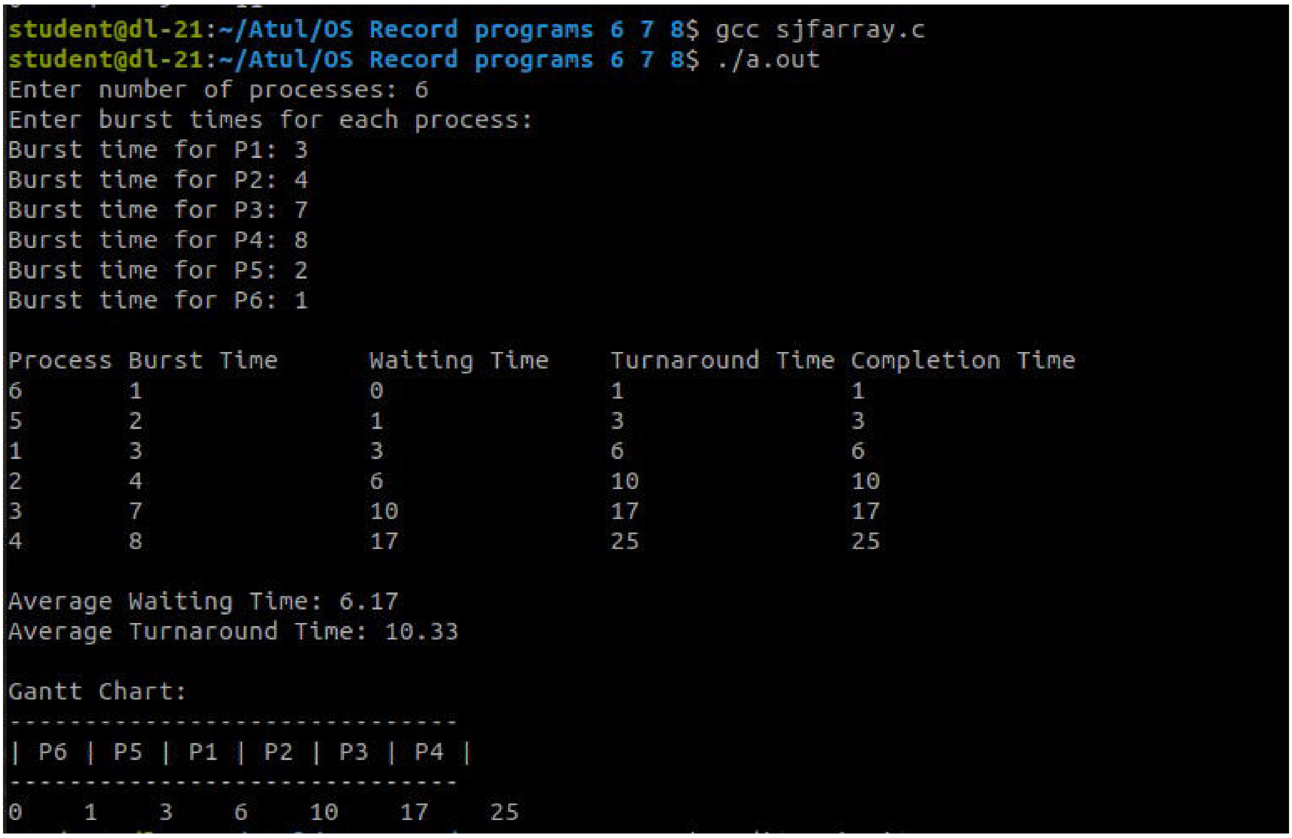
```c
#include <stdio.h>

void findWaitingTime(int processes[], int n, int bt[], int wt[]) {
    wt[0] = 0;

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

void findTurnAroundTime(int processes[], int n, int bt[], int wt[], int tat[]) {

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

void findCompletionTime(int processes[], int n, int bt[], int ct[]) {

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

void sortProcessesByBurstTime(int processes[], int bt[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (bt[i] > bt[j]) {
                int temp = bt[i];
                bt[i] = bt[j];
                bt[j] = temp;

                temp = processes[i];
                processes[i] = processes[j];
                processes[j] = temp;
            }
        }
    }
}

void findAvgTime(int processes[], int n, int bt[]) {
    int wt[n], tat[n], ct[n];
    findWaitingTime(processes, n, bt, wt);
    findTurnAroundTime(processes, n, bt, wt, tat);
    findCompletionTime(processes, n, bt, ct);

    int total_wt = 0, total_tat = 0;

    printf("\nProcess\tBurst Time\tWaiting Time\tTurnaround Time\tCompletion
Time\n");
    for (int i = 0; i < n; i++) {
        total_wt += wt[i];
        total_tat += tat[i];
        printf("%d\t%d\t\t%d\t\t%d\t\t%d\n", processes[i], bt[i], wt[i], tat[i],
ct[i]);
    }
    printf("\nAverage Waiting Time: %.2f", (float)total_wt / n);
    printf("\nAverage Turnaround Time: %.2f\n", (float)total_tat / n);
}

void ganttChart(int processes[], int n, int bt[], int ct[]) {
    printf("\nGantt Chart:\n");
    for (int i = 0; i < n; i++) {
      printf("-----"); }
    printf("\n");

    int current_time = 0;
    for (int i = 0; i < n; i++) {
        printf("| P%d ", processes[i]);
        current_time += bt[i];
    }
    printf("|\n");
    for (int i = 0; i < n; i++) {
      printf("-----");}
    printf("\n");

    current_time = 0;
    printf("0");
    for (int i = 0; i < n; i++) {
        current_time += bt[i];
        printf("    %d", current_time);
    }
    printf("\n");
}

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

    int processes[n], burst_time[n];
    for (int i = 0; i < n; i++) {
        processes[i] = i + 1;
    }
    printf("Enter burst times for each process:\n");
    for (int i = 0; i < n; i++) {
        printf("Burst time for P%d: ", processes[i]);
        scanf("%d", &burst_time[i]);
    }
    sortProcessesByBurstTime(processes, burst_time, n);
    findAvgTime(processes, n, burst_time);
    ganttChart(processes, n, burst_time, burst_time);
    return 0;
}
```



```
student@dl-21:~/Atul/OS Record programs 6 7 8$ gcc sjfarray.c
student@dl-21:~/Atul/OS Record programs 6 7 8$ ./a.out
Enter number of processes: 6
Enter burst times for each process:
Burst time for P1: 3
Burst time for P2: 4
Burst time for P3: 7
Burst time for P4: 8
Burst time for P5: 2
Burst time for P6: 1

Process Burst Time      Waiting Time    Turnaround Time Completion Time
6       1               0               1               1
5       2               1               3               3
1       3               3               6               6
2       4               6               10              10
3       7               10              17              17
4       8               17              25              25

Average Waiting Time: 6.17
Average Turnaround Time: 10.33

Gantt Chart:
------------------------------
| P6 | P5 | P1 | P2 | P3 | P4 |
------------------------------
0    1    3    6    10    17    25
```