# Задача A: Рукопожатия

Ответ — это число сочетаний из $n$ элементов по 2. Оно обозначается как $C_n^2$, $\text{choose}(n, 2)$ или $\binom{n}{2}$ и равно $n \cdot (n-1)/2$.

Действительно, количество рукопожатий — это количество способов выбрать двух людей из $n$, причём порядок не важен: пара $(a, b)$ и пара $(b, a)$ считаются вместе только один раз. Есть $n$ способов выбрать первого участника собрания, а из оставшихся есть $n-1$ способ выбрать второго. После этого каждую пару участников мы посчитали ровно два раза.

Возможно и такое решение. Пронумеруем людей числами от 1 до $n$. Пусть наша пара имеет вид $(a, b)$, где $a < b$. Если $a = 1$, то есть $n-1$ способ выбрать $b$, если $a = 2$, то таких способов $n-2$, и так далее. Наконец, если $a = n-1$, то остался единственный способ выбрать $b$: это $b = n$. Значит, ответ равен $1+2+3+\ldots+(n-1)$.

Ещё одно решение, не требующее формул: двумя вложенными циклами от 1 до $n$ посчитать количество пар $(a, b)$, в которых $1 \leqslant a < b \leqslant n$.

# Задача B: Краткое имя

**Решение 1:** Посмотрим на каждую тройку идущих подряд букв в имени. Если в ней первая буква не совпадает со второй, но совпадает с третьей, выведем эту тройку букв и завершим работу. Если этого так и не произошло, выведем «`none`».

**Решение 2:** Переберём все возможные короткие имена. Рассмотрим все пары различных букв $(\alpha, \beta)$, всего таких пар $26 \cdot 25 = 650$. Для каждой пары построим строку $\alpha\beta\alpha$ и попробуем найти её как подстроку в имени. Как только мы нашли какое-то краткое имя, выведем его и завершим работу. Если этого так и не произошло, выведем «`none`».

Требуется аккуратность: программа должна работать корректно, даже если исходное имя содержит меньше трёх букв.

# Задача C: Тип треугольника

Сначала проверим, вырожденный ли наш треугольник $ABC$. Для этого можно, например, вычислить псевдоскалярное произведение векторов $AB$ и $AC$: $(x_B - x_A) \cdot (y_C - y_A) - (x_C - x_A) \cdot (y_B - y_A)$. Если оно равно нулю, точки $A$, $B$ и $C$ лежат на одной прямой.

В невырожденном случае вычислим квадраты длин сторон треугольника: $a^2 = (x_C - x_B)^2 + (y_C - y_B)^2$, $b^2 = (x_A - x_C)^2 + (y_A - y_C)^2$, $c^2 = (x_B - x_A)^2 + (y_B - y_A)^2$. Отсортируем их так, что $a^2 \leqslant b^2 \leqslant c^2$. Если теперь $a^2 + b^2 = c^2$, треугольник прямоугольный, если $a^2 + b^2 > c^2$, то он остроугольный, а если $a^2 + b^2 < c^2$, то он тупоугольный.

Во всех проверках выше для вычислений использовались только целые числа, что гарантирует корректность. Проверки можно сделать и по-другому, но при вычислениях с вещественными числами следует учитывать, что они представляются в компьютере лишь с некоторой точностью, так что равенства и неравенства следует проверять с учётом возможной погрешности вычислений.

Требуется аккуратность: в вырожденном случае какие-то из заданных точек могут совпадать.

# Задача D: Две страны

Заметим, что добавляемая клетка должна быть пустой, а одна из четырёх соседних клеток должна быть клеткой страны Алевтиния. Всё, что нужно сделать — проверить эти условия для каждой клетки. После нахождения первой же подходящей клетки поставим в неё букву «`A`» и прекратим поиск. После поиска — успешного или нет — выведем получившуюся карту.

Следует проявить аккуратность при проверках соседей на краях карты. Например, для этого можно хранить карту в массиве размера $(\text{rows} + 2) \times (\text{cols} + 2)$, а первый и последний столбец, а также первую и последнюю строку, заполнить каким-нибудь символом, не равным «A». Другой вариант — при проверке соседа сначала проверять, что клетка с такими координатами есть на карте.

Решение, которое просто берёт первую попавшуюся пустую клетку и объявляет её территорией Алевтинии, не набирает полный балл: один из контрпримеров показан справа.

```
ABBBB
AB..B
ABB.B
AAAAB
```

## Задача E: Сообщение на сервере

Решение состоит в том, чтобы передать сообщение без изменений, а ещё суметь показать, где в полученных $2n+2$ цифрах оно начинается.

**Решение 1:** Передадим следующее: $n-1$ единицу, ноль, затем сообщение, а далее любые $n$ символов. Поскольку из $3n$ переданных символов останется хотя бы $2n+2$, из первых $n$ переданных символов («111...110») будут переданы либо все, либо все кроме первой единицы, $\ldots$, либо только последние два («10»), но никак не меньше. Значит, полученное имеет вид «111...110<*message*>...», и в начале есть хотя бы одна единица. Пропустим все единицы и первый ноль. Следующие $n$ символов — это и есть сообщение.

**Решение 2:** Обозначим отдельные цифры исходного сообщения как $m_1 m_2 \ldots m_n$. В качестве пакета передадим следующее: $m_2 m_3 \ldots m_n 0 m_1 m_2 \ldots m_n 1 m_1 m_2 \ldots m_{n-1}$. Заметим, что в этом пакете символы на расстоянии $n+1$ совпадают всегда, кроме специально добавленных нуля и единицы. Также заметим, что мы обязательно получим эти ноль и единицу. Значит, чтобы восстановить сообщение, в полученном нужно найти два различных символа на расстоянии в $n+1$ символ — тогда $n$ символов между ними и будут исходным сообщением.

Заметим, что оба решения работали бы, даже если бы оставалось всего $2n+1$ символов.

## Разработчики комплекта задач:

Иван Казменко (gassafm@gmail.com)
Наталья Гинзбург (naagin@gmail.com)