**Maps and Sets Exercise**

**Quick Question #1**

What does the following code return?

**new** Set([1,1,2,2,3,4])

(4){1,2,3,4}

**Quick Question #2**

What does the following code return?

[...**new** Set("referee")].join("")

// [...new Set(‘referee’] returns an array: [‘r’, ‘e’, ‘f’]

// .join(‘’) turns array into a string and returns ‘ref’

‘ref’

**Quick Questions #3**

What does the Map ***m*** look like after running the following code?

**// creates empty Map array**

**let** m = **new** Map();

m.set([1,2,3], **true**);

{Array(3) => true}

1. **key**: (3) [1, 2, 3]
2. **value**: true

m.set([1,2,3], **false**);

{Array(3) => false}

1. **key**: (3) [1, 2, 3]
2. **value**: false

**hasDuplicate**

Write a function called hasDuplicate which accepts an array and returns true or false if that array contains a duplicate

function hasDuplicate (*arr*){

    const newSet = new Set(*arr*);

*if*(newSet.size !== *arr*.length){

*return* true

    }

*return* false

}

hasDuplicate([1,3,2,1]) *// true*

hasDuplicate([1,5,-1,4]) *// false*

**vowelCount**

Write a function called vowelCount which accepts a string and returns a map where the keys are letters and the values are the count of the vowels in the string.

vowelCount('awesome') *// Map { 'a' => 1, 'e' => 2, 'o' => 1 }*

vowelCount('Colt') *// Map { 'o' => 1 }*

function isVowel(*char*){

*// returns true or false depending on if  char is a vowel or not*

*return* "aeiou".includes(*char*);

  }

  function vowelCount(*str*){

*// create an empty Map object to keep track of how many duplicate vowels there are in str*

    const vowelMap = new Map();

*// loop over each char of str*

*for*(let char of *str*){

*// turn all char to lowercase*

      let lowerCaseChar = char.toLowerCase()

*//   if lowerCaseChar is a vowel*

*if*(isVowel(lowerCaseChar)){

*// if Map object already includes a count of that char (vowel)*

*if*(vowelMap.has(lowerCaseChar)){

*// use 'set' to add a key/value pair to vowelMap with the lowerCaseChar as the key and vowelMap.get(lowerCaseChar) as the value incremented by 1*

          vowelMap.set(lowerCaseChar, vowelMap.get(lowerCaseChar) + 1);

        } *else* {

*// use 'set' to add new key/value pair to vowelMap with lowerCaseChar as the key and 1 as the value*

          vowelMap.set(lowerCaseChar, 1);

        }

      }

    }

*return* vowelMap;

  }