**PROGRAM**

#include<stdio.h>

#include<stdlib.h>

struct pr

{

int ID;

int bt;

int at;

};

void main()

{

int i,j,n,ct=0;

float awt=0,atat=0;

printf("Enter the number of processes :");

scanf("%d",&n);

struct pr p[n],swap;

printf("\n");

printf("Enter the arrival time and burst time of process :");

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

{

printf("\nP%d :",(i+1));

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

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

p[i].ID=i+1;

}

for(i=0;i<=n-1;i++)

{

for(j=0;j<n-1-i;j++)

{

if(p[j].at>p[j+1].at)

{

swap=p[j];

p[j]=p[j+1];

p[j+1]=swap;

}

}

}

int wt[n],tat[n];

printf("\nGantt chart\n");

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

printf("----");

printf("\n|");

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

{

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

}

printf("\n");

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

printf("----");

printf("\n0");

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

{

ct+=p[i].bt;

printf(" %d",ct);

}

printf("\nID\tAT\tBT\tCT\tTAT\tWT\n");

ct=0;

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

{

ct=ct+p[i].bt;

tat[i]=ct-p[i].at;

wt[i]=tat[i]-p[i].bt;

printf("\n%d\t%d\t%d\t%d\t%d\t%d\n",p[i].ID,p[i].at,p[i].bt,ct,tat[i],wt[i]);

atat+=tat[i];

awt+=wt[i];

}

atat/=n;

awt/=n;

printf("\nAverage turnaroundtime = %.2f\n",atat);

printf("\nAverage waitingtime = %.2f\n",awt);

}

**OUTPUT**

vishnunadh10733@ubuntu-H410M-H-V2:~/vishnunadh62$ gedit FCFS.c

vishnunadh10733@ubuntu-H410M-H-V2:~/vishnunadh62$ gcc FCFS.c

vishnunadh10733@ubuntu-H410M-H-V2:~/vishnunadh62$ ./a.out

Enter the number of processes :5

Enter the arrival time and burst time of process :

P1 :0 8

P2 :1 4

P3 :2 9

P4 :3 5

P5 :4 2

Gantt chart

--------------------

|P1 |P2 |P3 |P4 |P5 |

--------------------

0 8 12 21 26 28

ID AT BT CT TAT WT

1 0 8 8 8 0

2 1 4 12 11 7

3 2 9 21 19 10

4 3 5 26 23 18

5 4 2 28 24 22

Average turnaroundtime = 17.00

Average waitingtime = 11.40

**PROGRAM**

#include<stdio.h>

#include<stdlib.h>

int ct=0,n,b;

struct pr

{

int ID;

int bt;

int at;

int comp;

}p[10],swap;

int bursttime();

void main()

{

int i,j;

float awt,atat;

printf("Enter the number of processes :");

scanf("%d",&n);

printf("\n");

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

p[i].comp=0;

printf("Enter the arrival time and burst time of process :");

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

{

printf("\nP%d :",(i+1));

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

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

p[i].ID=i+1;

}

for(i=0;i<=n-1;i++)

{

for(j=0;j<n-1-i;j++)

{

if(p[j].at>p[j+1].at)

{

swap=p[j];

p[j]=p[j+1];

p[j+1]=swap;

}

}

}

int wt[n],tat[n];

printf("\nGantt chart\n");

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

printf("----");

printf("\n|");

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

{

b=bursttime();

printf("P%d |",p[b].ID);

ct=ct+p[b].bt;

p[b].comp=1;

}

printf("\n");

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

printf("----");

printf("\n0");

ct=0;

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

p[i].comp=0;

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

{

b=bursttime();

ct=ct+p[b].bt;

p[b].comp=1;

printf(" %d",ct);

}

printf("\n");

printf("\nID\tAT\tBT\tCT\tTAT\tWT\n");

ct=0;

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

p[i].comp=0;

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

{

b=bursttime();

p[b].comp=1;

ct=ct+p[b].bt;

tat[b]=ct-p[b].at;

wt[b]=tat[b]-p[b].bt;

printf("\n%d\t%d\t%d\t%d\t%d\t%d\n",p[b].ID,p[b].at,p[b].bt,ct,tat[b],wt[b]);

atat+=tat[b];

awt+=wt[b];

}

atat/=n;

awt/=n;

printf("\nAverage turnaroundtime = %.2f\n",atat);

printf("\nAverage waitingtime = %.2f\n",awt);

}

int bursttime()

{

int small;

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

{

if(p[i].at<=ct&&p[i].comp==0)

{

small=i;

break;

}

}

for(int k=0;k<n;k++)

{

if(p[small].bt>p[k].bt)

{

if(p[k].at<=ct&&p[k].comp==0)

small=k;

}

}

return small;

}

**OUTPUT**

vishnunadh10733@ubuntu-H410M-H-V2:~/vishnunadh62$ gedit SJF.c

vishnunadh10733@ubuntu-H410M-H-V2:~/vishnunadh62$ gcc SJF.c

vishnunadh10733@ubuntu-H410M-H-V2:~/vishnunadh62$ ./a.out

Enter the number of processes :4

Enter the arrival time and burst time of process :

P1 :0 2

P2 :1 4

P3 :3 3

P4 :2 7

Gantt chart

----------------

|P1 |P2 |P3 |P4 |

----------------

0 2 6 9 16

ID AT BT CT TAT WT

1 0 2 2 2 0

2 1 4 6 5 1

3 3 3 9 6 3

4 2 7 16 14 7

Average turnaroundtime = 6.75

Average waitingtime = 2.75

**PROGRAM**

#include<stdio.h>

#include<stdlib.h>

int ct=0,n,b;

struct proc

{

int ID;

int bt;

int at;

int comp;

int pt;

}p[10],swap;

int priority();

void main()

{

int i,j;

float awt=0,atat=0;

printf("Enter the number of processes :");

scanf("%d",&n);

printf("\n");

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

p[i].comp=0;

printf("Enter the priority,arrival time and burst time of process :");

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

{

printf("\nP%d :",(i+1));

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

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

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

p[i].ID=i+1;

}

for(i=0;i<=n-1;i++)

{

for(j=0;j<n-1-i;j++)

{

if(p[j].at>p[j+1].at)

{

swap=p[j];

p[j]=p[j+1];

p[j+1]=swap;

}

}

}

int wt[n],tat[n];

printf("\nGantt chart\n");

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

printf("----");

printf("\n|");

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

{

b=priority();

printf("P%d |",p[b].ID);

ct=ct+p[b].bt;

p[b].comp=1;

}

printf("\n");

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

printf("----");

printf("\n0");

ct=0;

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

p[i].comp=0;

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

{

b=priority();

ct=ct+p[b].bt;

p[b].comp=1;

printf(" %d",ct);

}

printf("\n");

printf("\nID\tPT\tAT\tBT\tCT\tTAT\tWT\n");

ct=0;

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

p[i].comp=0;

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

{

b=priority();

p[b].comp=1;

ct=ct+p[b].bt;

tat[b]=ct-p[b].at;

wt[b]=tat[b]-p[b].bt;

printf("\n%d\t%d\t%d\t%d\t%d\t%d\t%d\n",p[i].ID,p[i].pt,p[i].at,p[i].bt,ct,tat[b],wt[b]);

atat+=tat[b];

awt+=wt[b];

}

atat/=n;

awt/=n;

printf("\nAverage turnaroundtime = %.2f\n",atat);

printf("\nAverage waitingtime = %.2f\n",awt);

}

int priority()

{

int small;

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

{

if(p[i].at<=ct&&p[i].comp==0)

{

small=i;

break;

}

}

for(int k=0;k<n;k++)

{

if(p[small].pt>p[k].pt)

{

if(p[k].at<=ct&&p[k].comp==0)

small=k;

}

}

return small;

}

**OUTPUT**

vishnunadh10733@ubuntu-H410M-H-V2:~/vishnunadh62$ gedit PT.c

vishnunadh10733@ubuntu-H410M-H-V2:~/vishnunadh62$ gcc PT.c

vishnunadh10733@ubuntu-H410M-H-V2:~/vishnunadh62$ ./a.out

Enter the number of processes :5

Enter the priority,arrival time and burst time of process :

P1 :5 0 1

P2 :2 1 3

P3 :1 2 5

P4 : 4 3 7

P5 :3 4 9

Gantt chart

--------------------

|P1 |P2 |P3 |P5 |P4 |

--------------------

0 1 4 9 18 25

ID PT AT BT CT TAT WT

1 5 0 1 1 1 0

2 2 1 3 4 3 0

3 1 2 5 9 7 2

4 4 3 7 18 14 5

5 3 4 9 25 22 15

Average turnaroundtime = 9.40

Average waitingtime = 4.40

**PROGRAM**

#include<stdio.h>

#include<stdlib.h>

struct proc

{

int ID;

int bt;

int at;

int comp;

int bt1;

}p[10],swap;

void main()

{

int i,j,n,qt,completed,cta[10],cta1[10];

float awt=0,atat=0;

printf("Enter the number of processes :");

scanf("%d",&n);

printf("Enter the arrival time and burst time of process :");

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

{

printf("\nP%d :",(i+1));

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

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

p[i].ID=i+1;

p[i].bt1=p[i].bt;

}

printf("Enter the quantum time :");

scanf("%d",&qt);

for(i=0;i<=n-1;i++)

{

for(j=0;j<n-1-i;j++)

{

if(p[j].at>p[j+1].at)

{

swap=p[j];

p[j]=p[j+1];

p[j+1]=swap;

}

}

}

int ct=0,wt[n],tat[n];

printf("\nGantt chart\n");

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

printf("-----------");

printf("\n|");

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

p[i].comp=0;

completed=n;

j=0;

while(completed>0)

{

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

{

if(p[i].bt>qt&&p[i].comp==0)

{

ct+=qt;

cta[j]=ct;

j++;

cta1[i]=ct;

p[i].bt=p[i].bt-qt;

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

}

else if(p[i].bt<=qt&&p[i].comp==0)

{

ct+=p[i].bt;

cta[j]=ct;

j++;

cta1[i]=ct;

p[i].bt=0;

p[i].comp=1;

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

}

}

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

{

if(p[i].comp==1)

completed--;

}

}

printf("\n");

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

printf("-----");

printf("\n0");

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

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

printf("\n");

printf("\nID\tAT\tBT\tCT\tTAT\tWT\n");

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

{

tat[i]=cta1[i]-p[i].at;

wt[i]=tat[i]-p[i].bt1;

printf("\n%d\t%d\t%d\t%d\t%d\t%d\n",p[i].ID,p[i].at,p[i].bt1,cta1[i],tat[i],wt[i]);

atat+=tat[i];

awt+=wt[i];

}

atat/=n;

awt/=n;

printf("\nAverage turnaroundtime = %.2f\n",atat);

printf("\nAverage waitingtime = %.2f\n",awt);

}

**OUTPUT**

vishnunadh10733@ubuntu-H410M-H-V2:~/vishnunadh62$ gedit RR.c

vishnunadh10733@ubuntu-H410M-H-V2:~/vishnunadh62$ gcc RR.c

vishnunadh10733@ubuntu-H410M-H-V2:~/vishnunadh62$ ./a.out

Enter the number of processes :4

Enter the arrival time and burst time of process :

P1 :0 4

P2 :2 6

P3 :4 9

P4 :6 5

Enter the quantum time :3

Gantt chart

--------------------------------------------

| P1 | P2 | P3 | P4 | P1 | P2 | P3 | P4 | P3 |

---------------------------------------------

0 3 6 9 12 13 16 19 21 24

ID AT BT CT TAT WT

1 0 4 13 13 9

2 2 6 16 14 8

3 4 9 24 20 11

4 6 5 21 15 10

Average turnaroundtime = 15.50

Average waitingtime = 9.50