Write a disc scheduling

(a) FCFS

(b) SCAN

(c) CSCAN

(a)
```
# include <stdio.h>
# include <stdlib.h>
int main ()
{
  int RQ[100], i, n, TotalHeadMoment=0,
initial;
printf ("Enter the no. of requests \n");
scanf ("%d", &n);
printf ("Enter 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
      TotalHeadMoment);
  return 0;
}
```

(b)
```
# include <stdio
# include <stdio
int main ()
{
  int RQ[100], i, j, n, TotalHead movement =0,
initial, size, move;
printf ("Enter requests");
scanf ("%d", &n);
printf ("Enter requests sequence ");
for (i=0; i<n; i++)
    scanf ("%d", &RQ[i]);
printf ("Enter head position");
scanf ("%d", &initial);
printf ("Enter disk size");
scanf ("%d", &size);
printf ("Enter the movement direction");
scanf ("%d", &move);
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 mdiff;
for (i=0; i<n; i++)
{
```

```
if ( initial < RQ[i] )
{
    index = i;
    break;
}
}

if (move == 1)
{
  for ( i = index ; i < n ; i++ )
{
  TotalHeadMoment = TotalHeadMoment +
  abs ( RQ[i] - initial ) ;
    initial = RQ[i] ;
}
  ~~TotalHeadMoment = TotalHeadMoment +~~
  ~~abs ( RQ[i] - initial ) ;~~
  else
  {
    for ( i = index - 1 ; i >= 0 ; i - - )
{
  TotalHeadMoment = TotalHeadMoment +
  abs ( RQ[i] - initial ) ;
    initial = RQ[i] ;
}

  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
return 0;
}

(c) // begin for C-Scan disc scheduling
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 mdex;
for (i=0; i<n; i++)
{
 if (initial < RQ[i])
 {
  mdex = i;
  break;
 }
}

if (move == 1)
{ for (i = mdex; i<n; i++)
{ TotalHeadMovement = TotalHeadMovement
+ abs(RQ[i] - initial);
 initial = RQ[i];
}
```

```
TotalHeadMovement = TotalHeadMovement +
abs (size - RQ [i-1] - i);
TotalHeadMovement = TotalHeadMovement +
abs (size - 1 - 0);
initial = 0;
for (i = 0; i < index; i++)
{ TotalHeadMovement = TotalHeadMovement
+ abs (RQ [i] - initial);
initial = RQ [i];
}
}

printf ("Total Head movement is,
TotalHeadMovement);
return 0;
}
```

Output

1) FCFS

Enter number of requests

3

Enter the sequence

55 12 37 22

Enter head position

20

Total head movement is 118

2) SCAN & CSAN

enter the requests sequence

35 12 37 22 40

Enter head position

20

Enter total disk size

140

Enter the head movement direction for
high 1 for low 0

Total head movement for SCAN is 93
Total headmovement for CSAN is 172.

28/8/23

```
C:\Users\Avani\Desktop\1bm21cs036\jk\bin\Debug\jk.exe
Enter the number of tracks: 3
Enter the tracks to be traversed: 4
5
6

Tracks traversed        Difference between tracks
0                       4
4                       1
5                       1

Average header movements: 2.000000
```

```
C:\Users\Avani\Desktop\1bm21cs036\jk\bin\Debug\jk.exe
Enter the number of tracks to be traversed: 5
Enter the tracks: 23
43
56
78
44
Enter the header position: 78
Enter the total range: 4
Total movement is: -95

Process returned 0 (0x0)   execution time : 12.490 s
Press any key to continue.
```

```
C:\Users\Avani\Desktop\1bm21cs036\jk\bin\Debug\jk.exe
Enter the number of tracks to be traversed: 5
Enter the tracks: 23
45
22
67
55
Enter the header position: 55
Enter the total range: 3
Total movement is: -6

Process returned 0 (0x0)   execution time : 11.949 s
Press any key to continue.
```