

```

1
2  /**          operating system assignment
3
4          NAME : Daisana Teafava
5          ID   : 1100290
6  **/
7
8  #include <bits/stdc++.h>
9  using namespace std;
10
11 void fifo(vector<int> data, int head)
12 {
13     cout<<endl;
14     cout<<" ";
15     cout << head << " -> ";
16     int cost = 0;
17     int nhead = head;
18     int count = 0;
19
20     for (int i = 0; i < data.size() - 1; i++)
21     {
22         cout << data[i] << " -> ";
23         cost += abs(nhead - data[i]);
24         if (nhead != data[i])
25             count++;
26         nhead = data[i];
27     }
28     cout << data.back() << endl<<endl;
29     count++;
30     cost += abs(nhead - data.back());
31 }
32 void ascending(vector<int> data, int head, int *count, int *cost, int *nhead)
33 {
34     vector<int>::iterator headl = find(data.begin(), data.end(), head);
35     for (auto i = headl - 1; i >= data.begin(); i--)
36     {
37         cout << *i << " -> ";
38         *cost += abs(*nhead - *i);
39         if (*nhead != *i)
40             *count += 1;
41         *nhead = *i;
42     }
43 }
44 void descending(vector<int> data, int head, int *count, int *cost, int *nhead)
45 {
46     vector<int>::iterator headl = find(data.begin(), data.end(), head);
47     for (auto i = headl + 1; i < data.end(); i++)
48     {
49         cout << *i << " -> ";
50         *cost += abs(*nhead - *i);
51         if (*nhead != *i)
52             *count += 1;
53         *nhead = *i;
54     }
55     cout<<endl;
56 }
57 void cscan(vector<int> data, int head, bool isAscending)
58 {
59     data.push_back(head);
60     sort(data.begin(), data.end());
61     vector<int>::iterator headl = find(data.begin(), data.end(), head);
62     int cost = 0;
63     int nhead = head;
64     int count = 0;
65     cout<<endl;
66     cout<<" ";
67     if (isAscending)
68     {
69         for (auto i = headl; i >= data.begin(); i--)
70         {
71             cout << *i << " -> ";
72             cost += abs(nhead - *i);
73             if (nhead != *i)
74                 count += 1;
75             nhead = *i;
76         }
77         for (auto i = data.end() - 1; i > headl; i--)
78         {
79             cout << *i << " -> ";
80             cost += abs(nhead - *i);
81             if (nhead != *i)
82                 count += 1;
83             nhead = *i;
84         }
85     }
86     else

```

```

85     {
86         for (auto i = headl; i < data.end(); i++)
87         {
88             cout << *i << " -> ";
89             cost += abs(nhead - *i);
90             if (nhead != *i)
91                 count += 1;
92             nhead = *i;
93         }
94         for (auto i = data.begin(); i < headl; i++)
95         {
96             cout << *i << " -> ";
97             cost += abs(nhead - *i);
98             if (nhead != *i)
99                 count += 1;
100             nhead = *i;
101         }
102     }
103     cout<<endl<<endl;
104 }
105 void scan(vector<int> data, int head, bool isAscending)
106 {
107     data.push_back(head);
108     sort(data.begin(), data.end());
109     vector<int>::iterator headl = find(data.begin(), data.end(), head);
110     cout<<endl;
111     int cost = 0;
112     int nhead = head;
113     int count = 0;
114     cout<<" ";
115     cout << head << " -> ";
116
117     if (isAscending)
118     {
119         ascending(data, head, &count, &cost, &nhead);
120         descending(data, head, &count, &cost, &nhead);
121         cout<<endl;
122     }
123     else
124     {
125         ascending(data, head, &count, &cost, &nhead);
126         descending(data, head, &count, &cost, &nhead);
127     }
128     cout<<endl;
129 }
130 int main()
131 {
132     cout<<endl<<<< "OPERATING SYSTEM ASSIGNMENT"<<endl;
133     cout<<<< "HDD Scheduling"<<endl;
134     cout<<<< "Name : Dejene Tesfaye mulugeta"<<endl<<<< "ID : 1100290"<<endl<<endl;
135     cout<<<< "HDD scheduling Algorithm "<<endl<<endl;
136
137     int n;
138     cout << "Enter size of Queue : ";
139     cin >> n;
140     vector<int> data;
141     for (int i = 0; i < n; i++)
142     {
143         int temp;
144         cout << "Enter the Queue : ";
145         cin >> temp;
146         data.push_back(temp);
147     }
148     cout<<endl;
149     cout<<<< "Queue size is full"<<endl<<endl;
150     int head;
151     cout << "Enter head starting location : ";
152     cin >> head;
153     cout<<endl;
154     int isAscending;
155     cout <<endl<<<< "FIFO"<<endl;
156     fifo(data, head);
157     cout <<endl<<<< "SCAN"<<endl;
158     scan(data, head, isAscending == 2 ? 1 : 0);
159     cout <<endl<<<< "CSCAN"<<endl;
160     cscan(data, head, isAscending == 2 ? 1 : 0);
161
162     return 0;
163 }

```