Приветствуем!

В нашей компании убеждены, что хороший программист — это прежде всего личные качества, среди которых, на первом месте стоит сообразительность. Поэтому первым этапом хотим получить от вас решение нескольких довольно простых задачек на сообразительность. Прежде чем перейти к задачам, **небольшая вводная:**

1. Если вы успешно справитесь с заданием, на собеседовании будут дополнительные задачи на сообразительность в режиме реального времени, поэтому:

* пожалуйста, не пытайтесь найти решения в интернете и/или просить помочь знакомых
* не тратьте на решение задач слишком много времени — 2 часов более чем достаточно.

2. Для прохождения данного этапа не обязательно решить все задачи.

3. Ответы мы ожидаем получить в течение 3 дней с момента отправки данного письма. Но чем раньше вы их пришлете, тем лучше.

4. Если ничего не удастся решить, будем благодарны, если вы сообщите об этом.

**1 задание**

Есть 90 человек. Каждый из этих 90 человек относится к одной из двух категорий. Одни на все вопросы говорят правду, другие — всегда врут. У каждого ровно 1 любимое блюдо из списка (макароны/пельмени/вареники). Каждому задают три вопроса:

* *Ваше любимое блюдо — макароны?*
* *Ваше любимое блюдо — пельмени?*
* *Ваше любимое блюдо — вареники?*

Результаты получились такие:

* на первый вопрос утвердительно ответило 45 человек,
* на второй — 35,
* на третий — 30.

Сколько человек всегда говорят правду?

**2 задание**

Есть числа от трех до одиннадцати. Есть квадрат — 3 на 3 клетки:

а б в

г д е

ё ж з

*(Буквы ничего не значат, кроме того, что числа не повторяются)*

Можно ли расставить числа в клетки квадрата таким образом, что перемножения чисел в строках дает тот же результат, что и произведение чисел в столбцах с теми же номерами?

Если можно — расставьте, если нельзя — объясните почему.

**3 задание**

Ученые разработали новый материал неизвестной прочности. Они знают, что материал разбивается при падении с высоты от 1 метра до 5 000 метров. Но не знают, с какой именно высоты. Чтобы определить прочность, ученые поднимают предмет на некоторую высоту и сбрасывают его оттуда. Их задача — определить, начиная с какой именно высоты предмет начнет разбиваться.

Специальная платформа, с помощью которой они осуществляют эксперимент, скидывает предмет только с дискретных высот (1, 2, 3 ... 4999, 5000 метров — платформа не может скинуть предмет, например, с 2,5 метров. Точности в 1 метр ученым вполне достаточно). При падении с высоты "n" метров предмет уничтожается. Если же его сбрасывали с высоты ниже "n", то его можно использовать в повторных экспериментах.

Нужно АБСОЛЮТНО ТОЧНО найти ту высоту, начиная с которой предметы разрушаются. Сделать это нужно за МИНИМАЛЬНО возможное число экспериментов. У ученых при этом всего 2 предмета, но они абсолютно одинаковые. Каким образом этого можно достигнуть? Сколько экспериментов при этом максимально потребуется?

Также, чтобы понимать, что у вас есть минимальные навыки программирования, необходимо решить **несколько задач по программированию**.

Вы можете использовать для их решения любой комфортный вам язык. В качестве решения достаточно выслать текстовый файл с кодом написанных функции/методов.

**ВАЖНО! Укажите рядом с каждой задачей время, которое вы потратили на ее выполнение.**

1. Написать метод/функцию, который/которая на вход принимает массив городов. В качестве результата возвращает строку, где города разделены запятыми, а в конце стоит точка.

Пример:

«Москва, Санкт-Петербург, Воронеж.»

2. Написать метод/функцию, который/которая на вход принимает число (float), а на выходе получает число, округленное до пятерок.

Пример:

27 => 25, 27.8 => 30, 41.7 => 40.

3. Написать метод/функцию, который/которая на вход принимает число (int), а на выходе выдает слово “компьютер” в падеже, соответствующем указанному количеству. Например, «25 компьютеров», «41 компьютер», «1048 компьютеров».

4. Написать метод/функцию, который/которая на вход принимает целое число, а на выходе возвращает то, является ли число простым (не имеет делителей кроме 1 и самого себя).

5. Написать метод, который определяет, какие элементы присутствуют в двух экземплярах в каждом из массивов (= в двух и более, причем в каждом). На вход подаются два массива. На выходе массив с необходимыми совпадениями.

Пример:

[7, 17, 1, 9, 1, 17, 56, 56, 23], [56, 17, 17, 1, 23, 34, 23, 1, 8, 1]

На выходе [1, 17]