Median of Two Sorted Arrays

Program:

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

int max(int a, int b) {
    return a > b ? a : b;
}

int min(int a, int b) {
    return a < b ? a : b;
}

double findMedianSortedArrays(int* nums1, int m, int* nums2, int n) {
    if (m > n) { // Ensure m <= n
        return findMedianSortedArrays(nums2, n, nums1, m);
    }

    int imin = 0, imax = m, halfLen = (m + n + 1) / 2;
    while (imin <= imax) {
        int i = (imin + imax) / 2;
        int j = halfLen - i;
        if (i < m && nums1[i] < nums2[j - 1]) {
            imin = i + 1; // i is too small
        } else if (i > 0 && nums1[i - 1] > nums2[j]) {
            imax = i - 1; // i is too big
        } else { // i is perfect
            int maxOfLeft;
            if (i == 0) { maxOfLeft = nums2[j - 1]; }
```

```
            else if (j == 0) { maxOfLeft = nums1[i - 1]; }
            else { maxOfLeft = max(nums1[i - 1], nums2[j - 1]); }

            if ((m + n) % 2 == 1) {
                return maxOfLeft; // Odd case
            }

            int minOfRight;
            if (i == m) { minOfRight = nums2[j]; }
            else if (j == n) { minOfRight = nums1[i]; }
            else { minOfRight = min(nums1[i], nums2[j]); }

            return (maxOfLeft + minOfRight) / 2.0; // Even case
        }
    }

    return 0.0; // Should not reach here
}

int main() {
    int nums1[] = {1, 3};
    int nums2[] = {2};
    int m = sizeof(nums1) / sizeof(nums1[0]);
    int n = sizeof(nums2) / sizeof(nums2[0]);

    double median = findMedianSortedArrays(nums1, m, nums2, n);
    printf("The median is: %f\n", median);

    return 0;
```

}