PROGRAM – 2

```
#include<stdio.h>
#include<string.h>
int num;
struct node
{
    char process[2];
    int arrival;
    int burst;
}p[10];

struct table
{
    int burst;
    int arrival;
    int complete;
    int waiting;
    int turn_around;
    int response;
}p1[10];
void complete()
{
    int i,j,sum;
    for(i=0;i<num;i++)
    {
        sum=0;
        for(j=0;j<=i;j++)
        {
            sum=sum+p1[j].burst;
        }
        p1[i].complete=sum;
    }
}
void waiting()
{
    int i,j,sum;
    p1[0].waiting=p1[0].response=0;
    for(i=1;i<num;i++)
    {
        sum=0;
        for(j=0;j<i;j++)
        {
            sum=sum+p1[j].burst;
        }
        p1[j].waiting=sum-p1[j].arrival;
        p1[j].response=p1[j].waiting;
    }
}
void turn_around()
{
    int i;
```

```
    for(i=0;i<num;i++)
    {
        p1[i].turn_around=p1[i].burst+p1[i].waiting;
    }
}
int main()
{
    int i,j,min,rem,k,sum;
    struct node temp,p2[10];
    printf("Enter the no of processes to be included ");
    scanf("%d",&num);
    printf("Enter the details of processses \n");
    for(i=0;i<num;i++)
    {
        printf("-->Process = ");
        scanf("%s",p[i].process);
        printf("Arrival time : ");
        scanf("%d",&p[i].arrival);
        printf("Burst time : ");
        scanf("%d",&p[i].burst);
        p2[i]=p[i];
        printf("\n");
    }
    for(i=0;i<num;i++)
    {
        rem=0;
        temp=p[i];
        min=p[i].arrival;
        for(j=i+1;j<num;j++)
        {
            if(min>p[j].arrival)
            {
                rem=j;
                min=p[j].arrival;
            }
        }
        if(rem!=0)
        {
            p[i]=p[rem];
            p[rem]=temp;
        }
    }
    for(i=0;i<num;i++)
    {
        p1[i].arrival=p[i].arrival;
        p1[i].burst=p[i].burst;
    }
    waiting();
    complete();
    turn_around();
    printf("ID\tBT\tAT\tCT\tWT");
    printf("\tTAT\tRT\n");
    printf("\t---\t---\t---\t---");
    printf("\t---\t---\n");
```

```
    for(i=0;i<num;i++)
    {
        for(j=0;j<num;j++)
        {
            if(p2[i].arrival==p[j].arrival)
            {

printf("%s\t%d\t%d\t%d\t%d\t%d\t%d\n\n",p2[i].process,p1[j].burst,p1
[j].arrival,p1[j].complete,p1[j].waiting,p1[j].turn_around,p1[j].res
ponse);
            }
        }
    }
    for(k=0;k<num;k++)
    {
        sum=sum+p1[k].complete;
    }
    printf("Avg CT= %f\n",(float)sum/num);
    sum=0;
    for(k=0;k<num;k++)
    {
        sum=sum+p1[k].waiting;
    }
    printf("Avg WT= %f\n",(float)sum/num);
    sum=0;
    for(k=0;k<num;k++)
    {
        sum=sum+p1[k].turn_around;
    }
    printf("Avg TAT= %f\n",(float)sum/num);
    sum=0;
    for(k=0;k<num;k++)
    {
        sum=sum+p1[k].response;
    }
    printf("Avg RT= %f\n",(float)sum/num);
    printf("gantt chart is represented as\n");
    for(i=0;i<num;i++)
    {
        printf("| %s   ",p[i].process);
    }
    printf("\n0     ");
    for(i=0;i<num;i++)
    {
        printf("%d    ",p1[i].complete);
    }
}
```