```
//Disk Scheduling Algorithm: SCAN
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define max 25

void sort(int *arr, int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) { // Compare arr[j] with arr[j+1]
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

int main() {
    int tracks;
    printf("Enter number of tracks: ");
```

```
scanf("%d", &tracks);
int n;
printf("Enter number of track numbers in queue: ");
scanf("%d", &n);
int tnums[n];
int head;
printf("Enter initial position of read/write head: ");
scanf("%d", &head);
printf("Enter track numbers in queue: \n");
int left[max], right[max], lp = 0, rp = 0;
for (int i = 0; i < n; i++) {
    int s;
   printf("Enter track number %d: ", i + 1); // Added a colon after the prompt
    scanf("%d", &s);
    if (!(s > 0 && s < tracks)) {
        printf("Invalid track number...\nEnter again: ");
        scanf("%d", &s);
    } else {
        if (s <= 50) {
            left[lp] = s;
            lp++;
        } else if (s > 50) {
            right[rp] = s;
            rp++;
        } else {
            continue;
        }
    }
}
sort(left, lp);
sort(right, rp);
char dir[max];
printf("Enter direction: ");
scanf(" %s",dir);
int total = 0;
if(strcmp(dir,"Large")==0){
    printf("Right\n");
    for (int i = 0; i < rp; i++) {
        printf("abs(%d - %d) \n",head,right[i]);
        total += abs(head - right[i]);
        head = right[i];
    }
```

```
        total += abs(head - (tracks-1));
        head=tracks-1;
        printf("Left\n");
        for (int i = lp - 1; i >= 0; i--) { // Changed i++ to i--
            printf("abs(%d - %d) \n",head,left[i]);
            total += abs(head - left[i]);
            head = left[i];
        }
    }else if(strcmp(dir,"Small")==0){
        for (int i = lp - 1; i >= 0; i--) { // Changed i++ to i--
            printf("abs(%d - %d) \n",head,left[i]);
            total += abs(head - left[i]);
            head = left[i];
        }
        total+=head;
        head=0;
        for (int i = 0; i < rp; i++) {
            printf("abs(%d - %d) \n",head,right[i]);
            total += abs(head - right[i]);
            head = right[i];
        }
    }else{
        printf("Invalid direction...");
    }

    printf("Total number of track movements: %d", total);
    return 0;
}
/*Output:
Enter number of tracks: 200
Enter number of track numbers in queue: 7
Enter initial position of read/write head: 50
Enter track numbers in queue:
Enter track number 1: 82
Enter track number 2: 170
Enter track number 3: 43
Enter track number 4: 140
Enter track number 5: 24
Enter track number 6: 16
Enter track number 7: 190
Enter direction: Large
Right
```

```
abs(50 - 82)
abs(82 - 140)
abs(140 - 170)
abs(170 - 190)
Left
abs(199 - 43)
abs(43 - 24)
abs(24 - 16)
Total number of track movements: 332
*/
```