

## 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 returns a new array that contains:

1. the elements from the first half of `arr1`
2. then, all 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 return 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 return the following array:

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

In addition,

- you may assume that neither array is `null`
- you may not 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 not modify either of the arrays that are provided as parameters.

*Write your solution to problem #6 here:*

# CSE 121 Final Exam Reference Sheet

(DO NOT WRITE ANY WORK YOU WANTED GRADED ON THIS REFERENCE SHEET. IT WILL NOT BE GRADED)

```

type[] name = new type[length];
type[] name = {VAL1, VAL2, VAL3, ...};

type[][] name = new type[numRows][numColumns];
type[][] name = {
    {VAL1, VAL2, VAL3, ...},
    ...
    {VAL4, VAL5, VAL6, ...}
};
    
```

Using Arrays	Description
<code>int value = name[i]</code>	Get the value at index <code>i</code>
<code>name[i] = value</code>	Set the value at index <code>i</code>
<code>name.length</code>	Get the number of elements in <code>name</code>

Using 2D Arrays (rectangular)	Description
<code>int value = name[i][j]</code>	Get the value at row <code>i</code> , column <code>j</code>
<code>name[i][j] = value</code>	Set the value at row <code>i</code> , column <code>j</code>
<code>name.length</code>	Number of rows (number of inner arrays)
<code>name[0].length</code>	Number of columns (length of inner array)

String Method	Description
<code>charAt(i)</code>	Returns character in this String at index <code>i</code>
<code>contains(str)</code>	Returns <code>true</code> if this String contains <code>str</code> inside it, returns <code>false</code> otherwise
<code>startsWith(str)</code>	Returns <code>true</code> if this String starts with <code>str</code> , returns <code>false</code> otherwise
<code>endsWith(str)</code>	Returns <code>true</code> if this String ends with <code>str</code> , returns <code>false</code> otherwise
<code>equals(str)</code>	Returns <code>true</code> if this String is the same as <code>str</code> , returns <code>false</code> otherwise
<code>equalsIgnoreCase(str)</code>	Returns <code>true</code> if this String is the same as <code>str</code> ignoring capitalization, returns <code>false</code> otherwise
<code>indexOf(str)</code>	Returns the first index this String where <code>str</code> begins, returns <code>-1</code> if not found
<code>length()</code>	Returns the number of characters in this String
<code>replace(str, newStr)</code>	Returns a new String with all <code>str</code> in this String replaced with <code>newStr</code>
<code>substring(i)</code>	Returns characters in this String from index <code>i</code> (inclusive) to end (exclusive)
<code>substring(i, j)</code>	Returns characters in this String from index <code>i</code> (inclusive) to <code>j</code> (exclusive)
<code>toLowerCase()</code>	Returns an all-lowercase version of this String
<code>toUpperCase()</code>	Returns an all-uppercase version of this String

Random Method	Description
<code>nextInt(max)</code>	Returns a random integer from 0 (inclusive) to <code>max</code> (exclusive)
<code>nextInt(min, max)</code>	Returns a random integer from <code>min</code> (inclusive) to <code>max</code> (exclusive)

Math Method	Description
<code>Math.abs(val)</code>	Returns the absolute value of <code>val</code>
<code>Math.max(val1, val2)</code>	Returns the larger of the two values <code>val1</code> and <code>val2</code>
<code>Math.min(val1, val2)</code>	Returns the smaller of the two values <code>val1</code> and <code>val2</code>
<code>Math.pow(base, exp)</code>	Returns the value of <code>base</code> raised to the <code>exp</code> power
<code>Math.sqrt(val)</code>	Returns the square root of <code>val</code>

Scanner Method	Description
<code>next()</code>	Returns the next token as a String
<code>nextLine()</code>	Returns the entire line as a String
<code>nextInt()</code>	Returns the next token as an <code>int</code> , throws exception if cannot
<code>nextDouble()</code>	Returns the next token as a <code>double</code> , throws exception if cannot