• PRIORITY

1. Declare the variables
2. Declare the variable i,j as integer,totaltatime and totalwtime is equal to zero
3. Get the value of n and assign burst time for each process
4. Assign wtime[0] as zero and tatime[0] as btime[0] and inside the loop calculate wait time and turnaround time
5. Calculate total waiting time and total turnaround time and calculate average waiting time and average turnaround time by dividing it by totalnumber of process
6. Print total waiting time, total turnaround time, average waiting time, and average turnaround time
7. Stop the program

PROGRAM CODE

```
#include<stdio.h>
#include<stdlib.h>
struct process
{
        int no,bt,at,tat,wt,ct,prior,id;
}p[20];
int ready,n,a,ct,b,t,i,j,q[50],f=-1,r=-1;
float sum_wt=0.0,sum_tat=0.0,avg_wt,avg_tat;
void sort(int n)
{
struct process temp;
for(i=0;i<n-1;i++)
{
 for(j=0;j<n-i-1;j++)
        {
        if(p[j].at>p[j+1].at)
        {
                temp=p[j];
                p[j]=p[j+1];
                p[j+1]=temp;
        }   }  }
}
void FCFS(){
int flag;
```

```
printf("\nEnter the number of processes : ");
scanf("%d",&n);
  for(i=0;i<n;i++)
  {
  printf("\nEnter arrival time and burst time of process P%d : ",i);
  scanf("%d%d",&p[i].at,&p[i].bt);

   p[i].no=i+1;
  }
  sort(n);
   p[0].ct=p[0].at+p[0].bt;
        for(i=1;i<n;i++)
        {
                if(p[i].at>p[i-1].ct)
                {
                        p[i].ct=p[i].at+p[i].bt;
                }
                else
                {
                        p[i].ct=p[i-1].ct+p[i].bt;
                }
}
```

```
for(i=0;i<n;i++)
        {
        p[i].tat=p[i].ct-p[i].at;
                p[i].wt=p[i].tat-p[i].bt;
sum_wt+=p[i].wt;
sum_tat+=p[i].tat;
}
avg_wt=sum_wt/n;
avg_tat=sum_tat/n;
printf("\nPROCESS\t ARRIVAL TIME \t BURST TIME \t TURNARROUND TIME
 \t WAITING TIME\n");
for(i=0;i<n;i++)


{
printf("%d\t\t%d\t\t%d\t\t%d\t\t%d\n",p[i].no,p[i].at,p[i].bt,p[i].tat,p[i].wt);
}
printf("\nAverage waiting time is %.2f\n\n",avg_wt);
printf("Average turnarround time %.2f\n\n",avg_tat);
}

void SJF()
{
int count=0,t=0,short_p,temp[10],n,i;

floattotal_wt=0,total_tat=0,awt,atat;
        printf("\nEnter the number of
        proceses:\n"); scanf("%d",&n);
        for(i=0;i<n;i++)
        {
        printf("\nEnter arrival time and burst time of process P%d : ",i);
        scanf("%d%d",&p[i].at,&p[i].bt);
        temp[i]=p[i].bt;

        }
        p[19].bt=10000;
        for(t=0;count!=n;t++)
        {
```

```
        short_p=19;
    for(i=0;i<n;i++)
            {
                if(p[i].bt<p[short_p].bt&& (p[i].at<=t && p[i].bt>0))
                {
                    short_p=i;
                }

            }
            p[short_p].bt=p[short_p].bt-1;
            if(p[short_p].bt==0)
            {
                count++;
                p[short_p].wt=t+1-p[short_p].at-temp[short_p];
        p[short_p].tat=t+1-p[short_p].at;

        total_wt+=p[short_p].wt;
        total_tat+=p[short_p].tat;
        }
        }
        awt=total_wt/n;
        atat=total_tat/n;
     printf("process , wt, tat\n");
        for(i=0;i<n;i++)
        {
            printf("%d\t%d\t%d\n",i+1,p[i].wt,p[i].tat);
        }

        printf("Average waiting time :%.2f\n",awt);
        printf("\nAverage turnarround time:%.2f\n",atat);


    }
    void RR()
    {
    int queue[100];
```

```
  int F=-1;
  int R=-1;
          void insert(int n)
          {
if(F==-1)
F=0; R+=1;
                  queue[R]=n;
          }
          int delete()
          {
                  int n;
                  n=queue[F];
                  F+=1;
                  return n;
          }
                  int n,TQ,a,time=0;
                  int temp[10],exist[10]={0};
                  float total_wt=0,total_tat=0,avg_wt,avg_tat;
                  printf("\nEnter the number of process:\n");
  scanf("%d",&n);
                  for(int i=0;i<n;i++)

                  {

                           printf("\nEnter arrival time and burst time of process P%d : ",i);
                          scanf("%d%d",&p[i].at,&p[i].bt); p[i].id=i;

                          temp[i]=p[i].bt;
                  }printf("\nEnter the time quantum:\n");
                  scanf("%d",&TQ);
                  insert(0);
                  exist[0]=1;
```

```
        while(F<=R)

        {

            a=delete();

            if(p[a].bt>=TQ)

            {

                p[a].bt=p[a].bt-TQ;

                time+=TQ;

            }

            else

            {

                time+=p[a].bt;

                p[a].bt=0;

            }



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

            {

                if(exist[i]==0 && p[i].at<=time)

                {

                insert(i);

                exist[i]=1;

}

            }

            if(p[a].bt==0)

            {

                p[a].tat=time-p[a].at;

                p[a].wt=p[a].tat-temp[a];

                total_tat=total_tat+p[a].tat;

                total_wt=total_wt+p[a].wt;
            }

            else
```

```
{
                    insert(a);
                }
            }
avg_tat=total_tat/n;
avg_wt=total_wt/n;
            // printing of the answer
            printf("ID WT TAT\n");
            for(int i=0;i<n;i++)
            {
                printf("%d %d %d\n",p[i].id,p[i].wt,p[i].tat);
            }
            printf("Average waiting time of the processes is : %.2f\n",avg_wt);
            printf("\nAverage turn around time of the processes is : %.2f\n\n",avg_tat);
        }
 void Priority(){
        int i,n,temp[20],t,count=0,sp;
        float to_wt=0,to_tat=0,avg_wt,avg_tat;
        printf("Enter the no of processes : ");
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
             printf("\nEnter Arrival time and burst time,priority of process P%d : \n",i);
             scanf("%d%d%d",&p[i].at,&p[i].bt,&p[i].prior);
            p[i].no=i;
            temp[i]=p[i].bt;
        }
        p[9].prior=1000;
        for(t=0;count!=n;t++)
        {
            sp=9;
            for(i=0;i<n;i++)
            {
                if(p[sp].prior>p[i].prior && p[i].at<=t && p[i].bt>0)
                {
                    sp=i;
                }
            }
            p[sp].bt=p[sp].bt-1;
            if(p[sp].bt==0)
            {
                count++;
                p[sp].tat=t+1-p[sp].at;
                p[sp].wt=p[sp].tat-temp[sp];
                to_wt+=p[sp].wt;
                to_tat+=p[sp].tat;
            }
        }
        avg_tat=to_tat/n;
        avg_wt=to_wt/n;
        printf("P\tARRIVAL TIME\tBURST TIME\tWAITING TIME\tTURNARROUND
    TIME\tPRIORITY\n");
        for(i=0;i<n;i++)
        {

    printf("%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\n",i,p[i].at,temp[i],p[i].wt,p[i].tat,p[i].prior);
        }
        printf("Average turnarrounf time : %.2f\n",avg_tat);
        printf("\nAverage waiting time : %.2f\n",avg_wt);
    }
     int main()
    {
        int opt;
        do{
        printf("Enter the choice :\n 1.FCFS\n2.SJF\n3.RR\n4.Priority\n5.Exit\n");
        scanf("%d",&opt);
            switch(opt)
            {
                case 1:
                FCFS();
                break;
                case 2:
                SJF();
                break;
                case 3:
                RR();
                break;
                case 4:
                Priority();
                break;
                case 5:
                printf("Exit");
                break;
                default:
                printf("Enter the choice:");
                break;
        }
    }
    while(opt!=5);
    return 0;
    }
```