Swift (iOS) Challenges

Challenge 01

Are the letters unique?

Difficulty: Easy

Write a function that accepts a String as it's only parameter, and returns true if the string has only unique letters, taking letter case into account.

Sample input and output :

- **The string "No duplicates" should return true.**
- **The string "abcdefghijklmnopqrstuvwxyz" should return true.**
- **The string "AaBbCc" should return true because the challenge is case-sensitive.**
- **The string "Hello, world" should return false because of the double Ls and double Os.**

Challenge 02

Is a string a palindrome?

Difficulty: Easy

Write a function that accepts a String as its only parameter, and returns true if the string reads the same when reversed, ignoring case.

Sample input and output :

- **The string “rotator” should return true.**
- **The string “Rats live on no evil star” should return true.**
- **The string “Never odd or even” should return false; even though the letters are the same in reverse the spaces are in different places.**
- **The string “Hello, world” should return false because it reads “dlrow ,olleH” backwards.**

Challenge 03

Do two strings contain the same characters?

Difficulty: Easy

Write a function that accepts two String parameters, and returns true if they contain the same characters in any order taking into account letter case.

Sample input and output :

- **The strings “abca” and “abca” should return true.**
- **The strings “abc” and “cba” should return true.**
- **The strings “ a1 b2 ” and “ b1 a2 ” should return true.**
- **The strings “abc” and “abca” should return false.**
- **The strings “abc” and “Abc” should return false.**
- **The strings “abc” and “cbAa” should return false.**
- **The strings “abcc” and “abca” should return false.**

Challenge 04

Does one string contain another?

Difficulty: Easy

Write your own version of the contains() method on String that ignores letter case, and without using the existing contains() method.

Sample input and output :

- **The code "Hello, world".myContains("Hello") should return true.**
- **The code "Hello, world".myContains("WORLD") should return true.**
- **The code "Hello, world".myContains("Goodbye") should return false.**

Challenge 05

Count the Characters?

Difficulty: Easy

Write a function that accepts a string and character, and returns how many times a specific character appears, taking case into account.

Tip: If you can solve this without using a for-in loop, you can consider it a Tricky challenge.

Sample input and output :

- **The letter “a” appears twice in “I like animals”**
- **The letter “p” appears twice in “Whatsupp”**
- **The letter “i” appears four times in “Mississippi”.**

Challenge 06

Remove duplicate letters from a string

Difficulty: Easy

Write a function that accepts a string as its input, and returns the same string just with duplicate letters removed.

Tip: If you can solve this without using a for-in loop, you can consider it a Tricky challenge.

Sample input and output :

- **The string “wombat” should print “wombat”.**
- **The string “hello” should print “helo”.**
- **The string “Mississippi” should print “Misp”.**

Challenge 07

Condense whitespace

Difficulty: Easy

Write a function that returns a string with any consecutive spaces replaced with a single space or “_” (just for convenience).

Sample input and output :

I’ve marked spaces using “[space]” below for visual purposes:

- **The string “a[space][space][space]b[space][space][space]c” should return “a_b_c”.**
- **The string “[space][space][space][space]a” should return “_a”.**
- **The string “abc” should return “abc”.**

Challenge 08

String is rotated

Difficulty: Tricky

Write a function that accepts two strings, and returns true if one string is rotation of the other, taking letter case into account.

Tip: A string rotation is when you take a string, remove some letters from its end, then append them to the front. For example, “swift” rotated by two characters would be “ftswi”

Sample input and output :

- **The string “abcde” and “eabcd” should return true.**
- **The string “abcde” and “cdeab” should return true.**
- **The string “abcde” and “abced” should return false; this is not a string rotation.**
- **The string “abc” and “a” should return false; this is not a string rotation.**

Challenge 09

Find pangrams

Difficulty: Tricky

Write a function that returns true if it is given a string that is an English pangram, ignoring letter case.

Tip: A pangram is a string that contains every letter of the alphabet at least once.

Sample input and output :

- **The string “The quick brown fox jumps over the lazy dog” should return true.**
- **The string “The quick brown fox jumped over the lazy dog” should return false, because it’s missing the S.**

Challenge 10

Vowels and consonants

Difficulty: Tricky

Given a string in English, return a tuple containing the number of vowels and consonants.

Tip: Vowels are the letters, A, E, I, O, and U; consonants are the letters B, C, D, F, G, H, J, K, L, M, N, P, Q, R, S, T, V, W, X, Y, Z.

Sample input and output :

- **The input “Swift Coding Challenges” should return 6 vowels and 15 consonants.**
- **The input “Mississippi” should return 4 vowels and 7 consonants.**

Challenge 11

Three different letters

Difficulty: Tricky

Write a function that accepts two strings, and returns true if they are identical in length but have no more than three different letters, taking case and string order into account.

Sample input and output :

- **The strings “Clamp” and “Cramp” would return true, because there is one letter difference.**
- **The strings “Clamp” and “Crams” would return true, because there are two letter differences.**
- **The strings “Clamp” and “Grams” would return true, because there are three letter differences.**
- **The strings “Clamp” and “Grans” would return false, because there are four letter differences.**
- **The strings “Clamp” and “Clam” would return false, because they are different lengths.**
- **The strings “clamp” and “maple” should return false. Although they differ by only one letter, the letters that match are in different positions.**

Challenge 12

Find longest prefix

Difficulty: Tricky

Write a function that accepts a string of words with a similar prefix, separated by spaces, and returns the longest substring that prefixes all words.

Sample input and output :

- **The string “swift switch swill swim” should return “swi”.**
- **The string “flip flap flop” should return “fl”.**

Challenge 13

Run-length encoding

Difficulty: Taxing

Write a function that accepts a string as input, then returns how often each letter is repeated in a single run, taking case into account.

Tip: This approach is used in a simple lossless compression technique called run-length encoding.

Sample input and output :

- **The string “aabbcc” should return “a2b2c2”.**
- **The strings “aaabaaabaaa” should return “a3b1a3b1a3”**
- **The string “aaAAaa” should return “a2A2a2”**

Challenge 14

Reverse the words in a string

Difficulty: Tricky

Write a function that returns a string with each of its words reversed but in the original order, without using a loop.

Sample input and output :

- **The string “Swift Coding Challenges” should return “tfiwS gnidoC segnellahC”.**
- **The string “The quick brown fox” should return “ehT kciuq nworb xof”.**

Challenge 15

Count the numbers

Difficulty: Easy

Write an extension for collections of integers that returns the number of times a specific digit appears in any of its numbers.

Sample input and output :

- **The code [5, 15, 55, 515].countTheNumbers(count: "5") should return 6.**
- **The code [5, 15, 55, 515].countTheNumbers(count: "1") should return 2.**
- **The code [55555].countTheNumbers(count: "5") should return 5.**
- **The code [55555].countTheNumbers(count: "1") should return 0.**

Challenge 16

Integer disguised as string

Difficulty: Tricky

Write a function that accepts a string and returns true if it contains only numbers, i.e. the digits 0 through 9.

Sample input and output :

- **The input “01010101” should return true.**
- **The input “123456789” should return true.**
- **The letter “922337203685477580 8” should return true.**
- **The letter “1.01” should return false; “.” is not a number.**

Challenge 17

Add numbers inside a string

Difficulty: Tricky

Given a string that contains both letters and numbers, write a function that pulls out all the numbers then returns their sum.

Sample input and output :

- **The string “a1b2c3” should return 6 (1 + 2 + 3).**
- **The string “a10b20c30” should return 60 (10 + 20 + 30).**
- **The string “h8ers” should return “8”.**

Challenge 18

Subtract without subtract

Difficulty: Taxing

Create a function that subtracts one positive integer from another, without using "-" (assume that subtract <= from).

Sample input and output :

- **The code subtractWithoutSubtract(subtract: 5, from: 9) should return 4.**
- **The code subtractWithoutSubtract(subtract: 10, from: 30) should return 20.**

Challenge 19

Sort a string array by length

Difficulty: Easy

Extend collections with a function that returns an array of strings sorted by their lengths, longest first.

Sample input and output :

- The code ["a", "abc", "ab"].SortAStringArrayByLengt() should return ["abc", "ab", "a"].
- The code ["Germany", "Israel", "Ivory coast"].SortAStringArrayByLengt() should return ["Ivory coast", "Germany", "Israel"].
- The code [String]().SortAStringArrayByLengt() should return [].