DISK SCHEDULING ALGORITHM

```
#include <stdio.h>
#include <stdlib.h>

int m, n, start; // Global variables for disk specifications
int a[15];       // Global array for the request queue

int absolute(int a, int b)
{
   int c = a - b;
   if (c < 0)
      return -c;
   else
      return c;
}

void fcfs()
{
   printf("\nFCFS:\n");
   int count = 0;
   int x = start;
   printf("Scheduling services the request in the order that follows:\n%d\t", start);
   for (int i = 0; i < n; i++)
   {
      x -= a[i];
      if (x < 0)
         x = -x;
      count += x;
      x = a[i];
      printf("%d\t", x);
```

```
    }
    printf("\nTotal Head Movement: %d Cylinders\n", count);
}

void sstf()
{
    printf("\nSSTF:\n");
    int count = 0;
    int x = start;
    printf("Scheduling services the request in the order that follows:\n%d\t", start);
    for (int i = 0; i < n; i++)
    {
        int min = absolute(a[i], x);
        int pos = i;
        for (int j = i; j < n; j++)
        {
            if (min > absolute(x, a[j]))
            {
                pos = j;
                min = absolute(x, a[j]);
            }
        }
        count += absolute(x, a[pos]);
        x = a[pos];
        a[pos] = a[i];
        a[i] = x;
        printf("%d\t", x);
    }
    printf("\nTotal Head Movement: %d Cylinders\n", count);
}
```

```
//scan
void scan(int direction)
{
   printf("\nSCAN:\n");
   int count = 0;
   int pos = 0;

   for (int i = 0; i < n; i++)
   {
      for (int j = 0; j < n - i - 1; j++)
      {
         if (a[j] > a[j + 1])
         {
            int temp = a[j];
            a[j] = a[j + 1];
            a[j + 1] = temp;
         }
      }
   }

   for (int i = 0; i < n; i++)
   {
      if (a[i] < start)
         pos++;
   }

   int x = start;

   if (direction == 1) // Right direction
   {
      for (int i = pos; i < n; i++)
```

```
    {
        count += absolute(a[i], x);
        x = a[i];
        printf("%d\t", x);
    }
    if (x != m - 1)
    {
        count += absolute(x, m - 1);
        x = m - 1;
        printf("%d\t", x);
    }
    for (int i = pos - 1; i >= 0; i--)
    {
        count += absolute(a[i], x);
        x = a[i];
        printf("%d\t", x);
    }
}
else // Left direction
{
    for (int i = pos - 1; i >= 0; i--)
    {
        count += absolute(a[i], x);
        x = a[i];
        printf("%d\t", x);
    }
    if (x != 0)
    {
        count += absolute(x, 0);
        x = 0;
        printf("%d\t", x);
```

```
        }
        for (int i = pos; i < n; i++)
        {
            count += absolute(a[i], x);
            x = a[i];
            printf("%d\t", x);
        }
    }

    printf("\nTotal Head Movement: %d Cylinders\n", count);
}

void look(int direction)
{
    printf("\nLOOK:\n");
    int count = 0;
    int pos = 0;

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (a[j] > a[j + 1])
            {
                int temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
            }
        }
    }
```

```
for (int i = 0; i < n; i++)
{
   if (a[i] < start)
      pos++;
}

int x = start;

if (direction == 1) // Right direction
{
   for (int i = pos; i < n; i++)
   {
      count += absolute(a[i], x);
      x = a[i];
      printf("%d\t", x);
   }
   for (int i = pos - 1; i >= 0; i--)
   {
      count += absolute(a[i], x);
      x = a[i];
      printf("%d\t", x);
   }
}
else // Left direction
{
   for (int i = pos - 1; i >= 0; i--)
   {
      count += absolute(a[i], x);
      x = a[i];
      printf("%d\t", x);
   }
```

```
        for (int i = pos; i < n; i++)
        {
            count += absolute(a[i], x);
            x = a[i];
            printf("%d\t", x);
        }
    }

    printf("\nTotal Head Movement: %d Cylinders\n", count);
}




void cscan(int direction)
{
    printf("\nC-SCAN:\n");
    int count = 0;
    int pos = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (a[j] > a[j + 1])
            {
                int temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
            }
        }
    }
    for (int i = 0; i < n; i++)
```

```
{
    if (a[i] < start)
        pos++;
}

int x = start;

if (direction == 1) // Right direction
{
    for (int i = pos; i < n; i++)
    {
        count += absolute(x, a[i]);
        x = a[i];
        printf("%d\t", x);
    }
    count += absolute(m - 1, x);
    x = 0;
    printf("%d\t%d\t", m - 1, 0);
    for (int i = 0; i < pos; i++)
    {
        count += absolute(x, a[i]);
        x = a[i];
        printf("%d\t", x);
    }
}
else // Left direction
{
    for (int i = pos - 1; i >= 0; i--)
    {
        count += absolute(x, a[i]);
        x = a[i];
```

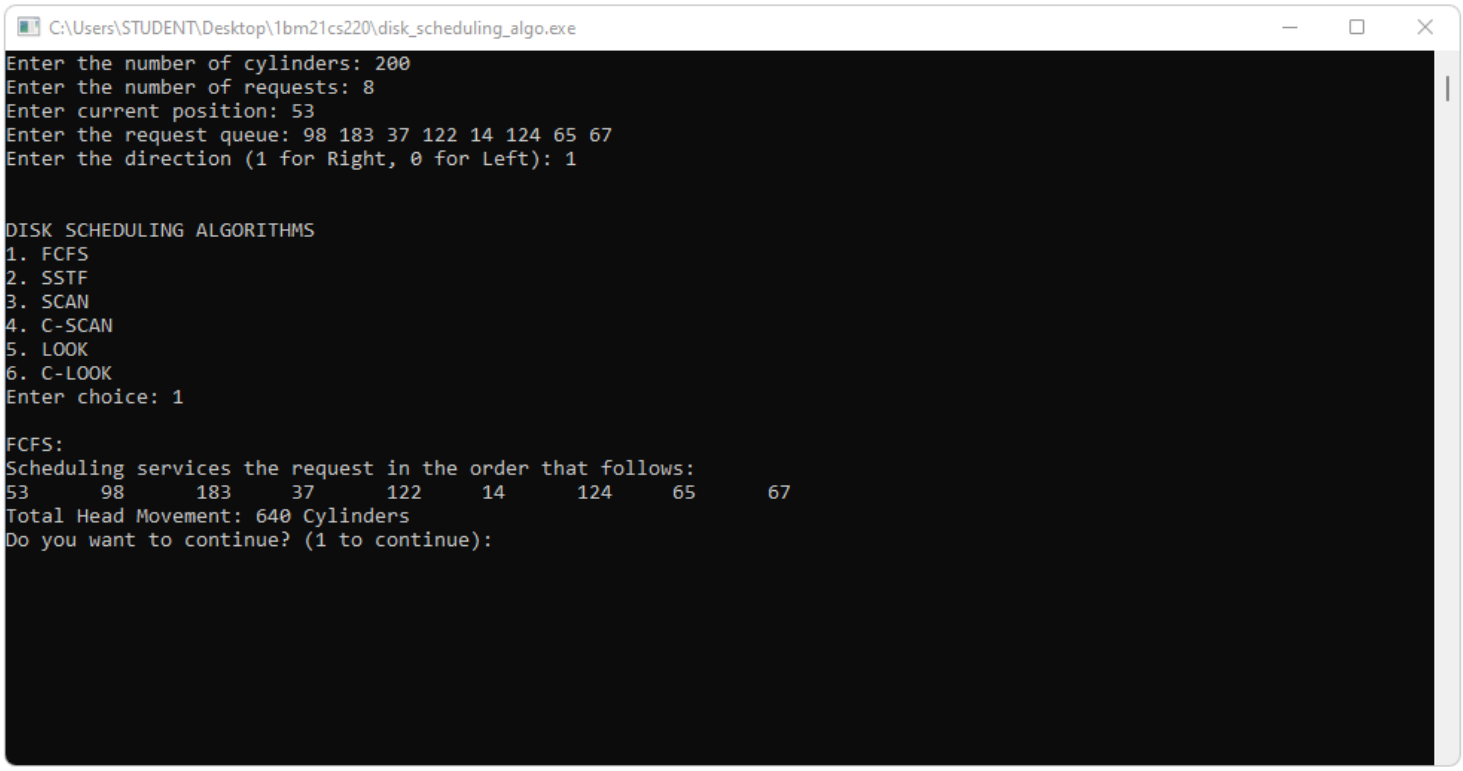
```
            printf("%d\t", x);
        }
        count += absolute(0, x);
        x = m - 1;
        printf("%d\t%d\t", 0, x);
        for (int i = n - 1; i >= pos; i--)
        {
            count += absolute(x, a[i]);
            x = a[i];
            printf("%d\t", x);
        }
    }

    printf("\nTotal Head Movement: %d Cylinders\n", count);
}

//look



void clook(int direction)
{
    printf("\nC-LOOK:\n");
    int count = 0;
    int pos = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (a[j] > a[j + 1])
            {
                int temp = a[j];
```

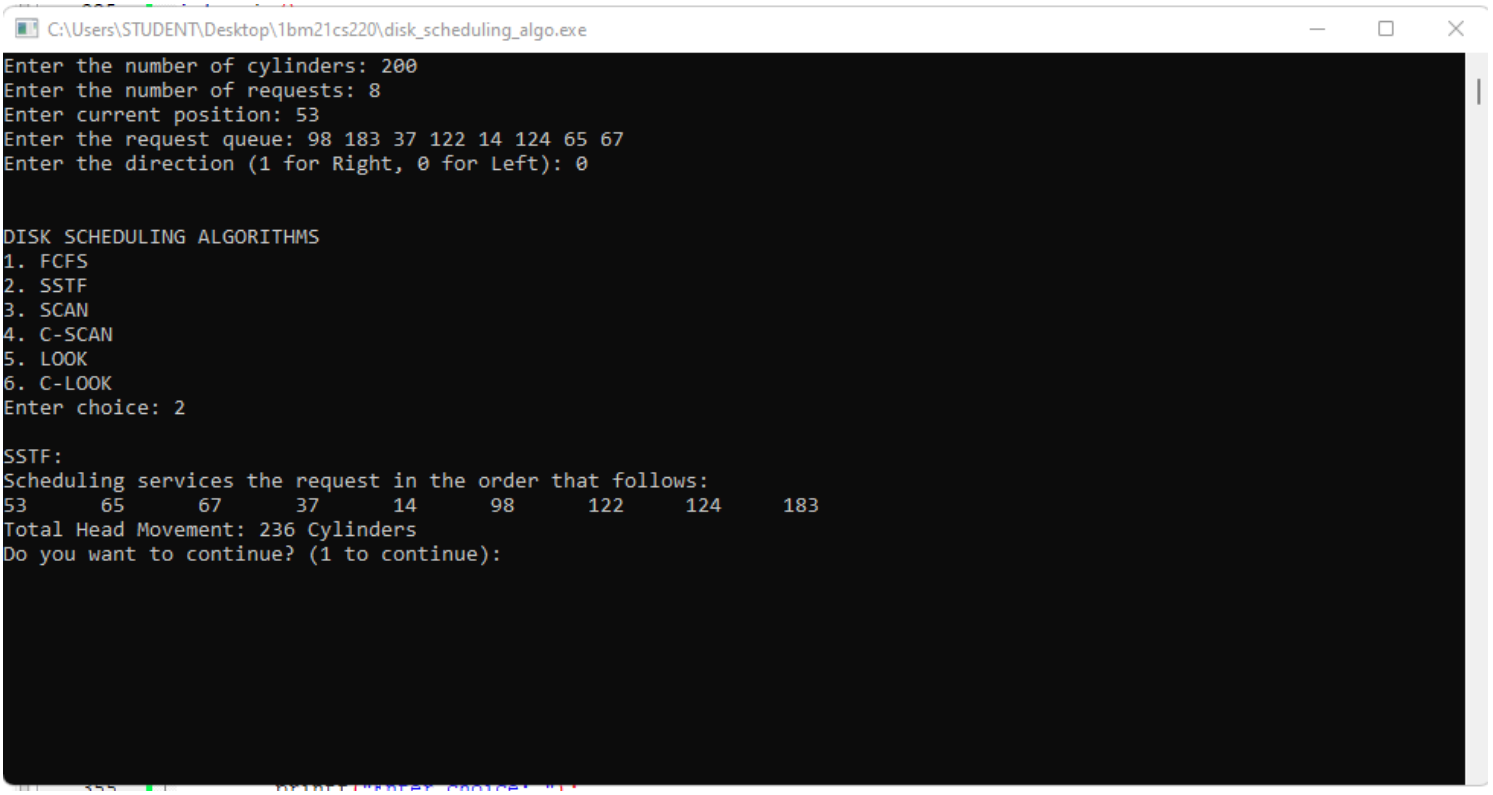
```
            a[j] = a[j + 1];
            a[j + 1] = temp;
        }
    }
}
for (int i = 0; i < n; i++)
{
    if (a[i] < start)
        pos++;
}

int x = start;

if (direction == 1) // Right direction
{
    for (int i = pos; i < n; i++)
    {
        count += absolute(x, a[i]);
        x = a[i];
        printf("%d\t", x);
    }
    for (int i = 0; i < pos; i++)
    {
        count += absolute(x, a[i]);
        x = a[i];
        printf("%d\t", x);
    }
}
else // Left direction
{
    for (int i = pos - 1; i >= 0; i--)
```

```
        {
            count += absolute(x, a[i]);
            x = a[i];
            printf("%d\t", x);
        }
        for (int i = n - 1; i >= pos; i--)
        {
            count += absolute(x, a[i]);
            x = a[i];
            printf("%d\t", x);
        }
    }

    printf("\nTotal Head Movement: %d Cylinders\n", count);
}

int main()
{
    int choice, direction;

    printf("Enter the number of cylinders: ");
    scanf("%d", &m);

    printf("Enter the number of requests: ");
    scanf("%d", &n);

    printf("Enter current position: ");
    scanf("%d", &start);

    printf("Enter the request queue: ");
    for (int i = 0; i < n; i++)
```

```
    {
        scanf("%d", &a[i]);
        if (a[i] >= m)
        {
            printf("\nInvalid input, re-enter: ");
            scanf("%d", &a[i]);
        }
    }

    printf("Enter the direction (1 for Right, 0 for Left): ");
    scanf("%d", &direction);

    do
    {
        printf("\n\nDISK SCHEDULING ALGORITHMS\n1. FCFS\n2. SSTF\n3. SCAN\n4. C-SCAN\n5.
LOOK\n6. C-LOOK\n");
        printf("Enter choice: ");
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            fcfs();
            break;
        case 2:
            sstf();
            break;
        case 3:
            scan(direction);
            break;
        case 4:
```

```
            cscan(direction);
            break;
        case 5:
            look(direction);
            break;
        case 6:
            clook(direction);
            break;
        default:
            printf("Invalid choice\n");
        }

        printf("Do you want to continue? (1 to continue): ");
        scanf("%d", &choice);
    } while (choice == 1);

    return 0;
}
```


C:\Users\STUDENT\Desktop\1bm21cs220\disk_scheduling_algo.exe
Enter the number of cylinders: 200
Enter the number of requests: 8
Enter current position: 53
Enter the request queue: 98 183 37 122 14 124 65 67
Enter the direction (1 for Right, 0 for Left): 1
DISK SCHEDULING ALGORITHMS
1. FCFS
2. SSTF
3. SCAN
4. C-SCAN
5. LOOK
6. C-LOOK
Enter choice: 1
FCFS:
Scheduling services the request in the order that follows:
53 98 183 37 122 14 124 65 67
Total Head Movement: 640 Cylinders
Do you want to continue? (1 to continue):


C:\Users\STUDENT\Desktop\1bm21cs220\disk_scheduling_algo.exe
Enter the number of cylinders: 200
Enter the number of requests: 8
Enter current position: 53
Enter the request queue: 98 183 37 122 14 124 65 67
Enter the direction (1 for Right, 0 for Left): 0
DISK SCHEDULING ALGORITHMS
1. FCFS
2. SSTF
3. SCAN
4. C-SCAN
5. LOOK
6. C-LOOK
Enter choice: 2
SSTF:
Scheduling services the request in the order that follows:
53 65 67 37 14 98 122 124 183
Total Head Movement: 236 Cylinders
Do you want to continue? (1 to continue):

```
DISK SCHEDULING ALGORITHMS
1. FCFS
2. SSTF
3. SCAN
4. C-SCAN
5. LOOK
6. C-LOOK
Enter choice: 3

SCAN:
37      14      0       65      67      98      122     124     183
Total Head Movement: 236 Cylinders
Do you want to continue? (1 to continue): 1


DISK SCHEDULING ALGORITHMS
1. FCFS
2. SSTF
3. SCAN
4. C-SCAN
5. LOOK
6. C-LOOK
Enter choice: 4

C-SCAN:
37      14      0       199     183     124     122     98      67      65
Total Head Movement: 187 Cylinders
Do you want to continue? (1 to continue): 1


DISK SCHEDULING ALGORITHMS
1. FCFS
2. SSTF
3. SCAN
4. C-SCAN
5. LOOK
6. C-LOOK
Enter choice: 5

LOOK:
37      14      65      67      98      122     124     183
Total Head Movement: 208 Cylinders
Do you want to continue? (1 to continue): 1


DISK SCHEDULING ALGORITHMS
1. FCFS
2. SSTF
3. SCAN
4. C-SCAN
5. LOOK
6. C-LOOK
Enter choice: 6

C-LOOK:
37      14      183     124     122     98      67      65
Total Head Movement: 326 Cylinders
Do you want to continue? (1 to continue):
```