. Write a program to simulate SCAN disk scheduling. Calculate total seek time. Print accepted input and output in tabular format

#include <stdio.h>

#include <stdlib.h>

void SCAN(int head, int requests[], int n, int disk\_size, int direction) {

int total\_seek\_time = 0;

int temp, i, j;

// Sort the request array

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

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

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

temp = requests[j];

requests[j] = requests[j + 1];

requests[j + 1] = temp;

}

}

}

// Find the position of the head in the sorted request array

int pos = 0;

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

if (head < requests[i]) {

pos = i;

break;

}

}

printf("\nSCAN Disk Scheduling\n");

printf("Initial Head Position: %d\n", head);

printf("Direction: %s\n", direction == 1 ? "Right" : "Left");

printf("\nRequest\tCurrent Head Position\tSeek Time\n");

printf("-------\t----------------------\t---------\n");

// Serve requests based on the direction

if (direction == 1) { // Moving towards the right

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

int seek\_time = abs(requests[i] - head);

total\_seek\_time += seek\_time;

printf("%d\t\t%d\t\t\t%d\n", requests[i], head, seek\_time);

head = requests[i];

}

// Go to the end of the disk if necessary

if (head != disk\_size - 1) {

int seek\_time = abs((disk\_size - 1) - head);

total\_seek\_time += seek\_time;

printf("%d\t\t%d\t\t\t%d\n", disk\_size - 1, head, seek\_time);

head = disk\_size - 1;

}

// Reverse direction and continue serving requests to the left

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

int seek\_time = abs(requests[i] - head);

total\_seek\_time += seek\_time;

printf("%d\t\t%d\t\t\t%d\n", requests[i], head, seek\_time);

head = requests[i];

}

} else { // Moving towards the left

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

int seek\_time = abs(requests[i] - head);

total\_seek\_time += seek\_time;

printf("%d\t\t%d\t\t\t%d\n", requests[i], head, seek\_time);

head = requests[i];

}

// Go to the start of the disk if necessary

if (head != 0) {

int seek\_time = abs(head - 0);

total\_seek\_time += seek\_time;

printf("%d\t\t%d\t\t\t%d\n", 0, head, seek\_time);

head = 0;

}

// Reverse direction and continue serving requests to the right

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

int seek\_time = abs(requests[i] - head);

total\_seek\_time += seek\_time;

printf("%d\t\t%d\t\t\t%d\n", requests[i], head, seek\_time);

head = requests[i];

}

}

printf("\nTotal Seek Time: %d\n", total\_seek\_time);

}

int main() {

int n, head, disk\_size, direction;

// Input number of requests

printf("Enter the number of disk requests: ");

scanf("%d", &n);

int requests[n];

// Input the disk requests

printf("Enter the disk requests:\n");

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

printf("Request %d: ", i + 1);

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

}

// Input initial head position

printf("Enter the initial head position: ");

scanf("%d", &head);

// Input the size of the disk

printf("Enter the disk size: ");

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

// Input direction (1 for right, 0 for left)

printf("Enter the initial direction (1 for right, 0 for left): ");

scanf("%d", &direction);

// Call the SCAN function to calculate seek time

SCAN(head, requests, n, disk\_size, direction);

return 0;

}