

1. Квадратное уравнение $ax^2 + bx + c = 0$ имеет либо 0, либо 1, либо 2 различных решения для действительных значений x . учитывая a , b и c , вы должны вернуть число решений в уравнение.

Пример:

```
solutions(1, 0, -1) → 2
// x2 - 1 = 0 has two solutions (x = 1 and x = -1).

solutions(1, 0, 0) → 1
// x2 = 0 has one solution (x = 0).

solutions(1, 0, 1) → 0
// x2 + 1 = 0 has no solutions.
```

2. Напишите функцию, которая возвращает позицию второго вхождения "zip" в строку, или -1, если оно не происходит по крайней мере дважды. Ваш код должен быть достаточно общим, чтобы передать все возможные случаи, когда "zip" может произойти в строке.

Пример:

```
findZip("all zip files are zipped") → 18

findZip("all zip files are compressed") → -1
```

Примечание:

Верхний регистр "Zip" - это не то же самое, что нижний регистр "zip".

3. Создайте функцию, которая проверяет, является ли целое число совершенным числом или нет. Совершенное число - это число, которое можно записать как сумму его множителей, исключая само число.

Например, 6-это идеальное число, так как $1 + 2 + 3 = 6$, где 1, 2 и 3-Все коэффициенты 6. Точно так же 28-это совершенное число, так как $1 + 2 + 4 + 7 + 14 = 28$.

Пример:

```
checkPerfect(6) → true

checkPerfect(28) → true

checkPerfect(496) → true

checkPerfect(12) → false

checkPerfect(97) → false
```

4. Создайте функцию, которая принимает строку и возвращает новую строку с заменой ее первого и последнего символов, за исключением трех условий:

- Если длина строки меньше двух, верните "несовместимо".
- Если первый и последний символы совпадают, верните "два-это пара".

Пример:

```
flipEndChars("Cat, dog, and mouse.") → ".at, dog, and mouseC"

flipEndChars("ada") → "Two's a pair."
```

```
flipEndChars("Ada") → "adA"
```

```
flipEndChars("z") → "Incompatible."
```

5. Создайте функцию, которая определяет, является ли строка допустимым шестнадцатеричным кодом.

Шестнадцатеричный код должен начинаться с фунтового ключа # и иметь длину ровно 6 символов. Каждый символ должен быть цифрой от 0-9 или буквенным символом от A-F. все буквенные символы могут быть прописными или строчными.

Пример:

```
isValidHexCode("#CD5C5C") → true

isValidHexCode("#EAECEE") → true

isValidHexCode("#eaecee") → true

isValidHexCode("#CD5C58C") → false
// Length exceeds 6

isValidHexCode("#CD5C5Z") → false
// Not all alphabetic characters in A-F

isValidHexCode("#CD5C&C") → false
// Contains unacceptable character

isValidHexCode("CD5C5C") → false
// Missing #
```

6. Напишите функцию, которая возвращает true, если два массива имеют одинаковое количество уникальных элементов, и false в противном случае.

Для примера:

```
arr1 = [1, 3, 4, 4, 4]
arr2 = [2, 5, 7]
```

В arr1 число 4 появляется трижды, что означает, что оно содержит три уникальных элемента: [1, 3, 4]. Поскольку arr1 и arr2 содержат одинаковое количество уникальных элементов, этот пример вернет значение true.

Пример:

```
same([1, 3, 4, 4, 4], [2, 5, 7]) → true

same([9, 8, 7, 6], [4, 4, 3, 1]) → false

same([2], [3, 3, 3, 3, 3]) → true
```

7. Число Капрекара-это положительное целое число, которое после возведения в квадрат и разбиения на две лексикографические части равно сумме двух полученных новых чисел:

- Если количество цифр квадратного числа четное, то левая и правая части будут иметь одинаковую длину.
- Если количество цифр квадратного числа нечетно, то правая часть будет самой длинной половиной, а левая-самой маленькой или равной нулю, если количество цифр равно 1.

– Учитывая положительное целое число n , реализуйте функцию, которая возвращает `true`, если это число Капрекара, и `false`, если это не так.

Пример:

```
isKaprekar(3) → false
// n2 = "9"
// Left + Right = 0 + 9 = 9 → 9 !== 3

isKaprekar(5) → false
// n2 = "25"
// Left + Right = 2 + 5 = 7 → 7 !== 5

isKaprekar(297) → true
// n2 = "88209"
// Left + Right = 88 + 209 = 297 → 297 === 297
```

Примечание:

Тривиально, 0 и 1-Это числа Капрекара, являющиеся единственными двумя числами, равными их квадрату.

8. Напишите функцию, которая возвращает самую длинную последовательность последовательных нулей в двоичной строке.

Пример:

```
longestZero("01100001011000") → "0000"

longestZero("100100100") → "00"

longestZero("11111") → ""
```

9. Если задано целое число, создайте функцию, которая возвращает следующее простое число. Если число простое, верните само число.

Пример:

```
nextPrime(12) → 13

nextPrime(24) → 29

nextPrime(11) → 11
// 11 is a prime, so we return the number itself.
```

10. Учитывая три числа, x , y и z , определите, являются ли они ребрами прямоугольного треугольника.

Пример:

```
rightTriangle(3, 4, 5) → true
// This is the classic example of a "nice" right angled triangle.

rightTriangle(145, 105, 100) → true
// This is a less famous example.

rightTriangle(70, 130, 110) → false
// This isn't a right angled triangle.
```