

Задача А. Золотые слитки

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Разбойники с большой дороги Джон и Боб ограбили караван и в качестве добычи получили три золотых слитка. Решив поделить добычу по-братски, Джон и Боб взвесили слитки и выяснили, что они весят x_1 , x_2 и x_3 фунтов, соответственно.

Теперь Джон и Боб хотят поделить слитки так, чтобы каждому из них досталось равное количество золота. Им не хотелось бы пилить слитки, но деваться некуда. Обсудив ситуацию, они решили, что если смогут, поделят добычу как есть, а если нет, то сумеют-таки распилить один слиток на две части. Распилить два или все три слитка они уже не смогут.

Помогите Джону и Бобу выбрать, какой слиток распилить на две части, и на какие части его следует распилить, чтобы после этого можно было поделить добычу поровну.

Формат входных данных

Первая строка входных данных содержит три целых числа: x_1 , x_2 и x_3 ($1 \leq x_i \leq 10^8$, сумма весов слитков чётна).

Формат выходных данных

Если невозможно распилить один слиток таким образом, что после этого можно поделить золото поровну, выведите -1 .

Если Джон и Боб и так могут поделить золото поровну, выведите 0 .

В противном случае на первой строке выведите число 1 , если следует распилить первый слиток, 2 , если следует распилить второй слиток, либо 3 , если следует распилить третий слиток. На второй строке выведите два положительных целых числа: веса частей, на которые следует распилить слиток. В сумме две части должны давать исходный вес слитка. Так как суммарный вес золота чётен, слиток всегда требуется распиливать на части, имеющие целый вес. Если возможных решений несколько, выведите любое.

Пример

стандартный ввод	стандартный вывод
2 3 3	2 2 1

Задача В. Автобус

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Каждое утро Антон едет на работу на автобусе.

Маршрут автобуса включает в себя n остановок, пронумерованных от 1 до n в порядке следования. Автобус проезжает от каждой остановки до следующей за одну минуту, а его временем стоянки можно пренебречь. Антон садится на первой остановке и выходит на последней.

В автобусе есть m сидений, которые расположены в один ряд и пронумерованы от 1 до m , ближайшее к входу сиденье имеет номер 1, а самое дальнее — номер m . На каждом сиденье может сидеть один пассажир, а также возле каждого сиденья может стоять один пассажир.

Когда человек входит в автобус, он садится на ближайшее ко входу свободное сиденье. Если они все заняты, он ищет ближайшее ко входу сиденье, рядом с которым никто не стоит, и встаёт у сидящего там человека над душой. Если такого места нет, он выходит из автобуса.

Каждый пассажир остаётся на своём месте до прибытия на нужную ему остановку. Стоящий пассажир остаётся стоять, даже если какое-то сиденье освободилось.

На каждой остановке из автобуса сначала выходят все пассажиры, которые собирались выйти на этой остановке, и только потом заходят новые.

Антон зашёл в автобус первым, он может сесть на любое сиденье и остаться на нём до конца поездки. Он хорошо знает, кто ещё будет ехать в автобусе, про каждого пассажира Антон знает, на какой остановке тот войдет в автобус и на какой выйдет. Помогите Антону выбрать место так, чтобы во время поездки у него как можно меньше суммарно стояли над душой.

Формат входных данных

В первой строке входных данных заданы три целых числа n , m и k — количество остановок, количество сидений в автобусе и количество пассажиров, кроме Антона ($2 \leq n \leq 10^9$, $1 \leq m \leq 2 \cdot 10^5$, $0 \leq k \leq 2 \cdot 10^5$).

В следующих k строках задано по 2 числа a_i и b_i , которые означают, что i -й пассажир собирается войти на a_i -й остановке и выйти на b_i -й ($1 \leq a_i < b_i \leq n$).

Если на одной остановке в автобус заходит несколько человек, они заходят в том порядке, в котором они перечислены во входных данных.

Формат выходных данных

Выведите два числа на одной строке — минимальное суммарное время в минутах, в течение которого у Антона будут стоять над душой, и номер места, на которое для этого нужно сесть. Если таких мест несколько, выведите ближайшее ко входу.

Пример

стандартный ввод	стандартный вывод
10 2 3 1 10 3 9 7 10	3 2

Задача С. Горные лыжи

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Суровая зима в Санкт-Барнаурге длится n дней. Таня очень любит кататься на лыжах и часто выезжает на близлежащий горнолыжный курорт в Тбиатыкенте. Так, про некоторые дни последней зимы Таня помнит, что была в этот день на горнолыжном курорте — ведь она выкладывала своё фото со склона в тот день в социальной сети *SkiForces*. Про другие дни никакой информации нет.

Известно, что Таня всегда ездит на курорт одинаково: она выезжает утром некоторого дня, проводит на курорте ровно k дней и возвращается вечером k -го дня поездки. При этом могло оказаться, что Таня снова поехала на курорт на следующий день после окончания предыдущей поездки. Те дни, когда Таня не ездила на горнолыжный курорт, она провела в городе.

Зима закончилась, и подруги говорят Тане, что она слишком много катается на лыжах. Чтобы понять, так ли это, Таня хочет выяснить, каким могло быть максимальное количество зимних дней, которые провела в городе.

Таня могла первый раз поехать на курорт до начала зимы и могла закончить последнюю поездку после окончания зимы.

Формат входных данных

В первой строке находятся три целых положительных числа n , k и m — продолжительность зимы в днях, длительность одной поездки на горнолыжный курорт в днях и количество дней, в которые Таня точно была на курорте ($1 \leq k \leq n \leq 10^9$, $1 \leq m \leq 2 \cdot 10^5$, $m \leq n$).

Во второй строке находятся m чисел d_1, d_2, \dots, d_m — номера дней в которые Таня точно была на курорте ($1 \leq d_i \leq n$). Каждый день перечислен не более одного раза.

Формат выходных данных

Выведите единственное целое число — максимальное количество зимних дней, в которые Таня не была на горнолыжном курорте.

Примеры

стандартный ввод	стандартный вывод
4 2 2 1 4	2
7 4 3 4 3 5	3
5 1 5 5 4 3 2 1	0
13 3 6 3 5 6 8 9 11	4

Замечание

В первом примере Таня могла быть на курорте два раза: первый раз начиная за день до зимы и заканчивая в 1-й день зимы; второй раз начиная в последний день зимы и заканчивая в 1-й день после зимы.

Таким образом, Таня могла провести в городе второй и третий дни зимы.

Во втором примере Таня могла быть на курорте один раз, например, начиная со второго дня зимы и заканчивая пятым днём зимы. Таким образом, Таня могла провести в городе три дня — в первый, в шестой и в седьмой дни зимы.

Задача D. IQ тест для роботов

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Лёша готовит своего робота к тестированию IQ. В 2116 году тестирование IQ для роботов проходит следующим образом. Роботу демонстрируется прямоугольная таблица, содержащая n строк и m столбцов, каждая клетка которой покрашена в какой-либо цвет.

Затем экзаменатор q раз просит робота выполнить следующее задание. Экзаменатор указывает на некоторую клетку в таблице, а робот в качестве ответа должен выбрать две другие клетки. При этом должны выполняться следующие условия.

- Ни одна из выбранных роботом клеток не должна совпадать с указанной экзаменатором.
- Одна из выбранных клеток должна лежать в одном столбце с указанной, а другая — в одной строке.
- Цвета двух выбранных роботом клеток должны быть различны.
- Расстояние между выбранными клетками должно быть минимально. Расстояние между клетками (r_1, c_1) и (r_2, c_2) определяется как $|r_1 - r_2| + |c_1 - c_2|$.

Бывает, что выбрать две клетки описанным образом невозможно, в этом случае в качестве ответа на задание робот должен сообщить об этом.

Лёша хочет научить своего робота справляться с заданием как можно лучше. Помогите ему запрограммировать робота.

Формат входных данных

В первой строке находятся целые числа n и m — количество строк и столбцов в таблице, соответственно ($2 \leq n, m \leq 500\,000$; $n \times m \leq 10^6$).

Следующие n строк содержат по m строчных латинских букв — описание таблицы, j -й символ i -й из этих строк задает цвет клетки (i, j) . Одинаковые буквы обозначают одинаковый цвет, а разные — разный.

В следующей строке находится число q — количество вопросов экзаменатора ($1 \leq q \leq 200\,000$).

Следующие q строк содержат описание вопросов. В i -й из этих строк находятся два числа x_i и y_i — номер строки и столбца, на пересечении которых находится клетка, для которой требуется найти ответ ($1 \leq x_i \leq n$, $1 \leq y_i \leq m$).

Формат выходных данных

Для каждого вопроса выведите ответ в отдельной строке. Если невозможно найти пару клеток, удовлетворяющих всем условиям, выведите -1 . Иначе выведите четыре числа r_1, c_1, r_2, c_2 — описание двух выбранных клеток. Если оптимальных ответов несколько, выведите любой.

Пример

стандартный ввод	стандартный вывод
3 4	-1
abbb	1 1 2 4
baab	3 2 2 1
babb	2 4 3 2
4	
1 1	
1 4	
3 1	
3 4	

Задача Е. Расшифровка ДНК

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Ученые работают на раскопках окаменелых останков древних существ на планете соседней звездной системы. В процессе исследования ученые пытаются понять, как именно цепочки ДНК различных существ составлялись из генов.

Цепочки ДНК всех исследуемых существ представляют собой последовательности нуклеотидов. Каждый нуклеотид обозначается строчной буквой латинского алфавита. Таким образом, цепочка ДНК представляет собой строку, составленную из строчных букв латинского алфавита.

Ген также представляет собой строку из строчных букв латинского алфавита. Известно, что в любом корректном наборе генов никакая строка не является префиксом другой строки.

Будем говорить, что цепочку ДНК d можно расшифровать с использованием набора генов G , если d можно представить как результат последовательной записи одного или нескольких генов: $d = g_1g_2 \dots g_k$, где g_i — гены из набора G . Один и тот же ген может входить в расшифровку ДНК несколько раз.

Для обработки информации ученым требуется разработать компьютерную систему, которая будет поддерживать корректный набор генов G и массив цепочек ДНК существ D . По мере анализа останков, ученые могут добавлять новый ген в набор G или добавлять новую цепочку ДНК в массив D . Гарантируется, что ни в какой момент времени не существует двух генов, один из которых является префиксом другого.

После каждой операции ученые хотят знать, какие цепочки ДНК в массиве D можно расшифровать, используя текущий набор генов G . После i -й операции система должна сообщать k_i — количество цепочек ДНК, находящихся в массиве D , которые впервые стало можно расшифровать после i -й операции, а затем k_i чисел — номера этих цепочек. Результат очередной операции должен быть получен до того, как станет известна следующая операция.

Помогите ученым разработать такую систему.

Формат входных данных

В первой строке находится число n — количество операций, которые необходимо выполнить ($1 \leq n \leq 100\,000$).

В следующих n строках находятся описания операций, i -я строка начинается с символа «+», если эта операция — добавление нового гена в набор G , или с символа «?», если эта операция — добавление цепочки ДНК в конец массива D . Далее через пробел находится строка x_i , состоящая из строчных латинских букв, которую необходимо использовать, чтобы получить строку s_i , которая задает добавляемый в этой операции ген или цепочку ДНК.

Для получения строки s_i из строки x_i , необходимо выполнить следующие действия. Если $i = 1$, то $s_i = x_i$. Иначе пусть число впервые расшифрованных цепочек ДНК после предыдущей операции равно k_{i-1} . Выполним k_{i-1} раз следующее действие: перенесем первый символ x_i в конец. Иначе говоря, выполним циклический сдвиг строки x_i влево на k_{i-1} . Получившаяся строка равна s_i — ген или цепочка ДНК, которую необходимо добавить на i -й операции.

Все строки не пусты, суммарный размер строк во всех операциях не превышает 10^6 .

Гарантируется, что ни в какой момент времени не существует двух генов, один из которых является префиксом другого.

Формат выходных данных

Выведите n строк.

В i -й строке выведите сначала число k_i — количество цепочек ДНК, находящихся в массиве D , которые впервые стало можно расшифровать после i -й операции, а затем k_i чисел — номера этих цепочек. Цепочки нумеруются с единицы в порядке добавления в массив D . Номера цепочек в одной строке можно выводить в любом порядке.

Пример

стандартный ввод	стандартный вывод
5	0
? abcabd	0
+ abc	1 2
? abcabc	0
? dabdab	2 1 3
+ abd	

Замечание

В первых трех операциях s_1 , s_2 и s_3 совпадают с соответствующими строками во вводе. Поскольку $k_3 = 1$, то для четвертой операции s_4 получается из строки $x_4 = \text{«dabdab»}$ циклическим сдвигом влево на 1, таким образом, в четвертой операции в массив D добавляется строка $s_4 = \text{«abdabd»}$. Наконец, $k_4 = 0$, поэтому $s_5 = x_5$.

Задача F. Преобразование таблицы

Ограничение по времени: 4 секунды
Ограничение по памяти: 512 мегабайт

Ваня занимается работой с большими данными. Его проект занимается обработкой гигантских таблиц статистических данных. Ваня отвечает за разработку модуля преобразования таблиц, который выполняет перестановку строк, столбцов и ячеек таблицы.

Модуль занимается обработкой таблицы, состоящей из h строк и w столбцов, строки пронумерованы сверху вниз от 0 до $h - 1$, столбцы пронумерованы слева направо от 0 до $w - 1$. Ячейка в i -й строке, j -м столбце таблицы обозначается как $[i, j]$. Исходно ячейка $[i, j]$ содержит число $i \times w + j$. На рис. 1. приведен пример исходного заполнения таблицы для $h = 3, w = 5$.

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14

Рис. 1. Пример исходного заполнения таблицы.

Модуль преобразования таблиц может выполнять следующие три типа операций.

Операция	Обозначение	Действие
Обмен столбцов	$c\ x\ y$	Поменять местами содержимое столбцов с номерами x и y
Обмен строк	$r\ x\ y$	Поменять местами содержимое строк с номерами x и y
Обмен ячеек	$f\ a\ b\ c\ d$	Поменять содержимое ячеек $[a, b]$ и $[c, d]$

На рис. 2. показано, как выглядит приведенная выше таблица после выполнения последовательности операций «с 0 1», «r 0 1», «f 0 0 1 2».

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14

выполняется операция «с 0 1»

1	0	2	3	4
6	5	7	8	9
11	10	12	13	14

выполняется операция «r 0 1»

6	5	7	8	9
1	0	2	3	4
11	10	12	13	14

выполняется операция «f 0 1 1 2»

6	2	7	8	9
1	0	5	3	4
11	10	12	13	14

Рис. 2. Пример преобразования таблицы

После выполнения всех операций Ваня вычисляет контрольную сумму для таблицы: сумма по всем ячейкам $[i, j]$ значений $(v[i][j] \times 17^i \times 19^j) \bmod (10^9 + 7)$. Здесь $v[i][j]$ означает значение в ячейке $[i, j]$, а операция «mod» означает операцию взятия остатка. Например, контрольная сумма для таблицы из примера вычисляется следующим образом: $(6 \times 17^0 \times 19^0 + 2 \times 17^0 \times 19^1 + \dots + 14 \times 17^2 \times 19^4) \bmod (10^9 + 7) = 564\,830\,737$.

Помогите Ване выполнить все операции и вычислить контрольную сумму таблицы, которая получится в итоге.

Поскольку входные данные для этой задачи слишком велики, чтобы задавать их непосредственно, для ввода вам потребуется процедура *расширения массива*. Эта процедура не имеет специфических особенностей, которые надо использовать в решении задачи, предполагаемое жюри решение этой задачи в явном виде генерирует все массивы и далее работает с ними так же, как если бы оно считало их из входных данных.

Опишем процедуру генерации массива длины n по массиву длины $2 \leq k \leq n$. Пусть задан массив целых неотрицательных чисел $A = (a[1], a[2], \dots, a[k])$. Массив целых чисел $Ax = (ax[1], ax[2], \dots, ax[n])$ будем называть расширением массива A по модулю r до размера n , если его элементы вычисляются по следующим формулам.

- Если $1 \leq i \leq k$, то $ax[i] = a[i]$.
- Если $k + 1 \leq i \leq n$, то $ax[i] = (10007 \times ax[i - 2] + 10009 \times ax[i - 1] + 87277) \bmod r$.

Здесь как «mod» также обозначена операция взятия остатка по модулю r .

Например, выполним расширение массива $A = (1, 4, 3)$ до 5 элементов по модулю 13.

$$ax[1] = a[1] = 1.$$

$$ax[2] = a[2] = 4.$$

$$ax[3] = a[3] = 3.$$

$$ax[4] = (10007 \times ax[2] + 10009 \times ax[3] + 87277) \bmod 13 = 157332 \bmod 13 = 6.$$

$$ax[5] = (10007 \times ax[3] + 10009 \times ax[4] + 87277) \bmod 13 = 177352 \bmod 13 = 6.$$

Таким образом, $Ax = (1, 4, 3, 6, 6)$.

Формат входных данных

Первая строка ввода содержит числа h, w, n — размеры таблицы и число преобразований, которые необходимо выполнить ($1 \leq h, w \leq 5\,000, 2 \leq n \leq 10^6$).

Вторая строка содержит строку s длины n — последовательность типов преобразований, $s[i] = \langle c \rangle$ задает операцию обмена столбцов, $s[i] = \langle r \rangle$ — операцию обмена строк, $s[i] = \langle f \rangle$ — операцию обмена ячеек.

Следующие четыре строки задают массивы A, B, C, D , соответственно. Каждый массив задается числом $k, 2 \leq k \leq n, k \leq 1000$, после чего следует k чисел — элементы массива. Элементы массивов A и C удовлетворяют ограничению $0 \leq a[i], c[i] \leq h - 1$, элементы массивов B и D удовлетворяют ограничению $0 \leq b[i], d[i] \leq w - 1$.

Пусть Ax является расширением массива A по модулю h до размера n , массив Bx является расширением массива B по модулю w до размера n , массив Cx является расширением массива C по модулю h до размера n и массив Dx является расширением массива D по модулю w до размера n .

Операции, которые требуется выполнить с таблицей, определяются следующим образом: тип i -й операции задается символом $s[i]$, а параметры получаются из массивов Ax, Bx, Cx, Dx .

- Если $s[i] = \langle c \rangle$, то i -я операция «с $Bx[i] \ Dx[i]$ »;
- Если $s[i] = \langle r \rangle$, то i -я операция «r $Ax[i] \ Cx[i]$ »;
- Если $s[i] = \langle f \rangle$, то i -я операция «f $Ax[i] \ Bx[i] \ Cx[i] \ Dx[i]$ »;

Формат выходных данных

Выведите одно число — контрольную сумму таблицы после выполнения всех преобразований.

Пример

стандартный ввод	стандартный вывод
3 5 3 crf 3 0 0 0 3 0 0 1 3 0 1 1 3 1 0 2	564830737

Задача G. Архивы джедаев

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Это интерактивная задача.

В Далёкой-Далёкой Галактике 10^{18} планет, и все они пронумерованы натуральными числами от 1 до 10^{18} .

В Архивах джедаев были сохранены сведения о всех планетах Галактики: по кругу в огромном зале были расположены ячейки, пронумерованные против часовой стрелки от 1 до 10^{18} . Исходно Архивы были устроены очень просто: в ячейке с номером i хранились сведения о планете с номером i . Поэтому, зная лишь номер планеты, всегда можно было легко найти информацию о ней.

Вот и Оби-Вану Кеноби понадобилось узнать координаты планеты Камино, которая имеет номер x . Однако, заглянув в Архивы, он с удивлением обнаружил, что кто-то удалил из Архивов m ячеек со сведениями о некоторых планетах. Кроме того, оставшиеся ячейки были заново пронумерованы, начиная с 1, против часовой стрелки (возможно, начиная не с той, с которой они нумеровались исходно), и теперь в ячейке с номером i могут храниться сведения о совсем другой планете.

Оби-Ван в затруднении. Ему нужно срочно найти данные о планете Камино. Оби-Ван может проверить содержимое ячейки и узнать, данные о какой планете в ней находятся. У Оби-Вана мало времени, поэтому он может проверить содержимое не более чем десяти ячеек.

Помогите ему выяснить, в какой ячейке находятся данные о планете Камино, если, конечно, они не были удалены из Архивов.

Протокол взаимодействия

Сначала вашей программе подаётся на вход в отдельной строке два числа: x — номер планеты Камино ($1 \leq x \leq 10^{18}$) и m — число удалённых из Архивов ячеек ($0 \leq m \leq 500$).

После этого ваша программа может делать запросы: «посмотреть, сведения о какой планете записаны в ячейке номер v ». Для этого нужно вывести в выходной поток на отдельной строке «? v » ($1 \leq v \leq 10^{18} - m$). В ответ на запрос во входном потоке на отдельной строке будет записано одно число — номер планеты, сведения о которой записаны в ячейке v . Вы можете сделать не более десяти таких запросов, иначе ваша программа получит вердикт «Wrong answer».

В конце вы должны вывести в выходной поток «! a », где a — это номер ячейки, в которой записаны сведения о планете Камино. Если данные о планете Камино были удалены из Архива, то следует вместо номера ячейки вывести -1 , таким образом последнее сообщение должно быть «! -1».

Замечание

После каждого действия вашей программы выводите символ перевода строки. Если вы используете «writeln» в Паскале, «cout << ... << endl» в C++, «System.out.println» в Java или «print» в Python, сброс потока вывода у вас происходит автоматически, дополнительно делать «flush» не обязательно. Если вы используете другой способ вывода, рекомендуется делать «flush», но все равно обязательно требуется выводить символ перевода строки.

Ниже приведены наиболее типичные причины получения тех или иных сообщений об ошибке.

Если ваша программа соблюдает протокол, но неверно определяет искомый номер ячейки либо выполняет слишком много запросов, вы получите результат «Wrong Answer».

Если ваша программа выводит некорректно отформатированные сообщения программе жюри, то вы получите результат «Presentation Error» либо «Wrong Answer».

Если ваша программа нарушила протокол и ждет ввода в то же время, когда его ждет и программа жюри, то вы получите результат «Idleness Limit Exceeded». Обратите внимание, что к такому же результату может привести и то, что вы не переводите строку после каждого выведенного сообщения или выводите не тем способом, который описан в начале раздела, и не делаете «flush».

Пример

стандартный ввод	стандартный вывод
16 3	
	? 1
10	
	? 2
12	
	? 3
13	
	? 4
16	
	! 4

Замечание

В примере из Архива были удалены данные о планетах с номерами 11, 14 и 15, а ячейки были перенумерованы, начиная с ячейки, содержащей сведения о планете с номером 10.

Задача Н. Музей

Ограничение по времени: 4 секунды
Ограничение по памяти: 512 мегабайт

В Федеральном Университете Флатландии открылся музей спортивных достижений. Главная гордость этого музея — зал, в котором расположены сувениры со всевозможных соревнований по спортивному программированию.

Программирование становится всё популярнее, и заинтересованных посетителей этого зала становится всё больше, поэтому директор музея начинает беспокоиться о сохранности своих сувениров. Для начала директор решил оградить одну или две зоны с сувенирами от посетителей так, чтобы они не могли свободно бродить между ними, а могли лишь смотреть со стороны.

Зал имеет форму выпуклого многоугольника, а сувениры — это точки внутри этого многоугольника. Директор хочет выбрать один или два треугольника с вершинами в углах зала, так чтобы эти треугольники не имели общей внутренней части (то есть площадь их пересечения была равна нулю), а каждый сувенир оказался внутри одного из треугольников.

Директор хочет доставить посетителям как можно меньше неудобств, поэтому суммарная площадь выбранных треугольников должна быть минимальной, при этом площадь каждого треугольника должна быть строго положительной.

Помогите директору оградить экспонаты.

Формат входных данных

Входные данные содержат один или несколько тестовых примеров.

В первой строке находится число t — количество тестовых примеров. Далее следуют t блоков, каждый из которых выглядит следующим образом.

В первой строке блока находится два целых положительных числа n и m — количество вершин в многоугольнике, форму которого имеет зал, и количество сувениров в зале ($3 \leq n \leq 2000$, $1 \leq m \leq 2000$).

В следующих n строках находятся по два целых числа xh_i и yh_i — координаты вершин многоугольника в декартовой системе координат ($-10^9 \leq xh_i, yh_i \leq 10^9$). Вершины даны в порядке обхода против часовой стрелки.

В следующих m строках находятся по два целых числа xs_i , ys_i — координаты сувениров ($-10^9 \leq xs_i, ys_i \leq 10^9$).

Гарантируется, что все сувениры находятся строго внутри зала, а также что никакие три из данных $n + m$ точек не лежат на одной прямой.

Сумма всех n и сумма всех m во всех тестовых примерах одних входных данных не превосходят 2000 каждая.

Формат выходных данных

Для каждого из t тестовых примеров выведите следующее.

Если невозможно выбрать один или два треугольника требуемым образом, выведите в отдельной строке -1 .

Иначе выведите в первой строке целое число k ($1 \leq k \leq 2$) — количество выбранных треугольников, и в каждой из следующих k строк выведите по три целых числа — номера вершин зала, которые являются вершинами треугольника.

Если оптимальных ответов несколько, выведите любой из них.

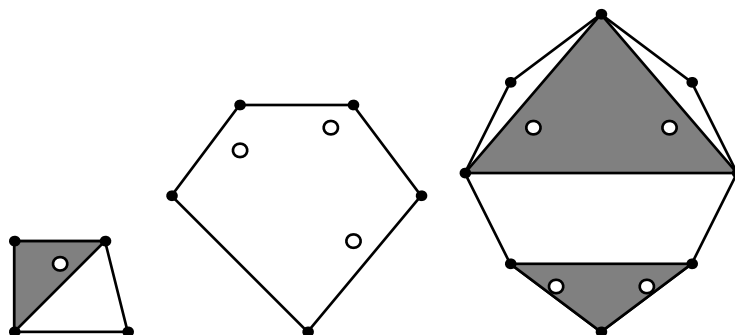
Обратите внимание, что требуется минимизировать только суммарную площадь выбранных треугольников.

Пример

стандартный ввод	стандартный вывод
3	1
4 1	1 4 3
0 0	-1
5 0	2
4 4	1 3 2
0 4	4 8 6
2 3	
5 3	
0 0	
6 -6	
11 0	
8 4	
3 4	
3 2	
7 3	
8 -2	
8 4	
-4 -4	
0 -7	
4 -4	
6 0	
4 4	
0 7	
-4 4	
-6 0	
-2 -5	
2 -5	
3 2	
-3 2	

Замечание

На рисунке показаны оптимальные способы выбрать треугольники в первом и третьем тестовых примерах, а также расположение сувениров во втором тестовом примере, где выбрать один или два треугольника нельзя.



Задача I. Обычный мальчик

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Петя — обычный мальчик. Каждый раз перед началом каждого учебного года у него появляется желание взяться за ум. И в этом году он не отступил от своей цели сразу после начала учебы. Внимательно слушая учителя и выполняя все домашние задания, он стал лучшим учеником в классе.

Учитель заметил успехи Пети и выдал ему самый сложный вариант контрольной по математике. Почти все задания мальчик решил за урок, но с одной так и не справился.

Дано целое положительное число x . Требуется найти число $y \geq x$, у которого не менее 100 делителей. Причем число y должно превышать число x не более чем на 1%, то есть должно выполняться неравенство $x \leq y \leq 1.01x$.

Помогите Пете решить эту сложную задачу.

Формат входных данных

Первая строка ввода содержит число x ($1 \leq x \leq 10^{16}$).

Формат выходных данных

Если подходящего числа не существует, то выведите -1 , иначе выведите любое подходящее число.

Примеры

стандартный ввод	стандартный вывод
5	-1
510000	510510

Замечание

Число 510510 имеет ровно 128 делителей.

Задача J. Полиглоты-интроверты

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Перед очередной лекцией на всемирной конференции полиглотов ее участники собрались в небольшой комнате. Всего в комнате n человек, каждый из которых разговаривает на некоторых из k языков. По стечению обстоятельств, все из них оказались интровертами, а поэтому общению с другими участниками предпочитают мирное прочтение книги.

Иногда, однако, один участник конференции решается-таки сообщить какую-то информацию другому. Два человека могут поговорить либо напрямую, либо через цепочку посредников. При общении напрямую два человека могут выбрать любой язык, которые знают оба собеседника. При общении через цепочку посредников любые два соседних человека в цепочке должны выбрать язык, на котором они будут говорить, среди тех, который знают оба собеседника. Таким образом, тот, кто хочет передать информацию, расскажет её первому посреднику, тот затем расскажет её второму посреднику, и так далее, пока информация не дойдёт до того, кому она предназначена.

Однако, когда происходит общение на каком-то языке, все в комнате, кто знает этот язык, слышат разговор на знакомом языке и отвлекаются от своей книги. Тот, кто передает информацию, хочет потревожить как можно меньше людей, поэтому планирует цепочку посредников таким образом, чтобы как можно меньше людей отвлеклось во время передачи информации.

Найдите для каждой пары людей (A, B) , какое минимальное людей можно отвлечь, чтобы передать информацию от A к B .

Формат входных данных

В первой строке входных данных содержатся 2 числа n и k — число людей в комнате и число различных языков, на которых они разговаривают ($2 \leq n \leq 300$, $1 \leq k \leq 300$).

Следующие n строк содержат описания людей, i -я из этих строк описывает языки, которые знает i -й человек, в следующем формате: $k_i a_{i,1} a_{i,2} \dots a_{i,k_i}$ — количество языков и сами языки в порядке возрастания номеров ($1 \leq k_i \leq k$, $1 \leq a_{i,j} \leq k$, $a_{i,j} < a_{i,j+1}$).

Формат выходных данных

Выведите n строк, в каждой по n чисел $f_{i,j}$, разделенных пробелами — минимальное количество людей, которые отвлекутся при оптимальном способе передаче информации от i -го человека к j -му. На главной диагонали должны стоять нули. Если передать информацию от i -го человека к j -му невозможно, выведите вместо соответствующего числа -1 .

Примеры

стандартный ввод	стандартный вывод
6 4	0 3 3 2 4 5
2 1 2	3 0 3 4 2 3
2 2 3	3 3 0 4 4 5
1 2	2 4 4 0 5 6
1 1	4 2 4 5 0 2
2 3 4	5 3 5 6 2 0
1 4	
4 3	0 2 2 -1
2 1 2	2 0 3 -1
1 1	2 3 0 -1
1 2	-1 -1 -1 0
1 3	

Задача К. «Исключающее или» наносит ответный удар

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Даны два неотрицательных целых числа a и n . Требуется найти такое минимальное неотрицательное число b , что $a \oplus b$ нацело делится на n .

Здесь \oplus обозначает операцию побитового «исключающего или» и соответствует операции «xor» в Паскале или «^» в других языках. Для вычисления побитового «исключающего или» двух чисел x и y необходимо записать каждое из них в двоичной системе счисления, дополнив, при необходимости, ведущими нулями слева. Результат в каждой позиции равен 1 в том случае, если в точности в одном из чисел в соответствующей позиции находится 1. К примеру, для чисел $x = 12$ и $y = 26$ результат равен 22:

$$\begin{array}{r|ccccc|l} x & 0 & 1 & 1 & 0 & 0 & 12 \\ y & 1 & 1 & 0 & 1 & 0 & 26 \\ \hline x \oplus y & 1 & 0 & 1 & 1 & 0 & 22 \end{array}$$

Формат входных данных

В первой строке входных данных содержится число t — количество тестовых примеров ($1 \leq t \leq 10^4$). В следующих t строках содержатся описания тестовых примеров. Каждое описание состоит из двух чисел a и n , разделенных пробелом ($1 \leq a, n \leq 10^{18}$).

Формат выходных данных

Для каждого из тестовых примеров требуется вывести единственное число — искомое b .

Пример

стандартный ввод	стандартный вывод
3	0
10 5	1
3 2	6
98 100	