**5/6**

1. Создайте функцию, которая возвращает true, если две строки имеют один и тот же буквенный шаблон, и false в противном случае.

Пример:

sameLetterPattern("ABAB", "CDCD") ➞ true

sameLetterPattern("ABCBA", "BCDCB") ➞ true

sameLetterPattern("FFGG", "CDCD") ➞ false

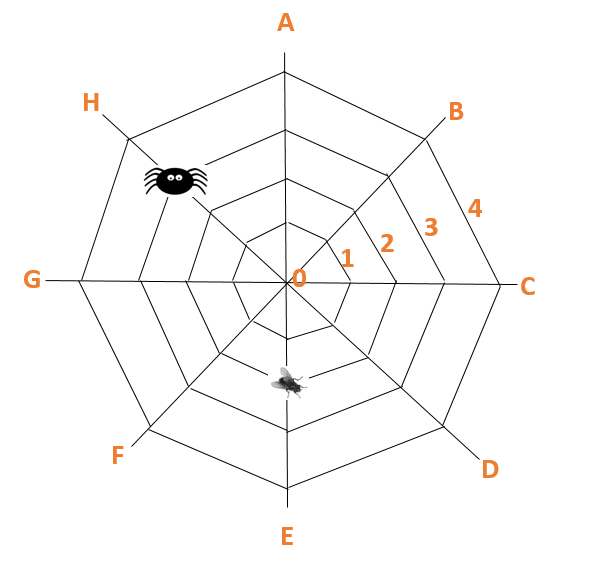
sameLetterPattern("FFFF", "ABCD") ➞ false

1. Паутина определяется кольцами, пронумерованными от 0 до 4 от центра, и радиалами, помеченными по часовой стрелке сверху как A-H.

Создайте функцию, которая принимает координаты паука и мухи и возвращает кратчайший путь для паука, чтобы добраться до мухи.

Стоит отметить, что кратчайший путь должен быть рассчитан "геометрически", а не путем подсчета количества точек, через которые проходит этот путь.

* Угол между каждой парой радиалов одинаков (45 градусов).
* Расстояние между каждой парой колец всегда одинаково (скажем, "x").



На приведенном выше рисунке координаты паука - "H3", а координаты мухи - "E2". Паук будет следовать по кратчайшему пути "H3-H2-H1-A0-E1-E2", чтобы добраться до мухи.

Пример:

spiderVsFly("H3", "E2") ➞ "H3-H2-H1-A0-E1-E2"

spiderVsFly("A4", "B2") ➞ "A4-A3-A2-B2"

spiderVsFly("A4", "C2") ➞ "A4-A3-A2-B2-C2"

1. Создайте функцию, которая будет рекурсивно подсчитывать количество цифр числа. Преобразование числа в строку не допускается, поэтому подход является рекурсивным.

digitsCount(4666) ➞ 4

digitsCount(544) ➞ 3

digitsCount(121317) ➞ 6

digitsCount(0) ➞ 1

digitsCount(12345) ➞ 5

digitsCount(1289396387328L) ➞ 13

1. Игроки пытаются набрать очки, формируя слова, используя буквы из 6-буквенного скремблированного слова. Они выигрывают раунд, если им удается успешно расшифровать слово из 6 букв.

Создайте функцию, которая принимает в массив уже угаданных слов расшифрованное 6-буквенное слово и возвращает общее количество очков, набранных игроком в определенном раунде, используя следующую рубрику:

3-буквенные слова – это 1 очко

4-буквенные слова – это 2 очка

5-буквенные слова – это 3 очка

6-буквенные слова – это 4 очка + 50 пт бонуса (за расшифровку слова)

Помните, что недопустимые слова (слова, которые не могут быть сформированы из 6-буквенных расшифрованных слов) считаются 0 очками.

Пример:

totalPoints(["cat", "create", "sat"], "caster") ➞ 2

// Since "create" is an invalid word.

totalPoints(["trance", "recant"], "recant") ➞ 108

// Since "trance" and "recant" score 54 pts each.

totalPoints(["dote", "dotes", "toes", "set", "dot", "dots", "sted"], "tossed") ➞ 13

// Since 2 + 3 + 2 + 1 + 1 + 2 + 2 = 13

Примечание:

- Если 6-буквенное слово имеет несколько анаграмм, считайте каждую 6-буквенную расшифровку дополнительными 54 очками. Например, если слово arches, а игрок угадал arches и chaser, добавьте 108 очков для обоих слов.

- Вы можете играть в Текстовый Твист здесь: <http://text-twist2.com>

1. Создайте функцию, которая получает каждую пару чисел из массива, который суммирует до восьми, и возвращает его как массив пар (отсортированный по возрастанию).

Пример:

sumsUp([1, 2, 3, 4, 5]) ➞ [[3, 5]]

sumsUp([1, 2, 3, 7, 9]) ➞ [[1, 7]]

sumsUp([10, 9, 7, 2, 8]) ➞ []

sumsUp([1, 6, 5, 4, 8, 2, 3, 7]) ➞ [[2, 6], [3, 5], [1, 7]]

// [6, 2] first to complete the cycle (to sum up to 8)

// [5, 3] follows

// [1, 7] lastly

// the pair that completes the cycle is always found on the left

// [2, 6], [3, 5], [1, 7] sorted according to cycle completeness, then pair-wise.

1. Какой процент вы можете набрать на тесте, который в одиночку снижает средний балл по классу на 5%? Учитывая массив оценок ваших одноклассников, создайте функцию, которая возвращает ответ. Округлите до ближайшего процента.

Пример:

takeDownAverage(["95%", "83%", "90%", "87%", "88%", "93%"]) ➞ "54%"

takeDownAverage(["10%"]) ➞ "0%"

takeDownAverage(["53%", "79%"]) ➞ "51%"

1. Создайте функцию, которая будет шифровать и дешифровать сообщения с использованием шифра Цезаря. Шифр Цезаря – это метод шифрования, в котором каждая буква в сообщении сдвигается на фиксированное количество позиций в алфавите. Например, если сдвиг составляет 3 позиции, то буква 'A' будет зашифрована как 'D', 'B' как 'E' и так далее.

Функция должна выполнять следующие действия:

1. Определять режим работы: шифрование или дешифрование сообщения.

2. Если пользователь хочет зашифровать сообщение, программа должна запросить само сообщение и сдвиг, на который нужно зашифровать текст.

3. Если пользователь хочет дешифровать сообщение, программа должна запросить зашифрованное сообщение и сдвиг, чтобы расшифровать его.

4. Обрабатывать сообщения только в верхнем регистре и оставлять другие символы (пробелы, цифры и специальные символы) без изменений.

Пример:

caesarCipher("encode", "hello world", 3) ➞ " KHOOR ZRUOG"

caesarCipher(["decode", "almost last task!", 4]) ➞ "EPQSWX PEWX XEWO!"

1. Создайте метод для рекурсивного вычисления количества различных способов как можно разместить k элементов из множества из n элементов без повторений. Это задача комбинаторики, и она часто используется в комбинаторных оптимизациях, таких как размещение задач на стеллажах или распределение ресурсов.

Метод принимает два параметра, где n - количество элементов в множестве, а k - количество элементов, которые нужно разместить (n >= k) и рассчитывает количество размещений по формуле размещений без повторений: n! / (n - k)!

Пример:

setSetup(5, 3) ➞ 60

setSetup(7, 3) ➞ 210

1. В этой задаче цель состоит в том, чтобы вычислить, сколько времени сейчас в двух разных городах. Вам дается строка cityA и связанная с ней строка timestamp (time in cityA) с датой, отформатированной в полной нотации США, как в этом примере:

"July 21, 1983 23:01"

Вы должны вернуть новую метку времени с датой и соответствующим временем в cityB, отформатированную как в этом примере:

"1983-7-22 23:01"

Список данных городов и их смещения по Гринвичу (среднее время по Гринвичу) приведены в таблице ниже.

| **GMT** | **City** |
| --- | --- |
| - 08:00 | Los Angeles |
| - 05:00 | New York |
| - 04:30 | Caracas |
| - 03:00 | Buenos Aires |
| 00:00 | London |
| + 01:00 | Rome |
| + 03:00 | Moscow |
| + 03:30 | Tehran |
| + 05:30 | New Delhi |
| + 08:00 | Beijing |
| + 10:00 | Canberra |

Пример:

timeDifference("Los Angeles", "April 1, 2011 23:23", "Canberra") ➞ "2011-4-2 17:23"

// Can be a new day.

timeDifference("London", "July 31, 1983 23:01", "Rome") ➞ "1983-8-1 00:01"

// Can be a new month.

timeDifference("New York", "December 31, 1970 13:40", "Beijing") ➞ "1971-1-1 02:40"

// Can be a new year.

Примечание:

- Обратите внимание на часы и минуты, ведущий 0 необходим в возвращаемой метке времени, когда они представляют собой одну цифру.

- Обратите внимание на города с получасовыми смещениями.

1. Новое число – это число, которое не является перестановкой любого меньшего числа. 869 – это не новое число, потому что это просто перестановка меньших чисел, 689 и 698. 509 – это новое число, потому что оно не может быть образовано перестановкой любого меньшего числа (ведущие нули не допускаются).

Напишите функцию, которая принимает неотрицательное целое число и возвращает true, если целое число является новым числом, и false, если это не так.

Пример:

isNew(3) ➞ true

isNew(30) ➞ true

isNew(321) ➞ false

isNew(123) ➞ true