Consider the following skeleton:

```
10    public class Interleaver
11 ▾  {
12         private String[] myString;
13
14         public String[] interleave2(String[] other)
15 ▾       {
16             String[] toRtn = new String[_____];
17             // your code here
18             for (_____)
19             {
20                 // your code here
21             }
22
23             return toRtn;
24         }
25    }
```

(a) Write a method that will interleave two arrays of the same length. (**skip and just do b if you got this**)

(ex)
Input
{1,2,3,4}
{5,6,7,8}
Output
{1,5,2,6,3,7,4,8}

(b) Write a method that will interleave two arrays of different lengths

(ex)
Input
{1,2,3,4}
{5,6,7,8}
Output
{1,5,2,6,3,7,4,8}

Input

{1,2}
{5,6,7,8}
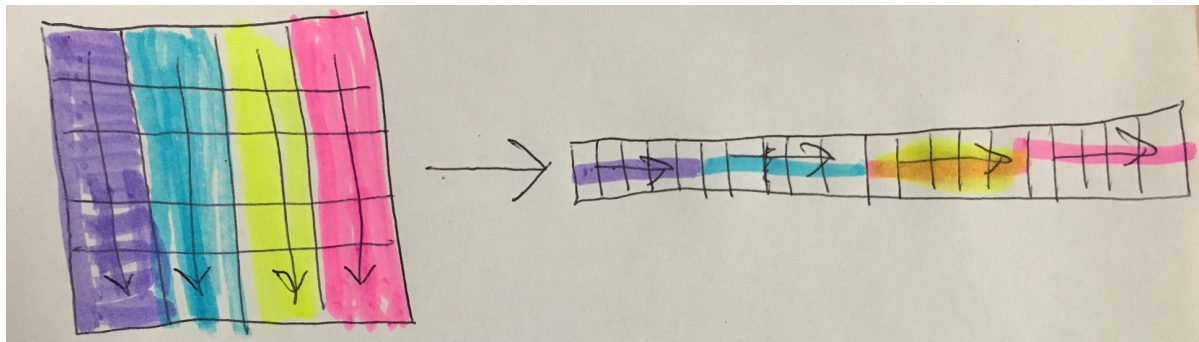Output
{1,5,2,6,7,8}

    (c) Write code to interleave 2 linked lists

    (d) Write code which interleave

    (e) "Vectorize" a 2D array. Assume the arrays are of the same length

```
49   public static String[][] mInterleave(String[][] entries)
50   {
51       // your code here
52   }
```

Hint: if you are having trouble visualizing the problem, consider the following visualization of what the code should do.

Deep CS Advice: Were you able to visualize what "interleaving" actually entailed without the visualization above? Was the visualization above useful? It's important to visualize what you are doing! Given some crazy transformation ("vectorize" is a transformation) where you take some arbitrary data structures and output some arbitrary data structures, could you code it?