EXERCISE TWO

```
#include <stdio.h>
// FCFS Scheduling
void fcfs(int n, int bt[]) {
 int wt[10], tat[10];
 wt[0] = 0;
 for(int i = 1; i < n; i++)
 wt[i] = wt[i-1] + bt[i-1];
 printf("\nFCFS Scheduling:\n");
 printf("Process\tBurst Time\tWaiting Time\tTurnaround Time\n");
 for(int i = 0; i < n; i++) {
 tat[i] = wt[i] + bt[i];
 printf("P%d\t\t%d\t\t%d\t\t%d\n", i+1, bt[i], wt[i], tat[i]);
 }
}
// Round Robin Scheduling
void roundRobin(int n, int bt[], int tq) {
 int rem_bt[10], wt[10], tat[10];
 int t = 0;
 for(int i = 0; i < n; i++) {
 rem_bt[i] = bt[i];
 wt[i] = 0;
 }
 while(1) {
 int done = 1;
 for(int i = 0; i < n; i++) {
 if(rem_bt[i] > 0) {
 done = 0;
 if(rem_bt[i] > tq) {
```

```
t += tq;
rem_bt[i] -= tq;
} else {
t += rem_bt[i];
wt[i] = t - bt[i];
rem_bt[i] = 0;
}
}
}
if(done == 1)
break;
}
 printf("\nRound Robin Scheduling (Time Quantum = %d):\n", tq);
printf("Process\tBurst Time\tWaiting Time\tTurnaround Time\n");
for(int i = 0; i < n; i++) {
tat[i] = bt[i] + wt[i];
printf("P%d\t\t%d\t\t%d\t\t%d\n", i+1, bt[i], wt[i], tat[i]);
}
}
int main() {
int n, tq;
int bt[10];
printf("Enter number of processes: ");
scanf("%d", &n);
printf("Enter burst times:\n");
for(int i = 0; i < n; i++) {
printf("P%d: ", i+1);
scanf("%d", &bt[i]);
}
fcfs(n, bt);
printf("\nEnter Time Quantum for Round Robin: ");
```

```
 scanf("%d", &tq);
 roundRobin(n, bt, tq);
  return 0;
}
```