b) SCAN

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

void scanDiskScheduling(int requests[], int n, int start, int diskSize, int direction) {
  int totalHeadMovement = 0;

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

  printf("Disk Scheduling (SCAN):\n");

  int left[n], right[n];
  int leftCount = 0, rightCount = 0;

  // Divide the requests into left and right of the starting position
  for (int i = 0; i < n; i++) {
    if (requests[i] < start) {
      left[leftCount++] = requests[i];
    } else {
      right[rightCount++] = requests[i];
    }
  }

  // Process the left side (if moving left)
  if (direction == 0) {
    for (int i = leftCount - 1; i >= 0; i--) {
      totalHeadMovement += abs(start - left[i]);
      printf("Move from %d to %d\n", start, left[i]);
      start = left[i];
    }
    totalHeadMovement += abs(start - 0); // Move to the beginning
    start = 0;
  }

  // Process the right side (if moving right)
  if (direction == 1) {
    for (int i = 0; i < rightCount; i++) {
      totalHeadMovement += abs(start - right[i]);
```

```
            printf("Move from %d to %d\n", start, right[i]);
            start = right[i];
        }
        totalHeadMovement += abs(start - (diskSize - 1)); // Move to the end
        start = diskSize - 1;
    }

    printf("\nTotal Head Movement = %d\n", totalHeadMovement);
}

int main() {
    int n, start, direction, diskSize;
    printf("Enter the number of disk requests: ");
    scanf("%d", &n);

    int requests[n];

    printf("Enter the disk requests:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &requests[i]);
    }

    printf("Enter the starting position of the disk head: ");
    scanf("%d", &start);

    printf("Enter the disk size (total number of tracks): ");
    scanf("%d", &diskSize);

    printf("Enter the direction of the head (0 for left, 1 for right): ");
    scanf("%d", &direction);

    scanDiskScheduling(requests, n, start, diskSize, direction);

    return 0;
}
```

Output

```
Enter the number of disk requests: 2
Enter the disk requests:
2
3
Enter the starting position of the disk head: 5
Enter the disk size (total number of tracks): 4
Enter the direction of the head (0 for left, 1 for right): 0
Disk Scheduling (SCAN):
Move from 5 to 3
Move from 3 to 2

Total Head Movement = 5


=== Code Execution Successful ===
```