**1/5**

1. При регистрации аккаунта предъявляются требования к логину и паролю. Напишите программу, проверяющую введенный пользователем логин, который должен состоять минимум из 7 символов, а также пароль, который должен содержать в себе цифру и не включать в себя символ «@».

Выводится одна строка: если все условия выполнены, то выводится «OK»; если длина логина недостаточна, то выводится «Некорректный логин»; если логин корректный, но в пароле отсутствует цифра или присутствует символ «@», то выводится «Некорректный пароль».

Пример:

Ввод:

admin

Вывод:

Некорректный логин

Ввод:

administrator

password1

Вывод:

ОК

1. Напишите программу, рассчитывающую стоимость сообщения, которое отправляется телеграммой, где каждый символ стоит 40 копеек.

Пример

Ввод:

Привет! Как дела?

Вывод:

6 р. 80 коп.

1. Напишите программу, генерирующую новый пароль на основе старого трехзначного пароля по следующим правилам:

находится сумма цифр, стоящих в двух старших разрядах (сотни и десятки); находится сумма цифр, стоящих в двух младших разрядах (десятки и единицы). Найденные две суммы записываются друг за другом в порядке убывания, формируя новый пароль.

Например, было введено число 167. Строим сумму старших разрядов – 1 + 6 = 7, строим сумму младших разрядов – 6 + 7 = 13. Полученные две суммы 7 и 13 записываем друг за другом в порядке не возрастания, те 137. Искомое число – 137.

1. Коля, Федя и Толя спорят кто из них должен стоять первый на построении на уроке физкультуры. Напишите программу, упорядочивающую учеников по росту. На вход подаются три строки с ростом, на выход – три упорядоченные строки.

Пример

Ввод:

132

144

126

Вывод:

144

132

126

1. В магазине действует скидка 15% на товары дороже 500 рублей. Напишите программу, частично имитирующую работу кассового аппарата: последовательно вводятся цены (действительные числа), последнее число – отрицательное – сигнал остановки; нужно вывести сначала общую стоимость товаров, затем размер скидки, затем итоговую цену с учетом скидки.

Пример

Ввод:

218.5

47

836

Вывод:

1101.5

125.4

976.1

1. Во многих задачах, связанных с компьютерами, особенно близких к аппаратной части, важную роль играют числа, являющиеся степенями двойки: 1, 2, 4, 8 и так далее. Напишите программу, которая проверяет, является ли введённое натуральное число степенью двойки. Если да, то выводится сама эта степень; если нет, выводится «НЕТ».

Пример

Ввод:

302231454903657293676544

Вывод:

78

1. Валера решил познать азы живописи и, как все начинающие художники, начал с рисования простых фигур в стиле кубизма. Помогите Валере написать программу построения прямоугольника n х m, состоящего из символов, введенных пользователем. Фигура должна быть пустой, а не заполненной. То есть она должна состоять только из контура.

На вход подаются два числа, каждое в отдельной строке — высота и ширина прямоугольника; на третьей строке символ, используемый для рисования контуров.

Пример

Ввод:

4

5

@

Вывод:

@@@@@

@ @

@ @

@@@@@

1. Напишите программу, имитирующую работу биржевого робота для автоматической торговли акциями.

Последовательно на отдельных строках вводится цена акций в первый, второй и последующие дни, ноль — сигнал остановки. В какой-то момент цена относительно предыдущего дня начинает расти. Мы покупаем акции в первый день, когда их цена превышает цену в предыдущий день. После этого в какой-то момент цена акций может начать уменьшаться. Мы продаём акции в первый же день, как только их цена становится меньше цены в предыдущий день. Возможно, после этого цены как-то ещё меняются.

Гарантируется, что среди введенных цен точно будет день, когда цена начнет расти, а после день, когда цена начнет падать. После продажи акций робот больше не участвует в торгах на бирже.

Программа должна вывести цену акций, по которой мы их купили, цену, по которой продали, и выгоду с каждой акции (возможно, отрицательную).

Пример

Ввод:

32

30

31

34

38

37

39

0

Вывод:

31 37 6

1. Напишите программу, которая заполняет массив из N элементов случайными целыми числами в диапазоне [A, B] и определяет количество элементов этого массива, у которых вторая цифра в десятичной записи (число десятков) – чётная. Размер массива и диапазон случайных чисел указывает пользователь.

В результате работы программы в первой строке должны быть выведены N элементов построенного массива, разделенные пробелами, а во второй строке – число элементов этого массива, у которых вторая цифра в десятичной записи (число десятков) – чётная.

Пример

Ввод:

100

150

5

Вывод:

100 101 143 110 113

3

1. Напишите программу, которая строит случайную перестановку чисел от 1 до N – массив из N натуральных чисел, в котором есть все числа от 1 до N; параметр N вводит пользователь. Программа должна вывести в одну строку элементы полученного случайного массива, разделив их пробелами. В массиве должны присутствовать все натуральные числа от 1 до N.

Пример

Ввод:

10

Вывод:

9 2 7 5 6 10 8 4 3 1

**2/5**

1. Вы подсчитываете очки за баскетбольный матч, учитывая количество забитых 2-х и 3-х очков, находите окончательные очки для команды и возвращаете это значение.

Пример:

points(13, 12) ➞ 62

1. Создайте функцию, которая возвращает true, если целое число равномерно делится на 5, и false в противном случае.

Пример:

divisibleByFive(5) ➞ true

divisibleByFive(37) ➞ false

1. У меня есть ведро с большим количеством темно-синей краски, и я хотел бы покрасить как можно больше стен. Создайте функцию, которая возвращает количество полных стен, которые я могу покрасить, прежде чем мне нужно будет отправиться в магазины, чтобы купить еще.

n - это количество квадратных метров, которые я могу нарисовать.

w и h-это ширина и высота одной стены в метрах.

Пример:

howManyWalls(54, 1, 43) ➞ 1

howManyWalls(46, 5, 4) ➞ 2

Примечание:

- Не считайте стену, если я не успею закончить покраску до того, как у меня закончится краска.

- Все стены будут иметь одинаковые размеры.

- Все числа будут целыми положительными.

1. Создайте функцию, которая принимает три аргумента prob, prize, pay и возвращает true, если prob \* prize > pay; в противном случае возвращает false.

Чтобы проиллюстрировать:

profitableGamble(0.2, 50, 9)

... должно давать true, так как чистая прибыль равна 1 (0.2 \* 50 - 9), и 1 > 0.

Пример:

profitableGamble(0.2, 50, 9) ➞ true

profitableGamble(0.9, 1, 2) ➞ false

profitableGamble(0.9, 3, 2) ➞ true

1. Создайте функцию, которая будет работать как оператор модуля % без использования оператора модуля. Оператор модуля-это способ определения остатка операции деления. Вместо того чтобы возвращать результат деления, операция по модулю возвращает остаток целого числа.

Пример:

mod(5, 2) ➞ 1

mod(218, 5) ➞ 3

mod(6, 3) ➞ 0

1. Лёша шел по прямой улице, по обеим сторонам которой стояло ровно n одинаковых домов. Номера домов на улице выглядят так:

1 | | 6

3 | | 4

5 | | 2

Он заметил, что четные дома увеличиваются справа, а нечетные уменьшаются слева.

Создайте функцию, которая принимает номер дома и длину улицы n и возвращает номер дома на противоположной стороне.

Пример:

oppositeHouse(1, 3) ➞ 6

oppositeHouse(2, 3) ➞ 5

oppositeHouse(3, 5) ➞ 8

oppositeHouse(5, 46) ➞ 88

1. Создайте функцию, которая принимает массив и возвращает разницу между наибольшим и наименьшим числами.

Пример:

differenceMaxMin([10, 4, 1, 4, -10, -50, 32, 21]) ➞ 82

// Smallest number is -50, biggest is 32.

differenceMaxMin([44, 32, 86, 19]) ➞ 67

// Smallest number is 19, biggest is 86.

1. Создайте функцию, которая принимает три целочисленных аргумента (a, b, c) и возвращает количество целых чисел, имеющих одинаковое значение.

Пример:

equal(3, 4, 3) ➞ 2

equal(1, 1, 1) ➞ 3

equal(3, 4, 1) ➞ 0

1. Создайте функцию, которая принимает строку, проверяет, имеет ли она одинаковое количество x и o и возвращает либо true, либо false.

Правила:

- Возвращает логическое значение (true или false).

- Верните true, если количество x и o одинаковы.

- Верните false, если они не одинаковы.

- Строка может содержать любой символ.

- Если "x" и "o" отсутствуют в строке, верните true.

Пример:

getXO("ooxx") ➞ true

getXO("xooxx") ➞ false

getXO("ooxXm") ➞ true

// Case insensitive.

getXO("zpzpzpp") ➞ true

// Returns true if no x and o.

getXO("zzoo") ➞ false

1. Напишите функцию, которая находит слово "бомба" в данной строке. Ответьте "ПРИГНИСЬ!", если найдешь, в противном случае:"Расслабься, бомбы нет".

Пример:

bomb("There is a bomb.") ➞ "DUCK!"

bomb("Hey, did you think there is a BOMB?") ➞ "DUCK!"

bomb("This goes boom!!!") ➞ "Relax, there's no bomb."

Примечание:

Строка "бомба" может появляться в разных случаях символов (например, в верхнем, нижнем регистре, смешанном).

**3/5**

1. Создайте функцию, имитирующую игру "камень, ножницы, бумага". Функция принимает входные данные обоих игроков (камень, ножницы или бумага), первый параметр от первого игрока, второй от второго игрока. Функция возвращает результат как таковой:

"Игрок 1 выигрывает"

"Игрок 2 выигрывает"

"НИЧЬЯ" (если оба входа одинаковы)

Правила игры камень, ножницы, бумага, если не известны:

Оба игрока должны сказать одновременно "камень", "бумага" или "ножницы".

Камень бьет ножницы, бумага бьет камень, ножницы бьют бумагу.

Пример:

rps("rock", "paper") ➞ "Player 2 wins"

rps("paper", "rock") ➞ "Player 1 wins"

rps("paper", "scissors") ➞ "Player 2 wins"

rps("scissors", "scissors") ➞ "TIE"

rps("scissors", "paper") ➞ "Player 1 wins"

1. Идет великая война между четными и нечетными числами. Многие уже погибли в этой войне, и ваша задача-положить этому конец. Вы должны определить, какая группа суммируется больше: четная или нечетная. Выигрывает большая группа.

Создайте функцию, которая берет массив целых чисел, суммирует четные и нечетные числа отдельно, а затем возвращает разницу между суммой четных и нечетных чисел.

Пример:

warOfNumbers([2, 8, 7, 5]) ➞ 2

// 2 + 8 = 10

// 7 + 5 = 12

// 12 is larger than 10

// So we return 12 - 10 = 2

warOfNumbers([12, 90, 75]) ➞ 27

warOfNumbers([5, 9, 45, 6, 2, 7, 34, 8, 6, 90, 5, 243]) ➞ 168

1. Учитывая строку, создайте функцию для обратного обращения. Все буквы в нижнем регистре должны быть прописными, и наоборот.

Пример:

reverseCase("Happy Birthday") ➞ "hAPPY bIRTHDAY"

reverseCase("MANY THANKS") ➞ "many thanks"

reverseCase("sPoNtAnEoUs") ➞ "SpOnTaNeOuS"

1. Создайте функцию, которая принимает строку из одного слова и выполняет следующие действия:

Конкатенирует inator до конца, если слово заканчивается согласным, в противном случае вместо него конкатенирует -inator

Добавляет длину слова исходного слова в конец, снабженный '000'.

Пример:

inatorInator("Shrink") ➞ "Shrinkinator 6000"

inatorInator("Doom") ➞ "Doominator 4000"

inatorInator("EvilClone") ➞ "EvilClone-inator 9000"

1. Напишите функцию, которая принимает три измерения кирпича: высоту(a), ширину(b) и глубину(c) и возвращает true, если этот кирпич может поместиться в отверстие с шириной(w) и высотой(h).

Пример:

doesBrickFit(1, 1, 1, 1, 1) ➞ true

doesBrickFit(1, 2, 1, 1, 1) ➞ true

doesBrickFit(1, 2, 2, 1, 1) ➞ false

Примечание:

- Вы можете повернуть кирпич любой стороной к отверстию.

- Мы предполагаем, что кирпич подходит, если его размеры равны размерам отверстия (то есть размер кирпича должен быть меньше или равен размеру отверстия, а не строго меньше).

- Нельзя класть кирпич под неортогональным углом.

1. Напишите функцию, которая принимает топливо (литры), расход топлива (литры/100 км), пассажиров, кондиционер (логическое значение) и возвращает максимальное расстояние, которое может проехать автомобиль.

топливо-это количество литров топлива в топливном баке.

Расход топлива-это базовый расход топлива на 100 км (только с водителем внутри).

Каждый дополнительный пассажир увеличивает базовый расход топлива на 5%.

Если кондиционер включен, то его общий (не базовый) расход топлива увеличивается на 10%.

Пример:

totalDistance(70.0, 7.0, 0, false) ➞ 1000.0

totalDistance(36.1, 8.6, 3, true) ➞ 331.83

totalDistance(55.5, 5.5, 5, false) ➞ 807.3

1. Создайте функцию, которая принимает массив чисел и возвращает среднее значение (average) всех этих чисел.

Пример:

mean([1, 0, 4, 5, 2, 4, 1, 2, 3, 3, 3]) ➞ 2.55

mean([2, 3, 2, 3]) ➞ 2.50

mean([3, 3, 3, 3, 3]) ➞ 3.00

1. Создайте функцию, которая принимает число в качестве входных данных и возвращает true, если сумма его цифр имеет ту же четность, что и все число. В противном случае верните false.

Пример:

parityAnalysis(243) ➞ true

// 243 is odd and so is 9 (2 + 4 + 3)

parityAnalysis(12) ➞ false

// 12 is even but 3 is odd (1 + 2)

parityAnalysis(3) ➞ true

// 3 is odd and 3 is odd and 3 is odd (3)

1. Создайте функцию, которая принимает массив чисел и возвращает "Бум!", если в массиве появляется цифра 7. В противном случае верните "в массиве нет 7".

Пример:

sevenBoom([1, 2, 3, 4, 5, 6, 7]) ➞ "Boom!"

// 7 contains the number seven.

sevenBoom([8, 6, 33, 100]) ➞ "there is no 7 in the array"

// None of the items contain 7 within them.

sevenBoom([2, 55, 60, 97, 86]) ➞ "Boom!"

// 97 contains the number seven.

1. Создайте функцию, которая определяет, могут ли элементы в массиве быть переупорядочены, чтобы сформировать последовательный список чисел, где каждое число появляется ровно один раз.

cons([5, 1, 4, 3, 2]) ➞ true

// Can be re-arranged to form [1, 2, 3, 4, 5]

cons([5, 1, 4, 3, 2, 8]) ➞ false

cons([5, 6, 7, 8, 9, 9]) ➞ false

// 9 appears twice

**4/5**

1. lPaeesh le pemu mnxit ehess rtnisg! О, извините, это должно было быть: Пожалуйста, помогите мне распутать эти строки!

Каким-то образом все строки перепутались, каждая пара символов поменялась местами. Помоги отменить это, чтобы снова понять строки.

Пример:

unmix("123456") ➞ "214365"

unmix("hTsii s aimex dpus rtni.g") ➞ "This is a mixed up string."

unmix("badce") ➞ "abcde"

1. Создать функцию, которая преобразует предложения, заканчивающиеся несколькими вопросительными знаками ? или восклицательными знаками ! в предложение, заканчивающееся только одним, без изменения пунктуации в середине предложений.

Пример:

noYelling("What went wrong?????????") ➞ "What went wrong?"

noYelling("Oh my goodness!!!") ➞ "Oh my goodness!"

noYelling("I just!!! can!!! not!!! believe!!! it!!!") ➞ "I just!!! can!!! not!!! believe!!! it!"

// Only change repeating punctuation at the end of the sentence.

noYelling("Oh my goodness!") ➞ "Oh my goodness!"

// Do not change sentences where there exists only one or zero exclamation marks/question marks.

noYelling("I just cannot believe it.") ➞ "I just cannot believe it."

Примечание:

- Меняйте только окончательную пунктуацию - оставляйте восклицательные или вопросительные знаки в середине предложения неизменными (см. Третий пример).

- Не беспокойтесь о смешанной пунктуации (нет случаев, которые заканчиваются чем-то вроде ?!??!).

- Сохраняйте предложения, в которых нет вопросительных/восклицательных знаков, одинаковыми.

1. Создайте функцию, которая заменяет все x в строке следующими способами:

Замените все x на "cks", ЕСЛИ ТОЛЬКО:

Слово начинается с "x", поэтому замените его на "z".

Слово-это просто буква "х", поэтому замените ее на " cks ".

Пример:

xPronounce("Inside the box was a xylophone") ➞ "Inside the bocks was a zylophone"

xPronounce("The x ray is excellent") ➞ "The ecks ray is eckscellent"

xPronounce("OMG x box unboxing video x D") ➞ "OMG ecks bocks unbocksing video ecks D"

Примечание:

- Все X являются строчными.

1. Учитывая массив целых чисел, верните наибольший разрыв между отсортированными элементами массива.

Например, рассмотрим массив:

[9, 4, 26, 26, 0, 0, 5, 20, 6, 25, 5]

... в котором после сортировки массив становится:

[0, 0, 4, 5, 5, 6, 9, 20, 25, 26, 26]

.. так что теперь мы видим, что самый большой разрыв в массиве находится между 9 и 20, что равно 11.

Пример:

largestGap([9, 4, 26, 26, 0, 0, 5, 20, 6, 25, 5]) ➞ 11

// After sorting: [0, 0, 4, 5, 5, 6, 9, 20, 25, 26, 26]

// Largest gap between 9 and 20 is 11

largestGap([14, 13, 7, 1, 4, 12, 3, 7, 7, 12, 11, 5, 7]) ➞ 4

// After sorting: [1, 3, 4, 5, 7, 7, 7, 7, 11, 12, 12, 13, 14]

// Largest gap between 7 and 11 is 4

largestGap([13, 3, 8, 5, 5, 2, 13, 6, 14, 2, 11, 4, 10, 8, 1, 9]) ➞ 2

// After sorting: [1, 2, 2, 3, 4, 5, 5, 6, 8, 8, 9, 10, 11, 13, 13, 14]

// Largest gap between 6 and 8 is 2

1. Это вызов обратного кодирования. Обычно вам дают явные указания о том, как создать функцию. Здесь вы должны сгенерировать свою собственную функцию, чтобы удовлетворить соотношение между входами и выходами.

Ваша задача состоит в том, чтобы создать функцию, которая при подаче входных данных ниже производит показанные примеры выходных данных.

Пример:

832 ➞ 594

51 ➞ 36

7977 ➞ 198

1 ➞ 0

665 ➞ 99

149 ➞ 0

1. Создайте функцию, которая принимает предложение в качестве входных данных и возвращает наиболее распространенную последнюю гласную всех слов в предложении в виде одной символьной строки.

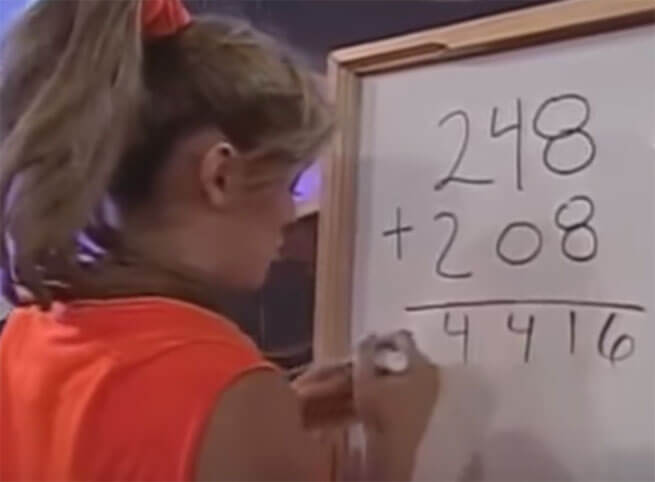
Пример:

commonLastVowel("Hello World!") ➞ "o"

commonLastVowel("Watch the characters dance!") ➞ "e"

commonLastVowel("OOI UUI EEI AAI") ➞ "i"

1. Для этой задачи забудьте, как сложить два числа вместе. Лучшее объяснение того, что нужно сделать для этой функции, - это этот мем:



Пример:

memeSum(26, 39) ➞ 515

// 2+3 = 5, 6+9 = 15

// 26 + 39 = 515

memeSum(122, 81) ➞ 1103

// 1+0 = 1, 2+8 = 10, 2+1 = 3

// 122 + 81 = 1103

memeSum(1222, 30277) ➞ 31499

1. Создайте функцию, которая удалит все повторяющиеся символы в слове, переданном этой функции. Не просто последовательные символы, а символы, повторяющиеся в любом месте строки.

Пример:

unrepeated("teshahset") ➞ "tesha"

unrepeated("hello") ➞ "helo"

unrepeated("aaaaa") ➞ "a"

unrepeated("WWE!!!") ➞ "WE!"

unrepeated("call 911") ➞ "cal 91"

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. 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. Последовательный прогон-это список соседних последовательных целых чисел. Этот список может быть как увеличивающимся, так и уменьшающимся. Создайте функцию, которая принимает массив чисел и возвращает длину самого длинного последовательного запуска.

longestRun([1, 2, 3, 5, 6, 7, 8, 9]) ➞ 5

// Two consecutive runs: [1, 2, 3] and [5, 6, 7, 8, 9] (longest).

Пример:

longestRun([1, 2, 3, 10, 11, 15]) ➞ 3

// Longest consecutive-run: [1, 2, 3].

longestRun([5, 4, 2, 1]) ➞ 2

// Longest consecutive-run: [5, 4] and [2, 1].

longestRun([3, 5, 7, 10, 15]) ➞ 1

**5/5**

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. Учитывая предложение с числами, представляющими расположение слова, встроенного в каждое слово, верните отсортированное предложение.

rearrange("Tesh3 th5e 1I lov2e way6 she7 j4ust i8s.") ➞ "I love Tesh just the way she is."

rearrange("the4 t3o man5 Happ1iest of6 no7 birt2hday steel8!") ➞ "Happiest birthday to the man of no steel!"

rearrange("is2 Thi1s T4est 3a") ➞ "This is a Test"

rearrange("4of Fo1r pe6ople g3ood th5e the2") ➞ "For the good of the people"

rearrange(" ") ➞ ""

1. Напишите функцию, которая делает первое число как можно больше, меняя его цифры на цифры во втором числе.

maxPossible(9328, 456) ➞ 9658

// 9658 is the largest possible number built from swaps from 456.

// 3 replaced with 6 and 2 replaced with 5.

Пример:

maxPossible(523, 76) ➞ 763

maxPossible(9132, 5564) ➞ 9655

maxPossible(8732, 91255) ➞ 9755

Примечание:

- Каждая цифра во втором числе может быть использована только один раз.

- Можно использовать ноль для всех цифр второго числа.

1. Создайте метод, который принимает массив arr и число n и возвращает массив из двух целых чисел из arr, произведение которых равно числу n следующего вида:

[value\_at\_lower\_index, value\_at\_higher\_index]

Убедитесь, что вы возвращаете массив из двух целых чисел, который точно делит n (произведение n) и что индексы совпадают с указанным выше форматом. Таким образом, сортировка значений основана на восходящих индексах.

Пример:

twoProduct([1, 2, 3, 9, 4, 5, 15, 3], 45) ➞ [9, 5]

// at index 5 which has the value 5 is a full match

// to the value at index 3 which is 9

// the closest gap between indices that equates

// to the product which is 45

twoProduct([1, 2, 3, 9, 4, 15, 3, 5], 45) ➞ [3, 15]

// at index 5 which has the value 15 is a full match

// to the value at index 2 which is 3

// the closest gap between indices that equates

// to the product which is 45

twoProduct([1, 2, -1, 4, 5, 6, 10, 7], 20) ➞ [4, 5]

// at index 4 which has the value 5 is a full match

// to the value at index 3 which is 4

// a full match can only be achieved if you've found the multiplier at an

// index lower than the current index, thus, 5 (@ index 4) has a pair lower

// than its index which is the value 4 (@ index 3), so, 5\*4 = 20, in which 5

// has a higher index (4) than 4 which has an index value of 3

twoProduct([1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 10) ➞ [2, 5]

twoProduct([100, 12, 4, 1, 2], 15) ➞ []

Примечание:

- Дубликатов не будет.

- Возвращает пустой массив, если совпадение не найдено.

- Всегда считайте, что пара рассматриваемого элемента (текущий элемент, на который указывали во время поиска) находится слева от него.

- Массив может иметь несколько решений (произведений n), поэтому возвращайте первое найденное полное совпадение (как описано выше).

1. Создайте рекурсивную функцию, которая проверяет, является ли число точной верхней границей факториала n. Если это так, верните массив точной факториальной границы и n, или иначе, пустой массив.

Пример:

isExact(6) ➞ [6, 3]

isExact(24) ➞ [24, 4]

isExact(125) ➞ []

isExact(720) ➞ [720, 6]

isExact(1024) ➞ []

isExact(40320) ➞ [40320, 8]

Примечание:

- Ожидается, что вы решите эту задачу с помощью рекурсии.

1. Деление на дробь часто приводит к бесконечно повторяющейся десятичной дроби.

1/3=.3333333... 1/7=.142857142857...

Создайте функцию, которая принимает десятичную дробь в строковой форме с повторяющейся частью в круглых скобках и возвращает эквивалентную дробь в строковой форме и в наименьших членах.

Пример:

fractions("0.(6)") ➞ "2/3"

fractions("1.(1)") ➞ "10/9"

fractions("3.(142857)") ➞ "22/7"

fractions("0.19(2367)") ➞ "5343/27775"

fractions("0.1097(3)") ➞ "823/7500"

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. 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