ARYAMAN MISHRA

19BCE1027

1.a)i) #include<stdio.h>

void print\_gantt\_chart(int n,int processes[],int bt[],int tat[],int p[])

{

int i, j;

// print top bar

printf(" ");

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

for(j=0; j<bt[i]; j++) printf("--");

printf(" ");

}

printf("\n|");

// printing process id in the middle

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

for(j=0; j<bt[i] - 1; j++) printf(" ");

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

for(j=0; j<bt[i] - 1; j++) printf(" ");

printf("|");

}

printf("\n ");

// printing bottom bar

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

for(j=0; j<bt[i]; j++) printf("--");

printf(" ");

}

printf("\n");

// printing the time line

printf("0");

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

for(j=0; j<bt[i]; j++) printf(" ");

if(tat[i] > 9) printf("\b"); // backspace : remove 1 space

printf("%d",tat[i]);

}

printf("\n");

}

void findWaitingTime(int processes[], int n,

int bt[], int wt[])

{

wt[0] = 0;

int i;

for(i=1;i<n;i++)

wt[i]=bt[i-1]+wt[i-1] ;

}

void findTurnAroundTime( int processes[], int n,

int bt[], int wt[], int tat[],int p[])

{

int i;

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

tat[i] = bt[i] + wt[i]; }

print\_gantt\_chart(n,processes,bt,tat,p);

}

void findavgTime( int processes[], int n, int bt[],int p[])

{

int wt[n], tat[n], total\_wt = 0, total\_tat = 0,i;

findWaitingTime(processes, n, bt, wt);

findTurnAroundTime(processes, n, bt, wt, tat,p);

printf("Patient ID Arrival Time Burst time Waiting time Turn around time\n");

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

{

total\_wt = total\_wt + wt[i];

total\_tat = total\_tat + tat[i];

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

printf(" 6:0%d ",processes[i]);

printf(" %d ", bt[i] );

printf(" %d",wt[i] );

printf(" %d\n",tat[i] );

}

float s=(float)total\_wt / (float)n;

float t=(float)total\_tat / (float)n;

printf("Average waiting time = %f",s);

printf("\n");

printf("Average turn around time = %f ",t);

}

int main()

{

int n,i;

printf("Enter number of process:");

scanf("%d",&n);

int processes[n],burst\_time[n],p[n];

printf("Enter time in minutes,process id and burst time.\n");

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

{

printf("Patient ID :");

scanf("%d",&p[i]);

printf("Arrival Time :");

scanf("%d",&processes[i]);

printf("Expected Consultation Time (in minutes):");

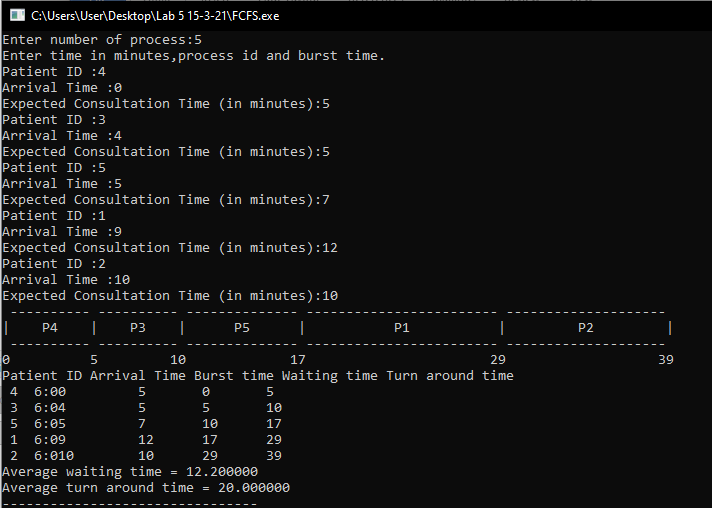
scanf("%d",&burst\_time[i]);

}

findavgTime(processes, n, burst\_time,p);

return 0;

}



ii) #include<stdio.h>

void print\_gantt\_chart(int n,int processes[],int bt[],int tat[],int wt[])

{

int i, j;

int last = bt[n-1] + ( n== 1 ? 0 : wt[n-1]);

// printing top bar

printf(" ");

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

for(j=0; j<bt[i]; j++) printf("--");

printf(" ");

}

printf("\n|");

// middle position

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

for(j=0; j<bt[i]-1; j++) printf(" ");

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

for(j=0; j<bt[i]-1; j++) printf(" ");

printf("|");

}

printf("\n ");

// bottom bar

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

for(j=0; j<bt[i]; j++) printf("--");

printf(" ");

}

printf("\n");

// printing waiting time

int minus = 0;

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

if(wt[i]>9) printf(" ");

printf("%d", wt[i]);

if(wt[i+1]>9){

minus = 1;

}

if(i+1 == n ) if (last>9) minus = 1;

for(j=0; j<bt[i]-minus; j++) printf(" ");

}

if(last>9) printf(" ");

printf("%d\n", last);

}

int main()

{

int n;

printf("Enter number of process:");

scanf("%d",&n);

int bt[n],p[n],wt[n],tat[n],i,j,total=0,pos,temp,processes[n];

float avg\_wt,avg\_tat;

printf("\nEnter Process ID and Burst Time:\n");

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

{

printf("Patient ID :");

scanf("%d",&processes[i]);

printf("Arrival Time:");

scanf("%d",&p[i]);

printf("Burst time:");

scanf("%d",&bt[i]);

}

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

{

pos=i;

for(j=i+1;j<n;j++)

{

if(bt[j]<bt[pos])

pos=j;

}

temp=bt[i];

bt[i]=bt[pos];

bt[pos]=temp;

temp=p[i];

p[i]=p[pos];

p[pos]=temp;

}

wt[0]=0;

for(i=1;i<n;i++)

{

wt[i]=0;

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

wt[i]+=bt[j];

total+=wt[i];

}

avg\_wt=(float)total/n;

total=0;

printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time\t Patient ID");

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

{

tat[i]=bt[i]+wt[i];

total+=tat[i];

printf("\np%d\t\t %d\t\t %d\t\t\t%d\t\t6:0%d\n",processes[i],bt[i],wt[i],tat[i],p[i]);

}

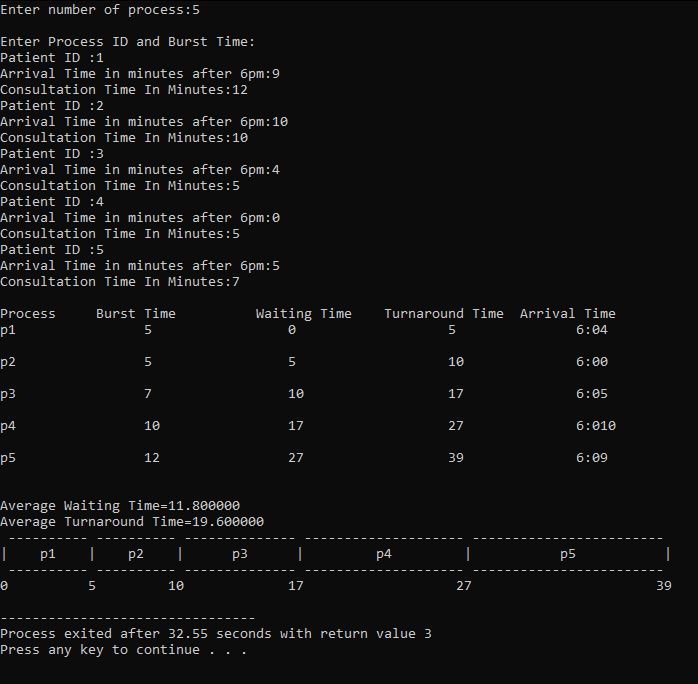
avg\_tat=(float)total/n;

printf("\n\nAverage Waiting Time=%f",avg\_wt);

printf("\nAverage Turnaround Time=%f\n",avg\_tat);

print\_gantt\_chart(n,p,bt,tat,wt);

}



Shortest Job First results in minimum average waiting time.

2.A) #include<stdio.h>

void print\_gantt\_chart(int n,int processes[],int bt[],int tat[],int p[])

{

int i, j;

// print top bar

printf(" ");

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

for(j=0; j<bt[i]; j++) printf("--");

printf(" ");

}

printf("\n|");

// printing process id in the middle

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

for(j=0; j<bt[i] - 1; j++) printf(" ");

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

for(j=0; j<bt[i] - 1; j++) printf(" ");

printf("|");

}

printf("\n ");

// printing bottom bar

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

for(j=0; j<bt[i]; j++) printf("--");

printf(" ");

}

printf("\n");

// printing the time line

printf("0");

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

for(j=0; j<bt[i]; j++) printf(" ");

if(tat[i] > 9) printf("\b"); // backspace : remove 1 space

printf("%d",tat[i]);

}

printf("\n");

}

void findWaitingTime(int processes[], int n,

int bt[], int wt[])

{

wt[0] = 0;

int i;

for(i=1;i<n;i++)

wt[i]=bt[i-1]+wt[i-1] ;

}

void findTurnAroundTime( int processes[], int n,

int bt[], int wt[], int tat[],int p[])

{

int i;

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

tat[i] = bt[i] + wt[i]; }

print\_gantt\_chart(n,processes,bt,tat,p);

}

void findavgTime( int processes[], int n, int bt[],int p[])

{

int wt[n], tat[n], total\_wt = 0, total\_tat = 0,i;

findWaitingTime(processes, n, bt, wt);

findTurnAroundTime(processes, n, bt, wt, tat,p);

printf("Family ID Arrival Time Burst time Waiting time Turn around time\n");

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

{

total\_wt = total\_wt + wt[i];

total\_tat = total\_tat + tat[i];

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

printf(" 7:0%d ",processes[i]);

printf(" %d ", bt[i] );

printf(" %d",wt[i] );

printf(" %d\n",tat[i] );

}

float s=(float)total\_wt / (float)n;

float t=(float)total\_tat / (float)n;

printf("Average waiting time = %f",s);

printf("\n");

printf("Average turn around time = %f ",t);

}

int main()

{

int n,i;

printf("Enter number of process:");

scanf("%d",&n);

int processes[n],burst\_time[n],p[n];

printf("Enter time in minutes,process id and burst time.\n");

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

{

printf("Family ID :");

scanf("%d",&p[i]);

printf("Arrival Time :");

scanf("%d",&processes[i]);

printf("Expected Dining time (in minutes):");

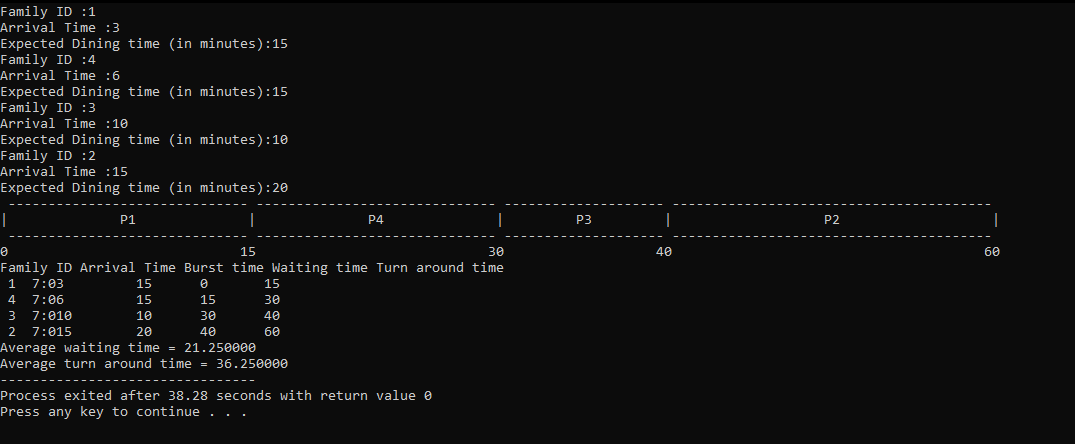
scanf("%d",&burst\_time[i]);

}

findavgTime(processes, n, burst\_time,p);

return 0;

}



2)b) #include<stdio.h>

void print\_gantt\_chart(int n,int processes[],int bt[],int tat[],int wt[],int p[])

{

int i, j;

int last = bt[n-1] + ( n== 1 ? 0 : wt[n-1]);

// printing top bar

printf(" ");

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

for(j=0; j<bt[i]; j++) printf("--");

printf(" ");

}

printf("\n|");

// middle position

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

for(j=0; j<bt[i]-1; j++) printf(" ");

printf("p%d",(i+1));

for(j=0; j<bt[i]-1; j++) printf(" ");

printf("|");

}

printf("\n ");

// bottom bar

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

for(j=0; j<bt[i]; j++) printf("--");

printf(" ");

}

printf("\n");

// printing waiting time

int minus = 0;

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

if(wt[i]>9) printf(" ");

printf("%d", wt[i]);

if(wt[i+1]>9){

minus = 1;

}

if(i+1 == n ) if (last>9) minus = 1;

for(j=0; j<bt[i]-minus; j++) printf(" ");

}

if(last>9) printf(" ");

printf("%d\n", last);

}

int main()

{

int n;

printf("Enter number of process:");

scanf("%d",&n);

int bt[n],p[n],wt[n],tat[n],i,j,total=0,pos,temp,processes[n];

float avg\_wt,avg\_tat;

printf("\nEnter Process ID and Burst Time:\n");

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

{

printf("Family ID :");

scanf("%d",&processes[i]);

printf("Arrival Time in minutes after 7pm:");

scanf("%d",&p[i]);

printf("Consultation Time In Minutes:");

scanf("%d",&bt[i]);

}

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

{

pos=i;

for(j=i+1;j<n;j++)

{

if(bt[j]<bt[pos])

pos=j;

}

temp=bt[i];

bt[i]=bt[pos];

bt[pos]=temp;

temp=p[i];

p[i]=p[pos];

p[pos]=temp;

}

wt[0]=0;

for(i=1;i<n;i++)

{

wt[i]=0;

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

wt[i]+=bt[j];

total+=wt[i];

}

avg\_wt=(float)total/n;

total=0;

printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time\t Arrival Time");

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

{

tat[i]=bt[i]+wt[i];

total+=tat[i];

printf("\np%d\t\t %d\t\t %d\t\t\t%d\t\t7:0%d\n",processes[i],bt[i],wt[i],tat[i],p[i]);

}

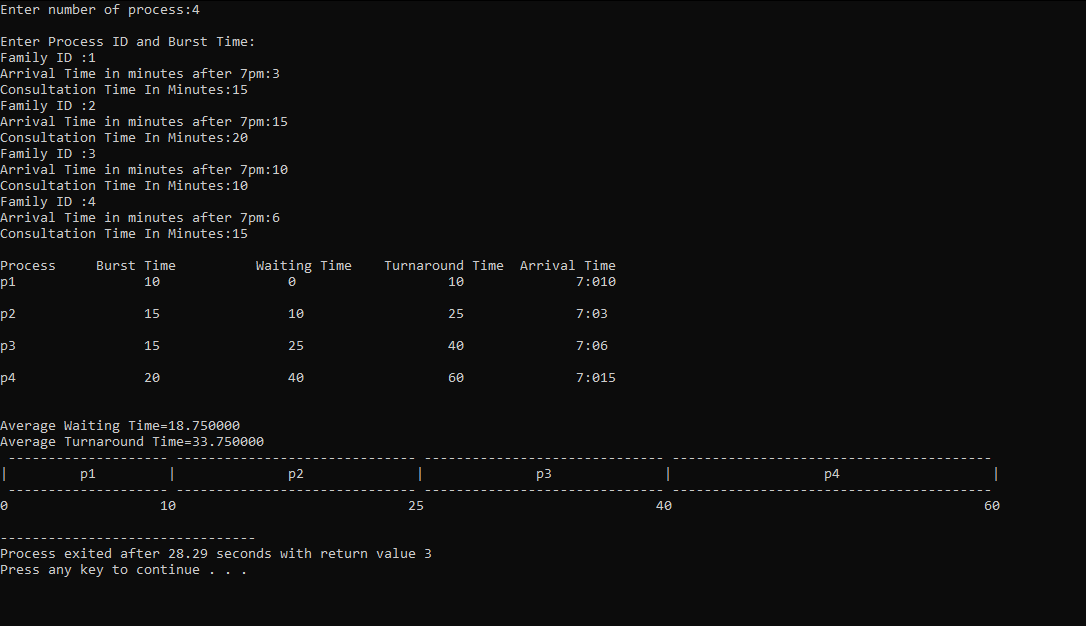
avg\_tat=(float)total/n;

printf("\n\nAverage Waiting Time=%f",avg\_wt);

printf("\nAverage Turnaround Time=%f\n",avg\_tat);

print\_gantt\_chart(n,p,bt,tat,wt,p);

}



Shortest Job First results in minimum average waiting time.