PROGRAM – 3

```
#include<stdio.h>
#include<string.h>
struct node{
    char arr[2];
    int arrival;
    int burst;
}a[10],A[10],a1[10];
int main()
{
    struct node A1[10];
    struct node temp;
    int num,range=0,i,min,first,k,j,l,m,max,sum,x,z,s,t;
    int ct[10];
    int tat[10];
    int wt[10];
    int rt[10];
    printf("\nEnter the total number of processes : ");
    scanf("%d",&num);
    printf("Enter the arrival and run time of the process \n");
    while(range<num)
    {
        scanf("%s %d
%d",a[range].arr,&a[range].arrival,&a[range].burst);
        range++;
    }
    min=a[0].arrival;
    for(i=1;i<num;i++)
    {
        if(min>a[i].arrival)
        {
            min=a[i].arrival;
        }
    }
    //printf("min=%d\n",min);
    k=0;
    for(i=0;i<num;i++)
    {
        if(a[i].arrival==min)
        {
            A1[k]=a[i];
            k++;
        }
    }
    if(k>1)
    {
        min=A1[0].burst;
        for(i=1;i<k;i++)
        {
            if(min>A1[i].burst)
            {
                min=A1[i].burst;
            A[0]=A1[i];
        }
    }
    sum=A[0].burst+A[0].arrival;
}
else if(k==1)
{
    A[0]=A1[0];
    sum=A[0].burst+A[0].arrival;
}
for(i=0;i<num;i++)
{
    if(strcmp(a[i].arr,A[0].arr) == 0)
    {
        temp=a[0];
        a[0]=a[i];
        a[i]=temp;
    }
}
for(x=1;x<num;x++)
{
    k=0;
    for(i=x;i<num;i++)//i=x;
    {
        if(sum>=a[i].arrival)
        {
            A1[k]=a[i];
            k++;
        }
    }
    //printf("k=%d\n",k);
    if(k>0)
    {
        s=0;
        max=A1[0].burst;
        for(j=1;j<k;j++)
        {
            if(max>A1[j].burst)
            {
                max=A1[j].burst;
                A[x]=A1[j];//A[x];
            }
        }
        for(z=0;z<k;z++)
        {
            if(max==A1[z].burst)
            {
                a1[s]=A1[z];
                s++;
            }
        }
        if(s>1)
        {
            min=a1[0].arrival;
```

```
                for(t=0;t<s;t++)
                {
                    if(a1[s].arrival<=min)
                    {
                        min=a1[s].arrival;
                        A[x]=a1[s];
                    }
                }
            }
            //printf("min=%d\n",min);
            if(k==1)
            {
                A[x]=A1[0];
            }
            sum=sum+A[x].burst;//A[x]
            for(i=0;i<num;i++)
            {
                if(strcmp(a[i].arr,A[x].arr) == 0)//A[x]
                {
                    temp=a[x];//a[x]
                    a[x]=a[i];
                    a[i]=temp;
                }
            }
        }
    }
    int sum1=A[0].arrival;
    for(i=0;i<num;i++)
    {
        sum1=sum1+a[i].burst;
        ct[i]=sum1;
        tat[i]=ct[i]-a[i].arrival;
        wt[i]=tat[i]-a[i].burst;
        if(wt[i]<0)
        {
            wt[i]=0;
        }
    }
    rt[0]=0;
    for(j=1;j<num;j++)
    {
        rt[j]=ct[j-1]-a[j].arrival;
        if(rt[j]<0)
        {
            rt[j]=0;
        }
    }
    float ctavg=0,tatavg=0,wtavg=0,rtavg=0;
    for(i=0;i<num;i++)
    {
        ctavg=ctavg+ct[i];
        tatavg=tatavg+tat[i];
        wtavg=wtavg+wt[i];
        rtavg=rtavg+rt[i];
    }
    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("ID\tat\tbt\tct\ttat\twt\trt\n");
    printf("--\t--\t--\t--\t--\t--\t--\n");
    for(i=1;i<num+1;i++)
    {
        for(j=0;j<num;j++)
        {
            if((a[j].arr[1]-'0')==i)
            {

printf("%s\t%d\t%d\t%d\t%d\t%d\t%d\n",a[j].arr,a[j].arrival,a[j].bur
st,ct[j],tat[j],wt[j],rt[j]);
            }
        }
    }
    printf("\nGhantt chart :\n");
    printf(" | %d  --> ",A[0].arrival);
    for(i=0;i<num;i++)
    {
        printf("  %s  -->  %d |",a[i].arr,ct[i]+A[0].arrival);
    }
    printf("\n");
}
```