

Task 1 - Magic Latin

Your task is move the first letter of each word to the end of it, then add "ay" to the end of the word. Leave punctuation marks untouched.

```
magicLatin('Magic latin is cool'); // agicMay atinlay siay oolcay
magicLatin('Hello world !');      // elloHay orldWay !
```

Task 2 - Valid Parentheses

Your task is to write a function that takes a string of parentheses, and determines if the order of the parentheses is valid. The function should return `true` if the string is valid, and `false` if it's invalid.

```
"()"          => true
")(()))"     => false
")(())(("    => false
"((()))((()())" => true
```

Task 3 - Binary number divisible by 5

You need to define a **regular expression** which tests if a given string representing a binary number is divisible by 5.

```
// 5 divisible by 5
divisibleByFive.test('101') === true

// 135 divisible by 5
divisibleByFive.test('10000111') === true

// 666 not divisible by 5
divisibleByFive.test('0000001010011010') === false
```

Task 4 - Next bigger number

You have to create a function that takes a positive integer number and returns the next bigger number formed by the same digits:

```
nextBigger(12) == 21
nextBigger(513) == 531
nextBigger(2017) == 2071
nextBigger(111) == -1
nextBigger(531) == -1
```