**Program:**

#include<stdio.h>

#define MAX 20

void main()

{

int i,n,bt[MAX],wt[MAX],tat[MAX],at[MAX],twt=0,ttat=0;

float awt,atat;

printf("\n Enter the proccess to be executed:");

scanf("%d",&n);

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

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

{

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

wt[i]=0;

}

printf("\n Enter arival time of each process:");

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

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

printf("\n Waiting time of process 1 is 0");

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

{

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

printf("\n Waiting time of process %d is %d ",i+1,wt[i]);

}

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

twt=twt+wt[i];

awt=(twt\*1.0)/n;

printf("\n Average waiting time of the processes are %f",awt);

tat[0]=wt[0]+bt[0];

printf("\n Turnaround time of process %d is %d",1,tat[0]);

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

{

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

printf("\n Turnaround time of process %d is %d",i+1,tat[i]);

}

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

ttat=ttat+tat[i];

atat=(ttat\*1.0)/n;

printf("\n Average turnaround time of the processes are %f",atat);

}

**Output:**

**Program:**

#include<stdio.h>

int main()

{

int bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,total=0,pos,temp,avg\_wt,avg\_tat;

printf("Enter Total Number of Process:");

scanf("%d",&n);

printf("\nEnter Burst Time and Priority\n");

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

{

printf("\nP[%d]\n",i+1);

printf("Burst Time:");

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

printf("Priority:");

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

p[i]=i+1;

}

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

{

pos=i;

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

{

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

pos=j;

}

temp=pr[i];

pr[i]=pr[pos];

pr[pos]=temp;

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=total/n;

total=0;

printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround 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",p[i],bt[i],wt[i],tat[i]);

}

avg\_tat=total/n;

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

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

return 0;

}

**Output:**