**EXPERIMENT 39**

**C-SCAN DISK SCHEDULING**

#include <stdio.h>

#include <stdlib.h>

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

// Simple Bubble Sort

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

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

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

int temp = arr[j];

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

arr[j+1] = temp;

}

}

int main() {

int size, i, j;

int disk\_size, head, total\_seek = 0;

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

scanf("%d", &size);

int requests[size];

printf("Enter the disk request sequence: ");

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

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

printf("Enter initial head position: ");

scanf("%d", &head);

printf("Enter disk size (last cylinder number): ");

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

// Sorting the requests

sort(requests, size);

// Splitting the request list into two parts:

// one with requests >= head, and another < head

int index;

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

if (requests[i] >= head) {

index = i;

break;

}

}

printf("\nC-SCAN Disk Scheduling Order:\n");

// Process requests from head to end

for (i = index; i < size; i++) {

printf("%d -> ", requests[i]);

total\_seek += abs(head - requests[i]);

head = requests[i];

}

// Go to end of disk

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

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

head = 0; // simulate moving from end to beginning

total\_seek += disk\_size - 1;

} else {

head = 0;

total\_seek += disk\_size - 1;

}

// Continue from beginning to the rest

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

printf("%d -> ", requests[i]);

total\_seek += abs(head - requests[i]);

head = requests[i];

}

printf("END\n");

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

return 0;

}