

Задача 1

Постановка

Назовём строку хорошей, если в ней содержатся только буквы 'd', 'g', 'o' и предыдущая буква не совпадает со следующей.

Примеры хороших строк: 'dog', 'god', 'dogdog'

Примеры плохих строк: 'good', 'ooo', 'doc'

Даны целые числа n и k . Пусть есть список всех хороших строк длины n . Будем считать, что строки отсортированы в лексикографическом порядке.

Необходимо вернуть k -ую строку этого списка. Если в списке менее k хороших строк, то вернуть None.

Входные данные

В строке находятся целые числа n и k .

Выходные данные

Хорошая строка под номером k .

Пример 1

Входные данные	Выходные данные
1 3	'o'

Пример 2

Входные данные	Выходные данные
3 4	'dog'

Пример 3

Входные данные	Выходные данные
1 4	None

Задача 2

Постановка

Пусть дана строка. Необходимо переставить в ней буквы так, чтобы одинаковые буквы не стояли рядом. Перестановку нужно сделать в лексикографическом порядке. Если это невозможно, то верните None.

Входные данные

Строка.

Выходные данные

- если перестановка существует, то измененная строка.
- если не существует, то None.

Пример 1

Входные данные	Выходные данные
'abbc'	'abcb'

Пример 2

Входные данные	Выходные данные
'bbbc'	None

Задача 3

Постановка

Некоторая строка закодирована шифром. Чтобы расшифровать эту строку, существует некоторая машина, которая читает один символ за раз и выполняет следующие действия:

- если прочитанный символ это буква, то она записывается на ленту
- если прочитанный символ это цифра n , то вся записанная строка печатается ещё $n-1$ раз.

Для некоторой зашифрованной строки необходимо вывести k -ую буквы расшифрованной строки.

Входные данные

Строка и целое число k .

Выходные данные

k -ая буква расшифрованной строки.

Пример 1

Входные данные	Выходные данные
'cute2dog3' 10	'o'

Пример 2

Входные данные	Выходные данные
'no23' 5	'n'