**Week – 7**

Simulate Disk scheduling algorithms

a) FCFS b) SSTF c) SCAN d) C-SCAN e) LOOK

**Code:**

#include<stdio.h>

#include<unistd.h>

#include<stdlib.h>

#include<limits.h>

int cylinders,curPos,arrLen;

void fcfs(int arr[]){

int temp=curPos,sum=0;

printf("Requests\theadMovements\t Total\n");

for(int i=0;i<arrLen;i++){

printf("%d\t",arr[i]);

printf("\t%d -> %d\t",temp,arr[i]);

int total = abs(temp-arr[i]);

printf("\t%d\n",total);

temp = arr[i];

sum+=total;

}

printf("Total no.of head movements:%d\n",sum);

}

void sstf(int arr[]){

int temp[200]={0};

int ans,presPos=curPos,tempLen=arrLen,index,sum=0;

printf("Requests\theadMovements\t Total\n");

while(tempLen--){

int mini = INT\_MAX;

for(int i=0;i<arrLen;i++){

if(temp[arr[i]] != -1){

ans = abs(curPos - arr[i]);

if(ans <= mini){

mini = ans;

index = i;

}

}

}

printf("%d\t",arr[index]);

printf("\t%d -> %d\t",presPos,arr[index]);

printf("\t%d\n",abs(presPos - arr[index]));

sum += abs(presPos - arr[index]);

presPos = arr[index];

temp[arr[index]] = -1;

}

printf("Total no.of head movements = %d\n",sum);

}

void sort(int tempArr[],int newLen){

for(int i=0;i<newLen;i++){

for(int j=0;j<newLen-i-1;j++){

if(tempArr[j] > tempArr[j+1]){

int temp = tempArr[j];

tempArr[j] = tempArr[j+1];

tempArr[j+1] = temp;

}

}

}

}

void scanRight(int arr[]) {

int newLen = arrLen + 3;

int tempArr[newLen];

for(int i = 0; i < arrLen; i++) {

tempArr[i] = arr[i];

}

tempArr[arrLen] = 0;

tempArr[arrLen + 1] = curPos;

tempArr[arrLen + 2] = cylinders - 1;

sort(tempArr, newLen);

int flag = 0, i, prePos, sum = 0;

printf("Requests\tHead Movements\tTotal\n");

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

if(tempArr[i] == curPos) {

prePos = i - 1;

flag = 1;

}

if(flag) {

printf("%d\t",tempArr[i]);

int pre;

if(i < newLen - 1) {

pre = abs(tempArr[i] - tempArr[i + 1]);

printf("\t%d -> %d\t",tempArr[i],tempArr[i + 1]);

printf("\t%d\n",pre);

} else {

pre = abs(tempArr[i] - tempArr[prePos]);

printf("\t%d -> %d\t",tempArr[i],tempArr[prePos]);

printf("%d\n",pre);

}

sum += pre;

}

}

for(int j = prePos; j > 0; j--) {

printf("%d\t", tempArr[j]);

int pre;

if(j > 0) {

pre = abs(tempArr[j] - tempArr[j - 1]);

printf("\t%d -> %d\t",tempArr[j],tempArr[j - 1]);

printf("%d\n",pre);

}

sum += pre;

}

printf("Total no. of head movements: %d\n", sum);

}

void scanLeft(int arr[]){

int newLen = arrLen+3;

int tempArr[newLen];

for(int i=0;i<arrLen;i++){

tempArr[i] = arr[i];

}

tempArr[arrLen] = 0;

tempArr[arrLen+1] = curPos;

tempArr[arrLen+2] = cylinders-1;

sort(tempArr,newLen);

int flag=0,i,prePos,sum=0;

printf("Requests\theadMovements\t Total\n");

for(i=newLen-1;i>=0;i--){

if(tempArr[i] == curPos){

prePos = i+1;

flag=1;

}

if(flag){

printf("%d\t",tempArr[i]);

int pre;

if(i > 0){

pre = abs(tempArr[i]-tempArr[i - 1]);

printf("\t%d -> %d\t",tempArr[i],tempArr[i-1]);

printf("%d\n",pre);

}else{

pre = abs(tempArr[i] - tempArr[prePos]);

printf("\t%d -> %d\t",tempArr[i],tempArr[prePos]);

printf("%d\n",pre);

}

sum += pre;

}

}

for(int j=prePos;j<newLen-1;j++){

printf("%d\t",tempArr[j]);

int pre;

pre = abs(tempArr[j]-tempArr[j+1]);

printf("\t%d -> %d\t",tempArr[j],tempArr[j+1]);

printf("%d\n",pre);

sum += pre;

}

printf("Total no.of head movements: %d\n",sum);

}

void scan(int arr[]){

int scanChoice;

printf("1.Towards right 2.Towards left:");

scanf("%d",&scanChoice);

switch(scanChoice){

case 1:

scanRight(arr);

break;

case 2:

scanLeft(arr);

break;

default:

break;

}

}

void c\_scanRight(int arr[]){

int newLen = arrLen+3;

int tempArr[newLen];

for(int i=0;i<arrLen;i++){

tempArr[i] = arr[i];

}

tempArr[arrLen] = 0;

tempArr[arrLen+1] = curPos;

tempArr[arrLen+2] = cylinders-1;

sort(tempArr,newLen);

int flag=0,i,sum=0;

printf("Requests\theadMovements\t Total\n");

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

if(tempArr[i] == curPos){

flag=1;

}

if(flag){

printf("%d\t",tempArr[i]);

int pre;

if(i < newLen-1){

pre = abs(tempArr[i]-tempArr[i + 1]);

printf("\t%d -> %d\t%d\n",tempArr[i],tempArr[i+1],pre);

}else{

pre = abs(tempArr[i] - tempArr[0]);

printf("\t%d -> %d\t%d\n", tempArr[i],tempArr[0],pre);

}

sum += pre;

}

}

for(int j=0;tempArr[j+1]!=curPos;j++){

printf("%d\t",tempArr[j]);

int pre;

pre = abs(tempArr[j]-tempArr[j-1]);

printf("\t%d -> %d\t%d\n",tempArr[j],tempArr[j+1],pre);

sum += pre;

}

printf("Total no.of head movements: %d\n",sum);

}

void c\_scanLeft(int arr[]){

int newLen = arrLen+3;

int tempArr[newLen];

for(int i=0;i<arrLen;i++){

tempArr[i] = arr[i];

}

tempArr[arrLen] = 0;

tempArr[arrLen+1] = curPos;

tempArr[arrLen+2] = cylinders-1;

sort(tempArr,newLen);

printf("Requests\theadMovements\t Total\n");

int i,flag=0,sum=0;

for(i=newLen-1;i>=0;i--){

if(tempArr[i] == curPos){

flag=1;

}

if(flag){

printf("%d\t",tempArr[i]);

int pre;

if(i > 0){

pre = abs(tempArr[i] - tempArr[i-1]);

printf("\t%d -> %d\t",tempArr[i],tempArr[i-1]);

printf("%d\n",pre);

}else{

pre = abs(tempArr[i] - tempArr[newLen-1]);

printf("\t%d -> %d\t",tempArr[i],tempArr[newLen-1]);

printf("%d\n",pre);

}

sum += pre;

}

}

for(int j=newLen-1;tempArr[j-1]!=curPos;j--){

printf("%d\t",tempArr[j]);

int pre;

pre = abs(tempArr[j]-tempArr[j-1]);

printf("\t%d -> %d\t",tempArr[j],tempArr[j-1]);

printf("%d\n",pre);

sum += pre;

}

printf("Total no.of head movements: %d\n",sum);

}

void c\_scan(int arr[]){

int scanChoice;

printf("1.Towards right 2.Towards left:");

scanf("%d",&scanChoice);

switch(scanChoice){

case 1:

c\_scanRight(arr);

break;

case 2:

c\_scanLeft(arr);

break;

default:

break;

}

}

void lookRight(int arr[]){

int newLen = arrLen+1;

int tempArr[newLen];

for(int i=0;i<arrLen;i++){

tempArr[i] = arr[i];

}

tempArr[arrLen] = curPos;

sort(tempArr,newLen);

int i,flag=0,sum=0,prePos;

printf("Requests\theadMovements\t Total\n");

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

if(tempArr[i] == curPos){

prePos = i-1;

flag = 1;

}

if(flag){

printf("%d\t",tempArr[i]);

int pre;

if(i < newLen-1){

pre = abs(tempArr[i] - tempArr[i+1]);

printf("\t%d -> %d\t%d\n",tempArr[i],tempArr[i+1],pre);

}else{

pre = abs(tempArr[i] - tempArr[prePos]);

printf("\t%d -> %d\t%d\n", tempArr[i],tempArr[prePos],pre);

}

sum += pre;

}

}

for(int j=prePos;j>0;j--){

printf("%d\t",tempArr[j]);

int pre;

if(j > 0){

pre = abs(tempArr[j]-tempArr[j-1]);

printf("\t%d -> %d\t%d\n",tempArr[j],tempArr[j-1],pre);

}

sum += pre;

}

printf("Total no.of head movements: %d\n",sum);

}

void lookLeft(int arr[]){

int newLen = arrLen+1;

int tempArr[newLen];

for(int i=0;i<arrLen;i++){

tempArr[i] = arr[i];

}

tempArr[arrLen] = curPos;

sort(tempArr,newLen);

int i,flag=0,sum=0,prePos;

printf("Requests\theadMovements\t Total\n");

for(i=newLen-1;i>=0;i--){

if(tempArr[i] == curPos){

prePos = i+1;

flag=1;

}

if(flag){

printf("%d\t",tempArr[i]);

int pre;

if(i > 0){

pre = abs(tempArr[i]-tempArr[i - 1]);

printf("\t%d -> %d\t",tempArr[i],tempArr[i-1]);

printf("%d\n",pre);

}else{

pre = abs(tempArr[i] - tempArr[prePos]);

printf("\t%d -> %d\t",tempArr[i],tempArr[prePos]);

printf("%d\n",pre);

}

sum += pre;

}

}

for(int j=prePos;j<newLen-1;j++){

printf("%d\t",tempArr[j]);

int pre;

pre = abs(tempArr[j]-tempArr[j+1]);

printf("\t%d -> %d\t",tempArr[j],tempArr[j+1]);

printf("%d\n",pre);

sum += pre;

}

printf("Total no.of head movements: %d\n",sum);

}

void look(int arr[]){

int scanChoice;

printf("1.Towards right 2.Towards left:");

scanf("%d",&scanChoice);

switch(scanChoice){

case 1:

lookRight(arr);

break;

case 2:

lookLeft(arr);

break;

default:

break;

}

}

int main(){

printf("Enter no.of cylinders:");

scanf("%d",&cylinders);

printf("Enter length of the array:");

scanf("%d",&arrLen);

int arr[arrLen];

printf("Enter requests:");

for(int i=0;i<arrLen;i++){

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

}

printf("Enter current position:");

scanf("%d",&curPos);

while(1){

printf("1.FCFS 2.SSTF 3.SCAN 4.C-SCAN 5.LOOK\n");

int choice;

printf("Enter choice:");

scanf("%d",&choice);

switch(choice){

case 1:

fcfs(arr);

break;

case 2:

sstf(arr);

break;

case 3:

scan(arr);

break;

case 4:

c\_scan(arr);

break;

case 5:

look(arr);

break;

default:

break;

}

int ch;

printf("Do you want to continue(1/0):");

scanf("%d",&ch);

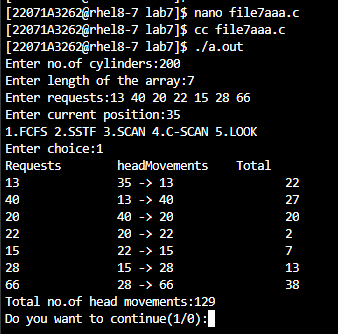
if(ch==0) break;

}

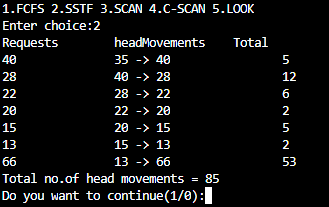
return 0;

}

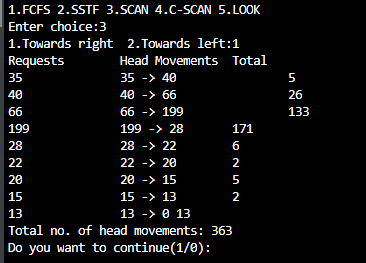
**Output for FCFS:**



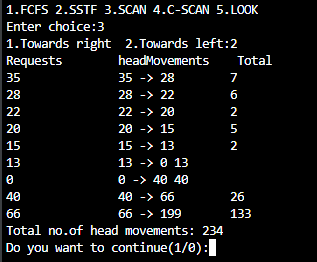
**Output for SSTF:**



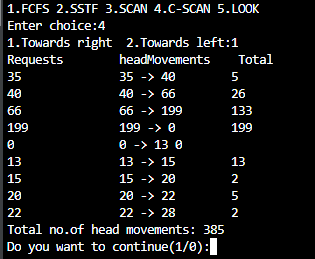
**Output for SCAN (Towards Right):**



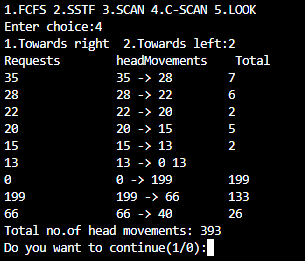
**Output for SCAN (Towards Left):**



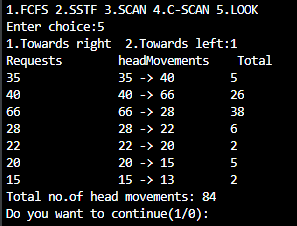
**Output for C-SCAN (Towards Right):**



**Output for C-SCAN (Towards Left):**



**Output for LOOK (Towards Right):**



**Output for LOOK (Towards Left):**

