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

#include <stdio.h>

#include <stdlib.h>

void SSTF(int head, int requests[], int n) {

int total\_seek\_time = 0;

int completed[n]; // Array to track completed requests

for (int i = 0; i < n; i++) completed[i] = 0; // Mark all requests as incomplete initially

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

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

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

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

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

int closest = -1;

int min\_seek\_time = 9999;

// Find the closest request

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

if (!completed[j]) { // Only consider uncompleted requests

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

if (seek\_time < min\_seek\_time) {

min\_seek\_time = seek\_time;

closest = j;

}

}

}

// Move the head to the closest request

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

total\_seek\_time += seek\_time;

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

head = requests[closest];

completed[closest] = 1; // Mark this request as completed

}

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

}

int main() {

int n, head;

// 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);

// Call the SSTF function to calculate seek time

SSTF(head, requests, n);

return 0;

}