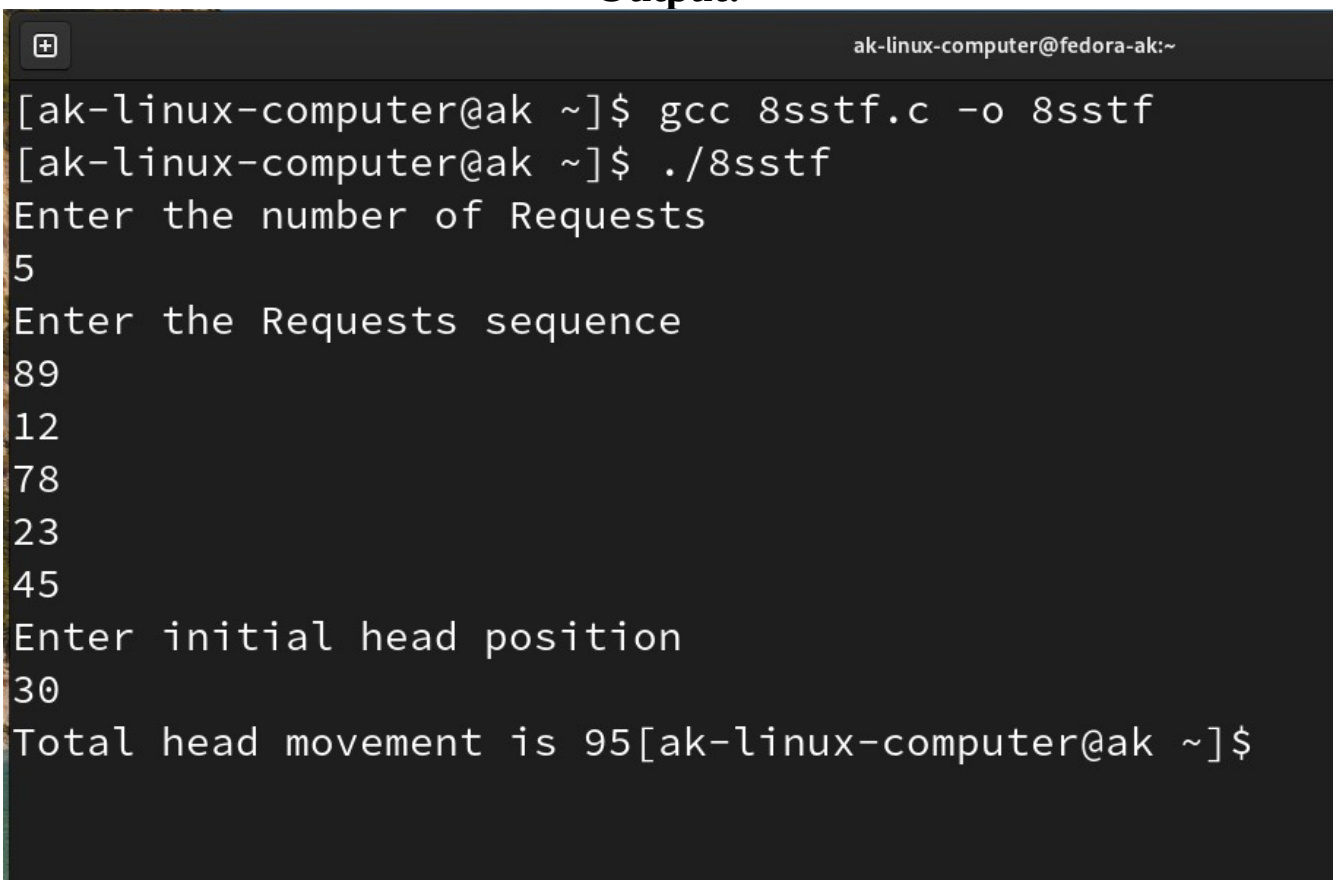
Assignment 8

SSTF code:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
   int RQ[100], i, n, TotalHeadMoment = 0, initial, count = 0;
   printf("Enter the number of Requests\n");
   scanf("%d", &n);
   printf("Enter the Requests sequence\n");
   for (i = 0; i < n; i++)
      scanf("%d", &RQ[i]);
   printf("Enter initial head position\n");
   scanf("%d", &initial);

   // logic for sstf disk scheduling

   /* loop will execute until all process is completed*/
   while (count != n)
   {
      int min = 1000, d, index;
      for (i = 0; i < n; i++)
      {
         d = abs(RQ[i] - initial);
         if (min > d)
         {
            min = d;
            index = i;
         }
      }
      TotalHeadMoment = TotalHeadMoment + min;
      initial = RQ[index];
      // 1000 is for max
      // you can use any number
      RQ[index] = 1000;
      count++;
   }

   printf("Total head movement is %d", TotalHeadMoment);
   return 0;
}
```

Output:

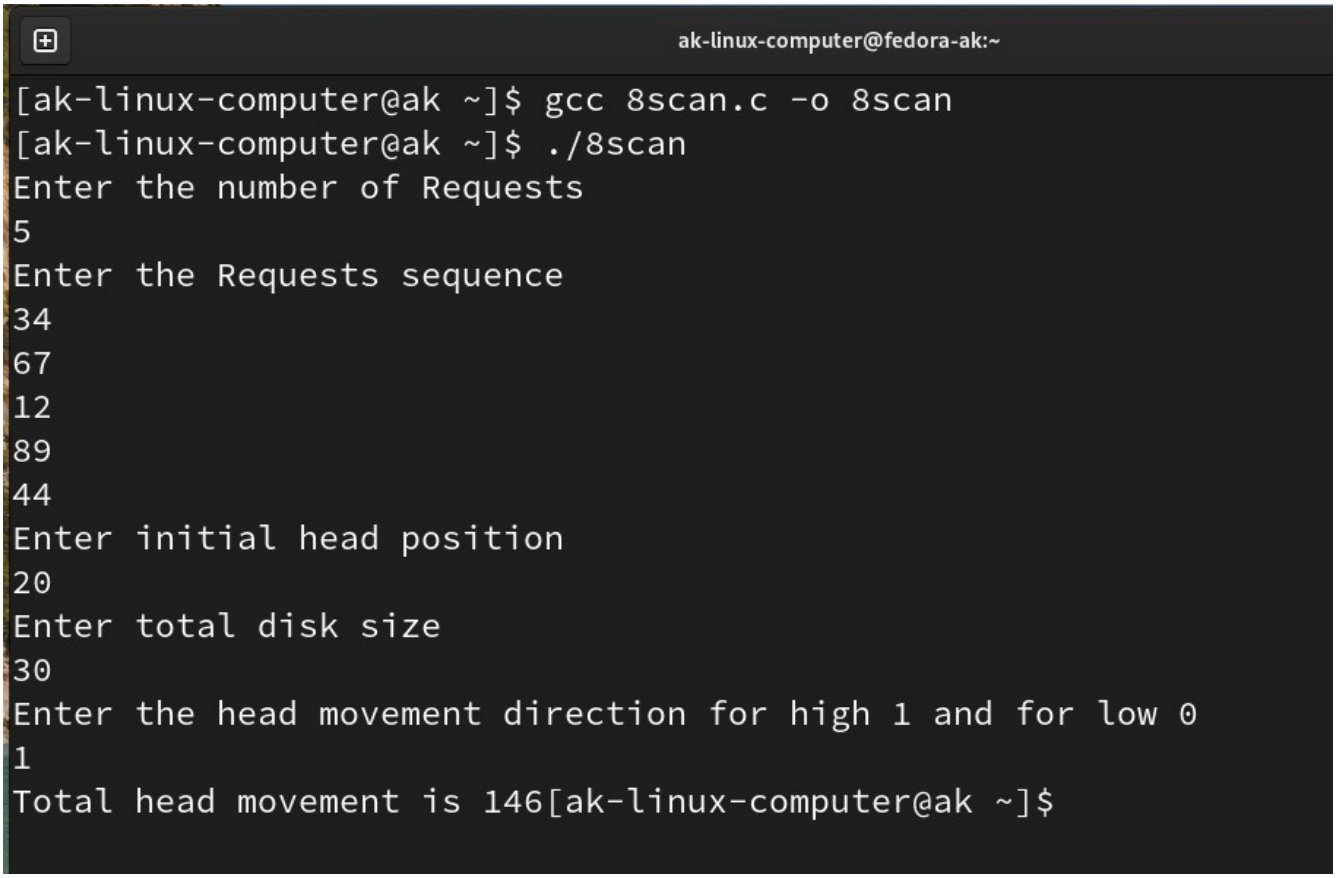

ak-linux-computer@fedora-ak:~
[ak-linux-computer@ak ~]$ gcc 8sstf.c -o 8sstf
[ak-linux-computer@ak ~]$ ./8sstf
Enter the number of Requests
5
Enter the Requests sequence
89
12
78
23
45
Enter initial head position
30
Total head movement is 95[ak-linux-computer@ak ~]$

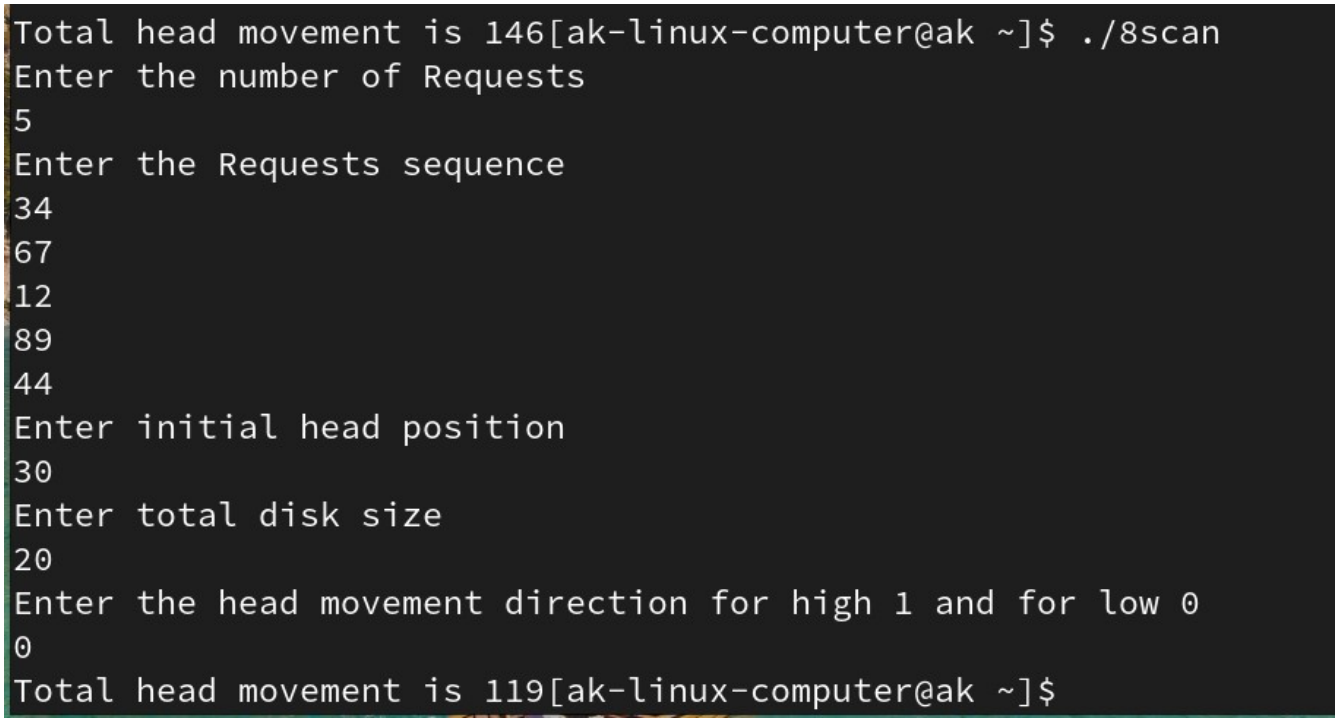
SCAN code:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
   int RQ[100], i, j, n, TotalHeadMoment = 0, initial, size, move;
   printf("Enter the number of Requests\n");
   scanf("%d", &n);
   printf("Enter the Requests sequence\n");
   for (i = 0; i < n; i++)
      scanf("%d", &RQ[i]);
   printf("Enter initial head position\n");
   scanf("%d", &initial);
   printf("Enter total disk size\n");
   scanf("%d", &size);
   printf("Enter the head movement direction for high 1 and for low 0\n");
   scanf("%d", &move);

   // logic for Scan disk scheduling

   /*logic for sort the request array */
   for (i = 0; i < n; i++)
   {
      for (j = 0; j < n - i - 1; j++)
      {
         if (RQ[j] > RQ[j + 1])
         {
            int temp;
            temp = RQ[j];
            RQ[j] = RQ[j + 1];
            RQ[j + 1] = temp;
         }
      }
   }

   int index;
   for (i = 0; i < n; i++)
   {
      if (initial < RQ[i])
      {
         index = i;
         break;
      }
   }

   // if movement is towards high value
   if (move == 1)
   {
```

```
        for (i = index; i < n; i++)
        {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }
        //  last movement for max size
        TotalHeadMoment = TotalHeadMoment + abs(size - RQ[i - 1] - 1);
        initial = size - 1;
        for (i = index - 1; i >= 0; i--)
        {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }
    }
    // if movement is towards low value
    else
    {
        for (i = index - 1; i >= 0; i--)
        {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }
        //  last movement for min size
        TotalHeadMoment = TotalHeadMoment + abs(RQ[i + 1] - 0);
        initial = 0;
        for (i = index; i < n; i++)
        {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }
    }

    printf("Total head movement is %d", TotalHeadMoment);
    return 0;
}
```

Output:


ak-linux-computer@fedora-ak:~
[ak-linux-computer@ak ~]$ gcc 8scan.c -o 8scan
[ak-linux-computer@ak ~]$ ./8scan
Enter the number of Requests
5
Enter the Requests sequence
34
67
12
89
44
Enter initial head position
20
Enter total disk size
30
Enter the head movement direction for high 1 and for low 0
1
Total head movement is 146[ak-linux-computer@ak ~]$

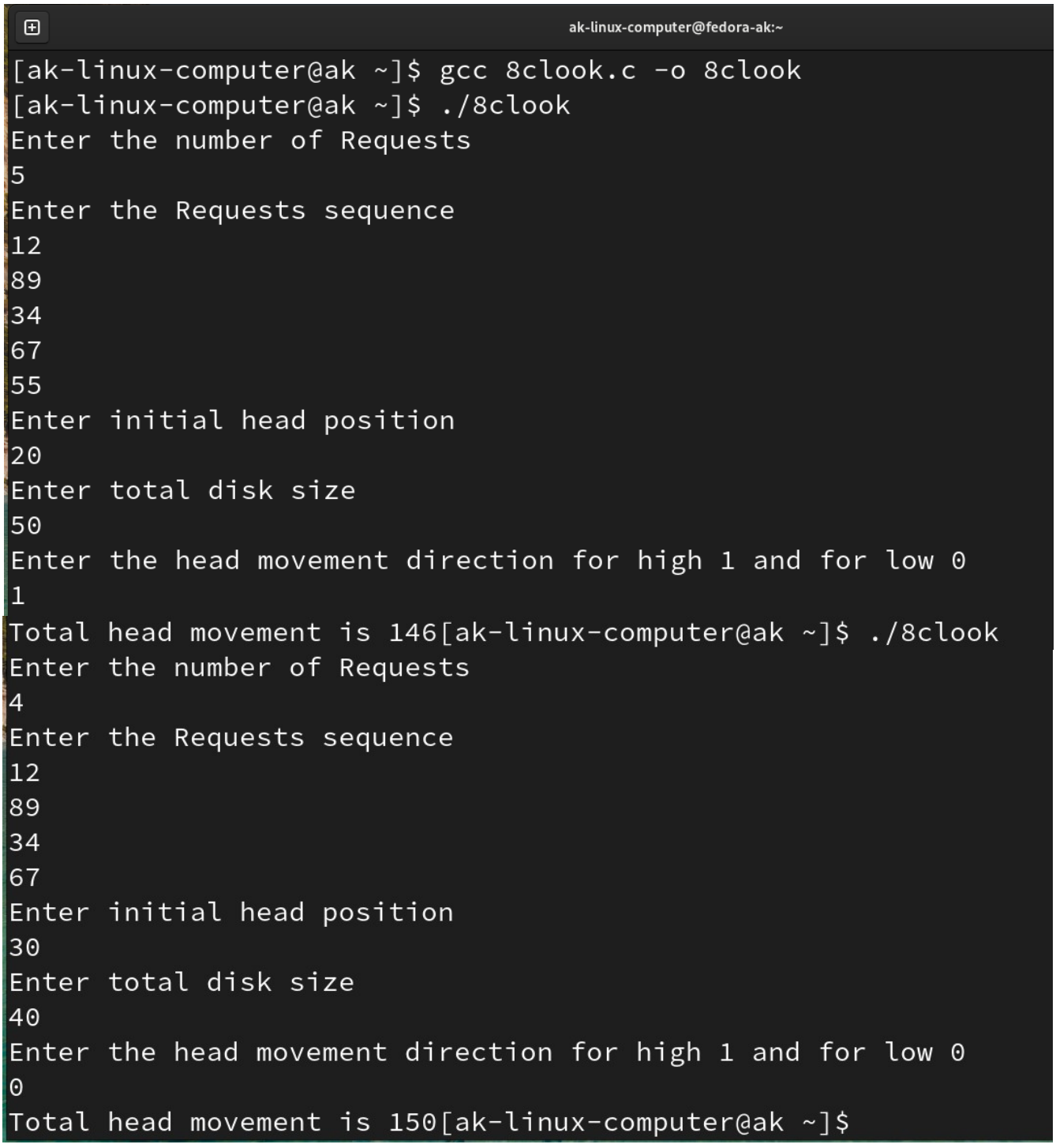

Total head movement is 146[ak-linux-computer@ak ~]$ ./8scan
Enter the number of Requests
5
Enter the Requests sequence
34
67
12
89
44
Enter initial head position
30
Enter total disk size
20
Enter the head movement direction for high 1 and for low 0
0
Total head movement is 119[ak-linux-computer@ak ~]$

C – LOOK code:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
   int RQ[100], i, j, n, TotalHeadMoment = 0, initial, size, move;
   printf("Enter the number of Requests\n");
   scanf("%d", &n);
   printf("Enter the Requests sequence\n");
   for (i = 0; i < n; i++)
      scanf("%d", &RQ[i]);
   printf("Enter initial head position\n");
   scanf("%d", &initial);
   printf("Enter total disk size\n");
   scanf("%d", &size);
   printf("Enter the head movement direction for high 1 and for low 0\n");
   scanf("%d", &move);

   // logic for C-look disk scheduling

   /*logic for sort the request array */
   for (i = 0; i < n; i++)
   {
      for (j = 0; j < n - i - 1; j++)
      {
         if (RQ[j] > RQ[j + 1])
         {
            int temp;
            temp = RQ[j];
            RQ[j] = RQ[j + 1];
            RQ[j + 1] = temp;
         }
      }
   }

   int index;
   for (i = 0; i < n; i++)
   {
      if (initial < RQ[i])
      {
         index = i;
         break;
      }
   }

   // if movement is towards high value
   if (move == 1)
   {
```

```
        for (i = index; i < n; i++)
        {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }

        for (i = 0; i < index; i++)
        {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }
    }
    // if movement is towards low value
    else
    {
        for (i = index - 1; i >= 0; i--)
        {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }

        for (i = n - 1; i >= index; i--)
        {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }
    }

    printf("Total head movement is %d", TotalHeadMoment);
    return 0;
}
```

Output:


ak-linux-computer@fedora-ak:~
[ak-linux-computer@ak ~]$ gcc 8clook.c -o 8clook
[ak-linux-computer@ak ~]$ ./8clook
Enter the number of Requests
5
Enter the Requests sequence
12
89
34
67
55
Enter initial head position
20
Enter total disk size
50
Enter the head movement direction for high 1 and for low 0
1
Total head movement is 146[ak-linux-computer@ak ~]$ ./8clook
Enter the number of Requests
4
Enter the Requests sequence
12
89
34
67
Enter initial head position
30
Enter total disk size
40
Enter the head movement direction for high 1 and for low 0
0
Total head movement is 150[ak-linux-computer@ak ~]$