Disk Scheduling Algorithms FCFS, SSTF, SCAN, LOOK

#include<stdio.h>

#include<math.h>

void fcfs(int noq, int qu[10], int st)

{

 int i,s=0;

 for(i=0;i<noq;i++)

 {

  s=s+abs(st-qu[i]);

  st=qu[i];

  }

 printf("\n Total seek time :%d",s);

}

void sstf(int noq, int qu[10], int st, int visit[10])

{

 int min,s=0,p,i;

 while(1)

{

  min=999;

  for(i=0;i<noq;i++)

   if (visit[i] == 0)

    {

      if(min > abs(st - qu[i]))

       {

        min = abs(st-qu[i]);

        p = i;

        }

      }

 if(min == 999)

  break;

  visit[p]=1;

  s=s + min;

  st = qu[p];

  }

 printf("\n Total seek time is: %d",s);

 }

void scan(int noq, int qu[10], int st, int ch)

{

 int i,j,s=0;

 for(i=0;i<noq;i++)

 {

  if(st < qu[i])

  {

   for(j=i-1; j>= 0;j--)

   {

    s=s+abs(st - qu[j]);

    st = qu[j];

    }

   if(ch == 3)

   {

   s = s + abs(st - 0);

   st = 0;

   }

  for(j = 1;j < noq;j++)

  {

   s= s + abs(st - qu[j]);

   st = qu[j];

   }

  break;

 }

}

printf("\n Total seek time : %d",s);

}

int main()

{

 int n,qu[20],st,i,j,t,noq,ch,visit[20];

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

 scanf("%d",&n);

 printf("enter number of queue elements");

 scanf("%d",&noq);

 printf("\n Enter the work queue");

 for(i=0;i<noq;i++)

 {

  scanf("%d",&qu[i]);

  visit[i] = 0;

  }

 printf("\n Enter the disk head starting posision: \n");

 scanf("%d",&st);

 while(1)

 {

  printf("\n\n\t\t MENU \n");

  printf("\n\n\t\t 1. FCFS \n");

  printf("\n\n\t\t 2. SSTF \n");

  printf("\n\n\t\t 3. SCAN \n");

  printf("\n\n\t\t 4. EXIT \n");

  printf("\nEnter your choice: ");

  scanf("%d",&ch);

  if(ch > 2)

   {

   for(i=0;i<noq;i++)

   for(j=i+1;j<noq;j++)

   if(qu[i]>qu[j])

   {

    t=qu[i];

    qu[i] = qu[j];

    qu[j] = t;

    }

   }

   switch(ch)

    {

    case 1: printf("\n FCFS \n");

            printf("\n\*\*\*\*\*\n");

            fcfs(noq,qu,st);

            break;

    case 2: printf("\n SSTF \n");

            printf("\n\*\*\*\*\*\n");

            sstf(noq,qu,st,visit);

            break;

    case 3: printf("\n SCAN \n");

            printf("\n\*\*\*\*\*\n");

            scan(noq,qu,st,ch);

            break;

    case 4: exit(0);

 }

 }

}

**Output**

"disksche.c" 122L, 2076C written

[anandh@localhost ~]$ cc disksche.c

[anandh@localhost ~]$ ./a.out

 Enter the maximum number of cylinders : 200

enter number of queue elements5

 Enter the work queue23

89

132

42

187

 Enter the disk head starting posision:

100

                  MENU

                 1. FCFS

                 2. SSTF

                 3. SCAN

                 4. EXIT

Enter your choice: 1

 FCFS

\*\*\*\*\*

 Total seek time : 421

                 MENU

                 1. FCFS

                 2. SSTF

                 3. SCAN

                 4. EXIT

Enter your choice: 2

 SSTF

\*\*\*\*\*

 Total seek time is: 273

                 MENU

                 1. FCFS

                 2. SSTF

                 3. SCAN

                 4. EXIT

Enter your choice: 3

 SCAN

\*\*\*\*\*

 Total seek time : 287

                 MENU

                 1. FCFS

                 2. SSTF

                 3. SCAN

                 4. EXIT

Enter your choice: 4

FCFS

/\*  
   FCFS Disk Scheduling Algorithm  
   Created by: Pirate  
\*/  
  
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
 int queue[100],n,head,i,j,k,seek=0,diff;  
 float avg;  
 // clrscr();  
  printf("\*\*\* FCFS Disk Scheduling Algorithm \*\*\*\n");  
 printf("Enter the size of Queue\t");  
 scanf("%d",&n);  
 printf("Enter the Queue\t");  
 for(i=1;i<=n;i++)  
  {  
     scanf("%d",&queue[i]);  
  }  
  printf("Enter the initial head position\t");  
 scanf("%d",&head);  
 queue[0]=head;  
 printf("\n");  
 for(j=0;j<=n-1;j++)  
 {  
    diff=abs(queue[j+1]-queue[j]);  
     seek+=diff;  
    printf("Move from %d to %d with Seek %d\n",queue[j],queue[j+1],diff);  
  }  
 printf("\nTotal Seek Time is %d\t",seek);  
  avg=seek/(float)n;  
 printf("\nAverage Seek Time is %f\t",avg);  
  getch();  
}

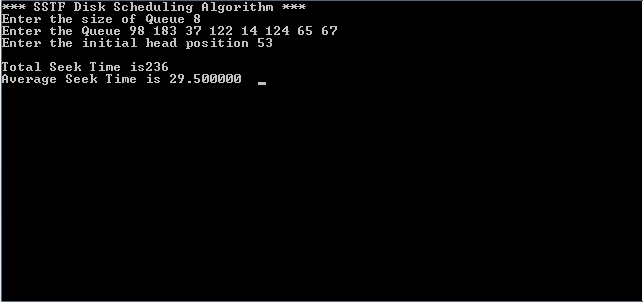
**Output**

[](http://3.bp.blogspot.com/-Mhd8nqoV9dQ/UpD4fr9_THI/AAAAAAAAArw/KyX5wdY-xuA/s1600/fifo.png)

SSTF

\*  
   SSTF Disk Scheduling Algorithm  
   Created By: Pirate  
\*/  
  
  
#include<stdio.h>  
#include<conio.h>  
#include<math.h>  
void main()  
{  
 int queue[100],t[100],head,seek=0,n,i,j,temp;  
 float avg;  
 // clrscr();  
 printf("\*\*\* SSTF Disk Scheduling Algorithm \*\*\*\n");  
 printf("Enter the size of Queue\t");  
 scanf("%d",&n);  
 printf("Enter the Queue\t");  
 for(i=0;i<n;i++)  
 {  
    scanf("%d",&queue[i]);  
 }  
 printf("Enter the initial head position\t");  
 scanf("%d",&head);  
  for(i=1;i<n;i++)  
  t[i]=abs(head-queue[i]);  
 for(i=0;i<n;i++)  
  {  
    for(j=i+1;j<n;j++)  
     {  
       if(t[i]>t[j])  
      {  
         temp=t[i];  
         t[i]=t[j];  
         t[j]=temp;  
         temp=queue[i];  
         queue[i]=queue[j];  
         queue[j]=temp;  
       }  
   }  
  }  
  for(i=1;i<n-1;i++)  
  {  
   seek=seek+abs(head-queue[i]);  
   head=queue[i];  
 }  
  printf("\nTotal Seek Time is%d\t",seek);  
 avg=seek/(float)n;  
 printf("\nAverage Seek Time is %f\t",avg);  
 getch();  
}

**Output**

[](http://3.bp.blogspot.com/-38DnKQHXuI4/UpD8TGSHyEI/AAAAAAAAAsA/DwUP22AsPfQ/s1600/sstf.png)

LOOK

#include<stdio.h>  
#include<conio.h>  
#include<math.h>  
#define max 20  
#define cymax 199  
  
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 CLOOK()  
{  
 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(cp == req)  
  { nposn = cyposn[0]; cp = 0; }  
  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 == cyposn[req] ? cyposn[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();  
 CLOOK();  
 getch();  
}

SCAN

void scan(int noq, int qu[10], int st, int ch)

{

 int i,j,s=0;

 for(i=0;i<noq;i++)

 {

  if(st < qu[i])

  {

   for(j=i-1; j>= 0;j--)

   {

    s=s+abs(st - qu[j]);

    st = qu[j];

    }

   if(ch == 3)

   {

   s = s + abs(st - 0);

   st = 0;

   }

  for(j = 1;j < noq;j++)

  {

   s= s + abs(st - qu[j]);

   st = qu[j];

   }

  break;

 }

}

printf("\n Total seek time : %d",s);

}

void main()

{

 int n,qu[20],st,i,j,t,noq,ch,visit[20];

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

 scanf("%d",&n);

 printf("enter number of queue elements");

 scanf("%d",&noq);

 printf("\n Enter the work queue");

 for(i=0;i<noq;i++)

 {

  scanf("%d",&qu[i]);

  visit[i] = 0;

  }

 printf("\n Enter the disk head starting posision: \n");

 scanf("%d",&st);

   for(i=0;i<noq;i++)

   for(j=i+1;j<noq;j++)

   if(qu[i]>qu[j])

   {

    t=qu[i];

    qu[i] = qu[j];

    qu[j] = t;

    }

   printf("\n SCAN \n");

            printf("\n\*\*\*\*\*\n");

            scan(noq,qu,st,ch);

 }

ALL

#include<stdio.h>  
  
int main()  
{  
    int len,start,que[20],i,j,dist=0,k,l,temp[20],tque[20],cnt=0,ends,ch;  
  
    printf("\nEnter the length of queue:");  
    scanf("%d",&len);  
    for(i=1;i<len+1;i++)  
    {  
        printf("\nEnter the cylinder block %d:",i);  
        scanf("%d",&que[i]);  
        tque[i]=que[i];  
    }  
    printf("\nEnter the starting block value:");  
    scanf("%d",&que[0]);  
    tque[0]=que[0];  
    printf("\nEnter the ends of disc block value:");  
    scanf("%d",&ends);  
    printf("\nHow u want SCAN Algorithm to start from?\n1:Back from start block\n2:Front from start block\nYour choice:");  
    scanf("%d",&ch);  
  
    //FCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFS  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FCFS\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");  
    printf("\nThe blocks visited are as follows->\n\n");  
  
    for(i=0;i<len+1;i++)  
    {  
        printf("%d ->> ",que[i]);  
        if(que[i+1]>que[i] && i+1!=len+1)  
        dist+=que[i+1]-que[i];  
        else if(tque[i+1]<=tque[i] && (i+1)!=k)  
        dist+=que[i]-que[i+1];  
  
    }  
    printf("\n\nTotal head movement=%d\n",dist);  
    //FCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFS  
  
    //SSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTF  
    dist=0;  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SSTF\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");  
    printf("\nThe blocks visited are as follows->\n\n");  
    for(i=0;i<len+1;i++)  
    {  
        if(que[i]>que[0])  
        temp[i]=que[i]-que[0];  
        else  
        temp[i]=que[0]-que[i];  
    }  
    //selection sort  
    for(i=0;i<len+1;i++)  
    {  
        for(j=1;j<len+1-i;j++)  
        if(temp[j-1]>temp[j])  
        {  
            k=temp[j-1];  
            l=tque[j-1];  
            temp[j-1]=temp[j];  
            tque[j-1]=tque[j];  
            temp[j]=k;  
            tque[j]=l;  
        }  
    }  
    //end of selection sort  
     for(i=0;i<len+1;i++)  
    {  
        printf("%d ->> ",tque[i]);  
        if(tque[i+1]>tque[i] && i+1!=len+1)  
        dist+=tque[i+1]-tque[i];  
        else if(i+1!=len+1)  
        dist+=tque[i]-tque[i+1];  
        temp[i]=0;  
  
    }  
    printf("\n\nTotal head movement=%d\n",dist);  
    //SSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTFSSTF  
  
    //SCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCAN  
    dist=0;  
  
    //restore value of tque  
    for(i=0;i<len+1;i++)  
    tque[i]=que[i];  
  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SCAN\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");  
    printf("\nThe blocks visited are as follows->\n\n");  
    //selection sort  
    for(i=0;i<len+1;i++)  
    for(j=1;j<len+1-i;j++)  
    {  
       if(tque[j-1]>tque[j])  
       {  
         k=tque[j-1];  
         tque[j-1]=tque[j];  
         tque[j]=k;  
       }  
    }  
    //end of selection sort  
    //Reversing que before start block and merging in main tque  
    for(i=0;i<len+1;i++)  
    {if(tque[i]==que[0]) break;}  
    tque[i]=tque[0];  
    for(j=1;j<i;j++)  
    {  
       k=tque[i-j];  
       tque[i-j]=tque[j];  
       tque[j]=k;  
    }  
    tque[0]=que[0];  
    k=i;  
    //end of reverse process  
    if(ch==1){  
    for(i=0;i<len+1;i++)  
    {  
        printf("%d ->> ",tque[i]);  
        if(tque[i+1]>tque[i] && i+1!=len+1)  
        dist+=tque[i+1]-tque[i];  
        else if(tque[i+1]<=tque[i] && (i+1)!=k)  
        dist+=tque[i]-tque[i+1];  
        if(k==i)  
        {printf("0 ->> ");dist+=2\*tque[i];}  
  
    }  
    }  
    else  
    {  
       dist=0;  
       printf("%d ->> ",tque[0]);  
       dist+=tque[k+1]-tque[0];  
       for(i=k+1;i<len+1;i++)  
       {  
        printf("%d ->> ",tque[i]);  
        if(tque[i+1]>tque[i] && i+1!=len+1)  
        dist+=tque[i+1]-tque[i];  
        else if((i+1)!=len+1)  
        dist+=tque[i]-tque[i+1];  
       }  
       printf("%d ->> ",ends);  
       dist+=(ends-tque[len]);  
       for(i=1;i<=k;i++)  
       {  
        dist+=ends-tque[i];  
        printf("%d ->> ",tque[i]);  
        if(tque[i+1]>tque[i] && (i+1)!=k+1)  
        dist+=tque[i+1]-tque[i];  
        else if(tque[i+1]<=tque[i] && (i+1)!=k)  
        dist+=tque[i]-tque[i+1];  
       }  
    }  
    printf("\n\nTotal head movement=%d\n",dist);  
    //SCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCANSCAN  
  
    //LOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOK  
  
    dist=0;  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LOOK\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");  
    printf("\nThe blocks visited are as follows->\n\n");  
    for(i=0;i<len+1;i++)  
    {  
        printf("%d ->> ",tque[i]);  
        if(tque[i+1]>tque[i] && i+1!=len+1)  
        dist+=tque[i+1]-tque[i];  
        else if(tque[i+1]<=tque[i] && (i+1)!=k)  
        dist+=tque[i]-tque[i+1];  
  
    }  
    printf("\n\nTotal head movement=%d\n",dist);  
    //LOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOKLOOK  
    return 0;