23. Design a C program to simulate SCAN disk scheduling algorithm.

19.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
int size = 8;
int disk_size = 200;
void CSCAN(int arr[], int head)
{
    int seek_count = 0;
    int distance, cur_track;
    vector<int> left, right;
    vector<int> seek_sequence;
    left.push_back(0);
    right.push_back(disk_size - 1);
    for (int i = 0; i < size; i++) {
        if (arr[i] < head)
            left.push_back(arr[i]);
        if (arr[i] > head)
            right.push_back(arr[i]);
    }
    std::sort(left.begin(), left.end());
    std::sort(right.begin(), right.end());
    for (int i = 0; i < right.size(); i++) {
        cur_track = right[i];
        seek_sequence.push_back(cur_track);
        distance = abs(cur_track - head);
        seek_count += distance;
        head = cur_track;
    }
    head = 0;
    seek_count += (disk_size - 1);
    for (int i = 0; i < left.size(); i++) {
        cur_track = left[i];
        seek_sequence.push_back(cur_track);
        distance = abs(cur_track - head);
        seek_count += distance;
        head = cur_track;
    }
    cout << "Total number of seek operations = "
        << seek_count << endl;
    cout << "Seek Sequence is" << endl;

    for (int i = 0; i < seek_sequence.size(); i++) {
        cout << seek_sequence[i] << endl;
    }
}
int main()
{
    int arr[size] = { 176, 79, 34, 60, 92, 11, 41, 114 };
    int head = 50;
    cout << "Initial position of head: " << head << endl;
    CSCAN(arr, head);
    return 0;
}
```

```
C:\Users\kalya\OneDrive\Desktop\19.exe
Initial position of head: 50
Total number of seek operations = 389
Seek Sequence is
60
79
92
114
176
199
0
11
34
41

--------------------------------
Process exited after 0.1269 seconds with return value 0
Press any key to continue . . .
```