6. Array Programming

Ronald (13 section)

Write a static method called **sumArrays** that sums the elements from a pair of arrays. Your method should take two parameters:

- `int[] nums1` - the first array of integers to consider
- `int[] nums2` - the second array of integers to consider

imp.

Your method should *return* a new array where each element of the new array contains the sum of the two integers at that index in the parameter arrays. If one array is longer than the other, the result array should contain the extra numbers from the longer array.

Edge.

For example, suppose the following code was executed:

```
int[] nums1 = {1, 2, 3, 4, 5};
int[] nums2 = {10, 20, 30, 40, 50};
int[] result = sumArrays(nums1, nums2);
```

// loop:

0 1 2 3 4

After this code is executed, the array result would contain the following values:

```
[11, 22, 33, 44, 55]
```

Note that the sum of the integers at each index was put into the result array (for instance 1 plus 10 is 11 and 30 plus 3 is 33).

As another example, suppose the following code was executed:

```
int[] nums1 = {1, 2, 3, 4, 5, 6, 7};
int[] nums2 = {10, 20, 30, 40, 50};
int[] result = sumArrays(nums1, nums2);
```

??

In this case, after this code is executed, `result` would contain the following values:

```
[11, 22, 33, 44, 55, 6, 7]
```

Notice that the extra elements 6 and 7 from the longer parameter array were also included in the result, but without being summed with anything.

You may assume that both parameter arrays contain at least one element.

Write your solution in the box on the next page.

Write your solution to problem #6 here:

```
public static int[] SumArrays(int[] nums1,
int[] nums2){
    int max = Math.max(nums1.length,nums2.length,
   int[ ] ret = new int[max ];

   for( int i =0; i<ret.length; i++){
      if ( i < nums1.length){
         ret[i] += nums1[i] ;

      }
      if (i < nums2.length){
         ret[i] += nums2[i];

      }


      {null - ->   }
   }


return ret;

}
```