Object literals. Array and String methods

1. Given an object. Invert it (keys become values and values become keys). If there is more than key for that given value create an array.

|  |  |
| --- | --- |
| **Input** | **Output** |
| { a: ‘1’, b: ‘2’ } | { 1: ‘a’, 2: ‘b’ } |
| { a: ‘1’, b: ‘2’, c: ‘2’ } | { 1: ‘a’, 2: [‘b’, ‘c’] } |
| { a: ‘1’, b: ‘2’, c: ‘2’, d: ‘2’ } | { 1: ‘a’, 2: [‘b’, ‘c’, ‘d’] } |

1. Given the list of the following readers:  
   [  
    { book: "Catcher in the Rye", readStatus: true, percent: 40},  
    { book: "Animal Farm", readStatus: true, percent: 20},  
    { book: "Solaris", readStatus: false, percent: 90 },  
    { book: "The Fall", readStatus: true, percent: 50 },  
    { book: "White Nights", readStatus: false, percent: 60 } ,  
    { book: "After Dark", readStatus: true, percent: 70 }  
   ];  
   Output the books sorted by the **percent** in descending order which **readStatus** is true.
2. Given two objects. Write a function that performs shallow compare.

|  |  |
| --- | --- |
| **Input** | **Output** |
| var a = { a: ‘1’ };  var b = { a: ‘1’};  shallowCompare(a, b) | true |
| var a = { a: ‘1’ };  var b = { a: ‘1’, b: ‘2’ };  shallowCompare(a, b) | false |
| var a = { };  var b = { };  shallowCompare(a, b) | true |

1. Given an object. Write a function that creates a deep copy of it.

|  |  |
| --- | --- |
| **Input** | **Output** |
| var a = { a: ‘1’, b: { a: 2} };  var b = deepCopy(a);  a.b.a = 123;  console.log(b.b.a !== 123); | true |

1. Given an array. Determine whether it consists only from uniques or not.
2. Determine if a word or phrase is an isogram. An isogram (also known as a "non pattern word") is a word or phrase without a repeating letter.
3. Given an array of numbers. Find the sum of numbers’ quadratic which are even.
4. Check whether string is palindrome, or not.
5. Given an array of integers, find the pair of adjacent elements that has the largest product and return that product.
6. Given an array of integers. All numbers are unique. Find the count of missing numbers between minimum and maximum elements to make integers sequence.
7. Implement the classic method for composing secret messages called a square code. Given an English text, output the encoded version of that text.  
   First, the input is normalized: the spaces and punctuation are removed from the English text and the message is downcased. Then, the normalized characters are broken into rows. The plain text should be organized into a rectangle. The size of the rectangle (r x c) should be decided by the length of the message, such that c >= r and c - r <= 1, where c is the number of columns and r is the number of rows. The coded message is obtained by reading down the columns going left to right.  
     
   ***Plain:*** *If man was meant to stay on the ground, god would have given us roots.  
   Ifmanwasmeanttostayonthegroundgodwouldhavegivenusroots****Cipher****:  
   ifmanwas  
   meanttos  
   tayonthe  
   groundgo  
   dwouldha  
   vegivenu  
   sroots*

1. Create an implementation of the rotational cipher, also sometimes called the Caesar cipher. The Caesar cipher is a simple shift cipher that relies on transposing all the letters in the alphabet using an integer key between 0 and 26. Using a key of 0 or 26 will always yield the same output due to modular arithmetic. The letter is shifted for as many values as the value of the key.  
     
   ***Plain****: abcdefghijklmnopqrstuvwxyz****Cipher****: nopqrstuvwxyzabcdefghijklm*