Problem 2: Write a program to implement the First Come First Serve scheduling algorithm and find the average turnaround time, waiting time, completion time and response time for overall process. Also Print Gantt chart for it.

Solution:

Source code:

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

typedef struct
{
   char process_name[3];
   int arrival_time;
   int burst_time;
   int complete_time;
   int turn_around_time;
   int wait_time;
   int response_time;
} process;

void print_process_table(process arr[],int n){
   int i;
   puts(" _____ _____ _____ _____ _____ _
_____ _____");
```

```
   puts("| Process Name | Arrival Time  | Burst Time | Complete Time | Turn Around Time | Wait Time | Response Time |");
   for(i=0; i<n;i++){
      puts("|_____|_____|_____|_____|_____|_____|_____|");
      printf("|    %3s    |    %3d    |    %3d   |    %3d    |    %4d    |   %3d   |    %3d    |\n",
      arr[i].process_name,arr[i].arrival_time,arr[i].burst_time,arr[i].complete_time,arr[i].turn_around_time,arr[i].wait_time,arr[i].response_time);
   }
   puts("|_____|_____|_____|_____|_____|_____|_____|");

}

void get_average(process arr[], int n){
   double tat=0,wt=0,rt=0;
   int i;
   for(i=0;i<n;i++){
      tat += (double)arr[i].turn_around_time;
      wt += (double)arr[i].wait_time;
      rt += (double)arr[i].response_time;
   }
   printf("Total time to Complete = %3d       Average Time to Complete = %.3f\n",arr[n-1].complete_time,(double)arr[n-1].complete_time/(double)n);
   printf("Total Turn Around Time = %.3f     Average Turn Around Time = %.3f\n",tat,tat/(double)n);
   printf("Total Waiting Time = %.3f        Average Waiting Time = %.3f\n",wt,wt/(double)n);
   printf("Total Response Time = %.3f       Average Response Time = %.3f\n",rt,rt/(double)n);
}

void gnatt(process arr[],int n){
```

```
int i,j;
// upper row
printf(" ");
for(i=0; i<n;i++){
   for(j=0;j<arr[i].burst_time+1;j++) printf("__");
   printf(" ");
}
printf("\n|");
// middle row
for(i=0;i<n;i++){
   for(j=0;j<arr[i].burst_time-1;j++){
      printf(" ");
   }
   printf("%3s",arr[i].process_name);
   for(j=0;j<arr[i].burst_time;j++){
      printf(" ");
   }
   printf("|");
}
printf("\n|");
// lower row
for(i=0; i<n;i++){
   for(j=0;j<arr[i].burst_time+1;j++) printf("__");
   printf("|");
}
printf("\n");
printf("0");
for(i=0; i<n; i++) {
   for(j=0; j<arr[i].burst_time+1; j++) printf("  ");
```

```
        if(arr[i].turn_around_time > 9) printf("\b");
        printf("%d", arr[i].turn_around_time);

    }
    printf("\n");
}

void main()
{
    int n =0,i;
    printf("Enter the number of processes\t");
    scanf("%d",&n);
    process arr[n];
    printf("Enter PROCESS_NAME ARRIVAL_TIME and BURST_TIME\n");
    for(i=0; i<n;i++)
    {
        scanf("%s %d %d",arr[i].process_name,&arr[i].arrival_time,&arr[i].burst_time);
    }
    // calculating completion time
    arr[0].complete_time=arr[0].burst_time + arr[0].arrival_time;
    arr[0].turn_around_time = arr[0].complete_time -  arr[0].arrival_time;
    arr[0].wait_time = arr[0].turn_around_time - arr[0].burst_time;
    arr[0].response_time = arr[0].wait_time;
    for(i=1; i<n;i++)
    {
        arr[i].complete_time = arr[i-1].complete_time + arr[i].burst_time;
        arr[i].turn_around_time = arr[i].complete_time-arr[i].arrival_time;
        arr[i].wait_time = arr[i].response_time =arr[i].turn_around_time-arr[i].burst_time;
    }
```

```
    print_process_table(arr,n);
    get_average(arr, n);
    puts("---------------------- GNATT CHART --------------------");
    gnatt(arr,n);
}
```

Output:

```
Enter the number of processes   5
Enter PROCESS_NAME ARRIVAL_TIME and BURST_TIME
p01 0 5
p02 4 7
p03 6 2
p04 8 4
p05 10 10
 _____________ _____________ _____________ _____________ _________________ _____________ _____________
| Process Name | Arrival Time | Burst Time | Complete Time | Turn Around Time | Wait Time | Response Time |
|_____________|_____________|_____________|_____________|_________________|_____________|_____________|
|      p01    |       0     |       5     |       5     |         5       |      0      |       0     |
|_____________|_____________|_____________|_____________|_________________|_____________|_____________|
|      p02    |       4     |       7     |      12     |         8       |      1      |       1     |
|_____________|_____________|_____________|_____________|_________________|_____________|_____________|
|      p03    |       6     |       2     |      14     |         8       |      6      |       6     |
|_____________|_____________|_____________|_____________|_________________|_____________|_____________|
|      p04    |       8     |       4     |      18     |        10       |      6      |       6     |
|_____________|_____________|_____________|_____________|_________________|_____________|_____________|
|      p05    |      10     |      10     |      28     |        18       |      8      |       8     |
|_____________|_____________|_____________|_____________|_________________|_____________|_____________|
Total time to Complete =  28        Average Time to Complete = 5.600
Total Turn Around Time = 49.000     Average Turn Around Time = 9.800
Total Waiting Time = 21.000         Average Waiting Time = 4.200
Total Response Time = 21.000        Average Response Time = 4.200
---------------------- GNATT CHART -------------------
 _____ _____________ _____ _____ _____________________
|  p01  |     p02     | p03 | p04 |         p05         |
|_____|_____________|_____|_____|_____________________|
0       5             8     8       10                  18

--------------------------------
Process exited after 40.79 seconds with return value 10
Press any key to continue . . .
```