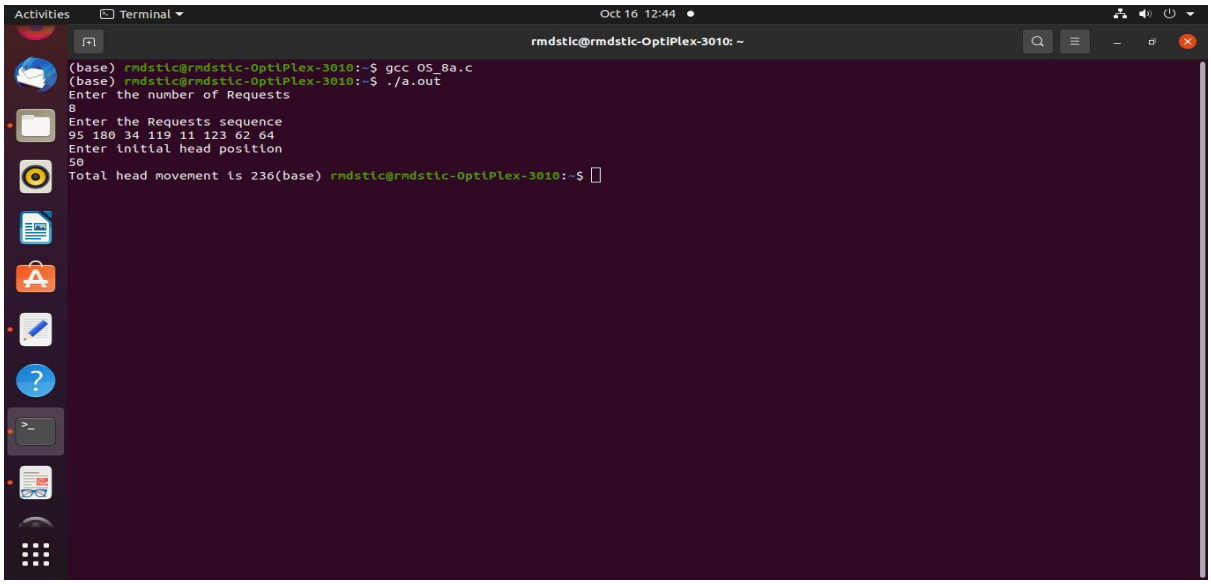
Assignment 8

Title: Implement the C program for Disk Scheduling Algorithms: SSTF, SCAN, C-Look considering the initial head position moving away from the spindle.

Code:- SSTF

```
#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);
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];
RQ[index] = 1000; count++;
}
printf("Total head movement is %d", TotalHeadMoment);
return 0;
}
```

Output:-

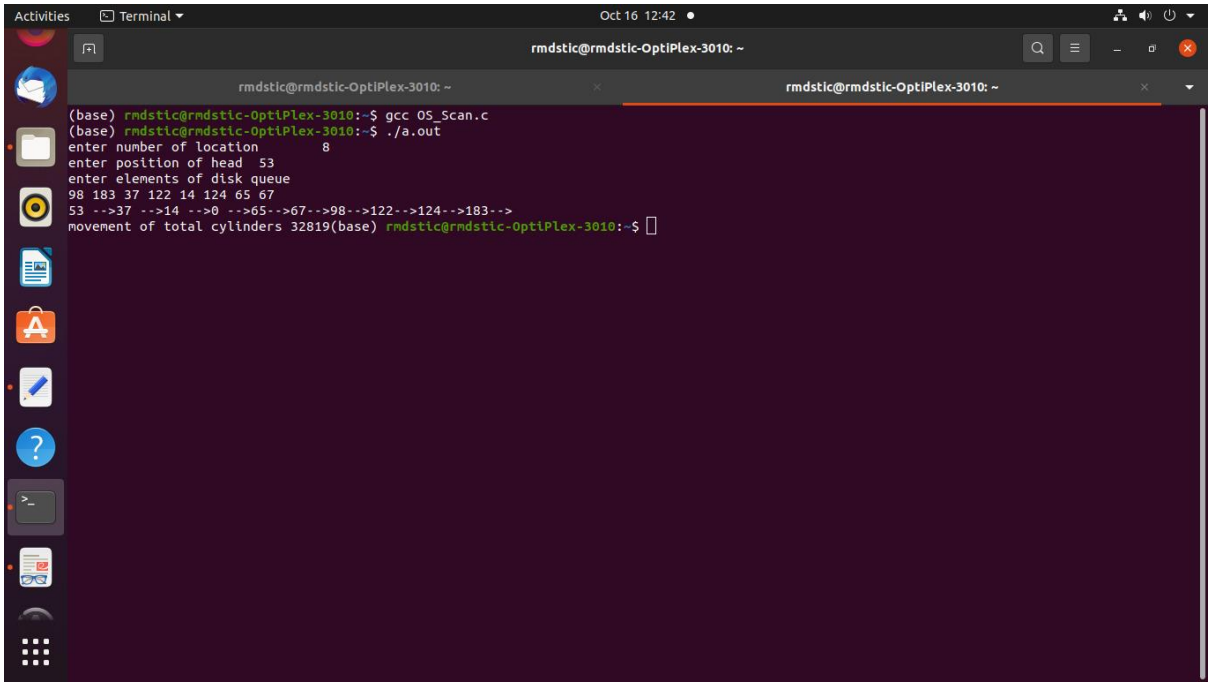


Code:-SCAN

```
#include<stdio.h>
int main()
{
int i, j, sum = 0, n; int d[20];
int disk;
 int temp, max;
int dloc;
printf("enter number of location\t");
scanf("%d", & n);
printf("enter position of head\t");
scanf("%d", & disk);
printf("enter elements of disk queue\n");
for (i = 0; i < n; i++)
{
scanf("%d", & d[i]);
}
d[n] = disk; n = n + 1;
for (i = 0; i < n; i++)
{
for (j = i; j < n; j++)
{ if (d[i] > d[j]) {
temp = d[i];
d[i] = d[j];
d[j] = temp;
}
}
}
max = d[n];
for (i = 0; i < n; i++)
          {
```

```
        if (disk == d[i]) {
                dloc = i;
        break;
        }
}
for (i = dloc; i >= 0; i--)
{
        printf("%d-->", d[i]);
}
        printf("0-->");
for (i = dloc + 1; i < n; i++)
{
        printf("%d-->", d[i]);
}
sum = disk + max;
printf("\nmovement of total cylinders %d", sum);
return 0;
}
```

Output:-



Code:- C-LOOK

```
#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);

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 (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];
}
}
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:-