Scheduling programs:

1.FCFS – without arrival times:

```
#include<stdio.h>
void main()
{
int i,n;
printf("enter number of processes : \n");
scanf("%d",&n);
int b[n],arr[n],com[n],tur[n],w[n];
float avg,t;
printf("enter burst times of each processes : \n");
for(i=0;i<n;i++)
{
scanf("%d",&b[i]);
}
for(i=0;i<n;i++)
{
arr[i]=0;
}
```

```
int s=0;

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

{

com[i]=s+b[i];

s=com[i];

}

s=0;

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

{

tur[i]=com[i]-arr[i];

s+=tur[i];

}

t=(float)s/n;

s=0;

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

{

w[i]=tur[i]-b[i];

s+=w[i];

}

avg=(float)s/n;
```

```
printf("process\tarrival\tcompletion\tturn around\tburst\twaiting\n");

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

{

printf("    %d\t   %d\t    %d\t\t   %d\t\t  %d\t
%d\n",i+1,arr[i],com[i],tur[i],b[i],w[i]);

}

printf("average waiting time : %.2f\n",avg);

printf("average turn around  : %.2f\n",t);

}
```

2.FCFS with arrival times

```
#include<stdio.h>

void swap(int *x,int *y)

{

   int t=*x;

   *x=*y;

   *y=t;

}

void main()

{

   int n;
```

```
printf("enter no.of proccesses :\n");
scanf("%d",&n);
int arrival[n],burst[n],process[n];
int i,j;
printf("enter arrival times\n");
for(i=0;i<n;i++)
{
    process[i]=i+1;
    scanf("%d",&arrival[i]);
}
printf("enter burst times\n");
for(i=0;i<n;i++)
{
    scanf("%d",&burst[i]);
}
for(i=0;i<n-1;i++)
{
    for(j=i+1;j<n;j++)
    {
        if(arrival[j]<arrival[i])
```

```
            {
                swap(&arrival[i],&arrival[j]);
                swap(&burst[i],&burst[j]);
                swap(&process[i],&process[j]);
            }
        }
    }
    int ct[n],wt[n],tat[n];
    int s=0,sw=0,stat=0;
    for(i=0;i<n;i++)
    {
        ct[i]=burst[i]+s;
        s+=burst[i];
        tat[i]=ct[i]-arrival[i];
        wt[i]=tat[i]-burst[i];
        sw+=wt[i];
        stat+=tat[i];
    }
    printf("process\tarrival\tburst\tcompletion\twaiting\tturn around\n");
    for(i=0;i<n;i++)
    {
        printf("p%d\t  %d\t%d\t  %d\t\t  %d\t
%d\n",process[i],arrival[i],burst[i],ct[i],wt[i],tat[i]);
    }
    printf("average waiting time is     : %.2f\n",(float)sw/n);
    printf("average turn around time is : %.2f\n",(float)stat/n);


}
```

3. SJF without arrival times

```
#include<stdio.h>
void swap(int *x,int *y)
{
    int t=*x;
    *x=*y;
    *y=t;
}
void main()
{
    int n;
```

```
printf("enter no.of processess\n");

scanf("%d",&n);

int arrival[n],burst[n],ct[n],wt[n],tat[n],p[n];

int i,j;

printf("enter burst times\n");

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

{

    p[i]=i+1;

    arrival[i]=0;

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

}

for(i=0;i<n-1;i++)

{

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

    {

        if(burst[j]<burst[i])

        {

            swap(&burst[i],&burst[j]);

            swap(&p[i],&p[j]);

        }
        }
    }
    int s=0,sw=0,stat=0;
  for(i=0;i<n;i++)
    {
        ct[i]=burst[i]+s;
        s+=burst[i];
        tat[i]=ct[i]-arrival[i];
        wt[i]=tat[i]-burst[i];
        sw+=wt[i];
        stat+=tat[i];
    }
    printf("process\tarrival\tburst\tcompletion\twaiting\tturn
around\n");
    for(i=0;i<n;i++)
    {
        printf("p%d\t  %d\t%d\t  %d\t\t  %d\t
%d\n",p[i],arrival[i],burst[i],ct[i],wt[i],tat[i]);
    }
    printf("average waiting time is     : %.2f\n",(float)sw/n);
    printf("average turn around time is : %.2f\n",(float)stat/n);
}
```

4. SJF with arrival times

```
#include<stdio.h>
void swap(int *x,int *y)
{
    int t=*x;
    *x=*y;
    *y=t;
}
void main()
{
    int n;
    printf("enter no.of processess:\n");
    scanf("%d",&n);
    int arrival[n],burst[n],ct[n],wt[n],tat[n],p[n];
    int i,j;
    printf("enter arrival times :\n");
    for(i=0;i<n;i++)
    {
        p[i]=i+1;

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

    }

    printf("enter burst times :\n");

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

    {

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

    }

    int min=arrival[0],flag=0;

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

    {

        if(arrival[i]<min)

        {

            min=arrival[i];

            flag=i;

        }

}

    swap(&arrival[0],&arrival[flag]);

    swap(&p[0],&p[flag]);

    swap(&burst[0],&burst[flag]);
```

```
int t=burst[0];

for(i=1;i<n-1;i++)
{
    min=1000;
    for(j=i;j<n;j++)
    {
    if(burst[j]<min && arrival[j]<=t)
    {
        min=burst[j];
        flag=j;
    }
    }
swap(&arrival[i],&arrival[flag]);
swap(&p[i],&p[flag]);
swap(&burst[i],&burst[flag]);
t+=burst[i];
}
int s=arrival[0],sw=0,stat=0;
for(i=0;i<n;i++)
```

```
    {
        ct[i]=burst[i]+s;
        s+=burst[i];
tat[i]=ct[i]-arrival[i];
        wt[i]=tat[i]-burst[i];
        sw+=wt[i];
        stat+=tat[i];
    }
    printf("process\tarrival\tburst\tcompletion\twaiting\tturn
around\n");
    for(i=0;i<n;i++)
    {
        printf("p%d\t  %d\t%d\t  %d\t\t  %d\t
%d\n",p[i],arrival[i],burst[i],ct[i],wt[i],tat[i]);
    }
    printf("average waiting time is     : %.2f\n",(float)sw/n);
    printf("average turn around time is : %.2f\n",(float)stat/n);
}
```

5.Priority scheduling

```
#include<stdio.h>
```

```
void swap(int *x,int *y)

{

int t=*x;

*x=*y;

*y=t;

}

void main()

{

int n;

printf("enter no.of proccesses\n");

scanf("%d",&n);

int arrival[n],burst[n],ct[n],wt[n],tat[n],p[n],pr[n];

int i,j;

printf("enter arrival times\n");

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

{

p[i]=i+1;

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

}

printf("enter burst times\n");
```

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

{

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

}

printf("enter priorities\n");

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

{

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

}

int min=arrival[0],flag=0;

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

{

if(arrival[i]<min)

{

min=arrival[i];

flag=i;

}

}

swap(&arrival[0],&arrival[flag]);

swap(&p[0],&p[flag]);
swap(&burst[0],&burst[flag]);
swap(&pr[0],&pr[flag]);
int t=burst[0];
for(i=1;i<n-1;i++)
    {
        min=1000;
        for(j=i;j<n;j++)
        {
        if(pr[j]<min && arrival[j]<=t)
        {
            min=pr[j];
            flag=j;
        }
        }
    swap(&arrival[i],&arrival[flag]);
    swap(&p[i],&p[flag]);
    swap(&burst[i],&burst[flag]);
    swap(&pr[i],&pr[flag]);
    t+=burst[i];
     }
```

```
int s=0,sw=0,stat=0;

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

    {

        ct[i]=burst[i]+s;

        s+=burst[i];

        tat[i]=ct[i]-arrival[i];

        wt[i]=tat[i]-burst[i];

        sw+=wt[i];

        stat+=tat[i];

    }

    printf("process\tarrival\tburst\tcompletion\twaiting\tturn
around\n");

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

    {

        printf("p%d\t  %d\t%d\t  %d\t\t  %d\t
%d\n",p[i],arrival[i],burst[i],ct[i],wt[i],tat[i]);

    }

    printf("average waiting time is     : %.2f\n",(float)sw/n);

    printf("average turn around time is : %.2f\n",(float)stat/n);

}
```