DISK SCHEDULING-2

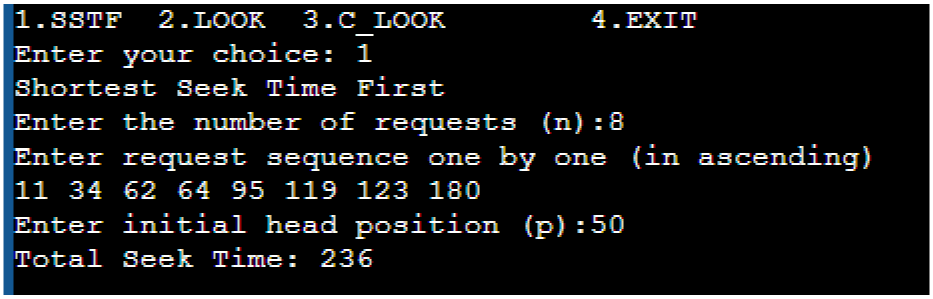
to simulate disk scheduling algorithms

a) SSTF
b)LOOK
c) C-LOOK

```
#include <stdio.h>
#include <stdlib.h>

void SSTF() {
    int n, head;
    printf("Shortest Seek Time First\n");
    printf("Enter the number of requests (n):");
    scanf("%d", &n);
    int rs[n];
    printf("Enter request sequence one by one (in ascending)\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &rs[i]);
    }
    printf("Enter initial head position (p):");
    scanf("%d", &head);
    int visited[n];
    for (int i = 0; i < n; i++) {
        visited[i] = 0;
    }
    int sum = 0;
    int current = head;
    for (int i = 0; i < n; i++) {
        int minDiff = 1e9;
        int index = -1;
        for (int j = 0; j < n; j++) {
            if (!visited[j]) {
                int diff = abs(current - rs[j]);
                if (diff < minDiff) {
                    minDiff = diff;
                    index = j;
                }
            }
        }
        visited[index] = 1;
        sum += minDiff;
        current = rs[index];
    }
    printf("Total Seek Time: %d\n", sum);
}

void LOOK(){
int RQ[100],i,j,n,TotalHeadMoment=0,initial,size,move;
    printf("Enter the number of Requests\n");
    scanf("%d",&n);
    printf("Enter the Requests sequence\n");
    for(i=0;i<n;i++)
     scanf("%d",&RQ[i]);
    printf("Enter initial head position\n");
    scanf("%d",&initial);
    printf("Enter the head movement direction for high 1 and for low 0\n");
    scanf("%d",&move);
    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(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];
        }
    }
    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",TotalHeadMoment);
}

void C_LOOK(){
int RQ[100],i,j,n,TotalHeadMoment=0,initial,size,move;
    printf("Enter the number of Requests\n");
    scanf("%d",&n);
    printf("Enter the Requests sequence\n");
    for(i=0;i<n;i++)
     scanf("%d",&RQ[i]);
    printf("Enter initial head position\n");
    scanf("%d",&initial);
    printf("Enter the head movement direction for high 1 and for low 0\n");
    scanf("%d",&move);
    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(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];
        }
    }
    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",TotalHeadMoment);
}

int main(){
    int ch;
    printf("1.SSTF\t2.LOOK\t3.C_LOOK\t4.EXIT");
    printf("\nEnter your choice: ");
    scanf("%d",&ch);
    switch(ch){
        case 1: SSTF();
                break;
        case 2: LOOK();
                break;
        case 3: C_LOOK();
                break;
        case 4: exit(0);
                break;
        default: printf("Invalid input");
                 break;
    }
    return(0);
}
```

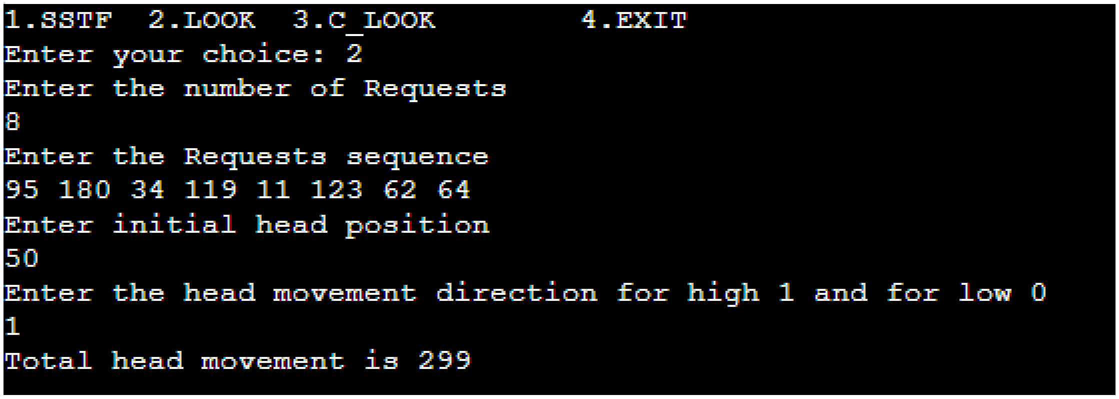
OUTPUT:

SSTF:



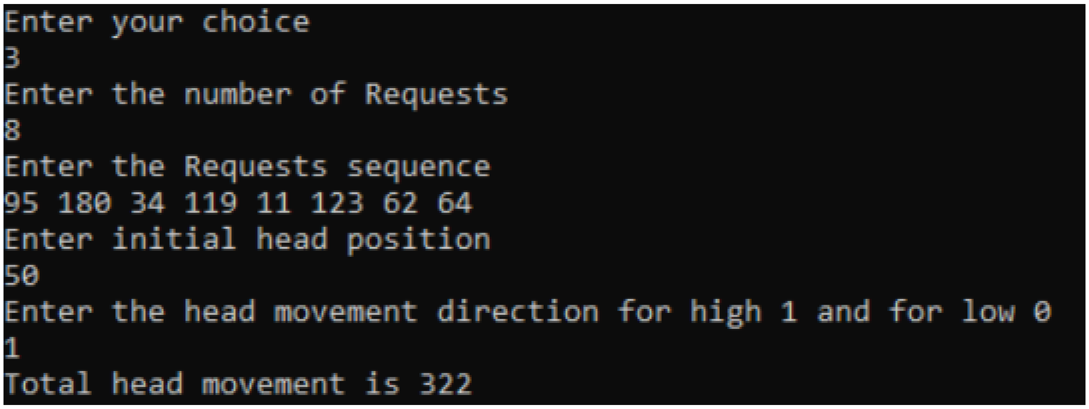
```
1.SSTF  2.LOOK  3.C_LOOK        4.EXIT
Enter your choice: 1
Shortest Seek Time First
Enter the number of requests (n):8
Enter request sequence one by one (in ascending)
11 34 62 64 95 119 123 180
Enter initial head position (p):50
Total Seek Time: 236
```

LOOK:



```
1.SSTF  2.LOOK  3.C_LOOK        4.EXIT
Enter your choice: 2
Enter the number of Requests
8
Enter the Requests sequence
95 180 34 119 11 123 62 64
Enter initial head position
50
Enter the head movement direction for high 1 and for low 0
1
Total head movement is 299
```

C_LOOK



```
Enter your choice
3
Enter the number of Requests
8
Enter the Requests sequence
95 180 34 119 11 123 62 64
Enter initial head position
50
Enter the head movement direction for high 1 and for low 0
1
Total head movement is 322
```