```
#include <stdio.h>
#include <stdlib.h>
#define MAX 10
typedef struct
{
   int pid;
   int burst_time;
   int waiting_time;
   int completion_time;
   int turnaround_time;
} Process;

void print_table(Process p[], int n);
void print_gantt_chart(Process p[], int n);
void avg_TAT_WT(Process p[],int n);
void sort_burst_time(Process p[],int n);
float sum_waiting_time,sum_turnaround_time;

int main()
{
   Process p[MAX];
   int i, j, n;

   printf("Enter total number of process: ");
   scanf("%d", &n);
   printf("Enter burst time for each process:\n");
   for(i=0; i<n; i++) {
      p[i].pid = i;
      printf("P[%d] : ", i);
      scanf("%d", &p[i].burst_time);
      p[i].waiting_time = p[i].turnaround_time = 0;
   }
   printf("The arrival time of each process is taken as 0ms");

   sort_burst_time(p,n);



   p[0].turnaround_time =p[0].completion_time= p[0].burst_time;

   for(i=1; i<n; i++) {
         int ct=0;
      for(int j=i;j>=0;j--)
      {
         ct=ct+p[j].burst_time;
      }
      p[i].completion_time=ct;

      p[i].waiting_time = p[i-1].completion_time;
      p[i].turnaround_time = p[i].waiting_time + p[i].burst_time;
   }
```

```
    // print table
    printf("\n"); // Empty line
    print_table(p, n);
    printf("\n"); // Empty Line


    // print Gantt chart
    printf("         GANTT CHART         \n");
    printf("         ***********         \n");
    print_gantt_chart(p, n);

    avg_TAT_WT(p,n);

    return 0;
}
void sort_burst_time(Process p[], int n)
{
    int i, j;
    Process temp;
    for (i = 0; i <n; i++)
    {
        for (j = 0; j <(n-i-1); j++)
        {
            if (p[j].burst_time>p[j+1].burst_time)
            {
                temp = p[j];
                p[j] = p[j + 1];
                p[j + 1] = temp;
            }
        }
    }
}


void print_table(Process p[], int n)
{
    int i;

    printf("+-----+------------+--------------+-----------------+\n");
    printf("| PID | Burst Time | Waiting Time | Turnaround Time |\n");
    printf("+-----+------------+--------------+-----------------+\n");

    for(i=0; i<n; i++) {
    printf("| %d   |    %d       |      %d        |        %d        |\n"
            , p[i].pid, p[i].burst_time, p[i].waiting_time, p[i].turnaround_time );
        printf("+-----+------------+--------------+-----------------+\n");
    }

}
```

```
void print_gantt_chart(Process p[], int n)
{
   int i, j;
   // print top bar
   printf(" ");
   for(i=0; i<n; i++) {
      for(j=0; j<p[i].burst_time; j++) printf("--");
      printf(" ");
   }
   printf("\n|");

   // printing process id in the middle
   for(i=0; i<n; i++) {
      for(j=0; j<p[i].burst_time - 1; j++) printf(" ");
      printf("P%d", p[i].pid);
      for(j=0; j<p[i].burst_time - 1; j++) printf(" ");
      printf("|");
   }
   printf("\n ");
   // printing bottom bar
   for(i=0; i<n; i++) {
      for(j=0; j<p[i].burst_time; j++) printf("--");
      printf(" ");
   }
   printf("\n");

   // printing the time line
   printf("0");
   for(i=0; i<n; i++) {
      for(j=0; j<p[i].burst_time; j++) printf("  ");
      printf("%d", p[i].completion_time);

   }
   printf("\n");

}
void avg_TAT_WT(Process p[],int n)
{
   for(int i=0; i<n; i++) {
         sum_waiting_time += p[i].waiting_time;
         sum_turnaround_time += p[i].turnaround_time;
   }
   printf("Total Waiting Time      : %.2f\n", sum_waiting_time);
   printf("Average Waiting Time    : %.2f\n", (sum_waiting_time /n));
   printf("Total Turnaround Time   : %.2f\n", sum_turnaround_time);
   printf("Average Turnaround Time : %.2f\n", (sum_turnaround_time / n));


}
```


cations
Places
Terminal
File Edit View Search Terminal Help
root@kali:~# cd Desktop
root@kali:~/Desktop# cd os
root@kali:~/Desktop/os# gcc sjf1.c
root@kali:~/Desktop/os# ./a.out
Enter total number of process: 3
Enter burst time for each process:
P[0] : 10
P[1] : 2
P[2] : 5
The arrival time of each process is taken as 0ms
+-----+------------+--------------+-----------------+
| PID | Burst Time | Waiting Time | Turnaround Time |
+-----+------------+--------------+-----------------+
| 1 | 2 | 0 | 2 |
+-----+------------+--------------+-----------------+
| 2 | 5 | 2 | 7 |
+-----+------------+--------------+-----------------+
| 0 | 10 | 7 | 17 |
+-----+------------+--------------+-----------------+
GANTT CHART
****
---- ---------- --------------------
| P1 | P2 | P0 |
---- ---------- --------------------
0 2 7 17
Total Waiting Time : 9.00
Average Waiting Time : 3.00
Total Turnaround Time : 26.00
Average Turnaround Time : 8.67
root@kali:~/Desktop/os#