**Lab 9**

**Question:**

Suppose the head of a moving- head disk with 200 tracks, numbered 0 to 199 is currently serving request at tracks 143 and has finished a request at track 125. The queue it requests is kept in the FIFO order 86, 147, 91, 177, 94, 150, 102, 175, 130.Write a program to calculate the total head movement using following algorithms.

* FCFS
* SSTF
* SCAN
* C-SCAN
* LOOK
* C-LOOK

**For FCFS:**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int RQ[100],i,n,TotalHeadMoment=0,initial;

printf("Enter the number of Requests: ");

scanf("%d",&n);

printf("Enter the Requests sequence: ");

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

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

printf("Enter initial head position: ");

scanf("%d",&initial);

// logic for FCFS disk scheduling

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

{

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

initial=RQ[i];

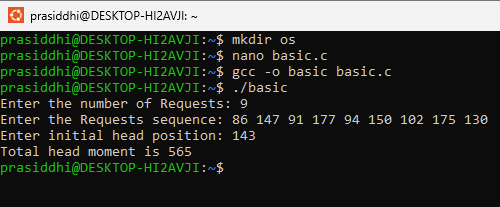
}

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

return 0;

}

**Output:**



**For SSTF:**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int RQ[100],i,n,TotalHeadMoment=0,initial,count=0;

printf("Enter the number of Requests: ");

scanf("%d",&n);

printf("Enter the Requests sequence: ");

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

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

printf("Enter initial head position: ");

scanf("%d",&initial);

// logic for sstf disk scheduling

/\* loop will execute until all process is completed\*/

while(count!=n)

{

int min=1000,d,index;

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

{

d=abs(RQ[i]-initial);

if(min>d)

{

min=d;

index=i;

}

29 }

TotalHeadMoment=TotalHeadMoment+min;

initial=RQ[index];

// 1000 is for max

// you can use any number

RQ[index]=1000;

count++;

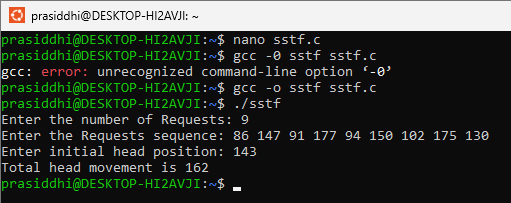
}

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

return 0;

}

**Output:**

****

**For SCAN**

#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: ");

scanf("%d",&n);

printf("Enter the Requests sequence: ");

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

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

printf("Enter initial head position: ");

scanf("%d",&initial);

printf("Enter total disk size: ");

scanf("%d",&size);

printf("Enter the head movement direction for high 1 and for low 0: ");

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];

}

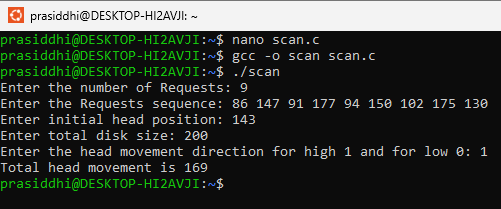
}

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

return 0;

}

**Output:**



**For C-SCAN**

#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: ");

scanf("%d",&n);

printf("Enter the Requests sequence: ");

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

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

printf("Enter initial head position: ");

scanf("%d",&initial);

printf("Enter total disk size: ");

scanf("%d",&size);

printf("Enter the head movement direction for high 1 and for low 0: ");

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];

}

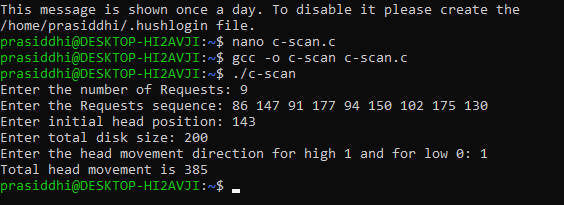
}

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

return 0;

}

**Output:**

****

**For look:**

#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: ");

scanf("%d",&n);

printf("Enter the Requests sequence: ");

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

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

printf("Enter initial head position: ");

scanf("%d",&initial);

printf("Enter total disk size: ");

scanf("%d",&size);

printf("Enter the head movement direction for high 1 and for low 0: ");

scanf("%d",&move);

// logic for look 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];

}

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];

}

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

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

initial=RQ[i];

}

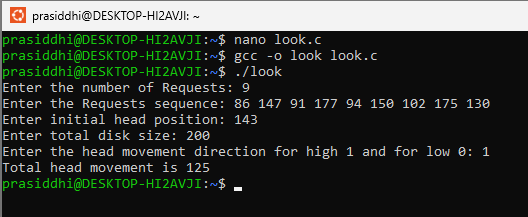
}

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

return 0;

}

**Output:**

****

**For C-LOOK**

#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: ");

scanf("%d",&n);

printf("Enter the Requests sequence: ");

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

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

printf("Enter initial head position: ");

scanf("%d",&initial);

printf("Enter total disk size: ");

scanf("%d",&size);

printf("Enter the head movement direction for high 1 and for low 0: ");

scanf("%d",&move);

// logic for C-look 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];

}

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];

}

for(i=n-1;i>=index;i--){

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

initial=RQ[i];

}

}

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

return 0;

}

**Output:**

