Scan

```
#include<stdio.h>
int main()
{
    int n,i,j,head,c,tot;
    printf("enter request no: ");
    scanf("%d",&n);
    int req[n+1];
    printf("enter request sequence: ");
    for(i=0;i<n;i++)
    {
        scanf("%d",&req[i]);
    }
    printf("enter initial head pos: ");
    scanf("%d",&head);
    printf("enter total disk size: ");
    scanf("%d",&tot);
    req[n]=head;
    for(i=0;i<n+1;i++)
    {
        for(j=0;j<n;j++)
        {
            if(req[j+1]<req[j])
            {
                int temp=req[j];
                req[j]=req[j+1];
                req[j+1]=temp;
            }
        }
    }
    printf("enter direction(forward-1/backward-0:) ");
    scanf("%d",&c);
    if(c==0)
    {
        printf("Total head Movement: %d",head+req[n]);
    }
    else
    {
        printf("Total head Movement: %d",(tot-head)+(tot-req[0]));
    }
}
```

Look

```
#include<stdio.h>
int main()
{
    int n,i,j,head,c,tot;
    printf("enter request no: ");
    scanf("%d",&n);
    int req[n+1];
    printf("enter request sequence: ");
    for(i=0;i<n;i++)
    {
        scanf("%d",&req[i]);
    }
    printf("enter initial head pos: ");
    scanf("%d",&head);
    req[n]=head;
    for(i=0;i<n+1;i++)
    {
        for(j=0;j<n;j++)
        {
            if(req[j+1]<req[j])
            {
                int temp=req[j];
                req[j]=req[j+1];
                req[j+1]=temp;
            }
        }
    }
    printf("enter direction(forward-1/backward-0:) ");
    scanf("%d",&c);
    if(c==0)
    {
        printf("Total head Movement: %d",(head-req[0])+(req[n]-req[0]));
    }
    else
    {
        printf("Total head Movement: %d",(req[n]-head)+(req[n]-req[0]));
    }
}
```

FCFS

```
#include<stdio.h>
int main()
{
    int n,i,j,head,c,tot,thm=0;
    printf("enter request no: ");
    scanf("%d",&n);
    int req[n];
    printf("enter request sequence: ");
    for(i=0;i<n;i++)
    {
        scanf("%d",&req[i]);
    }
    printf("enter initial head pos: ");
    scanf("%d",&head);
    for(i=0;i<n;i++)
    {
        thm+=abs(req[i]-head);
        head=req[i];
    }
    printf("Total head Movement: %d",thm);
}
```

Shortest Seek Time First

(record code)