Operating System Lab

CEN-493

Program - 15

Code :-

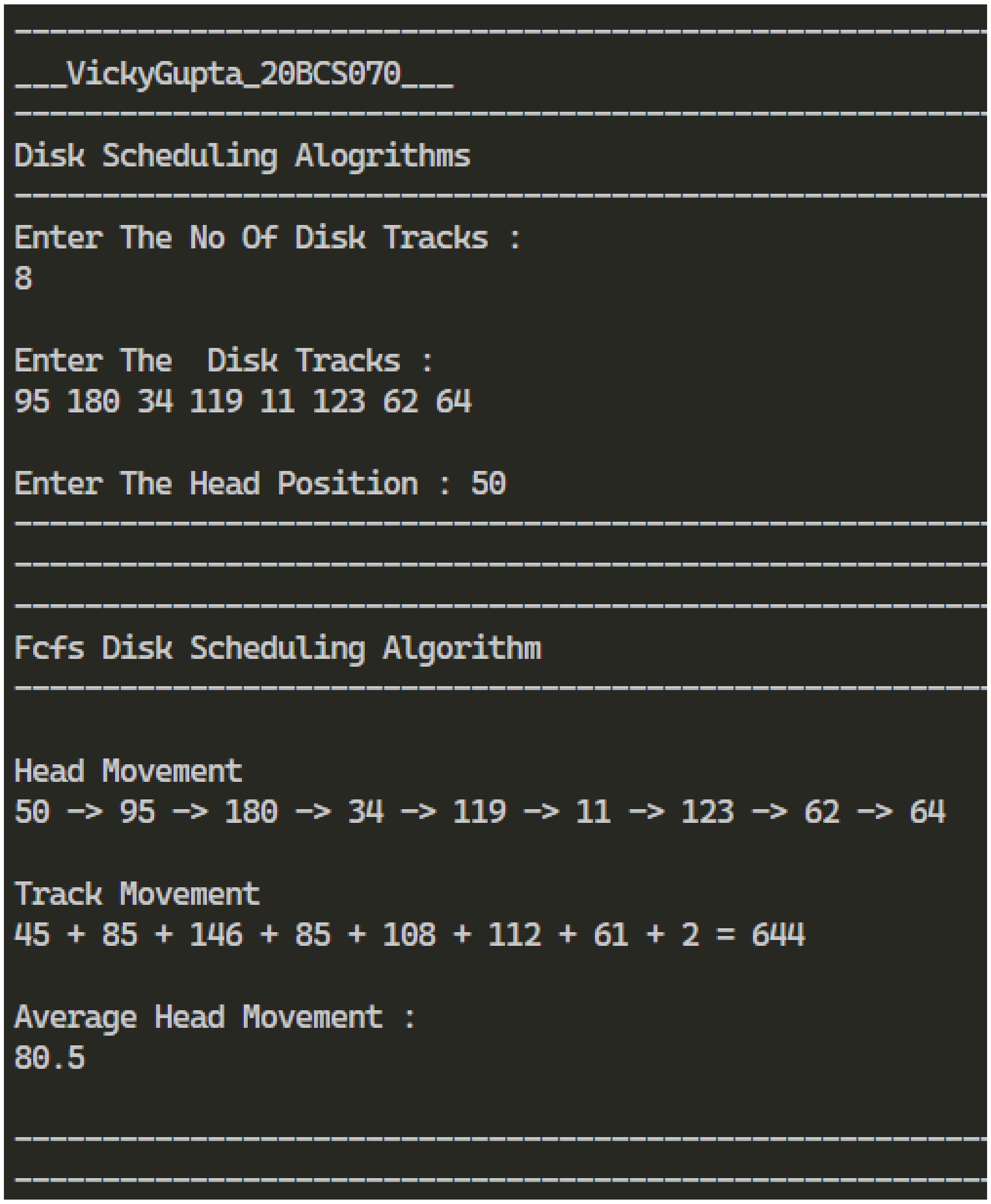
```
#include <iostream>
#include <math.h>
#include <vector>
#include <algorithm>
using namespace std;

void printLines()
{
    for (int i = 0; i < 120; i++)
    {
        cout << "-";
    }
    cout << "\n";
}
```

```
void printTheInfo(string info, int noOfDiskTracks,
vector<int> trackMovement, vector<int> headMovement)
{
    printLines();
    cout << info << "\n";
    printLines();
    int totalTrackMovement = 0;
    cout << "\nHead Movement\n";
    for (int i = 0; i < headMovement.size(); i++)
    {
        if (headMovement.size() - 1 == i)
            cout << headMovement[i] << " ";
        else
            cout << headMovement[i] << " -> ";
    }
    cout << "\n";

    cout << "\nTrack Movement\n";
    for (int i = 0; i < noOfDiskTracks; i++)
    {
        totalTrackMovement += trackMovement[i];
        if (i == noOfDiskTracks - 1)
            cout << trackMovement[i];
        else
            cout << trackMovement[i] << " + ";
    }
    cout << " = " << totalTrackMovement << "\n";
    float avgHeadMovement = (totalTrackMovement /
(float)noOfDiskTracks);
    cout << "\nAverage Head Movement : \n";
    cout << avgHeadMovement << "\n\n";
}

void fcfsDiskScheduling(int noOfDiskTracks, vector<int>
diskTracks, int headPosition)
{
    vector<int> headMovement, trackMovement;
    int prevHeadPosition = headPosition;
    headMovement.push_back(prevHeadPosition);
    for (int track = 0; track < noOfDiskTracks; track++)
    {
        headMovement.push_back(diskTracks[track]);
        trackMovement.push_back(abs(diskTracks[track] -
prevHeadPosition));
        prevHeadPosition = diskTracks[track];
    }

    printTheInfo("Fcfs Disk Scheduling Algorithm",
noOfDiskTracks, trackMovement, headMovement);
}

void sstfDiskScheduling(int noOfDiskTracks, vector<int>
diskTracks, int headPosition)
{
    vector<int> headMovement, trackMovement;
    int prevHeadPosition = headPosition;
    headMovement.push_back(prevHeadPosition);
    while (!diskTracks.empty())
    {
        int shortestSeekTime = 1e9, shortestSeekTimeIndex
= 0;
        for (int i = 0; i < diskTracks.size(); i++)
        {
            if (shortestSeekTime > abs(diskTracks[i] -
prevHeadPosition))
            {
                shortestSeekTime = abs(diskTracks[i] -
prevHeadPosition);
                shortestSeekTimeIndex = i;
            }
        }

        headMovement.push_back(diskTracks[shortestSeekTim
eIndex]);
        trackMovement.push_back(abs(diskTracks[shortestSe
ekTimeIndex] - prevHeadPosition));
        prevHeadPosition =
diskTracks[shortestSeekTimeIndex];
```

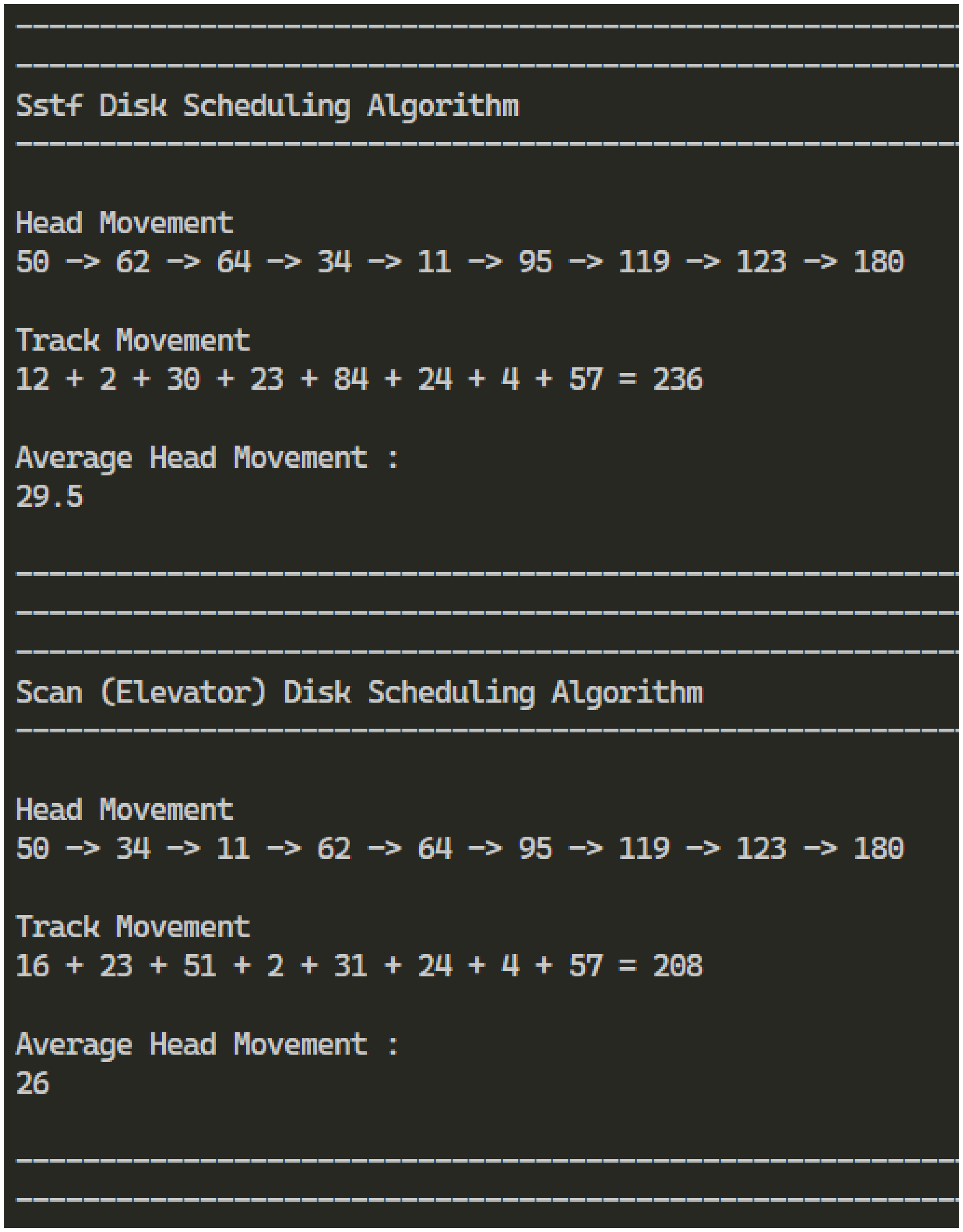
```
        diskTracks.erase(diskTracks.begin() +
shortestSeekTimeIndex);
    }

    printTheInfo("Sstf Disk Scheduling Algorithm",
noOfDiskTracks, trackMovement, headMovement);
}

void scanDiskScheduling(int noOfDiskTracks, vector<int>
diskTracks, int headPosition)
{
    vector<int> headMovement, trackMovement;
    int prevHeadPosition = headPosition;
    headMovement.push_back(prevHeadPosition);
    sort(diskTracks.begin(), diskTracks.end());

    int strtTrack = lower_bound(diskTracks.begin(),
diskTracks.end(), prevHeadPosition) - diskTracks.begin();
    if (diskTracks[strtTrack] > prevHeadPosition)
        strtTrack--;

    for (int track = strtTrack; track >= 0; track--)
    {
        headMovement.push_back(diskTracks[track]);
        trackMovement.push_back(abs(diskTracks[track] -
prevHeadPosition));
        prevHeadPosition = diskTracks[track];
    }
    for (int track = strtTrack + 1; track <
noOfDiskTracks; track++)
    {
        headMovement.push_back(diskTracks[track]);
        trackMovement.push_back(abs(diskTracks[track] -
prevHeadPosition));
        prevHeadPosition = diskTracks[track];
    }
    printTheInfo("Scan (Elevator) Disk Scheduling
Algorithm", noOfDiskTracks, trackMovement, headMovement);
}
```

```
int main()
{
    system("cls");

    printLines();
    cout << "___VickyGupta_20BCS070___\n";
    printLines();

    cout << "Disk Scheduling Alogrithms\n";
    printLines();

    int noOfDiskTracks;
    cout << "Enter The No Of Disk Tracks : \n";
    cin >> noOfDiskTracks;

    vector<int> diskTrack(noOfDiskTracks);
    cout << "\nEnter The  Disk Tracks :\n";
    for (int i = 0; i < noOfDiskTracks; i++)
    {
        cin >> diskTrack[i];
    }

    int headPosition;
    cout << "\nEnter The Head Position : ";
    cin >> headPosition;

    printLines();

    printLines();
    fcfsDiskScheduling(noOfDiskTracks, diskTrack,
headPosition);
    printLines();
    printLines();
    sstfDiskScheduling(noOfDiskTracks, diskTrack,
headPosition);
    printLines();
    printLines();
    scanDiskScheduling(noOfDiskTracks, diskTrack,
headPosition);
    printLines();
    printLines();

    return 0;
}
```

Output :-



```
-------------------------------------------------------
___VickyGupta_20BCS070___
-------------------------------------------------------
Disk Scheduling Alogrithms
-------------------------------------------------------
Enter The No Of Disk Tracks :
8

Enter The  Disk Tracks :
95 180 34 119 11 123 62 64

Enter The Head Position : 50
-------------------------------------------------------
-------------------------------------------------------
-------------------------------------------------------
Fcfs Disk Scheduling Algorithm
-------------------------------------------------------


Head Movement
50 -> 95 -> 180 -> 34 -> 119 -> 11 -> 123 -> 62 -> 64

Track Movement
45 + 85 + 146 + 85 + 108 + 112 + 61 + 2 = 644

Average Head Movement :
80.5


-------------------------------------------------------
-------------------------------------------------------
```



```
--------------------------------------------------------
--------------------------------------------------------
Sstf Disk Scheduling Algorithm
--------------------------------------------------------


Head Movement
50 -> 62 -> 64 -> 34 -> 11 -> 95 -> 119 -> 123 -> 180

Track Movement
12 + 2 + 30 + 23 + 84 + 24 + 4 + 57 = 236

Average Head Movement :
29.5


--------------------------------------------------------
--------------------------------------------------------
--------------------------------------------------------
Scan (Elevator) Disk Scheduling Algorithm
--------------------------------------------------------


Head Movement
50 -> 34 -> 11 -> 62 -> 64 -> 95 -> 119 -> 123 -> 180

Track Movement
16 + 23 + 51 + 2 + 31 + 24 + 4 + 57 = 208

Average Head Movement :
26


--------------------------------------------------------
--------------------------------------------------------
```