1. Find median of two sorted arrays

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

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

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

int median(int *arr, int size)
{
        return (size % 2 == 0)? (arr[size/2] + arr[size/2 - 1]) / 2 : arr[size/2];
}

int getMedianOfTwoArrays(int *arr1, int *arr2, int size)
{
        if(size == 1)
                return (arr1[0] + arr2[0]) / 2;
        if(size == 2)
                return (max(arr1[0], arr2[0]) + min(arr1[1], arr2[1])) / 2;

        //find median of two arrays
        int median1 = median(arr1, size);
        int median2 = median(arr2, size);

        if( median1 == median2 )
                return median1;

        if( median1 < median2)
                return (size % 2 == 0) ? getMedianOfTwoArrays(arr1 + size/2 - 1, arr2, size - size/2
+ 1) :
        getMedianOfTwoArrays(arr1 + size/2, arr2, size - size/2);

        return (size % 2 == 0) ? getMedianOfTwoArrays(arr2 + size/2 - 1, arr1, size - size/2 + 1):
        getMedianOfTwoArrays(arr2 + size/2, arr1, size - size/2);
}

int main()
{
        int *arr1, *arr2, size1, size2, index;
        printf("Enter number of elements in array 1\n");
        scanf("%d", &size1);
        printf("Enter number of elements in array 2\n");
        scanf("%d", &size2);
        //allocate memory for both arrays
	arr1 = (int *)malloc(size1 * sizeof(int));
	arr2 = (int *)malloc(size2 * sizeof(int));

	printf("Enter elements in array 1");
	for(index = 0; index < size1; index++)
		scanf("%d", &arr1[index]);
	printf("Enter elements in array 2");
	for(index = 0; index < size2; index++)
		scanf("%d", &arr2[index]);
	if (size1 == size2)
		printf("Median of Two sorted arrays is = %d",
			getMedianOfTwoArrays(arr1, arr2, size1));
	else
		printf("Given sizes are different");
	return 0;
}
```

Time Complexity : O(logn)
Space Complexity: O(1ogn)