#include <stdio.h>

// Define the process structure

struct process {

int name;

int arrival;

int burst;

int tat;

int waiting;

};

int main() {

int n;

float avgtt = 0.0, avgwait = 0.0;

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

scanf("%d", &n);

struct process arr[n];

// Input process details

for (int i = 0; i < n; i++) {

printf("Enter process number: ");

scanf("%d", &arr[i].name);

printf("Enter arrival time: ");

scanf("%d", &arr[i].arrival);

printf("Enter burst time: ");

scanf("%d", &arr[i].burst);

}

// Display initial process information

printf("pr\tarr\tburst\n");

for (int i = 0; i < n; i++) {

printf("%d\t%d\t%d\n", arr[i].name, arr[i].arrival, arr[i].burst);

}

// Sort processes by arrival time

for (int i = 0; i < n - 1; i++) {

for (int j = i + 1; j < n; j++) {

if (arr[i].arrival > arr[j].arrival) {

struct process temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

printf("Ascending order of arrival time:\n");

printf("pr\tarr\tburst\n");

for (int i = 0; i < n; i++) {

printf("%d\t%d\t%d\n", arr[i].name, arr[i].arrival, arr[i].burst);

}

// Calculate Turnaround Time (TAT) and Waiting Time (WT)

arr[0].tat = arr[0].arrival + arr[0].burst;

for (int i = 1; i < n; i++) {

if (arr[i - 1].tat >= arr[i].arrival) {

arr[i].tat = arr[i - 1].tat + arr[i].burst;

} else {

arr[i].tat = arr[i].arrival + arr[i].burst;

}

}

for (int i = 0; i < n; i++) {

arr[i].tat = arr[i].tat - arr[i].arrival; // Turnaround time

arr[i].waiting = arr[i].tat - arr[i].burst; // Waiting time

avgtt += arr[i].tat;

avgwait += arr[i].waiting;

}

avgtt /= n;

avgwait /= n;

// Display results

printf("pr\tarr\tburst\ttat\twait\n");

for (int i = 0; i < n; i++) {

printf("%d\t%d\t%d\t%d\t%d\n", arr[i].name, arr[i].arrival, arr[i].burst, arr[i].tat, arr[i].waiting);

}

printf("Average Turnaround Time: %.2f\n", avgtt);

printf("Average Waiting Time: %.2f\n", avgwait);

return 0;

}