

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  void fifo(vector<int> requests, int head) {
4      cout << head << " -> ";
5      int cost = 0;
6      int newHead = head;
7      int count = 0;
8      for (int i = 0; i < requests.size() - 1; i++)
9      {
10         cout << requests[i] << " -> ";
11         cost += abs(newHead - requests[i]);
12         if (newHead != requests[i])
13             count++;
14         newHead = requests[i];
15     }
16     cout << requests.back() << endl;
17     count++;
18     cost += abs(newHead - requests.back());
19     cout << "Total head movements = " << cost << endl;
20     cout << "Average head movements = " << (double)cost / count << endl; }
21 void ascending(vector<int> requests, int head, int *count, int *cost, int *newHead) {
22     vector<int>::iterator headLoc = find(requests.begin(), requests.end(), head);
23     for (auto i = headLoc - 1; i >= requests.begin(); i--)
24     {
25         cout << *i << " -> ";
26         *cost += abs(*newHead - *i);
27         if (*newHead != *i)
28             *count += 1;
29         *newHead = *i;
30     }
31 }
32 void descending(vector<int> requests, int head, int *count, int *cost, int *newHead) {
33     vector<int>::iterator headLoc = find(requests.begin(), requests.end(), head);
34     for (auto i = headLoc + 1; i < requests.end(); i++)
35     {
36         cout << *i << " -> ";
37         *cost += abs(*newHead - *i);
38         if (*newHead != *i)
39             *count += 1;
40         *newHead = *i;
41     }
42 }
43 void cscan(vector<int> requests, int head, bool isAscending) {
44     requests.push_back(head);
45     sort(requests.begin(), requests.end());
46     vector<int>::iterator headLoc = find(requests.begin(), requests.end(), head);
47     int cost = 0;
48     int newHead = head;
49     int count = 0;
50     if (isAscending)
51     {
52         for (auto i = headLoc; i >= requests.begin(); i--)
53         {
54             cout << *i << " -> ";
55             cost += abs(newHead - *i);
56             if (newHead != *i)
57                 count += 1;
58             newHead = *i;
59         }
60         for (auto i = requests.end() - 1; i > headLoc; i--)
61         {
62             cout << *i << " -> ";
63             cost += abs(newHead - *i);
64             if (newHead != *i)
65                 count += 1;
66             newHead = *i;
67         }
68     }
69     else
70     {
71         for (auto i = headLoc; i < requests.end(); i++)
72         {
73             cout << *i << " -> ";
74             cost += abs(newHead - *i);
75             if (newHead != *i)
76                 count += 1;
77             newHead = *i;
78         }
79         for (auto i = requests.begin(); i < headLoc; i++)
80         {
81             cout << *i << " -> ";
82             cost += abs(newHead - *i);
83             if (newHead != *i)
84                 count += 1;

```

```

85     newHead = *i;
86 }
87 }
88 cout << endl;
89 << "Total head movements = " << cost << endl;
90 cout << "Average head movements = " << (double)cost / count << endl; }
91 void scan(vector<int> requests, int head, bool isAscending) {
92     requests.push_back(head);
93     sort(requests.begin(), requests.end());
94     vector<int>::iterator headLoc = find(requests.begin(), requests.end(), head);
95     int cost = 0;
96     int newHead = head;
97     int count = 0;
98     cout << head << " -> ";
99     if (isAscending)
100     {
101         ascending(requests, head, &count, &cost, &newHead);
102         descending(requests, head, &count, &cost, &newHead);
103     }
104     else
105     {
106         descending(requests, head, &count, &cost, &newHead);
107         ascending(requests, head, &count, &cost, &newHead);
108     }
109     cout << endl;
110     << "Total head movements = " << cost << endl;
111     cout << "Average head movements = " << (double)cost / count << endl; }
112 int main()
113 {
114     int n, size;
115     cout << "please enter size of disk: ";
116     cin >> size;
117     cout << "please enter number of requests: ";
118     cin >> n;
119     cout << "please enter the requests: ";
120     vector<int> requests;
121     for (int i = 0; i < n; i++)
122     {
123         int temp;
124         cin >> temp;
125         requests.push_back(temp);
126     }
127     int head;
128     cout << "please enter head location: ";
129     cin >> head;
130     cout << "please choose the direction (1- ascending, 2- descending): ";
131     int isAscending;
132     cin >> isAscending;
133     cout << "FIFO" << endl;
134     fifo(requests, head);
135     cout << "SCAN" << endl;
136     scan(requests, head, isAscending == 2 ? 1 : 0);
137     cout << "CSCAN" << endl;
138     cscan(requests, head, isAscending == 2 ? 1 : 0);
139     return 0;
140 }
141

```