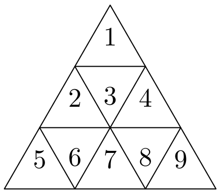
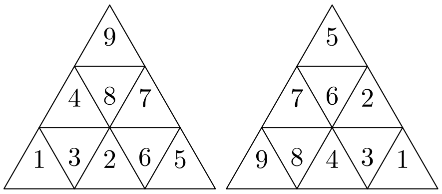
Задача A. Треугольный ключ

Имя входного файла:	`triangle.in`
Имя выходного файла:	`triangle.out`
Ограничение по времени:	2 секунд
Ограничение по памяти:	256 мегабайт
Описание тестов:	20 тестов по 5 баллов

У Индианы Джонса есть ключ от двери, ведущей к тайным богатствам инков. Ключ имеет форму правильного треугольника, который, в свою очередь, разбит на n^2 маленьких правильных треугольников, в каждом из которых написана одна десятичная цифра. Пример ключа приведен ниже на рисунке.



Джонс вставил треугольник в соответствующее треугольное отверстие в двери, и…она не открылась. Возможно, он вставил ключ неправильно, ведь его можно вставить тремя способами. Из-за того, что у Джонса осталась лишь одна попытка для того, чтоб открыть двери (в случае неудачи его убьют уже настигающие его кровожадные инки), он хочет знать как будет выглядеть ключ при его вставке в дверь другими двумя способами. После этого он сделает правильный выбор. Вас, как одного из двух своих помощников, он попросил показать ему только один из двух возможных вариантов. Внешний вид ключа, приведенного выше, при повороте против и по часовой стрелке изображен на рисунке ниже.



Поскольку вы не хотите смерти Индианы и хотите получить свою долю сокровищ, вам придется помочь ему!

Формат входного файла

В первой строке входного файла содержится одно целое число n ($1 \le n \le 100$). В следующих n строках описан сам треугольник. Строка входного файла, имеющая номер $i+1$, содержит $2i-1$ цифру — содержание i-й строки ключа-треугольника.

Последняя строка содержит слово `counterclockwise`, если треугольник необходимо повернуть против часовой стрелки, и `clockwise` в противном случае.

Формат выходного файла

Выведите вид ключа при его повороте в требуемую сторону. Треугольник опишите в том же формате, в котором это сделано во входном файле.

Примеры

triangle.in	triangle.out
3 1 2 3 4 5 6 7 8 9 counterclockwise	9 4 8 7 1 3 2 6 5
3 1 2 3 4 5 6 7 8 9 clockwise	5 7 6 2 9 8 4 3 1

Примечание

Решения, которые обрабатывают поворот только в одну сторону, будут оцениваться из 50 баллов.

Задача B. Держать строй - 2

Имя входного файла:	`inverse.in`
Имя выходного файла:	`inverse.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт
Описание тестов:	20 тестов по 5 баллов

В воинской части города Шатров продолжаются занятия по строевой подготовке. Молодой прапорщик Андрей Юрьевич уже приноровился проводить эти занятия, но тут его начальник Павел Андреевич стал давать ему новые, более сложные задания.

Как известно, обычно все солдаты выстраиваются в шеренгу по росту, начиная с самого низкого. Но Павлу Андреевич это показалось слишком скучным и он потребовал от Андрея Юрьевича выстроить солдат в другом порядке. Случайный порядок ему показался слишком хаотичным, поэтому он потребовал построения, в котором будет ровно k *инверсий*. *Инверсией* называется такая пара солдат, что более высокий из них стоит в шеренге раньше более низкого.

К счастью для Андрея Юрьевича, все солдаты разного роста и для каждого он знает, какой он по росту среди всех солдат. Исходя из этого, каждому солдату Андрей Юрьевич присвоил номер: первый номер получил самый низкий, а последний — самый высокий солдат.

Например, в построении $(3, 1, 2)$ есть две инверсии — пары $(3, 1)$ и $(3, 2)$, в которых более высокий солдат стоит раньше более низкого.

Андрей Юрьевич поручил Вам помочь ему в построении солдат в требуемом порядке, и для этого вы должны написать программу, находящую расстановку n солдат в порядке, в котором есть ровно k инверсий.

Формат входного файла

Первая и единственная строка входного файла содержит два целых числа n и k $(1 \le n \le 100\,000)$ — количество солдат в строю и требуемое количество инверсий.

Гарантируется, что существует построение солдат с требуемым числом инверсий, то есть $0 \le k \le \frac{n(n-1)}{2}$.

Формат выходного файла

В выходной файл выведите n целых чисел — построение солдат в требуемом порядке.

Примеры

inverse.in	inverse.out
3 2	3 1 2
3 0	1 2 3

Примечание

Решения, работающие при $n \le 10$, будут оцениваться из 30 баллов.

Решения, работающие при $n \le 1000$, будут оцениваться из 60 баллов.

Задача C. Волшебные грибы

Имя входного файла:	`mushrooms.in`
Имя выходного файла:	`mushrooms.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт
Описание тестов:	50 тестов по 2 балла

На планете Руук существует Большая Корпорация Маленьких Фей. Одним из видов деятельности, которым испокон веков занимаются ее сотрудницы, является посадка грядок с волшебными грибами. Каждый день, начиная с самого первого дня существования этой корпорации, феи создают одну новую грядку грибов. После этого с новой грядки два дня можно собирать споры, которыми размножаются эти грибы, а потом грядка будет поставлять уже только сам продукт — грибы.

Таким образом, если обозначить количество грибов, посаженных на грядке, созданной в день номер i, как c_i, то оно будет считаться по формуле $c_i = c_{i-1} + c_{i-2}$. Так, в первый и второй дни было посажено по одному грибу, в третий — два, в четвертый — три, в пятый — пять и так далее.

Волшебные грибы являются самыми ценными сувенирами, которые путешественник может привезти с планеты Руук. Поэтому первым, что делает любой приезжий, становится поиск грядки с волшебными грибами. Однако, в последнее время все чаще стали появляться сообщения о поддельных волшебных грибах. Тщательное расследование показало, что это является следствием действий Маленькой Корпорации Больших Фей, которая сажает грядки с грибами, внешне не отличимыми, но далеко не такими ценными, как волшебные. Причем, создавая очередную грядку, эти феи сажают туда такое количество грибов, какое их соперницы никогда не сажали и не смогут посадить.

Казалось бы, после выяснения этого факта отличать волшебные грядки от поддельных стало просто. Но обе корпорации существуют достаточно давно, количество грядок и грибов на них давно превысило все разумные пределы. Вас попросили написать программу, по количеству грибов на грядке сообщающую, является ли эта грядка волшебной.

Формат входного файла

Первая строка входного файла содержит одно число N ($1 \le N \le 1\,000\,000$) — количество исследуемых грядок. Следующие n строк содержат по одному целому числу a_i — количества грибов на исследуемых грядках. Все a_i положительны. Размер входного файла не превышает 1 Мб.

Формат выходного файла

Для каждого числа, данного во входном файле, выведите «`Yes`», если грядка с таким количеством грибов является волшебной, и «`No`» — если не является. Ответы разделяйте переводами строк.

Примеры

mushrooms.in	mushrooms.out
8 1 2 3 4 5 6 7 8	Yes Yes Yes No Yes No No Yes

Примечание

Решения, работающие для чисел, не превышающих $2^{63} - 1$, будут оцениваться из 30 баллов.

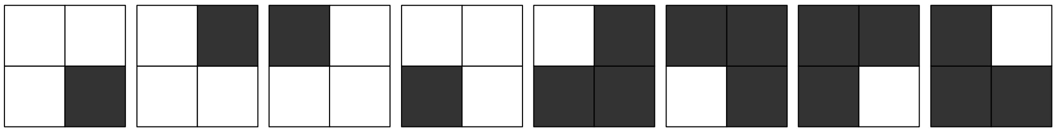
Решения, также работающие для входных данных, не превышающих 15 килобайт, будут оцениваться из еще 30 баллов.

Задача D. Сложность

Имя входного файла:	`complexity.in`
Имя выходного файла:	`complexity.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт
Описание тестов:	50 тестов по 2 балла

Федот — дизайнер, ему поручена ответственная работа по художественной укладке плитки черного и белого цвета. Его последнее задание — уложить черные и белые плитки в квадрате $n \times n$.

Федот любит свою работу и всегда тщательно готовится к каждому проекту. Федот считает, что два квадрата *похожи*, если один из них можно получить из другого несколько раз заменив цвета в какой-то строке или столбце на противоположные.



Все эти квадраты являются похожими, и никакой другой не похож на них

Федот заметил, что клиенты никогда не смотрят на всю работу целиком, обычно поле их зрения ограничивается квадратом $k \times k$. Для оценки эскизов он ввел специальную величину — *сложность*. Она равна числу пар не похожих друг на друга квадратов $k \times k$, которые встречаются в картине.

Методом проб и ошибок Федот установил, что клиентам нравятся картины определенной сложности. Слишком большая сложность похожа на хаос, а слишком малая навевает скуку, считает Федот.

Прежде чем класть плитку, Федот подготовил несколько эскизов. Помогите Федоту вычислить их сложность.

Формат входного файла

Первая строка входного файла содержит два целых числа n и k ($1 \le k \le n \le 500$). Следуюшие n строк содержат описание эскиза. Каждая из них имеет длину n и состоит из символов b и w, которые соответствуют белому и черному цветам плиток.

Формат выходного файла

В первой строке выходного файла выведите одно целое число q — сложность картины.

Примеры

complexity.in	complexity.out

`2 1`
`bw`
`wb`
	`0`

`3 2`
`bwb`
`wbb`
`bbw`
	`3`

Примечание

Решения, работающие при $n \le 10$, будут оцениваться из 30 баллов.

Решения, работающие при $n \le 100$, будут оцениваться из 60 баллов.