

PROGRAM-4

```
#include<stdio.h>
#include<string.h>
struct node{
    char arr[2];
    int arrival;
    int burst;
}a[10],A[10],f[10],f1[10];
struct graph{
    char arr1[2];
    int head;
    int tail;
}g[100],g1[100];
int main()
{
    struct node A1[10];
    struct node temp;
    int
num,range=0,i,min,j,sum,temp1,k,dup,tb=0,k1,l,m,sum1,min1,ho,minr,ma
xc,low=0;
    float ctavg=0,tatavg=0,wtavg=0,rtavg=0;
    int ct[10];
    int tat[10];
    int wt[10];
    int rt[10];
    struct graph temp2[100];
    int arr[10];
    printf("\nEnter the total number of processes : ");
    scanf("%d",&num);
    printf("Enter the Processes name , arrival and burst time \n");
    while(range<num)
    {
        scanf("%s %d
%d",a[range].arr,&a[range].arrival,&a[range].burst);
        range++;
    }
    for(i=0;i<num;i++)
    {
        tb=tb+a[i].burst;
        f[i]=a[i];
    }
    //printf("tb =%d \n",tb);
    for(sum=0;sum<tb;sum++)
    {
        dup=0;
        for(i=0;i<num;i++)
        {
            if(sum>=a[i].arrival)
            {
                if(a[i].burst!=0)
                {
                    A[dup]=a[i];
```

```

        dup++;
    }
}
if(dup > 1)
{
    min=A[0].burst;
    for(k1=1;k1<dup;k1++)
    {
        if(min>A[k1].burst)
        {
            min=A[k1].burst;
        }
    }
    sum1=0;
    for(l=0;l<dup;l++)
    {
        if(A[l].burst==min)
        {
            f1[sum1]=A[l];
            sum1++;
        }
    }
    if(sum1>1)
    {
        min1=f1[0].arr[1]-'0';
        for(m=1;m<sum1;m++)
        {
            if(min1>f1[m].arr[1]-'0')
            {
                min1=f1[m].arr[1]-'0';
            }
        }
        g[sum].arr1[0]=f1[0].arr[0];
        g[sum].arr1[1]=min1+'0';
        g[sum].head=sum+1;
        g[sum].tail=sum;
        for(j=0;j<num;j++)
        {
            if(a[j].arr[1]-'0'==min1)
            {
                a[j].burst=a[j].burst-1;
            }
        }
    }
}
else
{
    g[sum].arr1[0]=f1[0].arr[0];
    g[sum].arr1[1]=f1[0].arr[1];
    g[sum].head=sum+1;
    g[sum].tail=sum;
    for(j=0;j<num;j++)
    {
        if(a[j].arr[1]-'0'==f1[0].arr[1]-'0')

```

```

        {
            a[j].burst=a[j].burst-1;
        }
    }
}
else
{
    g[sum].arr1[0]=A[dup-1].arr[0];
    g[sum].arr1[1]=A[dup-1].arr[1];
    g[sum].head=sum+1;
    g[sum].tail=sum;
    for(j=0;j<num;j++)
    {
        if(a[j].arr[1]-'0'==A[dup-1].arr[1]-'0')
        {
            a[j].burst=a[j].burst-1;
        }
    }
}
}
for(i=1;i<num+1;i++)
{
    ho=0;
    for(j=0;j<tb;j++)
    {
        if(g[j].arr1[1]-'0'==i)
        {
            temp2[ho]=g[j];
            ho++;
        }
    }
    minr=temp2[0].tail;
    maxc=temp2[0].head;
    for(k=1;k<ho;k++)
    {
        if(minr>temp2[k].tail)
        {
            minr=temp2[k].tail;
        }
        if(maxc<temp2[k].head)
        {
            maxc=temp2[k].head;
        }
    }
    ct[i-1]=maxc;
    tat[i-1]=ct[i-1]-a[i-1].arrival;
    rt[i-1]=minr-a[i-1].arrival;
    if(rt[i-1]<0)
    {
        rt[i-1]=0;
    }
    wt[i-1]=tat[i-1]-f[i-1].burst;
    if(wt[i-1]<0)

```

```

        {
            wt[i-1]=0;
        }
    }
    for(i=0;i<num;i++)
    {
        ctavg=ctavg+ct[i];
        tatavg=tatavg+tat[i];
        wtavg=wtavg+wt[i];
        rtavg=rtavg+rt[i];
    }
    printf("\n");
    printf("AVERAGE COMPLETE TIME = %f\n\n",ctavg/num);
    printf("AVERAGE TAT TIME = %f\n\n",tatavg/num);
    printf("AVERAGE WAITING TIME = %f\n\n",wtavg/num);
    printf("AVERAGE RESPONSE TIME = %f\n\n",rtavg/num);
    printf(" process\tarr\tburst\tct\ttat\twt\ttrt\n\n");
    for(j=0;j<num;j++)
    {

printf("%s\t%d\t%d\t%d\t%d\t%d\t%d\n\n",f[j].arr,f[j].arrival,f[j].b
urst,ct[j],tat[j],wt[j],rt[j]);
    }
    printf("\nGhantt chart :\n");
    for(i=0;i<tb;i++)
    {
        if(g[i+1].arr1[1]-'0'==g[i].arr1[1]-'0')
        {
            g[i].head=g[i+1].head;
        }
        else
        {
            g1[low]=g[i];
            low++;
        }
    }
    for(i=0;i<low;i++)
    {
        if(i==0)
        {
            printf("%d %s %d |
",g[0].tail,g1[i].arr1,g1[i].head);
        }
        else
        {
            printf("%d %s %d |
",g1[i-1].head,g1[i].arr1,g1[i].head);
        }
    }
}

```