**Base-001**

Дана строка из символов латинского алфавита. Написать функцию transStr(str),

которая строчные гласные буквы делает прописными, а прописные – строчными.

Функция должна возвратить результат. Например :

console.log(transStr(‘colOr’)); // ‘cOlor’

console.log(transStr(‘sOrrY’)); // ‘sorry’

Решение :

function transStr(str) {  
 str = str || '';  
 let i,  
 len = str.length,  
 char,  
 result = '',  
 obj = {  
 a : 'A',  
 e : 'E',  
 o : 'O',  
 i : 'I',  
 u : 'U',  
 y : 'Y'  
 };  
  
 for (i = 0; i < len; i++) {  
 char = str[i].toLowerCase();  
 if (char in obj) {  
 result +=

(obj[char] === str[i]) ? str[i].toLowerCase() : str[i].toUpperCase();  
 }else{  
 result += str[i];  
 }  
 }  
 return result;  
}  
  
/\* =================== the testing function ======================= \*/  
function test (testedFunc, result, ...data) {  
 testedFunc(...data) === result ?  
 console.log(`+ + + + + + + : ${data} --> ${result}`) :  
 console.log(`- - - - - - - : ${data} --> ${result}`);  
}

/\* =================== tests ======================= \*/  
test(transStr, '', '');  
test(transStr, 'colOr', 'cOlor');  
test(transStr, ' cOOOOr alisA ', ' coooor AlIsa ');  
test(transStr, 'auaueoiuaueye', 'AUAUEOIUAUEYE');  
test(transStr, 'dfrtfdrfggtrfd', 'dfrtfdrfggtrfd');  
test(transStr, '^^^\*&%$###$#!@#@!~~', '^^^\*&%$###$#!@#@!~~');  
test(transStr, '', undefined);  
test(transStr, '', 12345);  
test(transStr, 'a', 'A', 'O');

**Base-002**

<https://www.codewars.com/kata/sum-of-digits-slash-digital-root>

In this kata, you must create a digital root function.

A digital root is the recursive sum of all the digits in a number. Given n, take the sum of the digits of n. If that value has two digits, continue reducing in this way until a single-digit number is produced. This is only applicable to the natural numbers.

Here's how it works:

digitalRoot(16)

=> 1 + 6

=> 7

digitalRoot(942)

=> 9 + 4 + 2

=> 15 ...

=> 1 + 5

=> 6

digitalRoot(132189)

=> 1 + 3 + 2 + 1 + 8 + 9

=> 24 ...

=> 2 + 4

=> 6

digitalRoot(493193)

=> 4 + 9 + 3 + 1 + 9 + 3

=> 29 ...

=> 2 + 9

=> 11 ...

=> 1 + 1

=> 2

function digitalRoot(n) {  
 while (n >= 10) {  
 let i,  
 result = 0,  
 str = n + '',  
 len = str.length;  
 for (i = 0; i < len; i++) {  
 result += +str[i];  
 }  
 n = result;  
 }  
 return n;  
}  
  
/\* =================== the testing function ======================= \*/  
function test (testedFunc, result, ...data) {  
 testedFunc(...data) === result ?  
 console.log(`+ + + + + + + : ${data} --> ${result}`) :  
 console.log(`- - - - - - - : ${data} --> ${result}`);  
}  
  
/\* =================== tests ======================= \*/  
test(digitalRoot, 7, 16);  
test(digitalRoot, 6, 942);  
test(digitalRoot, 6, 132189);  
test(digitalRoot, 2, 493193);