```
//Disk Scheduling Algorithm: SSTF
#include<stdio.h>
#include<stdlib.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);
    int total = 0;
    for (int i = lp - 1; i >= 0; i--) { // Changed i++ to i--
        total += abs(head - left[i]);
        head = left[i];
    }
    for (int i = 0; i < rp; i++) {
        total += abs(head - right[i]);
        head = right[i];
    }
    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
Total number of track movements: 208
*/
```