat += tat[i];
```

```
        printf("%d\t%d\t\t%d\t\t%d\n", i+1, bt[i], wt[i], tat[i]);
    }
    printf("\nAverage waiting time: %.2f\n", (float)total_wt/n);
    printf("Average turnaround time: %.2f\n", (float)total_tat/n);
}

int main() {
    int n, qt;
    printf("Enter the number of processes: ");
    scanf("%d", &n);
    int bt[n];
    printf("Enter the quantum time: ");
    scanf("%d", &qt);

    for (int i = 0; i < n; i++) {
        printf("Enter burst time for process %d: ", i+1);
        scanf("%d", &bt[i]);
    }
    roundRobin(n, bt, qt);
    return 0;
}
```

OUTPUT

```
Enter the number of processes: 4
Enter the quantum time: 5
Enter burst time for process 1: 3
Enter burst time for process 2: 5
Enter burst time for process 3: 4
Enter burst time for process 4: 9
PID  Burst Time  Waiting Time  Turnaround Time
1       3              0              3
2       5              3              8
3       4              8              12
4       9              12             21

Average waiting time: 5.75
Average turnaround time: 11.00
```