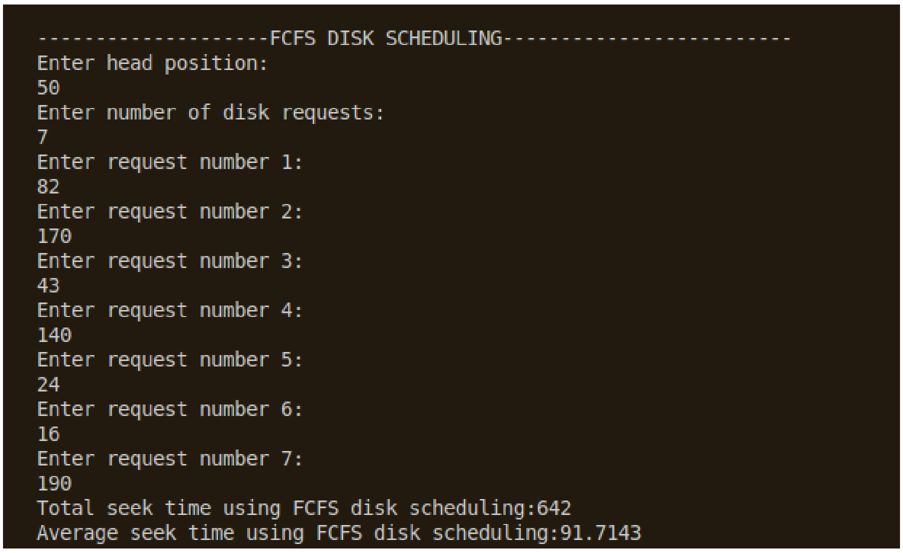
Practical 9: OS Lab

Sameer Gupta November 26, 2025

Q1. Write a CPP program to simulate the Disk scheduling algorithm (FCFS).

Solution:

```cpp
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cout<<"--------------------FCFS DISK SCHEDULING-------------------------"<<endl;
    int head;
    cout<<"Enter head position:"<<endl;
    cin>>head;
    cout<<"Enter number of disk requests:"<<endl;
    cin>>n;
    vector<int>diskRequests(n);
    for(int i=0;i<n;i++){
        cout<<"Enter request number "<<i+1<<":"<<endl;
        cin>>diskRequests[i];
    }
    int totalSeekTime=0;
    int currPos=head;
    for(int i=0;i<diskRequests.size();i++){
        totalSeekTime+=abs(diskRequests[i] - currPos);
        currPos=diskRequests[i];
    }
    cout<<"Total seek time using FCFS disk scheduling:"<<totalSeekTime<<endl;
    cout<<"Average seek time using FCFS disk scheduling:"<<totalSeekTime*1.0/n<<endl;
    return 0;
}
```



```
--------------------FCFS DISK SCHEDULING-------------------------
Enter head position:
50
Enter number of disk requests:
7
Enter request number 1:
82
Enter request number 2:
170
Enter request number 3:
43
Enter request number 4:
140
Enter request number 5:
24
Enter request number 6:
16
Enter request number 7:
190
Total seek time using FCFS disk scheduling:642
Average seek time using FCFS disk scheduling:91.7143
```

Figure 1: FCFS Output

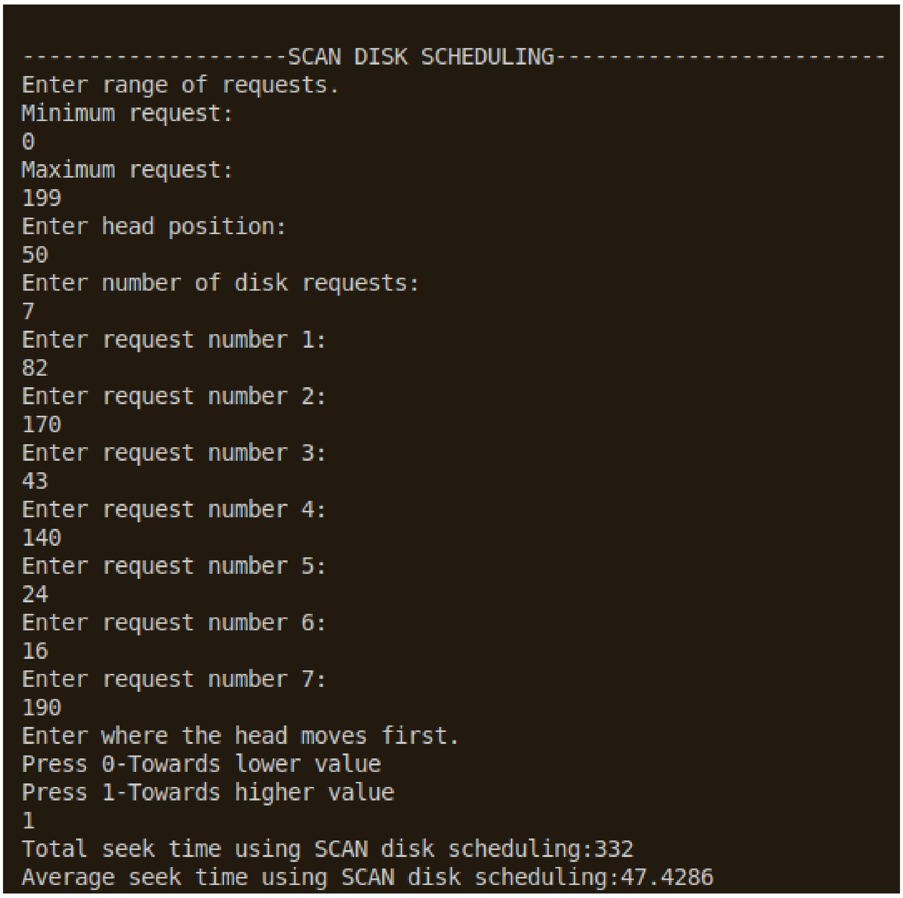
Q2. Write a CPP program to simulate the Disk scheduling algorithm SCAN.

Solution:

```cpp
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n,minRequest,maxRequest;
    cout<<"--------------------SCAN DISK SCHEDULING-------------------------"<<endl;
    cout<<"Enter range of requests.\nMinimum request:"<<endl;
    cin>>minRequest;
    cout<<"Maximum request:"<<endl;
    cin>>maxRequest;
    int head;
    cout<<"Enter head position:"<<endl;
    cin>>head;
    cout<<"Enter number of disk requests:"<<endl;
    cin>>n;
    vector<int>diskRequests(n);
    int maxUserRequest=minRequest,minUserRequest=maxRequest;
    for(int i=0;i<n;i++){
        cout<<"Enter request number "<<i+1<<":"<<endl;
        cin>>diskRequests[i];
        if(diskRequests[i]<minUserRequest) minUserRequest=diskRequests[i];
        if(diskRequests[i]>maxUserRequest) maxUserRequest=diskRequests[i];
    }
    bool isHigher;
    cout<<"Enter where the head moves first (0-Lower, 1-Higher):"<<endl;
    cin>>isHigher;
    int totalSeekTime;

    // SCAN Algorithm Logic
    if(isHigher)
        totalSeekTime=(maxRequest-head)+(maxRequest-minUserRequest);
    else
        totalSeekTime=(head-minRequest)+(maxUserRequest-minRequest);

    cout<<"Total seek time using SCAN disk scheduling:"<<totalSeekTime<<endl;
    cout<<"Average seek time using SCAN disk scheduling:"<<totalSeekTime*1.0/n<<endl;
    return 0;
}
```



```
--------------------SCAN DISK SCHEDULING-------------------------
Enter range of requests.
Minimum request:
0
Maximum request:
199
Enter head position:
50
Enter number of disk requests:
7
Enter request number 1:
82
Enter request number 2:
170
Enter request number 3:
43
Enter request number 4:
140
Enter request number 5:
24
Enter request number 6:
16
Enter request number 7:
190
Enter where the head moves first.
Press 0-Towards lower value
Press 1-Towards higher value
1
Total seek time using SCAN disk scheduling:332
Average seek time using SCAN disk scheduling:47.4286
```

Figure 2: SCAN Output

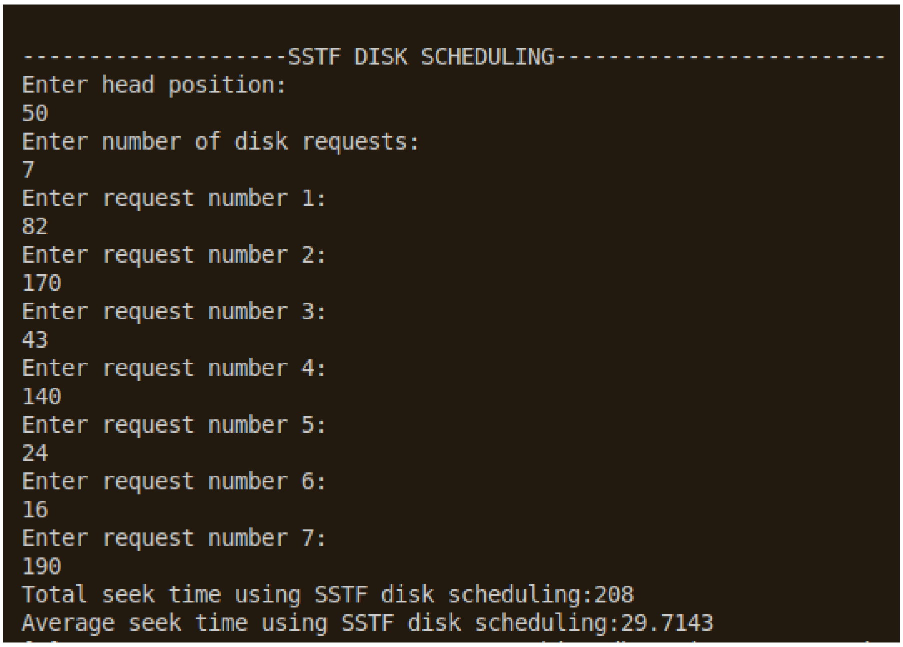
Q3. Write a CPP program to simulate the Disk scheduling algorithm SSTF.

Solution:

```cpp
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cout<<"--------------------SSTF DISK SCHEDULING-------------------------"<<endl;
    int head;
    cout<<"Enter head position:"<<endl;
    cin>>head;
    cout<<"Enter number of disk requests:"<<endl;
    cin>>n;
    vector<int>diskRequests(n);
    for(int i=0;i<n;i++){
        cout<<"Enter request number "<<i+1<<":"<<endl;
        cin>>diskRequests[i];
    }

    // Note: This logic sorts requests. Real SSTF finds nearest neighbor dynamically.
    sort(diskRequests.begin(),diskRequests.end());

    int totalSeekTime=0;
    int currPos=head;
    for(int i=0;i<diskRequests.size();i++){
        totalSeekTime+=abs(diskRequests[i] - currPos);
        currPos=diskRequests[i];
    }
    cout<<"Total seek time using SSTF disk scheduling:"<<totalSeekTime<<endl;
    cout<<"Average seek time using SSTF disk scheduling:"<<totalSeekTime*1.0/n<<endl;
    return 0;
}
```



```
--------------------SSTF DISK SCHEDULING------------------------
Enter head position:
50
Enter number of disk requests:
7
Enter request number 1:
82
Enter request number 2:
170
Enter request number 3:
43
Enter request number 4:
140
Enter request number 5:
24
Enter request number 6:
16
Enter request number 7:
190
Total seek time using SSTF disk scheduling:208
Average seek time using SSTF disk scheduling:29.7143
```

Figure 3: SSTF Output