1. **FCFS: first come first serve**

**Input:**

//fcfs

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

 int n,request[20],ch,tdhm=0,i;

 cout<<"Enter the number of requests: ";

 cin>>n;

 cout<<"\nEnter the requests: ";

 for (i=1;i<=n;i++)

 {

  cin>>request[i];

 }

 cout<<"\nEnter the current head position: ";

 cin>>ch;

 for (i=1;i<=n;i++)

 {

 tdhm= tdhm + abs(ch- request[i]);

  ch=request[i];

 }

 cout<<"\nThe total disk head movement is: "<<tdhm<<endl;

return 0;

}

**Output:**

Enter the number of requests: 8

Enter the requests: 98 183 37 122 14 124 65 67

Enter the current head position: 53

The total disk head movement is: 640

1. **Sstf: shortest seek time first**

**Input:**

//sstf

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

 int n,request[20],ch,tdhm=0,i,Count=1,dist=0,index;

 cout<<"Enter the number of requests: ";

 cin>>n;

 cout<<"\nEnter the requests: ";

 for (i=1;i<=n;i++)

 {

  cin>>request[i];

 }

 cout<<"\nEnter the current head position: ";

 cin>>ch;

 while (Count<=n)

 {

  int Min=9999;

  for(i=1;i<=n;i++)

  {

   dist=abs(ch-request[i]);

   if (dist<Min)

   {

    Min=dist;

    index=i;

   }

  }

  tdhm=tdhm+Min;

  ch=request[index];

  request[index]=9999;

  Count++;

 }

cout<<"\nThe total disk head movement is: "<<tdhm<<endl;

return 0;

 }

**Output**:

Enter the number of requests: 8

Enter the requests: 98 183 37 122 14 124 65 67

Enter the current head position: 53

The total disk head movement is: 236

1. **Scan**:

**Input:**

//scan

#include <iostream>

using namespace std;

void bubbleSort(int arr[], int n) {

    for (int i = 0; i < n - 1; i++) {

        for (int j = 0; j < n - i - 1; j++) {

            if (arr[j] > arr[j + 1]) {

                int temp = arr[j];

                arr[j] = arr[j + 1];

                arr[j + 1] = temp;

            }

        }

    }

}

void scanleft(int ch, int request[], int n, int startcyl, int largestcyl) {

    int tdhm = (ch - startcyl) + (largestcyl - startcyl);

    cout << "\nTotal Disk Head Movement (Left): " << tdhm << endl;

}

void scanright(int ch, int request[], int n, int endcyl, int smallestcyl) {

    int tdhm = (endcyl - ch) + (endcyl - smallestcyl);

    cout << "\nTotal Disk Head Movement (Right): " << tdhm << endl;

}

int main()

 {

    int n, request[20], startcyl, endcyl, ch, direction;

    cout << "\nEnter the number of requests: ";

    cin >> n;

    cout << "\nEnter the requests: ";

    for (int i = 0; i < n; i++) {

        cin >> request[i];

    }

    cout << "\nEnter the current head position: ";

    cin >> ch;

    cout << "\nEnter starting cylinder: ";

    cin >> startcyl;

    cout << "Enter ending cylinder: ";

    cin >> endcyl;

    bubbleSort(request, n);

    cout << "\nSorted requests: ";

    for (int i = 0; i < n; i++) {

        cout << request[i] << " ";

    }

    cout << endl;

    int smallestcyl = request[0];

    int largestcyl = request[n - 1];

    cout << "\nSmallest cylinder: " << smallestcyl << endl;

    cout << "Largest cylinder: " << largestcyl << endl;

    cout << "\nEnter the direction it is moving: \n1.Left\n2.Right\n\nYour choice: ";

    cin >> direction;

    switch (direction) {

        case 1:

            scanleft(ch, request, n, startcyl, largestcyl);

            break;

        case 2:

            scanright(ch, request, n, endcyl, smallestcyl);

            break;

        default:

            cout << "Invalid direction." << endl;

    }

    return 0;

}

**Output:**

Enter the number of requests: 8

Enter the requests: 98 183 37 122 14 124 65 67

Enter the current head position: 53

Enter starting cylinder: 0

Enter ending cylinder: 199

Sorted requests: 14 37 65 67 98 122 124 183

Smallest cylinder: 14

Largest cylinder: 183

Enter the direction it is moving:

1.Left

2.Right

Your choice: 2

Total Disk Head Movement (Right): 331

1. **C-scan:**

**Input:**

//cscan

#include <iostream>

using namespace std;

void bubbleSort(int arr[], int n) {

    for (int i = 0; i < n - 1; i++) {

        for (int j = 0; j < n - i - 1; j++) {

            if (arr[j] > arr[j + 1]) {

                int temp = arr[j];

                arr[j] = arr[j + 1];

                arr[j + 1] = temp;

            }

        }

    }

}

void scanleft(int ch, int request[], int n, int startcyl, int largestcyl,int endcyl,int rhead) {

    int tdhm = (ch - startcyl) + endcyl + (endcyl - rhead) ;

    cout << "\nTotal Disk Head Movement (Left): " << tdhm << endl;

}

void scanright(int ch, int request[], int n, int endcyl, int smallestcyl,int startcyl,int lhead) {

    int tdhm = (endcyl - ch) + endcyl + (lhead - startcyl);

    cout << "\nTotal Disk Head Movement (Right): " << tdhm << endl;

}

int main()

 {

    int n, request[20],a[20] ,startcyl, endcyl, ch, direction,lhead,rhead,i;

    cout << "\nEnter the number of requests: ";

    cin >> n;

    cout << "\nEnter the requests: ";

    for (int i = 0; i < n; i++) {

        cin >> request[i];

    }

    cout << "\nEnter the current head position: ";

    cin >> ch;

    cout << "\nEnter starting cylinder: ";

    cin >> startcyl;

    cout << "Enter ending cylinder: ";

    cin >> endcyl;

    for (int i = 0; i < n; i++)

     {

        a[i]=request[i] ;          //temporary array

    }

     a[n]=ch;

    bubbleSort(a, n+1);

for (i=0;i<=n;i++)

{

    if (a[i]==ch)

    {

     lhead=a[i-1];

     rhead=a[i+1];

    }

}

    bubbleSort(request, n);

    cout << "\nSorted requests: ";

    for (int i = 0; i < n; i++) {

        cout << request[i] << " ";

    }

    for (int i = 0; i < n; i++) {

        a[i]=request[i] ;

    }

    int smallestcyl = request[0];

    int largestcyl = request[n - 1];

    cout << "\nEnter the direction it is moving: \n1.Left\n2.Right\n\nYour choice: ";

    cin >> direction;

    switch (direction) {

        case 1:

            scanleft(ch, request, n, startcyl, largestcyl, endcyl,rhead);

            break;

        case 2:

            scanright(ch, request, n, endcyl, smallestcyl,startcyl,lhead);

            break;

        default:

            cout << "Invalid direction." << endl;

    }

    return 0;

}

**Output:**

Enter the number of requests: 8

Enter the requests: 98 183 37 122 14 124 65 67

Enter the current head position: 53

Enter starting cylinder: 0

Enter ending cylinder: 199

Sorted requests: 14 37 65 67 98 122 124 183

Enter the direction it is moving:

1.Left

2.Right

Your choice: 2

Total Disk Head Movement (Right): 382

1. **Look:**

**Input:**

//LOOK

#include <iostream>

using namespace std;

void bubbleSort(int arr[], int n) {

    for (int i = 0; i < n - 1; i++) {

        for (int j = 0; j < n - i - 1; j++) {

            if (arr[j] > arr[j + 1]) {

                int temp = arr[j];

                arr[j] = arr[j + 1];

                arr[j + 1] = temp;

            }

        }

    }

}

void scanleft(int ch, int smallestcyl, int largestcyl) {

    int tdhm = (ch - smallestcyl) + (largestcyl - smallestcyl) ;

    cout << "\nTotal Disk Head Movement (Left): " << tdhm << endl;

}

void scanright(int ch,int largestcyl, int smallestcyl) {

    int tdhm = (largestcyl - ch) + (largestcyl - smallestcyl) ;

    cout << "\nTotal Disk Head Movement (Right): " << tdhm << endl;

}

int main()

 {

    int n, request[20], ch, direction,i;

    cout << "\nEnter the number of requests: ";

    cin >> n;

    cout << "\nEnter the requests: ";

    for (i = 0; i < n; i++) {

        cin >> request[i];

    }

    cout << "\nEnter the current head position: ";

    cin >> ch;

    bubbleSort(request, n);

    cout << "Sorted requests: ";

    for (i = 0; i < n; i++) {

        cout << request[i] << " ";

    }

    int smallestcyl = request[0];

    int largestcyl = request[n - 1];

    cout << "\nEnter the direction it is moving: \n1.Left\n2.Right\n\nYour choice";

    cin >> direction;

    switch (direction) {

        case 1:

            scanleft(ch, smallestcyl, largestcyl);

            break;

        case 2:

            scanright(ch, largestcyl, smallestcyl);

            break;

        default:

            cout << "Invalid direction." << endl;

    }

    return 0;

}

**Output:**

Enter the number of requests: 8

Enter the requests: 98 183 37 122 14 124 65 67

Enter the current head position: 53

Sorted requests: 14 37 65 67 98 122 124 183

Enter the direction it is moving:

1.Left

2.Right

Your choice2

Total Disk Head Movement (Right): 299

1. **C-look:**

**Input:**

//CLOOK

#include <iostream>

using namespace std;

void bubbleSort(int arr[], int n) {

    for (int i = 0; i < n - 1; i++) {

        for (int j = 0; j < n - i - 1; j++) {

            if (arr[j] > arr[j + 1]) {

                int temp = arr[j];

                arr[j] = arr[j + 1];

                arr[j + 1] = temp;

            }

        }

    }

}

void scanleft(int ch, int smallestcyl, int largestcyl,int rhead) {

    int tdhm = (ch - smallestcyl) + (largestcyl - smallestcyl) + (largestcyl - rhead) ;

    cout << "\nTotal Disk Head Movement (Left): " << tdhm << endl;

}

void scanright(int ch,int largestcyl, int smallestcyl,int lhead) {

    int tdhm = (largestcyl - ch) + (largestcyl - smallestcyl) + (lhead - smallestcyl);

    cout << "\nTotal Disk Head Movement (Right): " << tdhm << endl;

}

int main()

 {

    int n, request[20],a[20] , ch, direction,lhead,rhead,i;

    cout << "\nEnter the number of requests: ";

    cin >> n;

    cout << "\nEnter the requests: ";

    for (i = 0; i < n; i++) {

        cin >> request[i];

    }

    cout << "\nEnter the current head position: ";

    cin >> ch;

    for (i = 0; i < n; i++)

     {

        a[i]=request[i] ;          //temporary array

    }

     a[n]=ch;

    bubbleSort(a, n+1);

for (i=0;i<=n;i++)

{

    if (a[i]==ch)

    {

     lhead=a[i-1];

     rhead=a[i+1];

    }

}

    bubbleSort(request, n);

    cout << "Sorted requests: ";

    for (i = 0; i < n; i++) {

        cout << request[i] << " ";

    }

    cout << endl;

    for (i = 0; i < n; i++) {

        a[i]=request[i] ;

    }

    int smallestcyl = request[0];

    int largestcyl = request[n - 1];

    cout << "\nEnter the direction it is moving: \n1.Left\n2.Right\n\nYour choice: " << endl;

    cin >> direction;

    switch (direction) {

        case 1:

            scanleft(ch, smallestcyl, largestcyl, rhead);

            break;

        case 2:

            scanright(ch, largestcyl, smallestcyl, lhead);

            break;

        default:

            cout << "Invalid direction." << endl;

    }

    return 0;

}

**Output:**

Enter the number of requests: 8

Enter the requests: 98 183 37 122 14 124 65 67

Enter the current head position: 53

Sorted requests: 14 37 65 67 98 122 124 183

Enter the direction it is moving:

1.Left

2.Right

Your choice:

2

Total Disk Head Movement (Right): 322