DISK SCHEDULING:

#include<stdio.h>

#include<conio.h>

#include<math.h>

int or[20],head,i,n,head\_movement=0,start,end;

void sort();

void fcfs();

void scan();

void look();

void cscan();

void clook();

void main()

{

int choice;

printf("enter the no of requests:\n");

scanf("%d",&n);

printf("enter the order of request:\n");

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

{

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

}

printf("enter the current read/write head position:\n");

scanf("%d",&head);

printf("enter the start and the end of read/write head:\n");

scanf("%d%d",&start,&end);

while(1)

{

printf("1.FCFS\t2.SSTF\t3.scan\t4.Look\t5.C-scan\t6.C-look\t7.Exit\n");

printf("enter your choice:\n");

scanf("%d",&choice);

switch(choice)

{

case 1:fcfs();

break;

case 2:sstf();

break;

case 3:scan();

break;

case 4:look();

break;

case 5:cscan();

break;

case 6:clook();

break;

case 7:exit(0);

break;

default:printf("Invalid choice\n");

}

}

getch();

}

void fcfs()

{

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

{

head\_movement+=fabs(head-or[i]);

head=or[i];

}

printf("head movement=%d\n",head\_movement);

}

void sstf()

{

int c=0,i,j,headm=0,k,t,temp,b[20];

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

{

b[i]=or[i];

}

b[n]=head;

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

{

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

{

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

{

temp=b[j];

b[j]=b[j+1];

b[j+1]=temp;

}

}

}

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

{

if(b[i]==head)

break;

else

c++;

}

j=c;

k=c;

t=j;

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

{

if((b[k+1]-b[t])<(b[t]-b[j-1]) && j>0)

{

headm+=(b[k+1]-b[t]);

k++;

t=k;

}

else if(j==0)

{

headm+=(b[k+1]-b[t]);

k++;

t=k;

}

else

{

headm+=(b[t]-b[j-1]);

j--;

t=j;

}

}

printf("SSTF-Total head movement=%d\n",headm);

}

void scan()

{

printf("enter the current read/write head position:\n");

scanf("%d",&head);

sort();

int j;

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

{

if(head<or[i])

break;

}

j=i;

head\_movement=fabs(head-end)+fabs(end-or[j-1]);

head=or[j-1];

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

{

head\_movement+=fabs(head-or[i]);

head=or[i];

}

printf("head movement=%d\n",head\_movement);

}

void look()

{

printf("enter the current read/write head position:\n");

scanf("%d",&head);

sort();

int j;

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

{

if(head<or[i])

break;

}

j=i;

head\_movement=fabs(head-or[n-1])+fabs(or[n-1]-or[j-1]);

head=or[j-1];

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

{

head\_movement+=fabs(head-or[i]);

head=or[i];

}

printf("head movement=%d\n",head\_movement);

}

void clook()

{

printf("enter the current read/write head position:\n");

scanf("%d",&head);

sort();

int j;

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

{

if(head<or[i])

break;

}

j=i;

head\_movement=fabs(head-or[n-1]);

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

{

head=or[i];

head\_movement+=fabs(head-or[i+1]);

}

printf("head movement=%d\n",head\_movement);

}

void cscan()

{

printf("enter the current read/write head position:\n");

scanf("%d",&head);

sort();

int j;

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

{

if(head<or[i])

break;

}

j=i;

head\_movement=fabs(head-end)+fabs(start-or[j-1]);

printf("head movement=%d\n",head\_movement);

}

void sort()

{

int temp,j;

for(i=0;i<n-1;i++)

{

for(j=0;j<n-i-1;j++)

{

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

{

temp=or[j];

or[j]=or[j+1];

or[j+1]=temp;

}

}

}

}

int absolute(int a, int b)

{

int c = a - b;

if (c < 0)

return -c;

else

return c;

}

OUTPUT:

