Anagram

result of rearranging the letters of a word to produce a new word, using all the original letters exactly once.

Example : geeks,keegs

Problem: Given a text and a pattern, print all occurrences of pattern and its permutations (or anagrams) in text.

```
text = "AAABABAA"
pattern = "AABA"
```

The first occurrence of 3 A's and 1 B is AAAB at index of 0:

```
text = "AAABABAA"
pattern = "AABA"
```

Our next occurrence of 3 A's and 1 B is AABA at index of 1:

```
text = "AAABABAA"
pattern = "AABA"
```

The next occurrence of 3 A's and 1 B is ABAA at index of 4:

```
text = "AAABABAA"
pattern = "AABA"
```

Thus, we say our Anagram exists at indices 0, 1 and 4.

```
/**
 * Write a function to determine if one String is
 * a case-insensitive anagram of another String
 * abcd bacd - true
 * abcd ddbbccaa - false
 * abcd bacf - false
 */
```

If we take all the letters and sort them alphabetically to determine 2 things:

- Letters match.
- Lengths match.

Use the **split()** method to split each string and turn it into an array.
Use the **join()** method to combine the elements of the array.
Because it's case sensitive, we use the **toLowerCase()** method.

Finally, we compare between paramaters str1 and str2... does sorted1 match sorted2 ?

```
let isAnagram = function (str1, str2) {
    let sorted1 = str1.split('').sort().join('').toLowerCase();
    let sorted2 = str2.split('').sort().join('').toLowerCase();
    return (sorted1 === sorted2);
}

console.log(isAnagram('hello', 'jello'));
console.log(isAnagram('hello', 'loelh'));
```

Checking the console, we get:

```
false
true
```