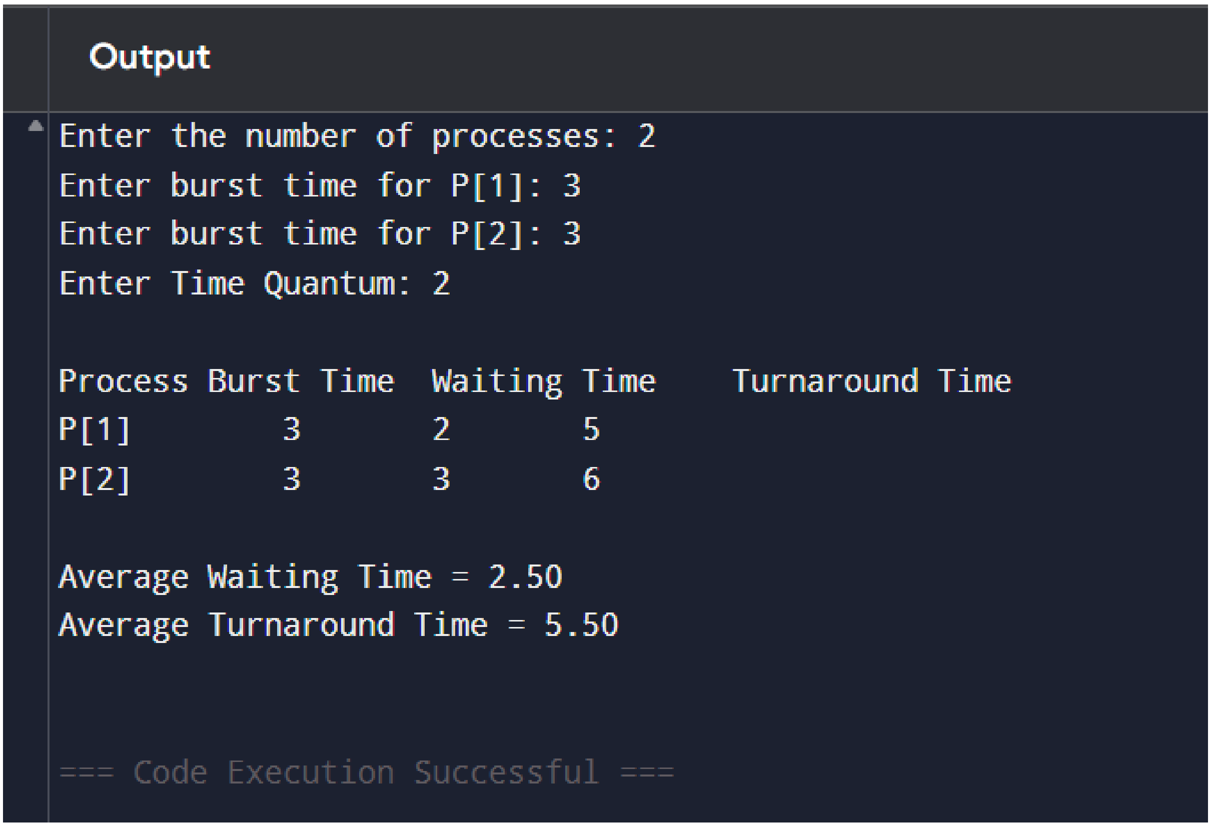
d) Round Robin (Experiment with different quantum sizes for RR algorithm)

```
#include <stdio.h>

int main() {
  int i, n, bt[20], rt[20], wt[20], tat[20];
  int time = 0, tq, remain, flag = 0;

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

  for(i = 0; i < n; i++) {
    printf("Enter burst time for P[%d]: ", i+1);
    scanf("%d", &bt[i]);
    rt[i] = bt[i];
  }

  printf("Enter Time Quantum: ");
  scanf("%d", &tq);

  while(remain > 0) {
    flag = 1;
    for(i = 0; i < n; i++) {
      if(rt[i] > 0) {
        flag = 0;
        if(rt[i] > tq) {
          time += tq;
          rt[i] -= tq;
        } else {
          time += rt[i];
          wt[i] = time - bt[i];
          rt[i] = 0;
          remain--;
        }
      }
    }
    if(flag) break;
  }

  for(i = 0; i < n; i++) {
    tat[i] = bt[i] + wt[i];
  }

  printf("\nProcess\tBurst Time\tWaiting Time\tTurnaround Time\n");
  for(i = 0; i < n; i++) {
        printf("P[%d]\t\t%d\t\t%d\t\t%d\n", i+1, bt[i], wt[i], tat[i]);
    }

    float avg_wt = 0, avg_tat = 0;
    for(i = 0; i < n; i++) {
        avg_wt += wt[i];
        avg_tat += tat[i];
    }

    printf("\nAverage Waiting Time = %.2f", avg_wt / n);
    printf("\nAverage Turnaround Time = %.2f\n", avg_tat / n);

    return 0;
}
```



```
Enter the number of processes: 2
Enter burst time for P[1]: 3
Enter burst time for P[2]: 3
Enter Time Quantum: 2

Process Burst Time  Waiting Time    Turnaround Time
P[1]        3       2       5
P[2]        3       3       6

Average Waiting Time = 2.50
Average Turnaround Time = 5.50



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