**Assignment Number 08**

**Name:** Mihir Unmesh Patil

**Roll NO**: TYCOC213

**Batch:** C/C-3

**CODE:**

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

int abs\_diff(int a, int b) {

    return abs(a - b);

}

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

    int total\_movement = 0;

    int current\_position = head;

    printf("FCFS Seek Sequence: %d ", head);

    for (int i = 0; i < n; i++) {

        total\_movement += abs\_diff(requests[i], current\_position);

        current\_position = requests[i];

        printf("%d ", requests[i]);

    }

    printf("\nFCFS Total Head Movement: %d\n", total\_movement);

}

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

    int total\_movement = 0;

    int current\_position = head;

    bool visited[n];

    for (int i = 0; i < n; i++) {

        visited[i] = false;

    }

    printf("SSTF Seek Sequence: %d ", head);

    for (int i = 0; i < n; i++) {

        int min\_diff = 99999;

        int next\_request = -1;

        for (int j = 0; j < n; j++) {

            if (!visited[j]) {

                int diff = abs\_diff(requests[j], current\_position);

                if (diff < min\_diff) {

                    min\_diff = diff;

                    next\_request = j;

                }

            }

        }

        if (next\_request != -1) {

            total\_movement += min\_diff;

            current\_position = requests[next\_request];

            visited[next\_request] = true;

            printf("%d ", current\_position);

        }

    }

    printf("\nSSTF Total Head Movement: %d\n", total\_movement);

}

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

    int total\_movement = 0;

    int current\_position = head;

    int seek\_sequence[n + 2];

    int seek\_count = 0;

    int left[n], right[n];

    int left\_count = 0, right\_count = 0;

    for (int i = 0; i < n; i++) {

        if (requests[i] < head) {

            left[left\_count++] = requests[i];

        } else {

            right[right\_count++] = requests[i];

        }

    }

    for (int i = 0; i < left\_count - 1; i++) {

        for (int j = 0; j < left\_count - i - 1; j++) {

            if (left[j] < left[j + 1]) {

                int temp = left[j];

                left[j] = left[j + 1];

                left[j + 1] = temp;

            }

        }

    }

    for (int i = 0; i < right\_count - 1; i++) {

        for (int j = 0; j < right\_count - i - 1; j++) {

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

                int temp = right[j];

                right[j] = right[j + 1];

                right[j + 1] = temp;

            }

        }

    }

    printf("SCAN Seek Sequence: %d ", head);

    if (direction == 'l') {

        for (int i = 0; i < left\_count; i++) {

            total\_movement += abs\_diff(current\_position, left[i]);

            current\_position = left[i];

            printf("%d ", current\_position);

        }

        total\_movement += abs\_diff(current\_position, 0);

        current\_position = 0;

        printf("%d ", current\_position);

        for (int i = 0; i < right\_count; i++) {

            total\_movement += abs\_diff(current\_position, right[i]);

            current\_position = right[i];

            printf("%d ", current\_position);

        }

    }

    else {

        for (int i = 0; i < right\_count; i++) {

            total\_movement += abs\_diff(current\_position, right[i]);

            current\_position = right[i];

            printf("%d ", current\_position);

        }

        total\_movement += abs\_diff(current\_position, disk\_size - 1);

        current\_position = disk\_size - 1;

        printf("%d ", current\_position);

        for (int i = 0; i < left\_count; i++) {

            total\_movement += abs\_diff(current\_position, left[i]);

            current\_position = left[i];

            printf("%d ", current\_position);

        }

    }

    printf("\nSCAN Total Head Movement: %d\n", total\_movement);

}

void c\_scan(int requests[], int n, int head, int disk\_size, char direction) {

    int total\_movement = 0;

    int current\_position = head;

    int seek\_sequence[n + 2];

    int seek\_count = 0;

    int left[n], right[n];

    int left\_count = 0, right\_count = 0;

    for (int i = 0; i < n; i++) {

        if (requests[i] < head) {

            left[left\_count++] = requests[i];

        } else {

            right[right\_count++] = requests[i];

        }

    }

    for (int i = 0; i < left\_count - 1; i++) {

        for (int j = 0; j < left\_count - i - 1; j++) {

            if (left[j] < left[j + 1]) {

                int temp = left[j];

                left[j] = left[j + 1];

                left[j + 1] = temp;

            }

        }

    }

    for (int i = 0; i < right\_count - 1; i++) {

        for (int j = 0; j < right\_count - i - 1; j++) {

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

                int temp = right[j];

                right[j] = right[j + 1];

                right[j + 1] = temp;

            }

        }

    }

    printf("C-SCAN Seek Sequence: %d ", head);

    if (direction == 'l') {

        for (int i = 0; i < left\_count; i++) {

            total\_movement += abs\_diff(current\_position, left[i]);

            current\_position = left[i];

            printf("%d ", current\_position);

        }

        total\_movement += abs\_diff(current\_position, 0);

        current\_position = 0;

        printf("%d ", current\_position);

        total\_movement += abs\_diff(current\_position, disk\_size-1);

        current\_position = disk\_size-1;

        printf("%d ", current\_position);

        current\_position = head;

        for (int i = 0; i < right\_count; i++) {

            total\_movement += abs\_diff(current\_position, right[i]);

            current\_position = right[i];

            printf("%d ", current\_position);

        }

    }

    else {

        for (int i = 0; i < right\_count; i++) {

            total\_movement += abs\_diff(current\_position, right[i]);

            current\_position = right[i];

            printf("%d ", current\_position);

        }

        total\_movement += abs\_diff(current\_position, disk\_size - 1);

        current\_position = disk\_size - 1;

        printf("%d ", current\_position);

        total\_movement += abs\_diff(current\_position, 0);

        current\_position = 0;

        printf("%d ", current\_position);

        current\_position = head;

        for (int i = 0; i < left\_count; i++) {

            total\_movement += abs\_diff(current\_position, left[i]);

            current\_position = left[i];

            printf("%d ", current\_position);

        }

    }

    printf("\nC-SCAN Total Head Movement: %d\n", total\_movement);

}

int main() {

    int n, head, disk\_size;

    char direction;

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

    scanf("%d", &n);

    int requests[n];

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

    for (int i = 0; i < n; i++) {

        scanf("%d", &requests[i]);

    }

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

    scanf("%d", &head);

    printf("Enter the disk size: ");

    scanf("%d", &disk\_size);

    printf("Enter the direction for SCAN and C-SCAN (l for left, r for right): ");

    scanf(" %c", &direction);

    printf("Disk Size: %d\n",disk\_size);

    fcfs(requests, n, head);

    sstf(requests, n, head);

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

    c\_scan(requests, n, head, disk\_size, direction);

    return 0;

}

OUTPUT:

