Name: T. Nikhil Kumar Reddy

Reg-No: 192372024

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.

Aim:

To design a program to simulate the **First Come First Serve (FCFS)** CPU scheduling algorithm, considering all processes are activated at time 0 and no I/O wait.

Algorithm:

1. Input the number of processes and their burst times.
2. Compute the completion time (CT) for each process.
   - CT[i]=CT[i−1]+BT[i]CT[i] = CT[i-1] + BT[i]CT[i]=CT[i−1]+BT[i] for i≥1i \geq 1i≥1.
3. Calculate Turnaround Time (TAT) and Waiting Time (WT):
   - TAT=CT−ArrivalTimeTAT = CT - ArrivalTimeTAT=CT−ArrivalTime
   - WT=TAT−BTWT = TAT - BTWT=TAT−BT
4. Display results including Completion Time, Turnaround Time, and Waiting Time.

Procedure:

1. Input process details (arrival times are 0 by default).
2. Iterate through processes in the order of arrival.
3. Use the FCFS formula to calculate the required times.
4. Output the computed metrics.

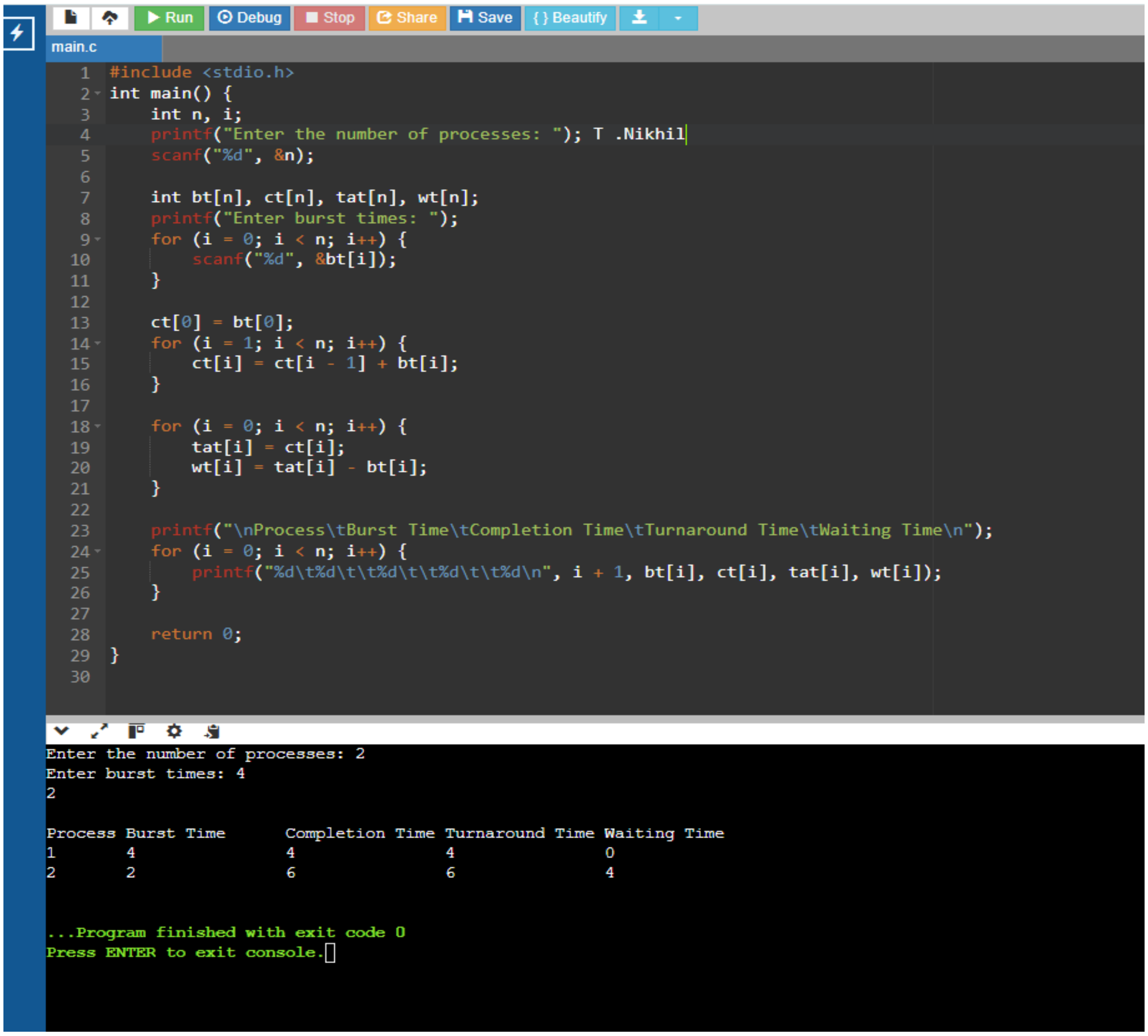
Code:

```
#include <stdio.h>

int main() {

   int n, i;

   printf("Enter the number of processes: ");

   scanf("%d", &n);

int bt[n], ct[n], tat[n], wt[n];

   printf("Enter burst times: ");
for (i = 0; i < n; i++) {
        scanf("%d", &bt[i]);
   }
ct[0] = bt[0];
   for (i = 1; i < n; i++) {
        ct[i] = ct[i - 1] + bt[i];
   }
for (i = 0; i < n; i++) {
        tat[i] = ct[i];
        wt[i] = tat[i] - bt[i];
   }
printf("\nProcess\tBurst Time\tCompletion Time\tTurnaround Time\tWaiting Time\n");
   for (i = 0; i < n; i++) {
        printf("%d\t%d\t\t%d\t\t%d\t\t%d\n", i + 1, bt[i], ct[i], tat[i], wt[i]);
   }
return 0;
}
```

Result

This simple implementation calculates the Completion Time (CT), Turnaround Time (TAT), and Waiting Time (WT) for all processes following FCFS scheduling.

Output:



```c
#include <stdio.h>
int main() {
    int n, i;
    printf("Enter the number of processes: "); T .Nikhil
    scanf("%d", &n);

    int bt[n], ct[n], tat[n], wt[n];
    printf("Enter burst times: ");
    for (i = 0; i < n; i++) {
        scanf("%d", &bt[i]);
    }

    ct[0] = bt[0];
    for (i = 1; i < n; i++) {
        ct[i] = ct[i - 1] + bt[i];
    }

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

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

    return 0;
}
```

```
Enter the number of processes: 2
Enter burst times: 4
2

Process Burst Time      Completion Time Turnaround Time Waiting Time
1       4               4               4               0
2       2               6               6               4


...Program finished with exit code 0
Press ENTER to exit console.
```