USN:1BM21CS222

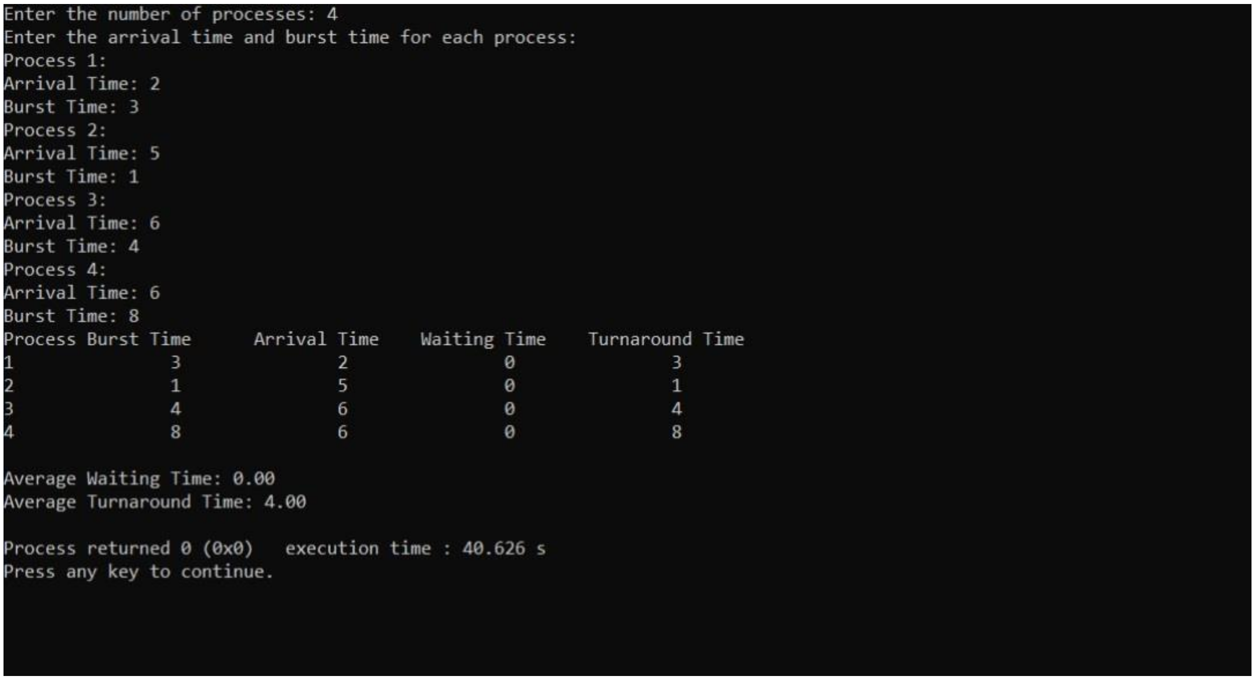
Code :

```
#include <stdio.h>

#define MAX_PROCESSES 10

Struct Process {
   Int processID;
   Int arrivalTime;
   Int burstTime;
};

Void findWaitingTime(struct Process processes[], int n, int waitingTime[])
{
   waitingTime[0] = 0;  // The first process always has a waiting time of 0

   for (int I = 1; I < n; i++) {
      waitingTime[i] = waitingTime[I – 1] + processes[I – 1].burstTime – processes[i].arrivalTime;

      if (waitingTime[i] < 0)
         waitingTime[i] = 0;
   }
}

Void findTurnaroundTime(struct Process processes[], int n, int waitingTime[], int turnaroundTime[])
{
   For (int I = 0; I < n; i++)
```

```
        turnaroundTime[i] = processes[i].burstTime + waitingTime[i];
}

Void findAverageTime(struct Process processes[], int n)
{
    Int waitingTime[MAX_PROCESSES], turnaroundTime[MAX_PROCESSES], totalWaitingTime = 0, totalTurnaroundTime = 0;

    findWaitingTime(processes, n, waitingTime);
    findTurnaroundTime(processes, n, waitingTime, turnaroundTime);

    printf("Process\tBurst Time\tArrival Time\tWaiting Time\tTurnaround Time\n");

    for (int I = 0; I < n; i++) {
        totalWaitingTime += waitingTime[i];
        totalTurnaroundTime += turnaroundTime[i];
        printf("%d\t\t%d\t\t%d\t\t%d\t\t%d\n", processes[i].processID, processes[i].burstTime, processes[i].arrivalTime,
               waitingTime[i], turnaroundTime[i]);
    }

    Float averageWaitingTime = (float)totalWaitingTime / n;
    Float averageTurnaroundTime = (float)totalTurnaroundTime / n;

    Printf("\nAverage Waiting Time: %.2f", averageWaitingTime);
    Printf("\nAverage Turnaround Time: %.2f\n", averageTurnaroundTime);
}

Int main()
```

```
{
    Int n;
    Printf("Enter the number of processes: ");
    Scanf("%d", &n);

    Struct Process processes[MAX_PROCESSES];

    Printf("Enter the arrival time and burst time for each process:\n");
    For (int I = 0; I < n; i++) {
        Printf("Process %d:\n", I + 1);
        Processes[i].processID = I + 1;
        Printf("Arrival Time: ");
        Scanf("%d", &processes[i].arrivalTime);
        Printf("Burst Time: ");
        Scanf("%d", &processes[i].burstTime);
    }

    findAverageTime(processes, n);

    return 0;
}
```

Output:



```
Enter the number of processes: 4
Enter the arrival time and burst time for each process:
Process 1:
Arrival Time: 2
Burst Time: 3
Process 2:
Arrival Time: 5
Burst Time: 1
Process 3:
Arrival Time: 6
Burst Time: 4
Process 4:
Arrival Time: 6
Burst Time: 8
Process Burst Time      Arrival Time    Waiting Time    Turnaround Time
1               3               2               0               3
2               1               5               0               1
3               4               6               0               4
4               8               6               0               8

Average Waiting Time: 0.00
Average Turnaround Time: 4.00

Process returned 0 (0x0)   execution time : 40.626 s
Press any key to continue.
```