EXP 39: Develop a C program to simulate C-SCAN disk scheduling algorithm.

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

void cscan(int request[], int n, int head, int disk_size) {
   int seek_count = 0;
   int distance;
   int i, j;

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

   // Find the index where requests[i] >= head
   int idx;
   for (i = 0; i < n; i++) {
      if (request[i] >= head) {
         idx = i;
         break;
      }
```

```
}

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

// Move to the right (higher cylinders)
for (i = idx; i < n; i++) {
    distance = abs(head - request[i]);
    seek_count += distance;
    head = request[i];
    printf(" -> %d", head);
}

// Jump to the start (0)
if (head != disk_size - 1) {
    seek_count += abs((disk_size - 1) - head);
    head = 0;
    seek_count += (disk_size - 1); // From end to 0
    printf(" -> %d", disk_size - 1);
    printf(" -> 0");
}

// Continue servicing from the beginning
for (i = 0; i < idx; i++) {
    distance = abs(head - request[i]);
    seek_count += distance;
    head = request[i];
    printf(" -> %d", head);
}
```

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

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

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

    int request[n];
    printf("Enter the disk request queue: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &request[i]);
    }

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

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

    cscan(request, n, head, disk_size);

    return 0;
}
```

Sample Output

```
Enter number of disk requests: 5
Enter the disk request queue: 34 56 78 98 32
Enter initial head position: 30
Enter total disk size : 20

Disk Head Movement:
30 -> 32 -> 34 -> 56 -> 78 -> 98 -> 19 -> 0
Total Seek Time: 166
Average Seek Time: 33.20

--------------------------------
Process exited after 30.38 seconds with return value 0
Press any key to continue . . .
```