#include<stdio.h>

#include<stdlib.h>

#define MAX 10

typedef struct

{

int pid;

int arrival\_time;

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);

float sum\_waiting\_time,sum\_turnaround\_time;

int main()

{

Process p[MAX];

int i,j,n;

printf("Enter total no.of processes: ");

scanf("%d",&n);

printf("Enter the arrival time for each processes:\n");

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

{

p[i].pid=i;

printf("p[%d] : ",i);

scanf("%d",&p[i].arrival\_time);

p[i].waiting\_time=p[i].turnaround\_time=0;

}

printf("Enter the burst time for each processes:\n");

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

{

printf("p[%d] : ",i);

scanf("%d",&p[i].burst\_time);

}

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].arrival\_time;

p[i].turnaround\_time= p[i].waiting\_time+p[i].burst\_time;

}

//print table

printf("\n");

print\_table(p,n);

printf("\n");

//print Gantt Chart

printf(" GANTT CHART \n");

printf(" \*\*\*\*\*\*\*\*\*\*\* \n");

print\_gantt\_chart(p,n);

avg\_TAT\_WT(p,n);

return 0;

}

void print\_table(Process p[], int n)

{

int i;

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

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

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

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

printf("| %d | %d | %d | %d | %d |\n"

, p[i].pid, p[i].arrival\_time, 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));

}

Output

