**38.Develop a C program to simulate C-SCAN disk scheduling algorithm Aim**: To Develop a C program to simulate C-SCAN disk scheduling algorithm **ALGORITHM:**-

* 1. Start at the current position of the disk head.
  2. Set the direction of movement to one side (e.g., right).
  3. While servicing requests in the selected direction:
     + Move the disk head to the next track in the current direction.
     + Calculate the seek time as the absolute difference between the newposition of the disk head and the previous position.
     + Add the seek time to the total seek time.
     + Update the previous position of the disk head to the currentposition.
  4. If there are no more requests in the current direction:

Move the disk head to the end of the disk in the current direction.

Change direction to the opposite side (e.g., left).

Continue servicing requests in the new direction.

* 1. Repeat step 3 and step 4 until all requests are serviced.
  2. After serving all the requests, calculate and display the total seektime.
  3. Calculate and display the average seek time, which is the total seektime divided by the number of requests.

# **Program:**

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

{

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);

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(move==1)

{

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

{

TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial); initial=RQ[i];

}

TotalHeadMoment=TotalHeadMoment+abs(size-RQ[i-1]-1); TotalHeadMoment=TotalHeadMoment+abs(size-1-0); initial=0;

for( i=0;i<index;i++){

TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial); initial=RQ[i];

}

}

Else

{

for(i=index-1;i>=0;i--)

{

TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial);

initial=RQ[i];

}

TotalHeadMoment=TotalHeadMoment+abs(RQ[i+1]-0); 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:**

