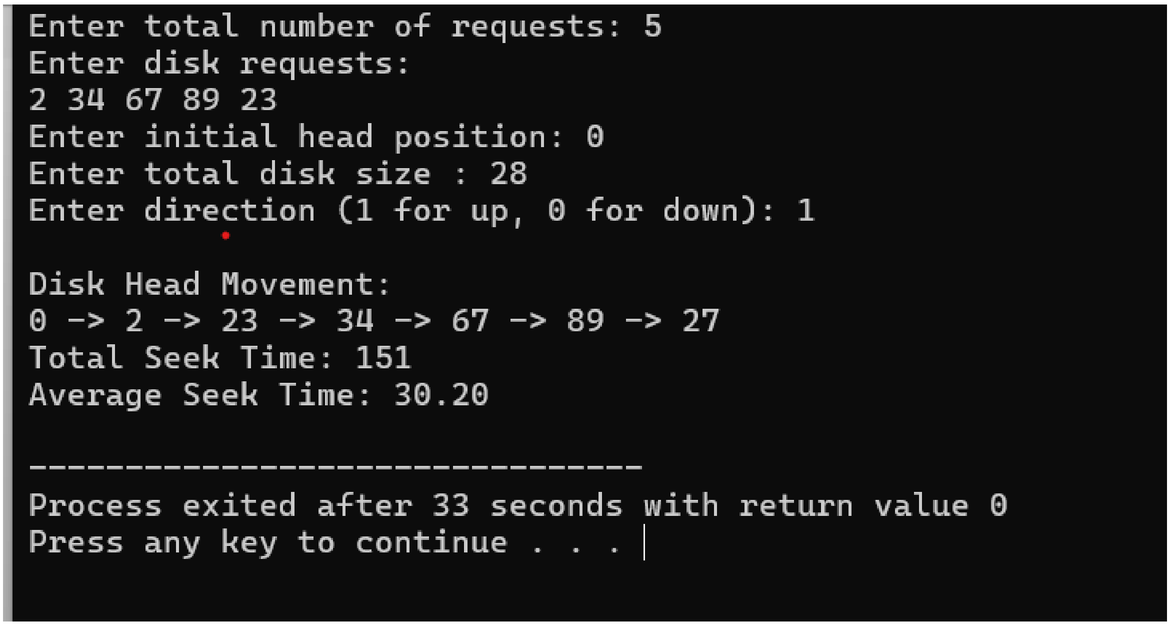
EXP 38: Design a C program to simulate SCAN disk scheduling algorithm.

```
#include <stdio.h>
#include <stdlib.h>

void scan(int requests[], int n, int head, int direction, int disk_size) {
    int seekCount = 0;
    int distance;
    int i, j;

    // Sort the request queue
    for (i = 0; i < n - 1; i++)
        for (j = i + 1; j < n; j++)
            if (requests[i] > requests[j]) {
                int temp = requests[i];
                requests[i] = requests[j];
                requests[j] = temp;
            }

    // Find the index where head is greater than or equal to request
    int idx;
    for (i = 0; i < n; i++) {
        if (requests[i] >= head) {
            idx = i;
            break;
        }
    }

    printf("\nDisk Head Movement:\n%d", head);
```

```
// If moving towards higher cylinder numbers
if (direction == 1) {
    // Move right
    for (i = idx; i < n; i++) {
        seekCount += abs(head - requests[i]);
        head = requests[i];
        printf(" -> %d", head);
    }

    // Go to end of disk
    if (head != disk_size - 1) {
        seekCount += abs(head - (disk_size - 1));
        head = disk_size - 1;
        printf(" -> %d", head);
    }

    // Reverse and move left
    for (i = idx - 1; i >= 0; i--) {
        seekCount += abs(head - requests[i]);
        head = requests[i];
        printf(" -> %d", head);
    }
} else { // direction == 0, moving left
    // Move left
    for (i = idx - 1; i >= 0; i--) {
        seekCount += abs(head - requests[i]);
        head = requests[i];
        printf(" -> %d", head);
        }

        // Go to start of disk
        if (head != 0) {
            seekCount += abs(head - 0);
            head = 0;
            printf(" -> %d", head);
        }

        // Reverse and move right
        for (i = idx; i < n; i++) {
            seekCount += abs(head - requests[i]);
            head = requests[i];
            printf(" -> %d", head);
        }
    }

    printf("\nTotal Seek Time: %d\n", seekCount);
    printf("Average Seek Time: %.2f\n", (float)seekCount / n);
}

int main() {
    int n, head, direction, disk_size;

    printf("Enter total number of requests: ");
    scanf("%d", &n);

    int requests[n];
    printf("Enter disk requests:\n");
    for (int i = 0; i < n; i++)
        scanf("%d", &requests[i]);

    printf("Enter initial head position: ");
    scanf("%d", &head);

    printf("Enter total disk size : ");
    scanf("%d", &disk_size);

    printf("Enter direction (1 for up, 0 for down): ");
    scanf("%d", &direction);

    scan(requests, n, head, direction, disk_size);

    return 0;
}
```

Sample Output



```
Enter total number of requests: 5
Enter disk requests:
2 34 67 89 23
Enter initial head position: 0
Enter total disk size : 28
Enter direction (1 for up, 0 for down): 1

Disk Head Movement:
0 -> 2 -> 23 -> 34 -> 67 -> 89 -> 27
Total Seek Time: 151
Average Seek Time: 30.20

--------------------------------
Process exited after 33 seconds with return value 0
Press any key to continue . . .
```