```cpp

/**             operating system assignment

                NAME : Dejene Tesfaye
                ID    : 1100290
**/

#include <bits/stdc++.h>
using namespace std;

void fifo(vector<int> data, int head)
{   cout<<endl;
    cout<<"   ";
  cout << head << " -> ";
  int cost = 0;
  int nhead = head;
  int count = 0;

  for (int i = 0; i < data.size() - 1; i++)
  {
    cout << data[i] << " -> ";
    cost += abs(nhead - data[i]);
    if (nhead != data[i])
      count++;
    nhead = data[i];
  }
  cout << data.back() << endl<<endl;
  count++;
  cost += abs(nhead - data.back());
}
void ascending(vector<int> data, int head, int *count, int *cost, int *nhead)
{
  vector<int>::iterator headl = find(data.begin(), data.end(), head);
  for (auto i = headl - 1; i >= data.begin(); i--)
  {
    cout << *i << " -> ";
    *cost += abs(*nhead - *i);
    if (*nhead != *i)
       *count += 1;
    *nhead = *i;
  }
}
void descending(vector<int> data, int head, int *count, int *cost, int *nhead)
{
  vector<int>::iterator headl = find(data.begin(), data.end(), head);
  for (auto i = headl + 1; i < data.end(); i++)
  {
    cout << *i << " -> ";
    *cost += abs(*nhead - *i);
    if (*nhead != *i)
      *count += 1;
    *nhead = *i;
  }cout<<endl;
}
void cscan(vector<int> data, int head, bool isAscending)
{
  data.push_back(head);
  sort(data.begin(), data.end());
  vector<int>::iterator headl = find(data.begin(), data.end(), head);
  int cost = 0;
  int nhead = head;
  int count = 0;
  cout<<endl;
  cout<<"   ";
  if (isAscending)
  {
    for (auto i = headl; i >= data.begin(); i--)
    {
      cout << *i << " -> ";
      cost += abs(nhead - *i);
      if (nhead != *i)
        count += 1;
      nhead = *i;
    }
    for (auto i = data.end() - 1; i > headl; i--)
    {
      cout << *i << " -> ";
      cost += abs(nhead - *i);
      if (nhead != *i)
        count += 1;
      nhead = *i;
    }
  }
  else
  {
    for (auto i = headl; i < data.end(); i++)
    {
      cout << *i << " -> ";
      cost += abs(nhead - *i);
      if (nhead != *i)
        count += 1;
      nhead = *i;
    }
    for (auto i = data.begin(); i < headl; i++)
    {
      cout << *i << " -> ";
      cost += abs(nhead - *i);
      if (nhead != *i)
        count += 1;
      nhead = *i;
    }
  }
cout<<endl<<endl;
}
void scan(vector<int> data, int head, bool isAscending)
{
  data.push_back(head);
  sort(data.begin(), data.end());
  vector<int>::iterator headl = find(data.begin(), data.end(), head);
  cout<<endl;
  int cost = 0;
  int nhead = head;
  int count = 0;
  cout<<"   ";
  cout << head << " -> ";

  if (isAscending)
  {
    ascending(data, head, &count, &cost, &nhead);
    descending(data, head, &count, &cost, &nhead);
    cout<<endl;
  }
  else
  { ascending(data, head, &count, &cost, &nhead);
    descending(data, head, &count, &cost, &nhead);
  }
cout<<endl;
}
int main()
{
    cout<<endl <<"                                            OPERATING SYSTEM ASSIGNMENT"<<endl;
    cout<<"                                                  HDD Scheduling"<<endl;
    cout<<"         Name : Dejene tesfaye mulugeta"<<endl<<"         ID   : 1100290"<<endl<<endl;
    cout<<"    HDD scheduling Algorithm  "<<endl<<endl;

  int n;
  cout << "Enter size of Queue : ";
  cin >> n;
  vector<int> data;
  for (int i = 0; i < n; i++)
  {
    int temp;
    cout << "Enter the Queue : ";
    cin >> temp;
   data.push_back(temp);
  }
  cout<<endl;
  cout<<"            Queue size is full"<<endl<<endl;
  int head;
  cout << "Enter head starting location : ";
  cin >> head;
  cout<<endl;
  int isAscending;
  cout <<endl<< "````````````````````````FIFO`````````````````````" << endl;
  fifo(data, head);
  cout <<endl<< "````````````````````````SCAN`````````````````````" << endl;
  scan(data, head, isAscending == 2 ? 1 : 0);
  cout <<endl<< "````````````````````````CSCAN````````````````````" << endl;
  cscan(data, head, isAscending == 2 ? 1 : 0);

  return 0;
}
```