

```

#include <stdio.h>

// Structure to represent a process
struct Process {
    int id;        // Process ID
    int arrivalTime; // Arrival time
    int burstTime;  // Burst time
    int completionTime; // Completion time
};

// Function to perform non-preemptive SJF scheduling
void SJF(struct Process processes[], int n) {
    // Sort processes based on arrival time
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (processes[j].arrivalTime > processes[j + 1].arrivalTime) {
                struct Process temp = processes[j];
                processes[j] = processes[j + 1];
                processes[j + 1] = temp;
            }
        }
    }

    int currentTime = 0;
    printf("\nGantt Chart:\n");
    printf("-----\n");

    // Process each job in the sorted order
    for (int i = 0; i < n; i++) {
        // Wait until arrival time
        if (currentTime < processes[i].arrivalTime) {

```

```

        currentTime = processes[i].arrivalTime;
    }

    // Execute the process
    printf("| P%d ", processes[i].id);
    currentTime += processes[i].burstTime;
    processes[i].completionTime = currentTime;
    printf("| ");
}
}

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

    struct Process processes[n]; // Array to store processes

    // Input the process details
    for (int i = 0; i < n; i++) {
        printf("Enter arrival time for process %d: ", i + 1);
        scanf("%d", &processes[i].arrivalTime);
        printf("Enter burst time for process %d: ", i + 1);
        scanf("%d", &processes[i].burstTime);
        processes[i].id = i + 1;
    }

    // Perform non-preemptive SJF scheduling
    SJF(processes, n);

    printf("\n\n");
}

```

```

// Print completion time of each process

printf("Process Completion Times:\n");

printf("-----\n");

printf("Process\tCompletion Time\n");

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

    printf("P%d\t%d\n", processes[i].id, processes[i].completionTime);

}

return 0;

}

```

```

C:\Users\kondur\OneDrive\ID  X  +  -  X
Enter the number of processes: 2
Enter arrival time for process 1: 2
Enter burst time for process 1: 3
Enter arrival time for process 2: 4
Enter burst time for process 2: 1

Gantt Chart:
| P1 || P2 |

Process Completion Times:
-----
Process Completion Time
P1      5
P2      6

-----
Process exited after 4.377 seconds with return value 0
Press any key to continue . . . |

```