Solution Review: Palindromic Arrays

This lesson explains the solution for the palindromic arrays exercise.

WE'LL COVER THE FOLLOWING

There are several ways to go about this challenge. You can choose any one of them depending on your preference.

Solution 1: Indexing

Since `newarray1` and `newarray2` had already been created, we could replace their values by indexing:

```
let inputArray1 = [| 0, 1, 2, 3 |];
let inputArray2 = [| 40, 20, 20 |];
let outputArray1 = Array.make(7, 0);
let outputArray2 = Array.make(5, 0);

/* Solution 1 */
outputArray1[0] = inputArray1[0];
outputArray1[1] = inputArray1[1];
outputArray1[2] = inputArray1[2];
outputArray1[3] = inputArray1[3];
outputArray1[4] = inputArray1[2];
outputArray1[5] = inputArray1[1];
outputArray1[6] = inputArray1[0];

outputArray2[0] = inputArray2[0];
outputArray2[1] = inputArray2[1];
outputArray2[2] = inputArray2[2];
outputArray2[3] = inputArray2[1];
outputArray2[4] = inputArray2[0];

Js.log(outputArray1);
Js.log(outputArray2);
```

Solution 2: Using `get()` and `set()` Properties

An alternative to directly assigning the values through indices would be to use `Array.set()` and `Array.get()`.

Since we are assigning values to the `outputArray`s, we will use them in the `Array.set()` function. The third part of `Array.set()` requires the value which we need to assign. Therefore, we'll use `Array.get()` on the `inputArray`s here.

```
let inputArray1 = [| 0, 1, 2, 3 |];
let inputArray2 = [| 40, 20, 20 |];
let outputArray1 = Array.make(7, 0);
let outputArray2 = Array.make(5, 0);

/* Solution 2 */
Array.set(outputArray1, 0, Array.get(inputArray1, 0));
Array.set(outputArray1, 1, Array.get(inputArray1, 1));
Array.set(outputArray1, 2, Array.get(inputArray1, 2));
Array.set(outputArray1, 3, Array.get(inputArray1, 3));
Array.set(outputArray1, 4, Array.get(inputArray1, 2));
Array.set(outputArray1, 5, Array.get(inputArray1, 1));
Array.set(outputArray1, 6, Array.get(inputArray1, 0));

Array.set(outputArray2, 0, Array.get(inputArray2, 0));
Array.set(outputArray2, 1, Array.get(inputArray2, 1));
Array.set(outputArray2, 2, Array.get(inputArray2, 2));
Array.set(outputArray2, 3, Array.get(inputArray2, 1));
Array.set(outputArray2, 4, Array.get(inputArray2, 0));

Js.log(outputArray1);
Js.log(outputArray2);
```

Solution 3: New `let` Binding

We could also create new arrays by redefining `newarray1` and `newarray2`. In this case, we would need to specify the values in the new `let` bindings:

```
let inputArray1 = [| 0, 1, 2, 3 |];
let inputArray2 = [| 40, 20, 20 |];
let outputArray1 = Array.make(7, 0);
let outputArray2 = Array.make(5, 0);

/* Solution 3 */
let outputArray1 = [| 0, 1, 2, 3, 2, 1, 0 |];
let outputArray2 = [| 40, 20, 20, 20, 40|];
```

```
Js.log(outputArray1);
Js.log(outputArray2);
```