Problem A. Nutella's Life

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Сайт `chefforces.at` только что опубликовал расписание контестов на следующий год. Всего планируется n контестов, изменений в расписании не планируется.

После изучения списка авторов каждого контеста Олег выписал n целых чисел a_i, по одному для каждого контеста — планируемые ощущения от каждого контеста. Если a_i положительно, контест добавит позитива, если же отрицательно — наоборот.

Однако Олег не хочет пропускать контесты, особенно он не хочет пропускать несколько контестов подряд. Более формально, если Олег решает пропустить контест и он уже пропустил x контестов, которые проходили непосредственно перед этим, позитив уменьшается на $x + 1$.

Наконец, Олег хочет, чтобы каждый контест приносил как минимум не меньше позитива, чем предыдущий, в котором он участвовал, то есть если Олег участвовал в контестах с номерами i и j, $i < j$, то должно выполняться условие $a_i \le a_j$.

Помогите Олегу составить расписание контестов таким образом, чтобы максимизировать полученный в итоге позитив.

Input

Первая строка входа содержит одно целое число n — количество контестов в расписании ($1 \le n \le 10^5$).

Вторая строка содержит n целых чисел a_i ($-10^9 \le a_i \le 10^9$).

Output

Выведите одно целое число: максимальный позитив, который Олег может получить от участия в контестах.

Examples

standard input	standard output
7 1 3 2 7 3 2 4	7
7 -3 -4 -2 -2 -6 -8 -1	-11

Problem B. Олег и анализ данных

Input file:	*стандартный ввод*
Output file:	*стандартный вывод*
Time limit:	2 секунды
Memory limit:	256 мебибайт

Сейчас у всех на слуху машинное обучение, нейронные сети и большие данные. Вот и студент Олег, желая быть в тренде, стал усердно заниматься анализом данных на языке Python. До чего же приятно прийти в коворкинг, добавить пару слоёв в нейросеть и, откинувшись на спинку кресла, попивать смузи, пока компьютер обрабатывает гигабайты данных! Но сегодня что-то пошло не так, и Олег просит вашей помощи.

Изначально у него был массив a, содержащий очень важные данные: целые числа от L до R. Затем Олег написал функцию $f(a, m)$, которая возвращает новый массив, в котором каждое число заменено на его остаток от деления на целое положиельное число m. Наконец, Олег опечатался и написал строчку $a = f(a, Q)$, тем самым удалив исходный массив a! Чтобы оценить масштаб трагедии, он хочет посчитать количество таких целых положительных чисел X, что, независимо от содержимого изначального массива a, результат работы функции $f(a, X)$ будет таким же, как и если бы Олег не написал ту злополучную строчку.

Если задача ещё не понятна, вот вам формальное условие. Требуется посчитать количество таких целых положительных чисел X, что для любого целого числа S из отрезка $[L, R]$ будет верно, что

$$((S \bmod Q) \bmod X) = (S \bmod X).$$

Input

В единственной строке даны три целых числа, разделённые пробелами, — L, R и Q соответственно ($1 \le L, R, Q \le 10^{12}$, $L \le R$).

Output

Если количество подходящих чисел X конечно — выведите его. Иначе выведите слово «`infinity`».

Examples

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

Problem C. Гирлянда

Input file:	*стандартный ввод*
Output file:	*стандартный вывод*
Time limit:	2 секунды
Memory limit:	256 мебибайт

Однажды вечером Никита отдыхал дома и наблюдал за новогодней гирляндой. Огоньки на гирлянде завораживающе мерцали: лампочки по неизвестным Никите правилам то загорались, то гасли. И тут Никита придумал эту задачу.

Новогодняя гирлянда состоит из n последовательно соединённых лампочек различных цветов. В любой момент времени каждая лампочка либо включена, либо выключена. Также известно, что все лампочки одного цвета находятся в одном и том же состоянии и меняют его синхронно.

Время от времени гирлянда меняется следующим образом: все лампочки заданного цвета меняют своё состояние на противоположное. После каждого такого изменения Никиту интересует количество «островков света» — максимальных по включению непрерывных отрезков зажжённых лампочек.

Изначально все лампочки **выключены**. Помогите Никите посчитать количество «островков света» в гирлянде после каждого её изменения.

Input

В первой строке заданы три числа n, k и q — количество лампочек в гирлянде, количество различных цветов и количество запросов ($1 \le n, q \le 2 \cdot 10^5$, $1 \le k \le n$).

В следующей строке задана последовательность из n чисел $c_1, c_2, \ldots, c_n$ — цвета лампочек в гирлянде ($1 \le c_i \le k$).

В каждой из следующих q строках задано одно число d_i — цвет группы лампочек, которая поменяла своё состояние ($1 \le d_i \le k$).

Output

Выведите q чисел, по одному на строке. В строке номер i должно быть выведено количество «островков света» в гирлянде после i-го запроса.

Example

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

Problem E. Спрут

Input file:	*стандартный ввод*
Output file:	*стандартный вывод*
Time limit:	2 секунды
Memory limit:	256 мебибайт

Влад занимается подводной съёмкой. Больше всего ему нравится фотографировать спрутов и осьминогов. Например, знаете ли вы, что у осьминогов три сердца? А то, что они могут изменять форму и окрас тела? А вот вам ещё удивите... Влад, отойди, наконец, от клавиатуры, хватит писать про спрутов!

Влад не только фотограф, но и отличный программист. Он разработал специальную программу для автоматического распознавания спрутов на фотографиях. На вход программа получает фотографию, а на выходе выдаёт представление спрута в виде *неориентированного графа*. Неформально, неориентированный граф — это множество точек и множество рёбер, их соединяющих. Граф, полученный распознаванием изображения спрута, всегда имеет специфический вид: три или более точки образуют *тело*, а остальные точки входят в состав *щупалец*, прикреплённых к точкам тела, причём к каждой точке тела прикреплено **не более одного** щупальца. Тело — это такая последовательность точек, в которой любые две соседние точки, а также первая и последняя точки, соединены ребром, при этом из каждой точки выходят или два ребра, или три (если к ней прикреплено щупальце). Щупальце — это такая последовательность точек, в которой любые две соседние точки соединены ребром, а из каждой точки выходят или два ребра, или одно (в каждом щупальце точка с одним ребром ровно одна). Кроме того, в таком графе между любыми двумя точками не более одного ребра, а любое ребро соединяет две различные точки.

Собрав все свои фотографии, Влад запустил процесс распознавания и, после нескольких часов ожидания, получил результат. Однако из-за хитрого бага в коде во все графы добавилось одно лишнее ребро! После перечитывания исходного кода своей программы Влад понял, что в каждый граф добавилось ровно одно ребро между двумя различными точками, между которыми ранее не было ребра. Помогите Владу найти лишнее ребро, и в благодарность он расскажет вам ещё много удивительных фактов про спрутов!

Input

В первой строке даны целые числа n и m, разделённые пробелом, — количество точек и количество рёбер в графе ($1 \le n, m \le 10^5$).

В следующих m строках описаны рёбра графа. Каждое ребро задаётся двумя различными целыми числами, разделёнными пробелом, — номерами точек, которые соединяет это ребро. Точки пронумерованы, начиная с единицы. Каждая пара точек встречается не более одного раза.

Гарантируется, что данный граф был получен добавлением ровно одного ребра к графу, полученному распознаванием изображения спрута.

Output

В единственной строке выведите два целых числа, разделённых пробелом, — номера точек, между которыми нужно удалить ребро. Номера можно выводить в любом порядке.

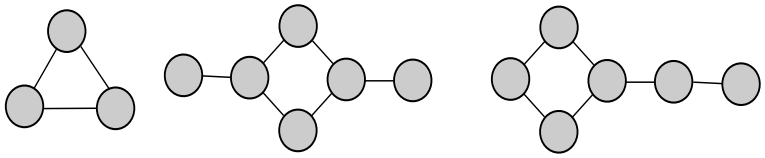
Если существует несколько вариантов ответа, выведите любой.

Example

стандартный ввод	стандартный вывод
7 8 1 2 2 3 3 4 4 1 3 5 5 6 5 7 3 6	3 2

Note

Примеры графов, которые **являются** корректными представлениями спрута:



Примеры графов, которые **не являются** корректными представлениями спрута:

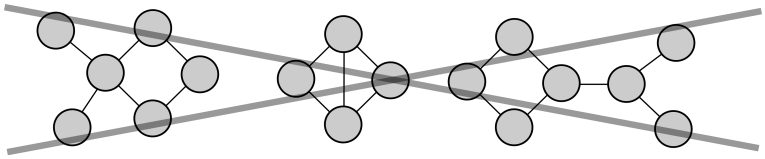
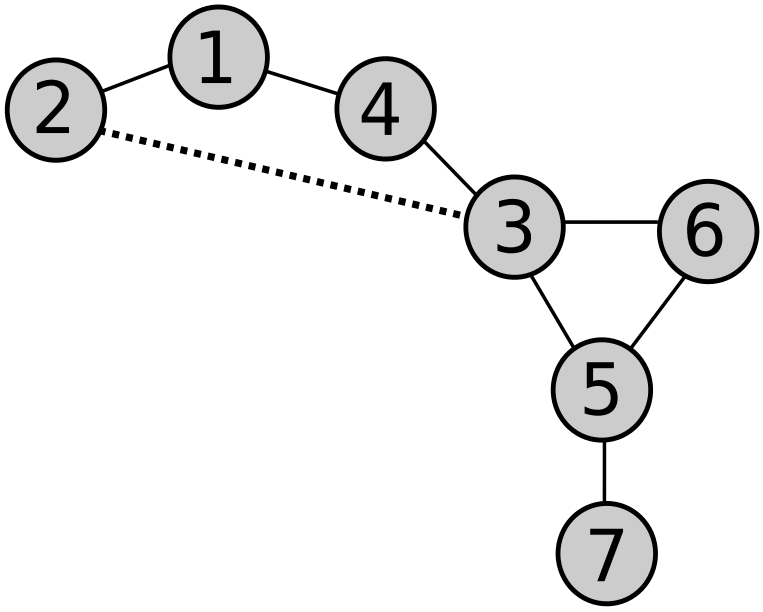


Иллюстрация первого теста:



Problem J. Читаемость

Input file:	*стандартный ввод*
Output file:	*стандартный вывод*
Time limit:	2 секунды
Memory limit:	256 мебибайт

Дана последовательность из n чисел от 1 до 10^9. Необходимо найти перестановку этих чисел, такую, что:

1. В каждой паре соседних чисел есть одно четное и одно нечетное
2. Сумма модулей разностей старой позиции и новой позиции по всем числам минимальна
3. Перестановка лексикографически минимальна среди всех перестановок, удовлетворяющих пунктам 1 и 2.

Гарантируется, что ответ существует.

Input

В первый строке дано число n ($1 \le n \le 10^5$).

Во второй строке следуют n чисел, разделенных пробелом — изначальная последовательность.

Output

Выведите n чисел, разделенные пробелом — перестановку последовательности.

Example

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

Problem K. Robotobor

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

После успеха одного из предыдущих проектов Вам предложено место CEO уральского филиала компании Rapid City Dynamics. На новом месте Вы решили начать разработку нового проекта — палиндробота.

Палиндробом может передвигаться по шахматной доске, состоящей из $n \times m$ клеток. Некоторые клетки заняты, так что робот не может в них зайти. Каждую секунду робот передвигается в клетку, которая имеет общую сторону с текущей клеткой.

Передвижение робота контролируется программой. Программа состоит из нуля или более строк. Каждая строка непуста и состоить из символов "`U`" (вверх), "`D`" (вниз), "`L`" (влево) или "`R`" (вправо). Каждый символ обозначает, что робот должен передвинуться в соседнюю клетку в соответствующем направлении. Программа выполняется построчно, каждая строка выполняется слева направо.

Программа является корректной тогда и только тогда, когда соблюдаются следующие требования:

- Во время выполнения программы робот не попадает на занятые клетки и не выходит за пределы доски.
- Длина каждой строки не превосходит 100 символов.
- Каждая строка является палиндромом, то есть читается одинаковым образом слева направо и справа налево.

Ваша задача — найти корректную программу, которая переводит палиндробота из клетки S в клетку F и состоит из наименьшего возможного количества строк. Обратите внимание, что минимизировать суммарную длину строк **не обязательно**.

Input

Первая строка входа содержит два целых числа n и m — количество строк и количество столбцов доски ($1 \le n, m \le 50$).

Последующие n строк задают доску. Каждая из этих строк содержит m символов — или символ "`.`" (пустое поле), или "`#`" (занятая клетка), "`S`" (начальная клетка) или "`F`" (конечная клетка). Гарантируется, что начальная и конечная клетки не заняты и что доска содержит ровно один символ "`S`" и ровно один символ "`F`".

Output

Если программы, удовлетворяющей условию задачи, не существует, выведите -1.

Иначе выведите соответствующую программу. Первая строка вывода должна содержать количество строк в программе k. Каждая из последующих k строк задаёт одну строку программы. Если корректных ответов несколько, выведите любой.

Examples

standard input	standard output
3 5 S.... .#... ..#F.	2 RR DRD
2 2 F# #S	-1

Problem L. Graph Isomorphism

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Даны два неориентированных графа $A = (V_A, E_A)$ и $B = (V_B, E_B)$, где набор вершин графа A есть $V_A = \{a_1, a_2, a_3, \ldots, a_{n_A}\}$, а набор вершин графа B есть $V_B = \{b_1, b_2, b_3, \ldots, b_{n_B}\}$. Графы A и B являются изоморфными тогда и только тогда, когда

1. количество рёбер графа A равно количеству рёбер графа B и количество вершин графа A равно количеству вершин графа B.
2. Существует такая биективная функция $f : V_A \to V_B$, что $\{u, v\} \in E_A$ тогда и только тогда, когда $\{f(u), f(v)\} \in E_B$.

Иначе говоря, можно перенумеровать вершины графа A и получить граф B.

Ваша задача — проверить, что два простых неориентированных графа из трёх вершин изоморфны.

Input

Первая строка входа содержит одно целое число T ($T \le 100$) — количество тестовых примеров.

Каждый тестовый пример начинается строкой, содержащей одно целое число m — количество рёбер m ($0 \le m \le 3$) в первом графе из 3 вершин (пронумерованных целыми числами от 1 до 3). Каждая из последующих m строк содержит по два различных целых числа u, v — номера вершин, соединённых ребром ($u \ne v$, $u, v \in \{1, 2, 3\}$).

Гарантируется, что любые две вершины соединены не более, чем одним ребром.

Далее следует описание второго графа в аналогичном формате.

Output

Если два графа изоморфны, выведите "`yes`". Иначе выведите "`no`".

Example

standard input	standard output
3 3 1 2 2 3 3 1 3 1 3 2 1 3 2 2 1 2 1 3 0 1 2 3 1 1 2	yes no yes

Problem M. Hamming Distance

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Расстояние Хэмминга $d_H(\vec{v}, \vec{u})$ между двумя n-мерными векторами $\vec{v} = (v_1, \ldots, v_n)$ и $\vec{u} = (u_1, \ldots, u_n)$ определяется как $d_H(\vec{v}, \vec{u}) = |\{i : v_i \neq u_i \text{ and } i \in \{1, \ldots, n\}\}|$, то есть как количество позиций, в которых соответствующие элементы различаются. Например, расстояние Хэмминга между $(1, 2, 3, 4, 5)$ и $(1, 0, 0, 4, 5)$ равно 2, так как эти две вектора различаются только во второй и третьей позициях.

Напишите программу, которая бы вычислаля расстояние Хэмминга между двумя заданными n-мерными векторами.

Input

Первая строка входа содержит одно целое число T ($T \leq 100$) — количество тестовых примеров.

Каждый тестовый пример состоит из трёх строк. Первая строка каждого тестового примера содержит целое число n ($0 < n \leq 50$) — размерность векторов. Вторая строка содержит n целых чисел $v_1, \ldots, v_n$, а третья строка содержит n целых чисел $u_1, \ldots, u_n$. Гарантируется, что $v_1, \ldots, v_n, u_1, \ldots, u_n \in \{0, 1, \ldots, 99\}$.

Output

Для каждого тестового примера выведите расстояние Хэмминга между $(v_1, \ldots, v_n)$ и $(u_1, \ldots, u_n)$.

Example

standard input	standard output
2 3 1 2 3 3 2 1 4 1 0 1 0 1 0 1 1	2 1

Problem N. X eNohpi

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 Mebibytes

«X eNohpi» — это новый гаджет от компании Elppa. У каждого X eNohpi есть серийный номер — целое число в интервале от 1 до 10^5. Компания Elppa проводит рекламную акцию: если покупается несколько X eNohpi, сумма серийных номеров которых делится на заданное целое положительное число M, то покупатель получает значительную скидку.

Вам заданы номера X eNohpi, доступных на складе некоторого магазина. Ваша задача — найти размер максимальной партии X eNohpi, которую можно купить с вышеуказанной скидкой в данном магазине. При этом покупка производится ровно один раз.

Input

В первой строке входного файла заданы два целых числа N и M ($1 \le N \le 500,\ 1 \le M \le 10^5$) — общее количество iPad на складах магазина и «бонусное» число M. Во второй строке заданы N различных целых чисел $S_1, \ldots, S_N$ ($0 \le S_i \le 10^5$) — серийные номера X eNohpi, присутствующих на складе магазина. Гарантируется, что хотя бы для одного подмножества X eNohpi со склада сумма серийных номеров делится на M (то есть скидка применима).

Output

Выведите одно число — максимальное количество X eNohpi из данного магазина, сумма серийных номеров которых делится на M.

Example

standard input	standard output
3 5 1 8 6	3
6 9 8 6 4 1 2 3	5

Problem O. Team Composition

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Как известно, в команде, участвующей в ICPC, должно быть три участника, которые работают во время контеста за одним компьютером.

Многие начинающие тренеры считают, что наиболее сильная команда получится, если собрать трёх студентов университета, имеющих максимальный рейтинг на `Topforces`.

Напишите программу, которая по заданным рейтингам участников поможет этим тренерам выбрать команду.

Input

Первая строка входа содержит одно целое число T ($T \le 100$) — количество тестовых примеров.

Каждый тестовый пример состоит из двух строк. Первая строка содержит одно целое число n ($3 \le n \le 60$), указывающее, что n студентов университета планируют участвовать в ICPC в этом сезоне. Вторая строка содержит n целых чисел $r_1, \ldots, r_n$, разделённых пробелами ($r_1, \ldots, r_n \in [1, 100]$) — рейтинги студентов на `TopForces`.

Output

Для каждого тестового примера в отдельной строке выведите максимальный суммарный рейтинг участников команды на `Topforces`, который может получить тренер.

Example

standard input	standard output
3 3 1 2 7 6 1 2 3 4 5 6 9 77 13 22 78 54 65 7 79 99	10 15 256

Problem P. Boxing

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 Mebibytes

В любительском боксе финалисты определяются в однокруговом турнире на выбывание. Победитель финала получает золотые медали, проигравший — серебряные.

Бронзовые медали разыгрываются в отдельном турнире, в котором участвуют те, кто на протяжении основного турнира проиграл одному из финалистов.

По заданным результатам основного турнира вычислите победителя, серебряного призёра и участников «бронзового» раунда.

Input

Входной файл состоит из не более, чем 60 тестовых примеров. Каждый тестовый пример начинается целым числом N. $N = 0$ сигнализирует о завершении входного файла (этот тестовый пример не должен быть обработан). В остальных случаях $2 \le N \le 6$.

Далее заданы $2^N - 1$ строк, состоящих из пар имён (строк длины не менее 1 и не более 10, состоящих из строчных и заглавных латинских букв), обозначающих, что участник, чьё имя указано первым, выиграл свой бой у участника, чьё имя указано вторым.

Гарантируется, что данные корректно описывают однокруговой турнир на выбывание и что имена никаких двух участников не совпадают.

Output

Для каждого тестового примера выведите в соответствии с примером `'Gold: '`, затем имя победителя, затем, на следующей строке, `'Silver: '` и имя победителя, затем, на третьей и последней для данного примера строке, `'Bronze round: '` и отсортированный по алфавиту (при сортировке любая строчная буква идёт после любой прописной) список участников турнира за бронзовые медали.

Example

standard input	standard output
3 A aB CC D F E H G A CC H F A H 0	Gold: A Silver: H Bronze Round: CC F G aB