c- look

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

int main() {
    int n, i, j, temp;
    int head, total = 0, direction;

    // Ask user for number of disk requests
    printf("Enter number of disk requests: ");
    scanf("%d", &n);

    int request[n]; // Array to store disk requests

    // Take disk request sequence as input
    printf("Enter the disk requests: ");
    for (i = 0; i < n; i++)
        scanf("%d", &request[i]);

    // Take initial head position
    printf("Enter initial head position: ");
    scanf("%d", &head);

    // Take direction (1 = moving towards higher cylinder numbers)
    printf("Enter head movement direction (1 for high, 0 for low): ");
    scanf("%d", &direction);

    // Sort the request array in ascending order (for easier scanning)
    for (i = 0; i < n - 1; i++) {
        for (j = 0; j < n - i - 1; j++) {
```

```
        if (request[j] > request[j + 1]) {
            temp = request[j];
            request[j] = request[j + 1];
            request[j + 1] = temp;
        }
    }
}

int pos;
// Find position where head would fit in sorted order
for (i = 0; i < n; i++) {
    if (head < request[i]) {
        pos = i;
        break;
    }
}

printf("\nSeek sequence: ");
int current = head;

// If direction is high (towards larger numbers)
if (direction == 1) {
    // First service all requests greater than current head
    for (i = pos; i < n; i++) {
        printf("%d -> ", request[i]);
        total += abs(current - request[i]);
        current = request[i];
    }
    // Jump to the lowest request (C-LOOK circular behavior)
    for (i = 0; i < pos; i++) {
        printf("%d -> ", request[i]);
```

```
            total += abs(current - request[i]);
            current = request[i];
        }
    }
    // If direction is low (towards smaller numbers)
    else {
        // First service all requests smaller than current head
        for (i = pos - 1; i >= 0; i--) {
            printf("%d -> ", request[i]);
            total += abs(current - request[i]);
            current = request[i];
        }
        // Jump to the highest request (C-LOOK circular behavior)
        for (i = n - 1; i >= pos; i--) {
            printf("%d -> ", request[i]);
            total += abs(current - request[i]);
            current = request[i];
        }
    }

    printf("\nTotal head movement = %d\n", total);
    printf("Average head movement = %.2f\n", (float)total / n);

    return 0;
}
```

```
Enter number of disk requests: 6
Enter the disk requests: 176 79 34 60 92 11
Enter initial head position: 50
Enter head movement direction (1 for high, 0 for low): 1
```

Seek sequence: 60 -> 79 -> 92 -> 176 -> 11 -> 34 ->

Total head movement = 314

Average head movement = 52.33

🔢 Step 1: Sort the Requests

We first sort all requests in ascending order so that head movement can be processed easily in one direction.

Sorted order:

11, 34, 60, 79, 92, 176

🎯 Step 2: Identify Where the Head Lies

Head = 50
We find where it fits in the sorted list —
it’s **between 34 and 60**.

So:

Lower requests: 11, 34

Higher requests: 60, 79, 92, 176

🚀 Step 3: Move in the Given Direction (1 = High)

Head is moving towards **higher cylinder numbers** (right side).
So we’ll first service **all higher requests** in order.

(a) Move from 50 → 60

Distance = |60 - 50| = **10**

(b) Move from 60 → 79

Distance = |79 - 60| = **19**

(c) Move from 79 → 92

Distance = |92 - 79| = **13**

(d) Move from 92 → 176

Distance = |176 - 92| = **84**

At this point, the head has reached the **last request in this direction**.

🔄 Step 4: Circular Jump (C-LOOK behavior)

After reaching the highest request (176),
C-LOOK **jumps back to the lowest pending request (11)** —
but this "jump" is counted as a **movement**, because it's a repositioning of the head.

(e) Jump from 176 → 11

Distance = |176 - 11| = **165**

🧭 Step 5: Continue in the Same Direction

Now we continue moving in the same "upward" direction —
servicing the remaining lower requests that were not yet handled.

(f) Move from 11 → 34

Distance = |34 - 11| = **23**

📊 Step 6: Total Head Movement

Step	From	To	Movement
1	50	60	10
2	60	79	19
3	79	92	13
4	92	176	84
5	176	11	165
6	11	34	23
Total			**314**

✅ **Total head movement = 314**
✅ **Average head movement = 314 / 6 = 52.33**