EXP 3: Design a CPU scheduling program with C using First Come First Served technique with the following considerations.

a. All processes are activated at time 0.

b. Assume that no process waits on I/O devices.

```
#include <stdio.h>

int main() {
    int n, i;
    int bt[20];      // Burst Time
    int at[20];      // Arrival Time (all 0)
    int wt[20];      // Waiting Time
    int tat[20];     // Turnaround Time
    int total_wt = 0, total_tat = 0;

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

    // Assume all arrival times are 0
    for (i = 0; i < n; i++) {
        at[i] = 0;
    }

    // Input burst times
    for (i = 0; i < n; i++) {
        printf("Enter Burst Time for Process %d: ", i + 1);
        scanf("%d", &bt[i]);
    }
```

```
    // First process has 0 waiting time
    wt[0] = 0;

    // Calculate waiting time for each process
    for (i = 1; i < n; i++) {
        wt[i] = wt[i - 1] + bt[i - 1];
    }

    // Calculate turnaround time
    for (i = 0; i < n; i++) {
        tat[i] = wt[i] + bt[i];
        total_wt += wt[i];
        total_tat += tat[i];
    }

    // Display the results including arrival time
    printf("\nProcess\tAT\tBT\tWT\tTAT\n");
    for (i = 0; i < n; i++) {
        printf("P%d\t%d\t%d\t%d\t%d\n", i + 1, at[i], bt[i], wt[i], tat[i]);
    }

    printf("\nAverage Waiting Time = %.2f", (float)total_wt / n);
    printf("\nAverage Turnaround Time = %.2f\n", (float)total_tat / n);

    return 0;
}
```

Sample Input

Enter number of processes: 3

Enter Burst Time for Process 1: 5

Enter Burst Time for Process 2: 7

Enter Burst Time for Process 3: 9

Sample Output

```
Process AT      BT      WT      TAT
P1      0       5       0       5
P2      0       7       5       12
P3      0       9       12      21

Average Waiting Time = 5.67
Average Turnaround Time = 12.67

--------------------------------
Process exited after 8.585 seconds with return value 0
Press any key to continue . . .
```