

```
public static int[] compute(int[] numbersArray) {  
    for (int i = 0; i <= numbersArray.length/2 - 1; i++) {  
        int tmp = numbersArray[numbersArray.length - i - 1];  
        numbersArray[numbersArray.length - i - 1] = numbersArray[i];  
        numbersArray[i] = tmp;  
    }  
    return numbersArray;  
}
```

The diagram illustrates the state of the `numbersArray` and the `tmp` variable during the execution of the `compute` method. It features several colored dots and connecting lines:

- A black dot is positioned above the `int[]` type declaration in the method signature.
- A purple dot is located above the `int i` declaration in the `for` loop.
- A red dot is placed above the `int tmp` declaration.
- A yellow dot is positioned above the `numbersArray` parameter in the `return` statement.
- A purple dot is located above the `numbersArray` parameter in the first assignment statement: `numbersArray[numbersArray.length - i - 1] = numbersArray[i];`
- An orange dot is placed above the `tmp` variable in the second assignment statement: `numbersArray[i] = tmp;`

Connections between these dots represent the flow of data:

- A purple line connects the purple dot above `int i` to the purple dot above the first `numbersArray` parameter.
- A red line connects the red dot above `int tmp` to the orange dot above `tmp` in the second assignment.
- A yellow line connects the yellow dot above the `numbersArray` parameter in the `return` statement to the purple dot above the first `numbersArray` parameter.
- A purple line connects the purple dot above the first `numbersArray` parameter to the orange dot above `tmp` in the second assignment.