***Завдання:***

В результаті у кожного мають бути функції для виконання 3х завдань,

1 - обов’язкове для всіх,

2 - variant mod 6

3 - (((group mod 10) + variant) mod 6) + 6

Завдання обов'язкове для всіх:

Фіскальний кодекс

Кожна особа в Італії має унікальний ідентифікаційний ідентифікаційний код, виданий національною податковою службою після реєстрації народження: Фіскальний кодекс (Codice Fiscale). [https://en.wikipedia.org/wiki/Italian\_fiscal\_code\_card (https://en.wikipedia.org/wiki/Italian\_fiscal\_code\_card)]

Враховуючи об'єкт, що містить особисті дані людини (ім'я, прізвище, стать та дата народження), поверніть 11 символів коду як рядок, виконуючи наступні кроки:

Створіть 3 символи прізвища, якщо воно має:

Принаймні 3 приголосні, тоді використовуються перші три приголосні. (Ньюмен -> NWM).

Менше ніж 3 приголосні, тоді голосні замінять відсутні літери в тому ж порядку, в якому вони відображаються (Fox -> FXO | Hope -> HPO).

Менше трьох літер тоді "X" займе третю позицію після приголосного та голосного (Yu -> YUX).

Створіть від імені 3 великі літери, якщо у них є:

Рівно 3 приголосних тоді приголосні вживаються в тому порядку, в якому вони з’являються (Метт -> МТТ).

Вживається більше 3 приголосних, тоді перший, третій та четвертий приголосні (Саманта -> СНТ | Томас -> ТМС).

Менше трьох приголосних, тоді голосні замінять відсутні літери в тому ж порядку, в якому вони відображаються (Bob -> BBO | Paula -> PLA).

Менше трьох літер тоді "X" займе третійю позицію після приголосного та голосного (Al -> LAX).

Утворіть 2 числа, 1 букву та 2 числа від дати народження та статі:

Візьміть дві останні цифри року народження (1985 -> 85).

Створіть лист, відповідний місяцю народження (січень -> A | грудень -> T), використовуючи таблицю для перерахунку, включену в код.

Для чоловіків приймають день народження, додаючи один нуль на початку, якщо він менше 10 (будь-який 9-й день -> 09 | будь-який 20-й день -> 20).

Для жінок беруть день народження і сумують до нього 40 (будь-який 9-й день -> 49 | будь-який 20-й день -> 60).

PS:

• Букви в відповіді мають бути в uppercase

• Дата народження в форматі D/M/YYYY

• Таблиця перетворення наведена в завданні

• Для людини створити клас

• Відображення місяця в літеру: // const months = { 1: "A", 2: "B", 3: "C", 4: "D", 5: "E", 6: "H", 7: "L", 8: "M", 9: "P", 10: "R", 11: "S", 12: "T" }

Приклад:

fiscalCode({

name: "Matt",

surname: "Edabit",

gender: "M",

dob: "1/1/1900"

}) ➞ "DBTMTT00A01"

fiscalCode({

name: "Helen",

surname: "Yu",

gender: "F",

dob: "1/12/1950"

}) ➞ "YUXHLN50T41"

Завдання по варіантах:

1 Числа харшад

Числа Харшада / Нівена - це додатні числа, які поділяються на суму їх цифр. Усі одноцифрові числа - це харшадські номери.

Наприклад, 27 - це число Харшада, як 2 + 7 = 9, а 9 - дільник 27.

Різні номери можуть виникати в послідовних кластерах. Числа від 1 до 10 - це харшадські числа.

Числа 132 і 133 - це обидва числа харшада. Числа 1014, 1015, 1016, 1017 - це харшадські числа.

Створіть функцію, яка приймає число і повертає чи дане число є числом Харшада

Приклад:

harshad(5) ➞ true

harshad(82) ➞ false

// 82 mod (8 + 2) != 0

2 Відстань до найближчого голосного

Напишіть функцію, яка займає рядок і для кожного символу повертає відстань до найближчого голосного. Якщо символ сам є голосним, поверніть 0.

мова роботи функції англійська, тому голосними є “a, e, i, o, u”, також всі букви будуть в lowercase.

Приклад:

distanceToNearestVowel("abcdabcd") ➞ [0, 1, 2, 1, 0, 1, 2, 3]

distanceToNearestVowel("shopper") ➞ [2, 1, 0, 1, 1, 0, 1]

3 Виткані струни

Створіть функцію, яка повертає масив заданого рядка, але зміщений пробілами. Ось кілька більш точних інструкцій:

Продовжуйте додавати пробіли зліва, поки у вас не буде однакової кількості пробілів, як довжина слова.

Потім продовжуйте видаляти пробіли, поки не досягнете початкового слова.

Приклад:

wiggleString("hello") ➞ [

"hello",

" hello",

" hello",

" hello",

" hello",

" hello"

" hello",

" hello",

" hello",

" hello",

"hello"

]

4 Перетворення об'єктів у масиви

Напишіть функцію, яка перетворює об’єкт у масив, де кожен елемент представляє пару ключ-значення.

ps: Поверніть порожній масив, якщо об’єкт порожній.

Приклад:

toArray({ a: 1, b: 2 }) ➞ [["a", 1], ["b", 2]]

toArray({ shrimp: 15, tots: 12 }) ➞ [["shrimp", 15], ["tots", 12]]

5 Візерунчастий ремінець

Браслет може мати 4 візерунка:

• горизонтальний: кожен елемент підряд однаковий.

• вертикальний: кожен елемент у стовпці однаковий.

• діагональна зліва: кожен елемент ідентичний тому, що знаходиться вгорі ліворуч або праворуч знизу.

• діагональна справа: кожен елемент ідентичний тому, що знаходиться вгорі праворуч або внизу зліва.

Напишіть функцію, яка повертає True, якщо розділ можна правильно класифікувати на один із 4 типів, в іншому випадку False.

Приклад:

isWristband([

["A", "A"],

["B", "B"],

["C", "C"]

]) ➞ true

// Part of horizontal wristband.

isWristband([

["A", "B"],

["A", "B"],

["A", "B"]

]) ➞ true

// Part of vertical wristband.

isWristband([

["A", "B", "C"],

["C", "A", "B"],

["B", "C", "A"],

["A", "B", "C"]

]) ➞ true

// Part of diagonal left wristband.

isWristband([

["A", "B", "C"],

["B", "C", "A"],

["C", "A", "B"],

["A", "B", "A"]

]) ➞ true

// Part of diagonal right wristband.

6 Сполучні слова

Напишіть функцію, яка з'єднує кожне попереднє слово з наступним словом спільними літерами. Повертайте отриманий рядок (видаляючи дублікати символів у накладенні) та мінімальну кількість спільних літер у всіх парах рядків.

Приклад:

join(["oven", "envier", "erase", "serious"]) ➞ ["ovenvieraserious", 2]

join(["move", "over", "very"]) ➞ ["movery", 3]

join(["to", "ops", "psy", "syllable"]) ➞ ["topsyllable", 1]

// "to" and "ops" share "o" (1)

// "ops" and "psy" share "ps" (2)

// "psy" and "syllable" share "sy" (2)

// the minimum overlap is 1

7 Шифр Цезаря

Юлій Цезар захищав свою конфіденційну інформацію, зашифрувавши її за допомогою шифру. Шифр Цезаря (https://en.wikipedia.org/wiki/Caesar\_cipher (https://en.wikipedia.org/wiki/Caesar\_cipher)) зміщує кожну букву на кілька літер. Якщо зміна приведе вас до кінця алфавіту, просто поверніть назад до передньої частини алфавіту. У разі обертання на 3, w, x, y і z буде відображатися на z, a, b і c.

Створіть функцію, яка займає рядок s (текст, який потрібно зашифрувати) та ціле число k (коефіцієнт обертання). Він повинен повернути зашифрований рядок.

PS: Всі символи на вході є ASCII символами

Приклад:

caesarCipher("middle-Outz", 2) ➞ "okffng-Qwvb"

// m -> o

// i -> k

// d -> f

// d -> f

// l -> n

// e -> g

// - -

// O -> Q

// u -> w

// t -> v

// z -> b

caesarCipher("Always-Look-on-the-Bright-Side-of-Life", 5)

➞ "Fqbfdx-Qttp-ts-ymj-Gwnlmy-Xnij-tk-Qnkj"

8 Зростання послідовних чисел

Напишіть функцію, яка повертає істину, якщо рядок складається з висхідних та послідовних чисел.

ps: Число може складатися з будь-якої кількості цифр, якщо цифри прилягають одна до одної, то рядок має щонайменше дві з них.

Приклад:

ascending("2324256") ➞ false

// No matter how this string is divided, the numbers are not consecutive.

ascending("444445") ➞ true

// Consecutive numbers 444 and 445.

9 Симетричні візерунки

Кетлін володіє прекрасним магазином килимів. Їй подобається групувати килими в 4 взаємовиключні категорії.

• недосконалий => imperfect

• горизонтально симетричний => horizontally symmetric

• вертикально симетричний => vertically symetric

• досконалий => perfect

Недосконалий килим - це ні горизонтально, ні вертикально симетрично. Ось приклад недосконалого килима:

[

["a", "a", "a", "a"],

["a", "a", "a", "a"],

["a", "a", "b", "b"]

]

Далі - горизонтально симетричний килим. Ви можете "скласти" килим через гіпотетичну вісь x, і обидві сторони були б однаковими. Горизонтально симетричний килим не є вертикально симетричним (інакше цей килимок був би класифікований як ідеальний).

[

["c", "a", "a", "a"],

["b", "b", "b", "b"],

["c", "a", "a", "a"]

]

Далі - вертикально симетричний килим. Ви можете "скласти" килим через гіпотетичну вісь y, і обидві сторони були б однаковими. Вертикально симетричний не є горизонтально симетричним (інакше цей килимок був би класифікований як ідеальний).

[

["a", "b", "a"],

["b", "b", "b"],

["a", "b", "a"],

["a", "b", "a"]

]

Нарешті, ідеальний килим - це вертикально і горизонтально симетричний. Тобто, складені або по довжині, або по ширині дадуть дві однакові частини.

[

["a", "b", "b", "a"],

["b", "b", "b", "b"],

["a", "b", "b", "a"]

]

Давши килим розміром m x n, визначте, чи він недосконалий, горизонтально симетричний, вертикально симетричний чи ідеальний. Килимки представлені за допомогою двовимірного масиву.

Приклад:

classifyRug([

["a", "a"],

["a", "a"]

]) ➞ "perfect"

classifyRug([

["a", "a", "b"],

["a", "a", "a"],

["b", "a", "a"]

]) ➞ "imperfect"

classifyRug([

["b", "a"],

["b", "a"]

]) ➞ "horizontally symmetric"

classifyRug([

["a", "a"],

["b", "b"]

]) ➞ "vertically symmetric"

10 Наполегливість

Аддитивна стійкість цілого числа, n - це кількість разів, яку вам доведеться замінити n сумою її цифр, поки n не стане одноцифровим цілим числом.

Мультиплікативна стійкість цілого числа, n - це кількість разів, коли вам доведеться замінити n на добуток його цифр, поки n не стане одноцифровим цілим числом.

Створіть дві функції, які беруть ціле число в якості аргументу і:

Поверніть її адитивну стійкість.

Поверніть його мультиплікативну стійкість.

Приклади:

additivePersistence(1679583) ➞ 3

// 1 + 6 + 7 + 9 + 5 + 8 + 3 = 39

// 3 + 9 = 12

// 1 + 2 = 3

// It takes 3 iterations to reach a single-digit number.

additivePersistence(123456) ➞ 2

// 1 + 2 + 3 + 4 + 5 + 6 = 21

// 2 + 1 = 3

additivePersistence(6) ➞ 0

// Because 6 is already a single-digit integer.

multiplicativePersistence(77) ➞ 4

// 7 x 7 = 49

// 4 x 9 = 36

// 3 x 6 = 18

// 1 x 8 = 8

// It takes 4 iterations to reach a single-digit number.

multiplicativePersistence(123456) ➞ 2

// 1 x 2 x 3 x 4 x 5 x 6 = 720

// 7 x 2 x 0 = 0

multiplicativePersistence(4) ➞ 0

// Because 4 is already a single-digit integer.

11 Знайдіть найчастіший елемент у масиві

Створіть функцію, яка займає масив і повертає елемент який зустрічається найчастіше.

Приклад:

findFrequent([3, 7, 3]) ➞ 3

findFrequent([null, "hello", true, null]) ➞ null

findFrequent([false, "up", "down", "left", "right", true, false]) ➞ false

12 Видаліть останню голосну

Напишіть функцію, яка видаляє останню голосну в кожному слові речення.

Функція повинна працювати з англійським алфавітом.

Голосні - a, e, i, o, u

Всі букви будуть в lowercase

Приклад:

removeLastVowel("Those who dare to fail miserably can achieve greatly.")

➞ "Thos wh dar t fal miserbly cn achiev gretly."

removeLastVowel("Love is a serious mental disease.")

➞ "Lov s serios mentl diseas"

removeLastVowel("Get busy living or get busy dying.")

➞ "Gt bsy livng r gt bsy dyng"

--