

Cycle-7: Simulate disk scheduling algorithms

a) Write a C program to simulate FCFS (First Come First Serve)

Disk scheduling algorithm

Aim: To write a C program for simulating FCFS (First come First Serve)

Disk Scheduling Algorithm

Program:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    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);

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

    printf("Total head moment is %d",TotalHeadMoment);
    return 0;
}
```

Output:-

```
Enter the number of Requests
8
Enter the Requests sequence
98 183 37 122 14 124 65 67
Enter initial head position
53
Total head moment is 640
```

b)Write a C program to simulate SCAN disk scheduling algorithm

```
#include<stdio.h>

#include<stdlib.h>

int main()
{
    int RQ[100],i,j,n,TotalHeadMoment=0,initial,size,move,index,temp;

    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 sort the request array */

    for(i=0;i<n;i++)
    {
        for(j=0;j<n-i-1;j++)
        {
            if(RQ[j]>RQ[j+1])
            {
                temp=RQ[j];
                RQ[j]=RQ[j+1];
                RQ[j+1]=temp;
            }
        }
    }

    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];

    }

} /*end of else*/


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

return 0;

}

```

Output:-

```
Enter the number of Requests
8
Enter the Requests sequence
98 183 37 122 14 124 65 67
Enter initial head position
53
Enter total disk size
200
Enter the head movement direction for high 1 and for low 0
0
Total head movement is 236_
```

c)Write a C program to simulate CSCAN disk scheduling algorithm

Aim: To write a C program for simulating CSCAN disk Scheduling Algorithm

Program:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int RQ[100],i,j,n,TotalHeadMoment=0,initial,size,move,index,temp;
    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])
            {
                temp=RQ[j];
                RQ[j]=RQ[j+1];
                RQ[j+1]=temp;
            }
        }
    }

    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);  
return 0;  
}
```

Output:-

```
Enter the number of Requests  
8  
Enter the Requests sequence  
98 183 37 122 14 124 65 67  
Enter initial head position  
53  
Enter total disk size  
200  
Enter the head movement direction for high 1 and for low 0  
1  
Total head movement is 382_
```