Q6: Array Programming

Write a static method called `insertMiddle` that takes in two integer arrays as parameters (we'll call them `arr1` and `arr2`) and <u>returns</u> a new array that contains:
1. the elements from the first half of `arr1`
2. then, <u>all</u> of the elements of `arr2`
3. finally, the rest of the elements in the second half of `arr1`

For example, consider the following two arrays:

```
int[] arr1 = {1, 2, 5, 6};
int[] arr2 = {3, 4};
```

A call to `insertMiddle(arr1, arr2)` should <u>return</u> the following array:

```
{1, 2, 3, 4, 5, 6}
```

It may be the case that the first array has an odd length. In this case, treat the first half as the "shorter half" of the two. For example, consider the following two arrays:

```
int[] arr3 = {2, 4, 6, 8, 10};
int[] arr4 = {1, 1, 1};
```

A call to `insertMiddle(arr3, arr4)` should <u>return</u> the following array:

```
{2, 4, 1, 1, 1, 6, 8, 10}
```

In addition,
- you may assume that neither array is `null`
- you may <u>not</u> assume that neither array is non-empty; in other words, your method should also work if one (or both) of the arrays has length `0`

As a reminder, you are restricted to the methods and classes provided on the reference sheet. Your method should <u>not</u> modify either of the arrays that are provided as parameters.

Here's the solution we went over in class:

```java
public static int[] insertMiddle(
  int[] arr1, int[] arr2
) {
  int[] result = new int[arr1.length + arr2.length];

  int firstHalf = arr1.length/2;

  for (int i = 0; i < firstHalf; i++) {
    result[i] = arr1[i];
  }

  for (int i = 0; i < firstHalf; i++) {
    result[i + firstHalf] = arr2[i];
  }

  for (int i = firstHalf; i < arr1.length; i++) {
    result[i + arr2.length] = arr1[i];
  }

  return result;
}
```