

**COMP 110/L**  
**Fall 2022**

**Lecture 18 Handout**

1.) The `printEachElement` method takes a two-dimensional array of `String` elements, and prints each on its own separate line. An example is shown below:

```
String[][] arr = new String[][]{
    new String[]{ "foo", "bar" },
    new String[]{ "baz" },
    new String[] { "alpha", "beta", "gamma" }
};
printEachElement(arr);
```

**Output:**

```
foo
bar
baz
alpha
beta
gamma
```

**Implement** `printEachElement` below. The signature is provided for you. As a hint, you'll need to write a loop inside of a loop, where the outer loop tracks which row you are on, and the inner loop tracks which column you are on for the given row.

```
public static void printEachElement(String[][] elems) {
```

2.) Consider the following method, which is supposed to copy a given array:

```
public static int[] copyArray(int[] input) {  
    int[] output = new int[input.length];  
    for (int index = 0; index < input.length; index++) {  
        output[index] = input[index];  
    }  
    return output;  
}
```

Using `assertArrayEquals` and/or `fail`, write three tests:

- One that makes sure the return value of `copyArray` is empty if the input array was empty (needs `assertArrayEquals`)
- One that makes sure that the return value of `copyArray` is the array `{1, 2, 3 }` if given the array `{ 1, 2, 3 }` (needs `assertArrayEquals`)
- One that makes sure that the return value of `copyArray` does not have negative length. The input can be anything. (needs `fail`)