Recursion Assignment

Problem 1: Recursion

```
// Write a recursive function called flatTheArray which accepts an array of arrays
// and returns a new array with all values flattened.

// flatTheArray([1, 2, 3, [4, 5] ]) // [1, 2, 3, 4, 5]
// flatTheArray([1, [2, [3, 4], [[5]]]]) // [1, 2, 3, 4, 5]
// flatTheArray([[1],[2],[3]]) // [1,2,3]
// flatTheArray([[[[1], [[[2]]], [[[[[[[3]]]]]]]]]]) // [1,2,3]

const flatTheArray = (arr) => {
}
```

Problem 2: Recursion

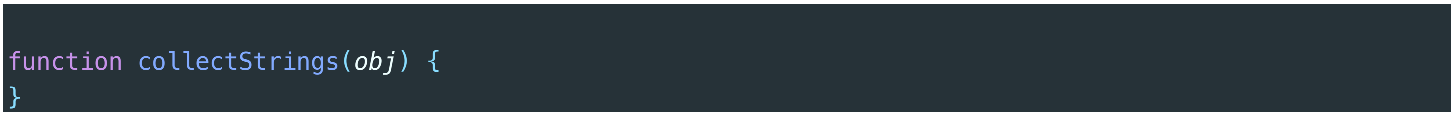
```
// Write a recursive function called capitalizeWords.
// Given an array of words, return a new array containing each word capitalized.

// let words = ['tony', 'kim'];
// capitalizedAllLetters(words); // ['TONY', 'KIM']

const capitalizeAllLetters = (array) => {
}
```

Problem 3: Recursion

```
// example 3: factorial

// factorial(1) // 1
// factorial(2) // 2
// factorial(7) // 5040

// Write a function factorial which accepts a number and returns the factorial of that number.
// A factorial is the product of an integer and all the integers below it;
// e.g., factorial four ( 4! ) is equal to 24, because 4 * 3 * 2 * 1 equals 24.  factorial zero
(0!) is always 1.

function factorial(x){
}
```

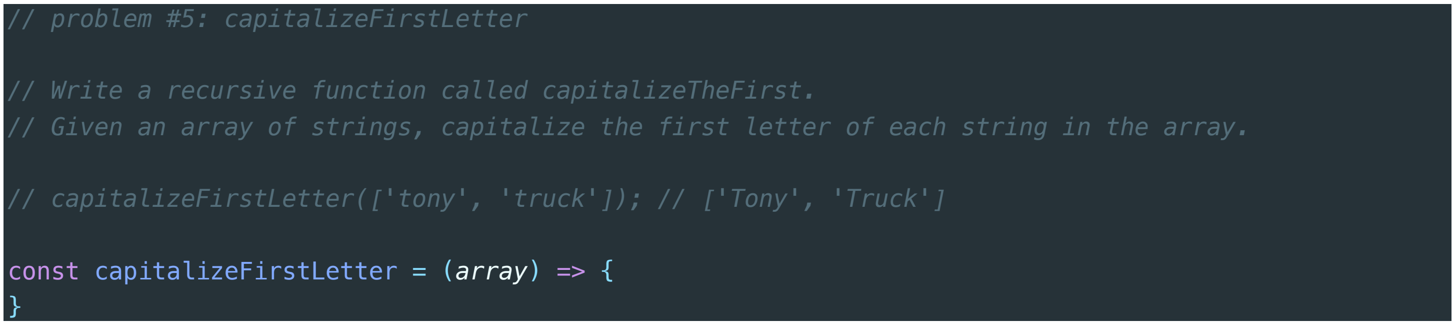
Problem 4: Recursion

```
// problem 4: collect Strings

// Write a function called collectStrings which accepts an object and returns
// an array of all the values in the object that have a typeof string

collectStrings(obj) // ["foo", "bar", "baz"])

//recursion with helper
```



```
function collectStrings(obj) {
}
```

Problem 5: Recursion



```
// problem #5: capitalizeFirstLetter

// Write a recursive function called capitalizeTheFirst.
// Given an array of strings, capitalize the first letter of each string in the array.

// capitalizeFirstLetter(['tony', 'truck']); // ['Tony', 'Truck']

const capitalizeFirstLetter = (array) => {
}
```