Assignment 8

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

void fcfs(vector<int> req, int head) {
  int distance = abs(head - req[0]);

  cout << "Order of servicing requests: " << req[0] << " ";
  for (int i = 0; i < req.size() - 1; i++) {
    distance += abs(req[i] - req[i + 1]);
    cout << req[i + 1] << " ";
  }
  cout << "\nTotal head movement: " << distance << endl;
}

void sstf(vector<int> req, int head) {
  int distance = 0;
  vector<int> orderService;

  while (!req.empty()) {
    auto it = min_element(req.begin(), req.end(), [&head](int a, int
b) {
      return abs(a - head) < abs(b - head);
    });

    orderService.push_back(*it); // Add closest request
    distance += abs(head - *it); // Calculate the movement
    head = *it;                  // Move head
    req.erase(it);               // Remove the request
  }

  cout << "Total head movement: " << distance << endl;
```

```
  cout << "Order of servicing requests: ";
  for (int i : orderService) {
    cout << i << " ";
  }
  cout << endl;
}

void scan(vector<int> req, int head, int ntracks) {
  sort(req.begin(), req.end());
  int distance = 0;

  distance = (ntracks - head) + (ntracks - req[0]);
  cout << "Total head movement: " << distance << endl;

  vector<int> orderService;
  int start=-1;
  for (int i = 0; i < req.size(); i++) {
    if (head < req[i]) {
      start = i;
      break;
    }
  }
  // Going upwards
  for (int i = start; i < req.size(); i++) {
    orderService.push_back(req[i]);
  }
  // Going downwards
  for (int i = start - 1; i >= 0; i--) {
    orderService.push_back(req[i]);
  }

  cout << "Order of servicing requests: ";
```

```
  for (int i : orderService) {
    cout << i << " ";
  }
  cout << endl;
}

void clook(vector<int> req, int head, int ntracks) {
  sort(req.begin(), req.end());
  int distance = 0;
  int start=-1;
  for (int i = 0; i < req.size(); i++) {
    if (head < req[i]) {
      start = i;
      break;
    }
  }

  distance += (req.back() - head) + (req.back() - req[0]);
  cout << "Total head movement: " << distance << endl;
  vector<int> orderService;

  for (int i = start; i < req.size(); i++) {
    orderService.push_back(req[i]);
  }

  for (int i = 0; i < start; i++) {
    orderService.push_back(req[i]);
  }
  cout << "Order of servicing requests: ";
  for (int i : orderService) {
    cout << i << " ";
  }
```

```
  cout << endl;
}

int main() {
  int ntracks, ndrequest, head, choice;
  cout << "Enter total number of tracks : ";
  cin >> ntracks;
  cout << "Enter total number of requests : ";
  cin >> ndrequest;

  vector<int> req(ndrequest);

  cout << "Enter disc requests in FCFS order : ";
  for (int i = 0; i < ndrequest; i++) {
    cin >> req[i];
  }
  cout << "Enter head position : ";
  cin >> head;

  while (1) {
    cout << "Enter your choice : " << endl;
    cout << "1. FCFS\n2. SSTF\n3. SCAN\n4. CLOOK\n5. EXIT" << endl;
    cin >> choice;

    switch (choice) {
    case 1:
      fcfs(req, head);
      break;

    case 2:
      sstf(req, head);
      break;
```

```
    case 3:
      scan(req, head, ntracks);
      break;

    case 4:
      clook(req, head, ntracks);
      break;

    case 5:
      exit(0);
      break;

    default:
      cout << "Invalid choice" << endl;
    }
  }

  return 0;
}
```

```
Output :
Enter total number of tracks : 199
Enter total number of requests : 7
Enter disc requests in FCFS order : 82 170 43 140 24 16 190
Enter head position : 50
Enter your choice :
1. FCFS
2. SSTF
3. SCAN
4. CLOOK
5. EXIT
```

```
1
Order of servicing requests: 82 170 43 140 24 16 190
Total head movement: 642
Enter your choice :
1. FCFS
2. SSTF
3. SCAN
4. CLOOK
5. EXIT
2
Total head movement: 208
Order of servicing requests: 43 24 16 82 140 170 190
Enter your choice :
1. FCFS
2. SSTF
3. SCAN
4. CLOOK
5. EXIT
3
Total head movement: 332
Order of servicing requests: 82 140 170 190 43 24 16
Enter your choice :
1. FCFS
2. SSTF
3. SCAN
4. CLOOK
5. EXIT
4
Total head movement: 314
Order of servicing requests: 82 140 170 190 16 24 43
```