**Problem 3:** Write a program to implement the shortest job first non-preemptive 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 gantt(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].complete\_time);

    }

    printf("\n");

}

void swap(process arr[],int ind1, int ind2){

    process temp = arr[ind1];

    arr[ind1] = arr[ind2];

    arr[ind2] = temp;

}

void main()

{

    int n =0,i,ct=0, mt=0,j, temp;

    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

   for(j=0;j<n;j++){

        mt=arr[j].burst\_time;

        for(i=j+1;i<n;i++){

            if(arr[i].arrival\_time<=ct && arr[i].burst\_time<mt){

                swap(arr,j,i);

            }

            if(ct<arr[i].arrival\_time){

                break;

            }

        }

        if(j==0){

            temp=0;

        }else{

            temp = arr[j-1].complete\_time;

        }

        arr[j].complete\_time=arr[j].burst\_time+temp;

        arr[j].turn\_around\_time = arr[j].complete\_time -  arr[j].arrival\_time;

        arr[j].wait\_time = arr[j].turn\_around\_time - arr[j].burst\_time;

        arr[j].response\_time = arr[j].wait\_time;

        ct=arr[j].complete\_time;

   }

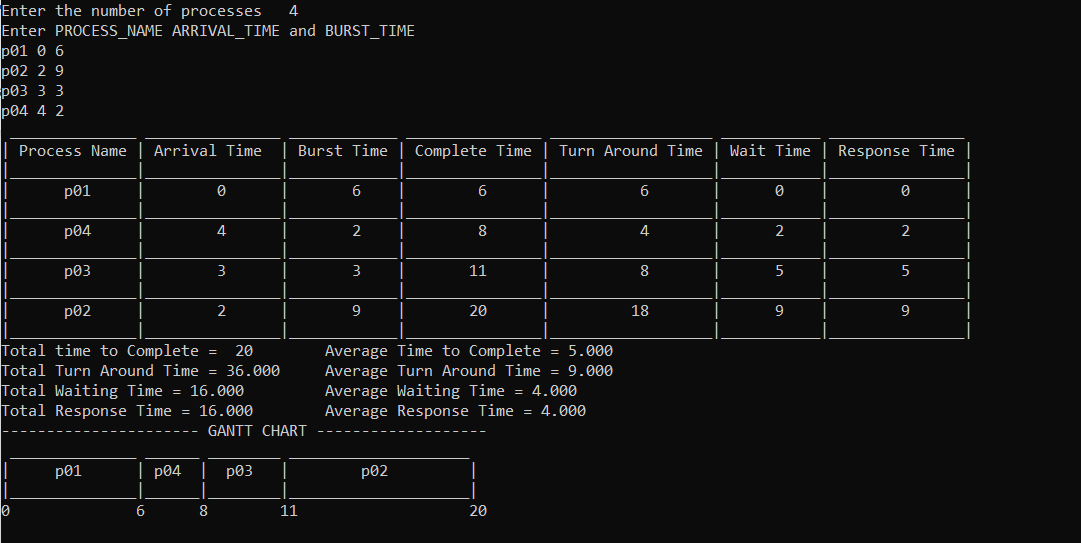
    print\_process\_table(arr,n);

    get\_average(arr, n);

    puts("---------------------- GANTT CHART -------------------");

    gantt(arr,n);

}

Output: