Program 15:

(a) Write a program to implement the C-SCAN elevator disk scheduling algorithm. The program should give detail about each disk movement from starting head position (input from the user) and calculate average head movement.

(b) Write a program to implement the C-LOOK elevator disk scheduling algorithm. The program should give detail about each disk movement from starting head position (input from the user) and calculate average head movement.

Answer:

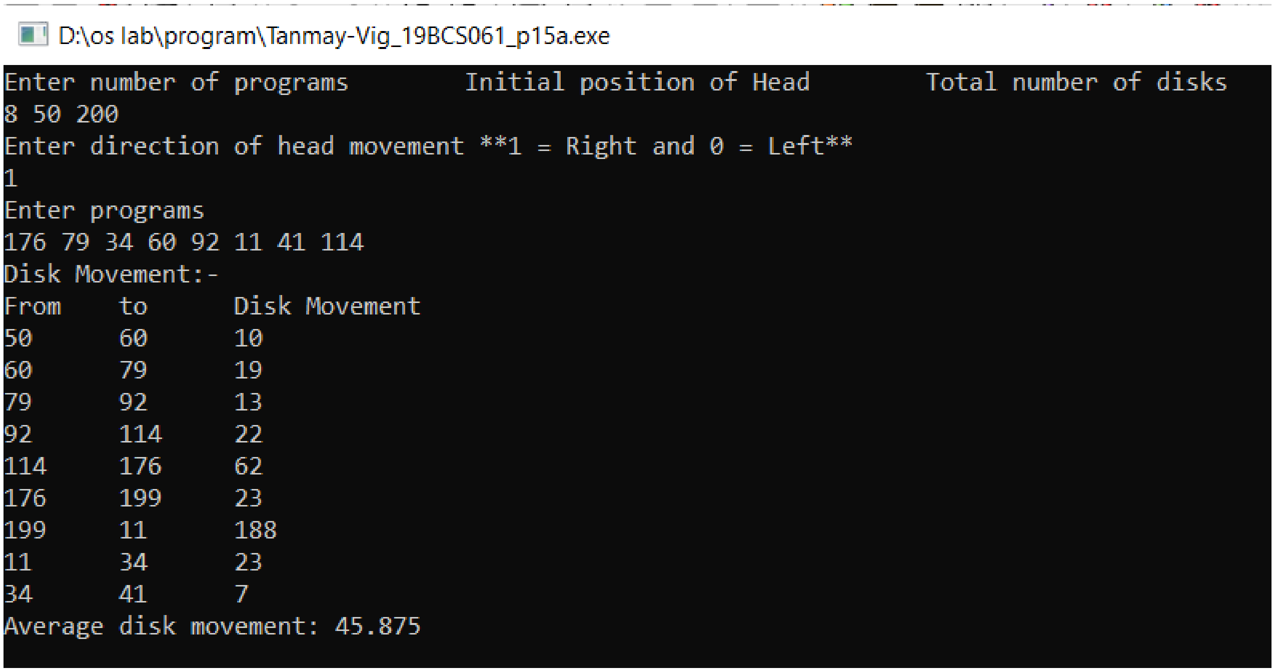
(a) Source Code:

```
#include <iostream>
#include <vector>
#include <cmath>
#include <algorithm>

using namespace std;
int search_(vector<int> programs,int lo, int hi, int x){// find point where arr[mid]<=head &&
arr[mid+1]>head
if(lo<hi){
   int mid=(hi-lo)/2;
   if(programs[mid]==x){
      return mid;
   }else if(programs[mid]<x){
      if((mid+1)==(int)programs.size() || programs[mid+1]>x) return mid;
      else search_(programs,lo,mid-1,x);
   }else{
      if((mid)==0 || programs[mid-1]<x) return mid;
      else search_(programs,mid+1,hi,x);
   }
}
return -1;
}
```

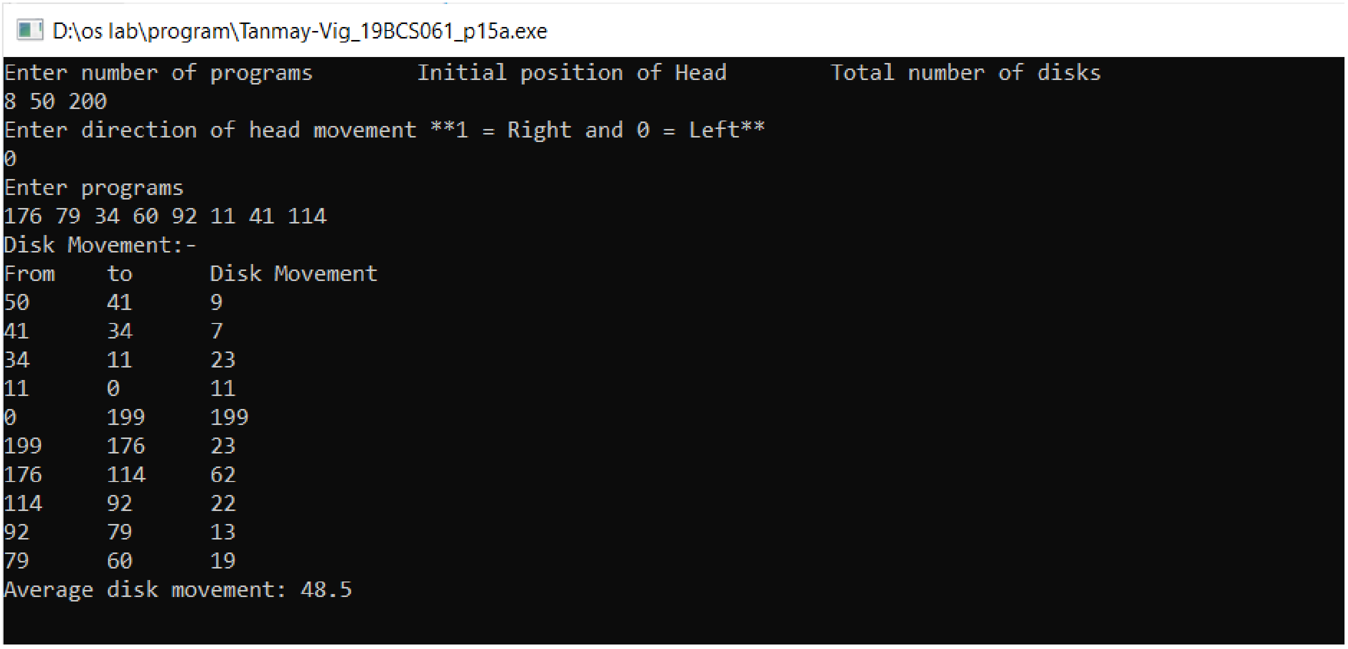
```
int piv(vector<int> &arr, int lo, int hi){
   int i=lo, p=arr[hi];
   for (int j=lo; j<=hi;j++){
      if(arr[j]<p){
         swap(arr[j],arr[i]);
         i+=1;
      }
   }
   swap(arr[hi],arr[i]);
   return i;
}

void sort_(vector<int> &arr,int lo, int hi){
   if(lo<hi){
      int p=piv(arr,lo,hi);
      sort_(arr,lo,p-1);
      sort_(arr,p+1,hi);
   }
}


int left_move(vector<int> programs, int from, int to, int pos){
   int sum=0;
   for(int i=from; i>=to; i--){
      sum+=abs(programs[i]-pos);
      cout<<pos<<"\t"<<programs[i]<<"\t"<<abs(programs[i]-pos)<<endl;
      pos=programs[i];
   }
   return sum;
}

int right_move(vector<int> programs, int from , int to, int pos){
   int sum=0;
   for(int i=from+1;i<=to;i++){
      sum+=abs(programs[i]-pos);
      cout<<pos<<"\t"<<programs[i]<<"\t"<<abs(programs[i]-pos)<<endl;
      pos=programs[i];
   }
   return sum;
}

int algo(vector<int> programs,int pos,int dir,int disk){
   programs.push_back(disk-1);
   programs.push_back(0);
   int sum=0,diff, n=(int)programs.size();

   sort_(programs,0,n-1); //sorting
   int ind=search_(programs,0,n-1,pos); //searching nearest index(0 based)
```

```
    if(programs[ind]>pos) ind-=1;

    cout<<"Disk Movement:-"<<endl;
    cout<<"From\tto\tDisk Movement"<<endl;

    if(dir==0){
        // for left side
        sum+=left_move(programs,ind, 0, pos);
        sum+=left_move(programs,n-1,ind+1,0);
    }else{
        // for left side
        sum+=right_move(programs,ind,n-1,pos);
        sum+=right_move(programs,0,ind,disk-1);
    }
    return sum;
}
int main()
{
    int n,pos,dir,disk;
    cout << "Enter number of programs\tInitial position of Head\tTotal number of disks"<<endl;
    cin>>n>>pos>>disk;
    cout << "Enter direction of head movement **1 = Right and 0 = Left**"<<endl;
    cin>>dir;
    vector<int> programs(n);
    cout<<"Enter programs"<<endl;
    for (int i=0; i<n;i++){
        cin>>programs[i];
    }
    int total_movements=algo(programs,pos,dir,disk);
    total_movements=(float)total_movements;
    cout<<"Average disk movement: "<<total_movements/(float)n<<endl;
    return 0;
}
```

Output:



```
Enter number of programs        Initial position of Head        Total number of disks
8 50 200
Enter direction of head movement **1 = Right and 0 = Left**
1
Enter programs
176 79 34 60 92 11 41 114
Disk Movement:-
From    to      Disk Movement
50      60      10
60      79      19
79      92      13
92      114     22
114     176     62
176     199     23
199     11      188
11      34      23
34      41      7
Average disk movement: 45.875
```



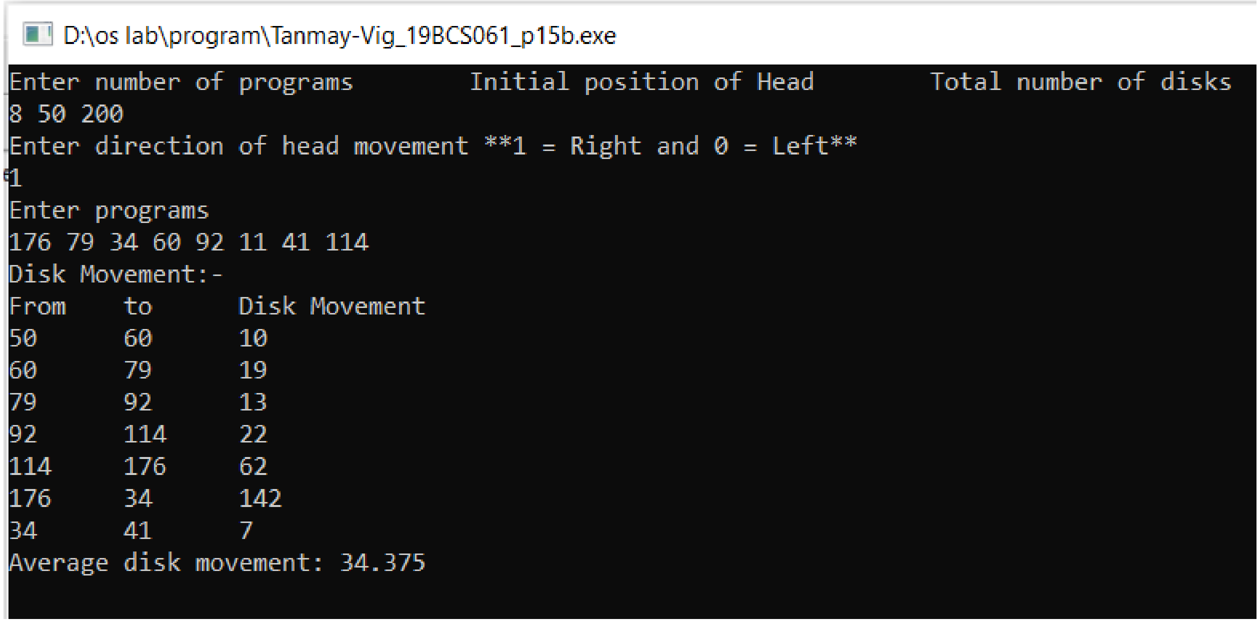
```
Enter number of programs        Initial position of Head        Total number of disks
8 50 200
Enter direction of head movement **1 = Right and 0 = Left**
0
Enter programs
176 79 34 60 92 11 41 114
Disk Movement:-
From    to      Disk Movement
50      41      9
41      34      7
34      11      23
11      0       11
0       199     199
199     176     23
176     114     62
114     92      22
92      79      13
79      60      19
Average disk movement: 48.5
```

(b) Source code:

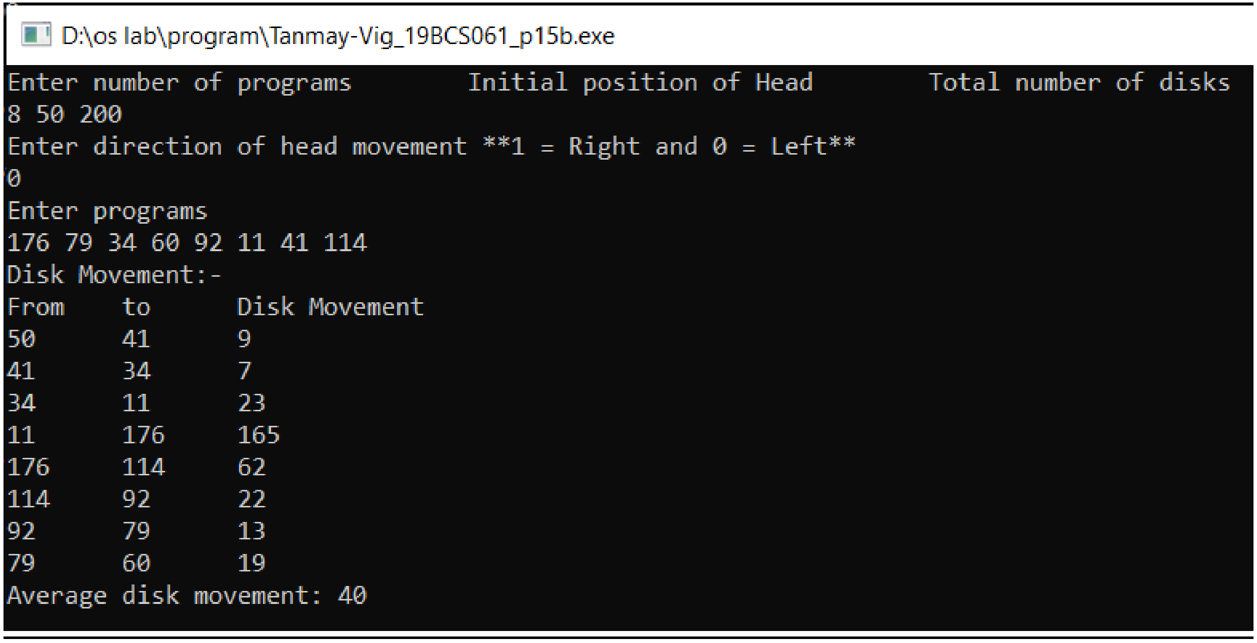
```
#include <iostream>
#include <vector>
#include <cmath>
#include <algorithm>

using namespace std;
int search_(vector<int> programs,int lo, int hi, int x){// find point where arr[mid]<=head && arr[mid+1]>head
if(lo<hi){
   int mid=(hi-lo)/2;
   if(programs[mid]==x){
      return mid;
   }else if(programs[mid]<x){
      if((mid+1)==(int)programs.size() || programs[mid+1]>x) return mid;
      else search_(programs,lo,mid-1,x);
   }else{
      if((mid)==0 || programs[mid-1]<x) return mid;
      else search_(programs,mid+1,hi,x);
   }
}
return -1;
}

int piv(vector<int> &arr, int lo, int hi){
   int i=lo, p=arr[hi];
   for (int j=lo; j<=hi;j++){
      if(arr[j]<p){
         swap(arr[j],arr[i]);
         i+=1;
      }
   }
   swap(arr[hi],arr[i]);
   return i;
}

void sort_(vector<int> &arr,int lo, int hi){
   if(lo<hi){
      int p=piv(arr,lo,hi);
      sort_(arr,lo,p-1);
      sort_(arr,p+1,hi);
   }
}


int left_move(vector<int> programs, int from, int to, int pos){
   int sum=0;
   for(int i=from; i>=to; i--){
      sum+=abs(programs[i]-pos);
      cout<<pos<<"\t"<<programs[i]<<"\t"<<abs(programs[i]-pos)<<endl;
      pos=programs[i];
   }
   return sum;
}

int right_move(vector<int> programs, int from , int to, int pos){
   int sum=0;
   for(int i=from+1;i<=to;i++){
      sum+=abs(programs[i]-pos);
      cout<<pos<<"\t"<<programs[i]<<"\t"<<abs(programs[i]-pos)<<endl;
      pos=programs[i];
   }
   return sum;
```

```
}

int algo(vector<int> programs,int pos,int dir,int disk){
   int sum=0,diff, n=(int)programs.size();

   sort_(programs,0,n-1); //sorting
   int ind=search_(programs,0,n-1,pos); //searching nearest index(0 based)
   if(programs[ind]>pos) ind-=1;
   cout<<"Disk Movement:-"<<endl;
   cout<<"From\tto\tDisk Movement"<<endl;

   if(dir==0){
      // for left side
      sum+=left_move(programs,ind, 0, pos);
      sum+=left_move(programs,n-1,ind+1,programs[0]);
   }else{
      // for left side
      sum+=right_move(programs,ind,n-1,pos);
      sum+=right_move(programs,0,ind,programs[n-1]);
   }
   return sum;
}
int main()
{
   int n,pos,dir,disk;
   cout << "Enter number of programs\tInitial position of Head\tTotal number of disks"<<endl;
   cin>>n>>pos>>disk;
   cout << "Enter direction of head movement **1 = Right and 0 = Left**"<<endl;
   cin>>dir;
   vector<int> programs(n);
   cout<<"Enter programs"<<endl;
   for (int i=0; i<n;i++){
      cin>>programs[i];
   }
   int total_movements=algo(programs,pos,dir,disk);
   total_movements=(float)total_movements;
   cout<<"Average disk movement: "<<total_movements/(float)n<<endl;
   return 0;
}
```

Output:



```
Enter number of programs        Initial position of Head        Total number of disks
8 50 200
Enter direction of head movement **1 = Right and 0 = Left**
1
Enter programs
176 79 34 60 92 11 41 114
Disk Movement:-
From    to      Disk Movement
50      60      10
60      79      19
79      92      13
92      114     22
114     176     62
176     34      142
34      41      7
Average disk movement: 34.375
```



```
Enter number of programs        Initial position of Head        Total number of disks
8 50 200
Enter direction of head movement **1 = Right and 0 = Left**
0
Enter programs
176 79 34 60 92 11 41 114
Disk Movement:-
From    to      Disk Movement
50      41      9
41      34      7
34      11      23
11      176     165
176     114     62
114     92      22
92      79      13
79      60      19
Average disk movement: 40
```