### Write a C program to simulate the following CPU scheduling algorithm to find turnaround time and waiting time for First Come First Serve (FCFS).

#include <stdio.h>  
void sort(int proc\_id[], int at[], int bt[], int n) {  
    int temp;  
    for (int i = 0; i < n - 1; i++) {  
        for (int j = i + 1; j < n; j++) {  
            if (at[j] < at[i]) {  
                 
                temp = at[i];  
                at[i] = at[j];  
                at[j] = temp;  
                 
            temp = bt[i];  
                bt[i] = bt[j];  
                bt[j] = temp;  
                 
 temp = proc\_id[i];  
                proc\_id[i] = proc\_id[j];  
                proc\_id[j] = temp;  
            }  
        }  
    }  
}  
  
int main() {  
    int n, c = 0;  
    printf("Enter number of processes: ");  
    scanf("%d", &n);  
     
    int proc\_id[n], at[n], bt[n], ct[n], tat[n], wt[n];  
    double avg\_tat = 0.0, ttat = 0.0, avg\_wt = 0.0, twt = 0.0;  
     
    for (int i = 0; i < n; i++)  
        proc\_id[i] = i + 1;  
     
    printf("Enter arrival times:\n");  
    for (int i = 0; i < n; i++)  
        scanf("%d", &at[i]);  
     
    printf("Enter burst times:\n");  
    for (int i = 0; i < n; i++)  
        scanf("%d", &bt[i]);  
     
    sort(proc\_id, at, bt, n);  
       for (int i = 0; i < n; i++) {  
        if (c >= at[i]) {  
            c += bt[i];  
        } else {  
            c = at[i] + bt[i];  
        }  
        ct[i] = c;  
    }  
       for (int i = 0; i < n; i++) {  
        tat[i] = ct[i] - at[i];  
        wt[i] = tat[i] - bt[i];  
        ttat += tat[i];  
        twt += wt[i];  
    }  
     
    avg\_tat = ttat / (double)n;  
    avg\_wt = twt / (double)n;  
     
    printf("FCFS scheduling:\n");  
    printf("PID\tAT\tBT\tCT\tTAT\tWT\n");  
    for (int i = 0; i < n; i++) {  
        printf("%d\t%d\t%d\t%d\t%d\t%d\n", proc\_id[i], at[i], bt[i], ct[i], tat[i], wt[i]);  
    }  
     
    printf("\nAverage turnaround time: %.2lf ms\n", avg\_tat);  
    printf("Average waiting time: %.2lf ms\n", avg\_wt);  
     
    return 0;  
}

### OUTPUT:

