```
#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 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]);
    }

    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 fcfsarray.c
student@dl-21:~/Atul/OS Record programs 6 7 8$ ./a.out
Enter number of processes: 3
Enter burst times for each process:
Burst time for P1: 4
Burst time for P2: 5
Burst time for P3: 2

Process Burst Time      Waiting Time    Turnaround Time Completion Time
1       4               0               4               4
2       5               4               9               9
3       2               9               11              11

Average Waiting Time: 4.33
Average Turnaround Time: 8.00

Gantt Chart:
---------------
| P1 | P2 | P3 |
---------------
0    4    9    11
```