**Experiment No:** 9

**Experiment Name:**Implementation of C-Scan  Disk Scheduling algorithm.

**Objectives:**

C-Scan

1. Similar to scan algorithm but the head returns to cylinder 0 when it reaches the end of the disk
2. Treats the cylinder list as a circular list that wraps around the disk
3. Waiting time is more uniform for cylinders near the edge of the disk

**CODE:**

#include<stdio.h>

#include<math.h>

#define max 20

#define cymax 199

//implement by piyal\_IT-15021

int i,j,req,ttl\_tracks=0,cp,np,cposn,nposn;

int cyposn[max],temp;

void input()

{

do

{

//clreol();

printf("\n Enter the current header position : ");

scanf("%d",&cposn);

}

while(cposn>cymax || cposn <=0);

printf("\n Enter the %d I/O Requests : ",req);

cyposn[0] = cposn;

for(i=1; i<=req; i++)

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

}

void CSCAN()

{

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

{

for(j=0; j<req-i; j++)

{

if(cyposn[j] > cyposn[j+1])

{

temp = cyposn[j];

cyposn[j] = cyposn[j+1];

cyposn[j+1] = temp;

}

}

}

cp=0;

do

{

if(cyposn[cp] == cposn)

break;

cp++;

}

while(cp!=req);

printf("\nS.No. Current Position Next Position Displacement \n");

printf("---------------------------------------------------------- \n\n");

i=0,j=cp;

cposn = cyposn[cp];

do

{

if(cposn == cyposn[req])

{

nposn = 199;

cp = -1;

}

else

nposn = cyposn[++cp];

printf(" %d\t\t%d\t\t%d\t\t%d\n",++i,cposn,nposn,abs(cposn-nposn));

ttl\_tracks += (abs(cposn-nposn));

cposn = nposn == 199 ? 0 : nposn;

}

while(nposn != cyposn[j-1]);

printf("---------------------------------------------------------- \n\n");

printf(" Total Tracks Displaced : %d",ttl\_tracks);

}

void main()

{

do

{

//clrscr();

printf("\n Enter the number of requests : ");

scanf("%d",&req);

}

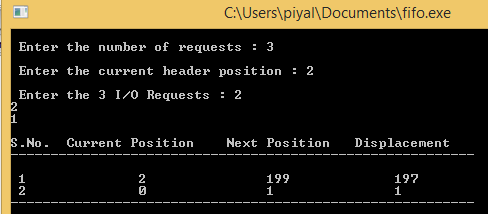
while(req>max || req <=0);

input();

CSCAN();

}

**Output:**



**Conclusion:**

In this lab report we learn about C-Scan Disk Scheduling algorithm.. We also learn how to implement C-Scan Disk Scheduling algorithm by using C program And testing the program different input and find output.We get proper output form this experiment.