#include <stdio.h>

#include <stdlib.h>

#include <math.h>

typedef struct Process {

int id;

int arrivalTime;

int burstTime;

} Process;

typedef struct DiskRequest {

int trackNumber;

int served;

} DiskRequest;

void sstf(DiskRequest requests[], int numRequests, int initialPosition) {

int currentHeadPosition = initialPosition;

int totalMovement = 0;

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

printf("Sequence: %d ", currentHeadPosition);

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

int minDistance = \_\_INT\_MAX\_\_;

int nextRequestIndex = -1;

for (int j = 0; j < numRequests; j++) {

if (!requests[j].served) {

int distance = abs(currentHeadPosition - requests[j].trackNumber);

if (distance < minDistance) {

minDistance = distance;

nextRequestIndex = j;

}

}

}

requests[nextRequestIndex].served = 1;

currentHeadPosition = requests[nextRequestIndex].trackNumber;

totalMovement += minDistance;

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

}

printf("\nTotal Head Movement: %d\n", totalMovement);

}

int main() {

DiskRequest requests[] = {{452, 0}, {789, 0}, {235, 0}, {4123, 0}, {1250, 0}, {45, 0}, {85, 0}, {678, 0}, {1509, 0}, {1470, 0}, {45, 0}};

int numRequests = sizeof(requests) / sizeof(requests[0]);

int initialPosition = 98;

sstf(requests, numRequests, initialPosition);

return 0;

}