

## Задача А. Смена стиля

Имя входного файла:            стандартный ввод  
Имя выходного файла:          стандартный вывод  
Ограничение по времени:      2 секунды  
Ограничение по памяти:        512 мегабайт

Желтому нужно выполнить задание по перезагрузке компьютера. У него есть специальная `reboot`-флешка с программой. Все переменные в этой программе написаны в стиле `CamelCase` или `camelCase`. Желтого это не устраивает, и он хочет чтобы все переменные были написаны в стиле `snake_case`.

Оба этих стиля используются для того, чтобы в строке без пробелов написать фразу из нескольких слов. В `camelCase` (`CamelCase`) для этого первая буква каждого слова из фразы, кроме возможно первого, делается заглавной, все остальные буквы остаются маленькими, и затем слова записываются друг за другом без пробелов. В `snake_case` все буквы слов оставляются маленькими и слова разделяются символом «`_`».

Помогите Желтому преобразовать несколько названий переменных из `camelCase` (`CamelCase`) в `snake_case`.

### Формат входных данных

В первой строке дано одно целое число  $n$  — количество названий переменных, для которых Желтый хочет поменять стиль ( $1 \leq n \leq 100$ ).

Далее даны  $n$  строк  $s_i$ , каждая из которых состоит из латинских букв — названия переменных ( $1 \leq |s_i| \leq 1000$ ).

### Формат выходных данных

Выведите название каждой переменной, преобразованное в `snake_case`

### Пример

стандартный ввод	стандартный вывод
4	<code>camel_case</code>
<code>camelCase</code>	<code>camel_case</code>
<code>CamelCase</code>	<code>to_be_or_not_to_be</code>
<code>toBeOrNotToBe</code>	<code>a_b_c_d_e</code>
<code>ABCDE</code>	

## Задача В. Результаты контеста

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Члены корабля решили поучаствовать в олимпиаде по программированию. Соревнование проходит по правилам ICPC.

Вам дана последовательность посылок участника в том порядке, в котором участник их делал. Для каждой посылки известен момент времени, в который она была сделана, номер задачи и вердикт тестирующей системы. Требуется посчитать количество задач, сданных участником, и его суммарный штраф.

Участник мог посылать задачу и после того, как её сдал. Штраф для задачи считается по формуле  $t + 20 \cdot k$ , где  $t$  — время первой успешной сдачи задачи в минутах,  $k$  — количество неправильных попыток перед первой успешной сдачей, не считая CE. Суммарный штраф считается как сумма штрафов по всем сданным задачам. За несданные задачи штраф не начисляется.

### Формат входных данных

В первой строке дано одно целое число  $n$  — количество посылок участника ( $1 \leq n \leq 100\,000$ ).

В следующих  $n$  строках дано описание посылок. Описание  $i$ -й посылки дается в формате «HH:MM X V», где:

- «HH:MM» — время посылки. Две цифры, обозначающие количество часов, и две цифры, обозначающие количество минут от начала контеста ( $0 \leq \text{HH} \leq 4$ ,  $0 \leq \text{MM} \leq 59$ ).
- «X» — номер задачи, заглавная латинская буква от 'A' до 'Z'.
- «V» — вердикт, значение из множества OK, WA, TL, ML, RE, CE, SV. Вердикт OK соответствует успешной посылке.

### Формат выходных данных

В единственной строке выведите через пробел два целых числа — количество сданных задач и суммарный штраф.

### Пример

стандартный ввод	стандартный вывод
7 00:05 A RE 00:07 A WA 00:09 A TL 00:29 B OK 00:55 D WA 01:08 A CE 01:10 A OK	2 159

## Задача С. Собака, предатель и кабеля

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Теперь у предателя есть собака! Палуба космического корабля может быть представлена в виде клетчатого поля  $n \times m$ , строки которого пронумерованы от 1 до  $n$  сверху вниз, а столбцы — от 1 до  $m$  слева направо. Между некоторыми соседними по стороне клетками располагаются отрезки кабеля.

В начале игры, собака находится в клетке  $(1, 1)$ , то есть в верхней левой клетке. Она выбирает одного из игроков, пусть выбранный игрок находится в клетке  $(x, y)$ . Собака выбирает один из кратчайших маршрутов от своего стартового положения до клетки  $(x, y)$  (за один ход собака может переместиться из клетки в соседнюю по стороне). После чего, собака и игрок начинают по-очереди делать ходы. Собака бежит по выбранному в самом начале маршруту, а игрок бежит ей навстречу по тому же маршруту с конца. Первый ход делает собака. Этот процесс продолжается до тех пор, пока собака и игрок не окажутся в одной клетке. Каждый раз, когда собака перебегает отрезок кабеля, она его перекусывает.

Помогите игрокам определить, какое максимальное количество отрезков кабеля может перекусить собака.

### Формат входных данных

В первой строке дано три целых числа  $n$ ,  $m$  и  $k$  — размеры поля и количество отрезков кабеля ( $1 \leq n, m \leq 200\,000$ ;  $n \cdot m \leq 200\,000$ ,  $0 \leq k \leq n \cdot (m - 1) + (n - 1) \cdot m$ ).

Далее дано описание  $k$  отрезков кабелей. Каждый отрезок описывается четырьмя целыми числами  $x_1, y_1, x_2$  и  $y_2$  ( $1 \leq x_1, x_2 \leq n$ ;  $1 \leq y_1, y_2 \leq m$ ). Эти числа задают позиции двух соседних по стороне клеток  $(x_1, y_1)$  и  $(x_2, y_2)$ , на границе между которыми находится отрезок кабеля. Гарантируется, что клетки  $(x_1, y_1)$  и  $(x_2, y_2)$  являются соседними по стороне.

В следующей строке дано одно целое число  $q$  — количество положений игроков, для которых нужно вычислить ответ ( $1 \leq q \leq 20$ ).

В следующих  $q$  строках дано по два целых числа  $x$  и  $y$  — позиция игрока ( $1 \leq x \leq n$ ,  $1 \leq y \leq m$ ).

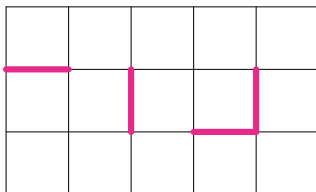
### Формат выходных данных

Выведите  $q$  строк, в  $i$ -й строке одно число — максимальное количество отрезков кабеля, которое может перекусить собака в  $i$ -м случае.

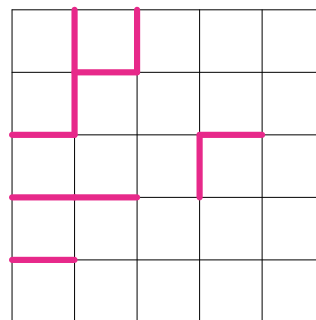
## Примеры

стандартный ввод	стандартный вывод
3 5 4 1 1 2 1 2 2 2 3 2 4 3 4 2 4 2 5 5 1 1 2 3 1 4 3 3 3 5	0 1 0 1 2
5 5 10 3 1 2 1 3 4 3 3 1 3 1 2 4 2 3 2 5 1 4 1 3 1 4 1 3 4 2 4 2 2 2 1 2 2 1 2 1 2 1 1 5 5 5 2 3 2 1 4 1 1 5	3 2 0 1 2

## Замечание



(а) В первом примере провода располагаются следующим образом



(б) Во втором примере провода располагаются следующим образом

## Задача D. Домино

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Вы с друзьями договорились сыграть в «Among Us». В назначенное время все зашли в игру. Все, кроме Арсения, который как всегда опаздывает. Находясь в комнате ожидания, вы решили поиграть в виртуальное домино, которое как раз в экспериментальном режиме было добавлено в последнем обновлении игры.

Так как режим еще экспериментальный, игра максимально примитивна: на столе лежит две доминошки. Каждая доминошка представляет собой прямоугольник, разделенный на две половины. В каждой половине доминошки нарисовано от нуля до шести точек. Игрокам нужно определить, можно ли совместить эти две доминошки. Доминошки можно совместить тогда и только тогда, когда существует такое целое неотрицательное число  $k$ , что на хотя бы одной из половинок каждой доминошки нарисовано ровно  $k$  точек.

Эта игра показалась вам чересчур примитивной, поэтому вы решили написать программу, которая будет автоматически в нее выигрывать. Сделайте это!

### Формат входных данных

Входные данные содержат две строки, каждая из которых содержит описание доминошки. Описание доминошки состоит из нескольких точек (символ «.», ASCII-код 46) и ровно одной вертикальной черты (символ «|», ASCII-код 124). Точки по одну сторону от вертикальной черты лежат в одной половине доминошки, а по другую — во второй половине.

Гарантируется, что каждая половина каждой доминошки содержит от 0 до 6 точек.

### Формат выходных данных

Выведите «Yes», если доминошки можно совместить, и «No» в противном случае.

### Примеры

стандартный ввод	стандартный вывод
.. .... .... ...	Yes
. ... .. ....	No
 	Yes

## Задача Е. Настройка коммуникаций

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

*По мотивам задачи о среднем геометрическом из прошлой интернет-олимпиады.*

Всем членам экипажа приходится выполнять кучу заданий, чтобы выиграть у импосторов, но иногда задания бывают слишком сложными, а члены экипажа слишком устают после выполнения всех коротких и средних заданий, поэтому они просят помощи у вас.

Текущее задание состоит в следующем: у прибора есть три натуральных параметра  $a$ ,  $b$  и  $c$ , которые были сброшены к заводским настройкам на значения  $x = g(a, b)$ ,  $y = g(a, c)$  и  $z = g(b, c)$  соответственно, где  $g$  — округленное вниз среднее геометрическое, то есть

$$g(p, q) = \lfloor \sqrt{pq} \rfloor$$

Помогите по текущим значениям  $x$ ,  $y$  и  $z$  определить любые подходящие  $a$ ,  $b$  и  $c$  (если они существуют), чтобы член экипажа смог правильно настроить прибор и выполнить задание. Поскольку членов экипажа много, и каждый настраивает свой прибор, вам понадобится решить задачу сразу несколько раз подряд!

### Формат входных данных

В первой строке дано одно целое число  $n$  — количество членов экипажа, которым нужно помочь ( $1 \leq n \leq 100\,000$ ).

Следующие  $n$  строк содержат по три целых числа  $x$ ,  $y$  и  $z$  каждая ( $1 \leq x, y, z \leq 10^9$ ).

### Формат выходных данных

Для каждой тройки чисел  $x$ ,  $y$ ,  $z$  выведите подходящие числа  $a$ ,  $b$  и  $c$ . Если подходящих ответов несколько, выведите любой. Если ответа нет, выведите три числа 0.

### Примеры

стандартный ввод	стандартный вывод
1 7 9 6	11 5 8
1 10 11 12	9 12 14

## Задача F. Электронный замок

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Предатели опять заблокировали двери. И чтобы их открыть, необходимо ввести код.

Поле для ввода кода представляет из себя бесконечную последовательность 7-сегментных индикаторов. Известно, что кодом является самое большое натуральное число, для отображения которого используется не более  $n$  горящих сегментов.



Рис. 2: Примеры того, как на 7-сегментном индикаторе отображаются все цифры.

Помогите команде выяснить код.

### Формат входных данных

В первой строке дано одно целое число  $n$  — максимальное количество горящих сегментов при отображении числа ( $2 \leq n \leq 100\,000$ ).

### Формат выходных данных

Выведите одно число — необходимый код.

### Пример

стандартный ввод	стандартный вывод
6	111

### Замечание

Так как у каждой единицы включено только два сегмента, то всего в числе 111 используется 6 включенных сегментов.

## Задача G. Место преступления

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Члены экипажа обнаружили место преступления предателя. Теперь команда корабля хочет оградить это место преступления.

Место преступления выглядит как выпуклый многоугольник  $A$ . Команда корабля хочет оградить его другим выпуклым многоугольником  $B$ . Причем, у  $B$  должно быть минимальное возможное число вершин. И все вершины многоугольника  $A$  должны лежать на границе многоугольника  $B$ .

Помогите команде выбрать такой многоугольник  $B$ .

### Формат входных данных

В первой строке дано одно целое число  $t$  — количество тестовых наборов ( $1 \leq t \leq 1000$ ). Далее дано описание  $t$  тестовых наборов.

В первой строке тестового набора дано одно целое число  $n$  — количество вершин в многоугольнике  $A$  ( $3 \leq n \leq 100$ ).

В следующих  $n$  строках даны по два целых числа  $x_i$  и  $y_i$  — координаты  $i$ -й вершины многоугольника ( $|x_i|, |y_i| \leq 1000$ ). Вершины многоугольника даны в порядке обхода против часовой стрелки. Гарантируется, что многоугольник является строго выпуклым. То есть, в том числе, никакие три его последовательные вершины не лежат на одной прямой.

### Формат выходных данных

Для каждого тестового набора сначала выведите одно целое число  $m$  — количество вершин в найденном многоугольнике  $B$ .

В следующих  $m$  строках выведите по два вещественных числа  $x_i$  и  $y_i$  — координаты  $i$ -й вершины многоугольника. Выводите вершины многоугольника в порядке обхода против часовой стрелки. Выведенный многоугольник должен быть строго выпуклым. А также, все вершины исходного многоугольника должны находиться на расстоянии не более  $10^{-6}$  от границы выведенного многоугольника.

### Пример

стандартный ввод	стандартный вывод
3	3
3	0.0 0.0
1 0	2.0 0.0
1 1	0.0 2.0
0 1	3
4	0.0 0.0
0 0	2.0 0.0
1 0	0.0 2.0
1 1	3
0 1	3.0 0.0
5	3.0 4.0
1 0	-1.0 0.0
3 0	
3 2	
1 2	
0 1	



## Задача Н. Похожие имена

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Как-то раз  $n$  друзей собрались сыграть в «Among us», но при этом они хотели, чтобы каждый человек, заходящий в лобби, понимал, что они играют вместе. Для этого они решили выбрать никнеймы с похожим началом, но поскольку каждому дорог его текущий никнейм, никто не хочет его сильно изменять.

В качестве компромисса было принято следующее решение: каждый игрок сдвинет свой никнейм по циклу на какое-то количество символов так, чтобы общий префикс никнеймов всех игроков был как можно длиннее. *Циклическим сдвигом* строки  $s = s_0s_1 \dots s_n$  называется строка вида  $s^i = s_i s_{i+1} \dots s_n s_0 s_1 \dots s_{i-1}$ , а *префиксом* — строка вида  $p^i = s_0 s_1 \dots s_i$ .

Так вот, возвращаясь к никнеймам: решить эту задачу предстоит вам, потому что игроки — не программисты, и для них это слишком сложно. Помогите им найти максимальный общий префикс, который можно получить, сдвинув их никнеймы по циклу.

### Формат входных данных

В первой строке задано число  $n$  — количество игроков ( $1 \leq n \leq 10^5$ ).

В следующих  $n$  строках заданы никнеймы игроков: на  $i$ -й строке дан никнейм  $i$ -го игрока  $s_i$  — последовательность строчных латинских букв ( $1 \leq |s_i| \leq 10^5$ ). Гарантируется, что сумма длин всех никнеймов не превосходит  $10^5$ .

### Формат выходных данных

Найдите длину наибольшего общего префикса, который могут получить игроки, применив к своим никнеймам какие-то циклические сдвиги.

### Примеры

стандартный ввод	стандартный вывод
4 abacada abracadabra rxacadd dzzzaca	4
2 abacaba acabaab	7

## Задача I. Теория Рамсея

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Карта для Among Us представляет собой неориентированный граф, вершины которого — комнаты, а рёбра — двусторонние тоннели между ними. У разработчика этой игры, Рамсея, есть теория, что карта должна удовлетворять некоторым свойствам, чтобы на ней было интересно играть.

В игре будет  $k$  предателей и  $l$  рядовых членов экипажа. Если в графе есть  $l$ -клика — набор из  $l$  вершин, каждая пара которых соединена ребром — то члены экипажа просто распределятся между ними, и в случае убийства кого-то из них все игроки из соседних комнат незамедлительно сбегутся, обнаружат убийцу и накажут его. Такая игра будет довольно неинтересной.

С другой стороны, если в графе есть  $k$ -антиклика — набор из  $k$  вершин, каждая пара которых не соединена ребром — то предатели смогут встать в её вершины, заманивать хороших игроков и их там убивать, и, скорее всего, им удастся при этом оставаться незамеченными. Такая стратегия, по мнению Рамсея, тоже сделает игру неинтересной.

Напишите программу, которая найдёт в графе  $l$ -клику или  $k$ -антиклику или определит, что их в графе нет (и тогда игра обещает быть захватывающей).

### Формат входных данных

В первой строке находится четыре числа  $n, m, k, l$ , разделённых пробелами — количество вершин и рёбер графа, а также размеры искомых антиклики и клики ( $1 \leq n \leq 300\,000$ ;  $0 \leq m \leq 300\,000$ ;  $1 \leq k, l \leq \min(5, n)$ ).

В следующих  $m$  строках находится по два целых числа  $a_i, b_i$ , разделённых пробелами — концы очередного ребра графа ( $1 \leq a_i < b_i \leq n$ ). Гарантируется, что все рёбра различны.

### Формат выходных данных

Если вы нашли набор вершин, являющийся или  $k$ -антикликой, или  $l$ -кликой, выведите номера всех этих вершин через пробел. Если же такого набора вершин нет, выведите «-1».

### Примеры

стандартный ввод	стандартный вывод
5 5 3 3 1 2 2 3 3 4 4 5 1 5	-1
4 0 2 4	1 2
4 0 4 2	1 2 3 4
5 10 1 4 1 2 2 3 3 4 4 5 1 3 2 4 3 5 1 4 2 5 1 5	1

## Замечание

Первый пример выглядит как пятиугольник, в котором провели все стороны, но не провели диагоналей. В нём нет ни треугольников, ни антитреугольников, поэтому ответ «-1».

Во втором и третьем примере граф пуст. Во втором примере требуется либо найти антиребро, либо 4-клик вершинах. Ответом послужит любая пара различных чисел от 1 до 4. В третьем примере всё наоборот — надо найти либо 4-антиклик, либо ребро. Так как рёбер нет, единственным правильным ответом на этот пример является набор из всех чисел от 1 до 4 (перечисленных в произвольном порядке).

В четвёртом примере дан полный граф, и корректным ответом является любая четвёрка его вершин (так как она будет его кликой). Однако набор из одной вершины всегда является как кликой, так и антикликой; следовательно, раз в примере разрешено вывести 1-антиклик, любое одновершинное множество — также корректный ответ.

## Задача J. Супер-счастливые билетки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Чтобы космический корабль мог взлететь, каждому члену экипажа нужно выдать билетик. На каждом билетике должна быть написана последовательность из  $n$  цифр, где  $n$  — четное положительное число.

Даниил считает билет супер-счастливым, если выполняются два свойства:

- Сумма первых  $\frac{n}{2}$  цифр равна сумме последних  $\frac{n}{2}$  цифр.
- Сумма цифр на четных позициях равна сумме цифр на нечетных позициях.

Помогите Даниилу посчитать количество различных супер-счастливых билетиков по модулю 998 244 353.

### Формат входных данных

В первой строке дано одно число  $n$  — длина билетика ( $2 \leq n \leq 200\,000$ ). Гарантируется, что  $n$  чётное.

### Формат выходных данных

Выведите количество супер-счастливых билетиков длины  $n$  по модулю 998 244 353.

### Примеры

стандартный ввод	стандартный вывод
2	10
8	448900