# LAB 13

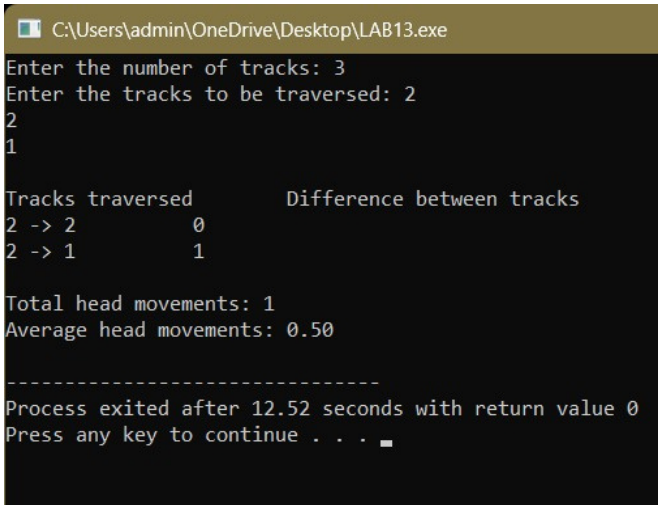
## CODE:

### A) FCFS

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


int main() {
    int t[20], tohm[20], n, i, tot = 0;
    float avhm;

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

    printf("Enter the tracks to be traversed: ");
    for (i = 0; i < n; i++) {
        scanf("%d", &t[i]);
    }

    // Calculate the head movements
    for (i = 0; i < n - 1; i++) {
        tohm[i] = abs(t[i + 1] - t[i]);
        tot += tohm[i];
    }

    avhm = (float)tot / (n - 1);

    printf("\nTracks traversed\tDifference between tracks\n");
    for (i = 0; i < n - 1; i++) {
```

```
        printf("%d -> %d\t\t%d\n", t[i], t[i + 1], tohm[i]);
    }

    printf("\nTotal head movements: %d", tot);
    printf("\nAverage head movements: %.2f\n", avhm);

    return 0;
}
```

OUTPUT:



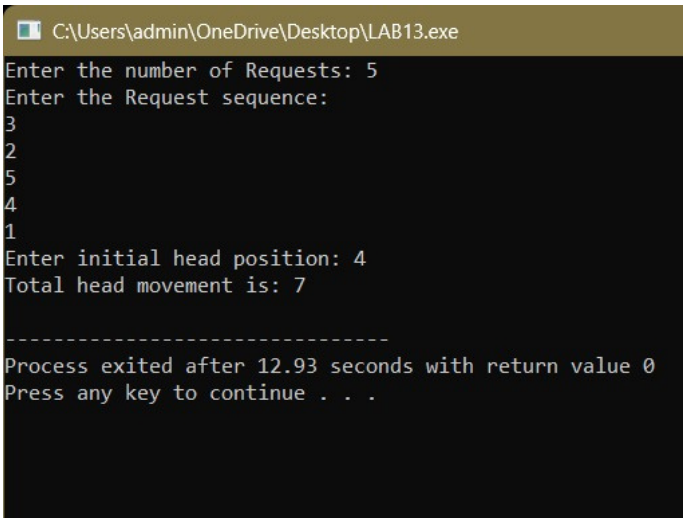

B) SSTF

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

int main() {
    int RQ[100], n, initial, TotalHeadMovement = 0, count = 0;
    int i, min, d, index, visited[100] = {0};

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

```
printf("Enter the Request sequence:\n");
for (i = 0; i < n; i++) {
   scanf("%d", &RQ[i]);
}

printf("Enter initial head position: ");
scanf("%d", &initial);

while (count < n) {
   min = 100000; // Large value
   index = -1;

   for (i = 0; i < n; i++) {
      if (!visited[i]) {
         d = abs(RQ[i] - initial);
         if (d < min) {
         min = d;
         index = i;
         }
      }
   }

   visited[index] = 1;
   TotalHeadMovement += min;
   initial = RQ[index];
   count++;
}
```

```
    printf("Total head movement is: %d\n", TotalHeadMovement);

    return 0;
}
```

OUTPUT:




C) SCAN

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

int main() {
    int t[100], n, head, i, j, temp;
    int totalMovement = 0;
    int direction;

    printf("Enter the number of tracks to be traversed: ");
    scanf("%d", &n);

    printf("Enter the position of the head: ");
```

```
scanf("%d", &head);

  t[0] = head;

  printf("Enter the track numbers:\n");
  for (i = 1; i <= n; i++) {
     scanf("%d", &t[i]);
  }

  n++; // include the head in the track list

  // Sorting the track array
  for (i = 0; i < n - 1; i++) {
     for (j = 0; j < n - i - 1; j++) {
        if (t[j] > t[j + 1]) {
           temp = t[j];
           t[j] = t[j + 1];
           t[j + 1] = temp;
        }
     }
  }

  // Ask direction: 0 for left, 1 for right
  printf("Enter head movement direction (0 for left, 1 for right): ");
  scanf("%d", &direction);

  // Find the index of the head
  int index;
```

```
 for (i = 0; i < n; i++) {
    if (t[i] == head) {
       index = i;
       break;
    }
}

printf("Order of servicing tracks:\n");

if (direction == 1) {
    // Move right
    for (i = index; i < n; i++) {
       printf("%d ", t[i]);
    }
    for (i = index - 1; i >= 0; i--) {
       printf("%d ", t[i]);
    }
} else {
    // Move left
    for (i = index; i >= 0; i--) {
       printf("%d ", t[i]);
    }
    for (i = index + 1; i < n; i++) {
       printf("%d ", t[i]);
    }
}

// Calculate total head movement
```

```
    for (i = 0; i < n - 1; i++) {
        totalMovement += abs(t[i + 1] - t[i]);
    }

    printf("\nTotal head movement: %d\n", totalMovement);

    return 0;
}
```

OUTPUT: