

# DISK SCHEDULING ALGORITHMS

```
#include<stdio.h>
#include<stdlib.h>
void FCFS(){
    int
    RQ[100],i,n,TotalHeadMoment=0,initial;
    printf("Enter the number of Requests\n");
    scanf("%d",&n); printf("Enter the Requests
    sequence\n"); for(i=0;i<n;i++)
        scanf("%d",&RQ[i]);
    printf("Enter initial head position\n");
    scanf("%d",&initial);

    // logic for FCFS disk scheduling

    for(i=0;i<n;i++)
    {
        TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial);
        initial=RQ[i];
    }

    printf("Total head moment is %d",TotalHeadMoment);
}

void SCAN(){
    int
    RQ[100],i,j,n,TotalHeadMoment=0,initial,size,move;
    printf("Enter the number of Requests\n");
    scanf("%d",&n); printf("Enter the Requests
    sequence\n"); for(i=0;i<n;i++)
        scanf("%d",&RQ[i]); printf("Enter
    initial head position\n");
```

```
scanf("%d",&initial); printf("Enter  
total disk size\n");  
scanf("%d",&size); printf("Enter the  
head movement direction for high 1  
and for low 0\n");  
scanf("%d",&move);
```

```
// logic for Scan disk scheduling
```

```
/*logic for sort the request array */  
for(i=0;i<n;i++)  
{ for(j=0;j<n-i-1;j++)  
  {  
    if(RQ[j]>RQ[j+1])  
    {  
      int temp;  
      temp=RQ[j];  
      RQ[j]=RQ[j+1];  
      RQ[j+1]=temp;  
    }  
  }  
}
```

```
int index;  
for(i=0;i<n;i++)  
{ if(initial<RQ[i])  
  {  
    index=i;  
    break;  
  }  
}
```

```
// if movement is towards high value
```

```

if(move==1)
{ for(i=index;i<n;i++)
{
    TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial);
    initial=RQ[i];
}
// last movement for max size
TotalHeadMoment=TotalHeadMoment+abs(size-RQ[i-1]-
1); initial = size-1; for(i=index-1;i>=0;i--)
{
    TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial);
    initial=RQ[i];

}
}
// if movement is towards low value
else
{ for(i=index-1;i>=0;i--)
{
    TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial);
    initial=RQ[i];
}
// last movement for min size
TotalHeadMoment=TotalHeadMoment+abs(RQ[i+1]-
0); initial =0; for(i=index;i<n;i++)
{
    TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial);
    initial=RQ[i];

}
}
printf("Total head movement is %d",TotalHeadMoment);
}
void C_SCAN(){

```

```

int
RQ[100],i,j,n,TotalHeadMoment=0,initial,size,move;
printf("Enter the number of Requests\n");
scanf("%d",&n); printf("Enter the Requests
sequence\n"); for(i=0;i<n;i++)
    scanf("%d",&RQ[i]);
printf("Enter initial head position\n"); scanf("%d",&initial); printf("Enter
total disk size\n"); scanf("%d",&size); printf("Enter the head
movement direction for high 1 and for low 0\n"); scanf("%d",&move);

```

```

// logic for C-Scan disk scheduling

```

```

    /*logic for sort the request array */
    for(i=0;i<n;i++)
    { for( j=0;j<n-i-1;j++)
        {
            if(RQ[j]>RQ[j+1])
            {
                int temp;
                temp=RQ[j];
                RQ[j]=RQ[j+1];
                RQ[j+1]=temp;
            }
        }
    }
    int index;
    for(i=0;i<n;i++)
    { if(initial<RQ[i])
        {
            index=i;
            break;
        }
    }
}

```

```

// if movement is towards high value
if(move==1)
{ for(i=index;i<n;i++)
{
    TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial);
    initial=RQ[i];
}
// last movement for max size
TotalHeadMoment=TotalHeadMoment+abs(size-RQ[i-1]-1);
/*movement max to min disk */
TotalHeadMoment=TotalHeadMoment+abs(size-1-
0); initial=0; for( i=0;i<index;i++)
{
    TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial);
    initial=RQ[i];

}
}
// if movement is towards low value
else
{ for(i=index-1;i>=0;i--)
{
    TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial);
    initial=RQ[i];
}
// last movement for min size
TotalHeadMoment=TotalHeadMoment+abs(RQ[i+1]-0);
/*movement min to max disk */
TotalHeadMoment=TotalHeadMoment+abs(size-1-
0); initial =size-1; for(i=n-1;i>=index;i--)
{
    TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial);
    initial=RQ[i];
}
}

```

```

    }
}

printf("Total head movement is %d",TotalHeadMoment);
}

void main(){ int ch; printf("\n 1.FCFS\t 2.SCAN\t
3.C-SCAN\t 4.EXIT\n"); while(1){
    printf("\nEnter your choice\n");
    scanf("%d",&ch); switch(ch){
        case 1:FCFS(); break;
        case 2:SCAN();
            break;
        case 3:C_SCAN();
            break;
        case 4:exit(0);
        default:printf("Invalid choice\n");
    }
}
}

```

**OUTPUT:**

1.FCFS 2.SCAN 3.C-SCAN 4.EXIT

Enter your choice

1

Enter the number of Requests

8

Enter the Requests sequence

95 180 34 119 11 123 62 64

Enter initial head position

50

Total head moment is 644

Enter your choice

2

Enter the number of Requests

8

Enter the Requests sequence

95 180 34 119 11 123 62 64

Enter initial head position

50

Enter total disk size

200

Enter the head movement direction for high 1 and for low 0

1

Total head movement is 337

Enter your choice

3

Enter the number of Requests

8

Enter the Requests sequence

95 180 34 119 11 123 62 64

Enter initial head position

50

Enter total disk size

200

Enter the head movement direction for high 1 and for low 0

1

Total head movement is 382

Enter your choice