1. Develop a C program to simulate C-SCAN disk scheduling algorithm.

#include <stdio.h>

#include <stdlib.h>

void sort(int arr[], int n) {

for(int i=0; i<n-1; i++)

for(int j=0; j<n-i-1; j++)

if(arr[j] > arr[j+1]) {

int t = arr[j];

arr[j] = arr[j+1];

arr[j+1] = t;

}

}

int main() {

int req[100], n, head, disk\_size, dir;

int left[100], right[100];

int l = 0, r = 0, total\_seek = 0, i, cur;

printf("Enter number of disk I/O requests: ");

scanf("%d", &n);

printf("Enter request sequence:\n");

for(i = 0; i < n; i++)

scanf("%d", &req[i]);

printf("Enter initial head position: ");

scanf("%d", &head);

printf("Enter total disk size: ");

scanf("%d", &disk\_size);

printf("Enter direction (0 = left, 1 = right): ");

scanf("%d", &dir);

for(i = 0; i < n; i++) {

if(req[i] < head)

left[l++] = req[i];

else

right[r++] = req[i];

}

sort(left, l);

sort(right, r);

cur = head;

printf("\nSeek sequence:\n");

if(dir == 1) {

for(i = 0; i < r; i++) {

printf("%d -> ", cur);

total\_seek += abs(right[i] - cur);

cur = right[i];

}

printf("%d -> ", cur);

total\_seek += abs((disk\_size - 1) - cur);

cur = 0;

total\_seek += disk\_size - 1;

for(i = 0; i < l; i++) {

printf("%d -> ", cur);

total\_seek += abs(left[i] - cur);

cur = left[i];

}

} else {

for(i = l - 1; i >= 0; i--) {

printf("%d -> ", cur);

total\_seek += abs(left[i] - cur);

cur = left[i];

}

printf("%d -> ", cur);

total\_seek += cur;

cur = disk\_size - 1;

total\_seek += cur;

for(i = r - 1; i >= 0; i--) {

printf("%d -> ", cur);

total\_seek += abs(right[i] - cur);

cur = right[i];

}

}

printf("END\n");

printf("Total Seek Time = %d\n", total\_seek);

printf("Average Seek Time = %.2f\n", (float)total\_seek / n);

return 0;

}