LAB EXERCISE 7

Part 1: Arrays

1. **Create a function called flattenArray.** It should take a nested array as an argument and flatten it into a single array. Use the .flat() method, which concatenates all sub-array elements into a new array recursively up to the specified depth.
2. **reate a function called checkInclusion.** It should take two arguments: an array of numbers and a specific number. The function should return true if the number is included in the array, and false otherwise. Use the .includes() method, which checks if an array contains a specific value.
3. **Create a function called concatenateArrays.** It should take two arrays as arguments and return a new array that combines the elements of both arrays. Use The .concat() method, which merges two or more arrays. It does not change the existing arrays but instead returns a new array.
4. Create a function called **sortStrings**. It should take an array of strings as an argument and return a new array where the strings are sorted alphabetically.

Part 2: Strings

1. Create a function called **mixUp.** It should take in two strings, and return the concatenation of the two strings (separated by a space) slicing out and swapping the first 2 characters of each. You can assume that the strings are at least 2 characters long.
    - For example: mixUp('mix', pod'): 'pox mid'
2. Create a function called **verbing.** It should take a single argument, a string. If its length is at least 3, it should add 'ing' to its end, unless it already ends in 'ing', in which case it should add 'ly' instead. If the string length is less than 3, it should leave it unchanged.
    - For example: verbing('swim'): 'swimming', verbing('swimming'): 'swimmingly'
3. Create a function called **isPalindrome.** It should take a single string as an argument and return true if the string is a palindrome (reads the same backward and forward), and false otherwise.