Disk1.c

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

void fcfs(int requests[], int n, int head) {
    int total = 0;
    printf("FCFS Order: ");
    for (int i = 0; i < n; i++) {
        printf("%d -> ", requests[i]);
        total += abs(requests[i] - head);
        head = requests[i];
    }
    printf("\nTotal Head Movements: %d\n", total);
}

void sstf(int requests[], int n, int head) {
    int total = 0, completed = 0, visited[n];
    for (int i = 0; i < n; i++) visited[i] = 0;
    printf("SSTF Order: ");
    while (completed < n) {
        int closest = -1, min_dist = INT_MAX;
        for (int i = 0; i < n; i++) {
            if (!visited[i] && abs(requests[i] - head) < min_dist) {
                min_dist = abs(requests[i] - head);
                closest = i;
            }
        }
        printf("%d -> ", requests[closest]);
        total += min_dist;
        head = requests[closest];
        visited[closest] = 1;
        completed++;
    }
    printf("\nTotal Head Movements: %d\n", total);
}

void scan(int requests[], int n, int head, int disk_size) {
    int total = 0;
    for (int i = 0; i < n - 1; i++)
        for (int j = i + 1; j < n; j++)
            if (requests[i] > requests[j]) {
                int temp = requests[i];
                requests[i] = requests[j];
                requests[j] = temp;
            }
    printf("SCAN Order: ");
    int i = 0;
    while (i < n && requests[i] < head) i++;
    for (int j = i; j < n; j++) {
        printf("%d -> ", requests[j]);
        total += abs(requests[j] - head);
        head = requests[j];
    }
    total += abs(disk_size - 1 - head);
    head = disk_size - 1;
    printf("%d -> ", head);
    for (int j = i - 1; j >= 0; j--) {
        printf("%d -> ", requests[j]);
        total += abs(requests[j] - head);
        head = requests[j];
    }
    printf("\nTotal Head Movements: %d\n", total);
}

void c_look(int requests[], int n, int head, int disk_size) {
    int total = 0;
    for (int i = 0; i < n - 1; i++)
        for (int j = i + 1; j < n; j++)
            if (requests[i] > requests[j]) {
                int temp = requests[i];
                requests[i] = requests[j];
                requests[j] = temp;
            }
    printf("C-LOOK Order: ");
    int i = 0;
    while (i < n && requests[i] < head) i++;
    for (int j = i; j < n; j++) {
        printf("%d -> ", requests[j]);
        total += abs(requests[j] - head);
        head = requests[j];
    }
    for (int j = 0; j < i; j++) {
        printf("%d -> ", requests[j]);
        total += abs(requests[j] - head);
        head = requests[j];
    }
    printf("\nTotal Head Movements: %d\n", total);
}

int main() {
    int n, head, disk_size, choice;
    printf("Enter disk size and initial head position: ");
    scanf("%d %d", &disk_size, &head);
    printf("Enter number of requests: ");
    scanf("%d", &n);
    int requests[n];
    printf("Enter requests:\n");
    for (int i = 0; i < n; i++) scanf("%d", &requests[i]);

    printf("\nChoose Algorithm:\n1. FCFS\n2. SSTF\n3. SCAN\n4. C-LOOK\nChoice: ");
    scanf("%d", &choice);
    switch (choice) {
        case 1: fcfs(requests, n, head); break;
        case 2: sstf(requests, n, head); break;
        case 3: scan(requests, n, head, disk_size); break;
        case 4: c_look(requests, n, head, disk_size); break;
        default: printf("Invalid choice.\n"); return 1;
    }

    return 0;
}
```