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

#define MAX_SIZE 50

// Global array and size
int arr[MAX_SIZE];
int size;

// Function to perform bubble sort in ascending order
void* sort_ascending(void* arg) {
    for (int i = 0; i < size - 1; i++) {
        for (int j = 0; j < size - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }

    printf("Ascending Order: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
```

```
    return NULL;
}

// Function to perform bubble sort in descending order
void* sort_descending(void* arg) {
    for (int i = 0; i < size - 1; i++) {
        for (int j = 0; j < size - i - 1; j++) {
            if (arr[j] < arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }

    printf("Descending Order: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
    return NULL;
}

int main() {
    pthread_t thread1, thread2;

    printf("Enter the size of the array: ");
    scanf("%d", &size);
```

```
    if (size <= 0 || size > MAX_SIZE) {
        printf("Invalid size. Please enter a size between 1 and 50.\n");
        return 1;
    }

    printf("Enter array elements: ");
    for (int i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }

    // Create two threads: one for ascending sort, one for descending sort
    if (pthread_create(&thread1, NULL, sort_ascending, NULL)) {
        printf("Error creating thread 1\n");
        return 1;
    }

    if (pthread_create(&thread2, NULL, sort_descending, NULL)) {
        printf("Error creating thread 2\n");
        return 1;
    }

    // Wait for both threads to finish
    pthread_join(thread1, NULL);
    pthread_join(thread2, NULL);

    return 0;
}
```