

Problem K. Ants

Input file: **ants.in**
Output file: **ants.out**
Time limit: 2 seconds
Memory limit: 64 Mebibytes

При движении муравьи образуют колонны так, что каждый муравей, кроме первого находится строго позади другого муравья. В случае если две колонны встречаются, муравьи начинают перепрыгивать друг через друга.

С момента встречи колонн, каждую секунду происходит следующее. Если какие-то два муравья, двигающиеся друг навстречу другу, оказываются рядом, то они меняются позициями (один перепрыгивает через другого). Восстановите порядок муравьёв по прошествии T секунд.

Input

В первой строке входного файла содержатся два целых числа N_1 и N_2 — количество муравьёв в первой и во второй колоннах соответственно. Следующие две строки задают порядок муравьёв в первой и второй колоннах (начиная с лидера колонны). Каждый муравей обозначен заглавной латинской буквой, причём никакие два муравья не обозначены одной и той же буквой вне зависимости от того, в какой колонне они идут. Последняя строка входного файла содержит целое число T ($0 \leq T \leq 50$).

Output

В выходной файл выведите в одной строке порядок муравьёв после T секунд. Считать, что первая колонна муравьёв движется слева направо, а вторая — справа налево.

Examples

<code>ants.in</code>	<code>ants.out</code>
3 3 ABC XYZ 0	CBAXYZ
3 3 ABC XYZ 2	CXBYAZ
3 4 XLQ CRUZ 3	CQRLUXZ

Problem L. Bard

Input file: **bard.in**
Output file: **bard.out**
Time limit: **2 seconds**
Memory limit: **64 Mebibytes**

Каждый вечер жители небольшой деревни собираются у костра и поют песни. Среди жителей есть бард, который, если он присутствует у костра, исполняет новую песню, которую ещё никто из жителей не слышал, после чего жители расходятся по домам. Если бард отсутствует, то жители поют друг другу все песни барда, которые они уже слышали.

По заданному списку жителей, присутствовавших у костра в течение E последовательных вечеров, выведите список тех жителей, которые слышали знают все песни, исполнявшиеся за это время.

Input

Первая строка входного файла содержит целое число N , $1 \leq N \leq 100$ — количество жителей. Жители занумерованы от 1 до N , причём первый номер присвоен барду. Вторая строка содержит целое число E , $1 \leq E \leq 50$ — количество вечеров. Каждая из последующих E строк содержит список жителей, которые были у костра в соответствующий вечер. Каждая строка начинается с целого положительного числа K , $2 \leq K \leq N$ — количества присутствующих в этот вечер жителей, далее идут K чисел — номера жителей. Гарантируется, что бард появится у костра как минимум один раз.

Output

Выведите по одному на каждой строке в порядке возрастания, начиная с барда, номера тех жителей, которые знают все песни, исполненные бардом за E вечеров.

Example

<code>bard.in</code>	<code>bard.out</code>
4	1
3	2
2 1 2	4
3 2 3 4	
3 4 2 1	
8	1
5	2
4 1 3 5 4	6
2 5 6	8
3 6 7 8	
2 6 2	
4 2 6 8 1	

Problem M. Circles

Input file: `circles.in`
Output file: `circles.out`
Time limit: 2 seconds
Memory limit: 64 Mebibytes

В 19 веке немецкий математик Герман Минковский разработал геометрию, в которой расстояние между двумя точками $A(x_1, y_1)$ и $B(x_2, y_2)$ определяется следующим образом:

$$D(A, B) = |x_1 - x_2| + |y_1 - y_2|$$

Остальные определения такие же, как и в евклидовой геометрии, включая и определение окружности:

Окружностью называется геометрическое место точек на плоскости, расположенных на фиксированном расстоянии (называемом радиусом) от фиксированной точки (называемой центром окружности).

По заданному радиусу R выведите площадь круга в евклидовой геометрии и в геометрии Минковского

Input

Во входном файле задано одно число — радиус R ($1 \leq R \leq 10000$)

Output

Первой строкой выходного файла выведите площадь круга заданного радиуса в евклидовой геометрии, второй — в геометрии Минковского с точностью 10^{-4} .

Examples

	<code>circles.in</code>	<code>circles.out</code>
1		3.141593 2.00
21		1385.442360 882.00000

Problem N. Guards

Input file: guard.in
Output file: guard.out
Time limit: 2 seconds
Memory limit: 64 Mebibytes

Около военной базы расположена система траншей. Каждую из траншей можно представить как отрезок. Ночью были выставлены три часовых. Два часовых могут видеть друг друга, если существует траншея (или цепочка траншей), полностью закрывающая отрезок, который соединяет точки расположения часовых, и между ними нет третьего часового.

С точки зрения безопасности, часовые должны быть расставлены таким образом, чтобы каждый часовой видел двух других. Сколькими возможными способами это можно сделать?

Input

Первая строка входного файла содержит целое число N ($1 \leq N \leq 20$) — количество траншей. Каждая из последующих N строк содержит описание одной траншеи: четыре целых положительных числа X_1, Y_1, X_2, Y_2 (каждое из них не превосходит 1000), где X_1 и Y_1 — координаты одного конца траншеи, а X_2 и Y_2 — координаты другого конца траншеи.

Output

Выведите количество способов расстановки часовых.

Examples

guard.in	guard.out
6 0 0 1 0 0 0 0 1 1 0 1 1 0 1 1 1 0 0 1 1 1 0 0 1	8
4 5 1 7 1 1 1 5 1 4 0 4 4 7 0 3 4	1

Problem O. Polygon

Input file: polygon.in
Output file: polygon.out
Time limit: 2 seconds
Memory limit: 64 Mebibytes

Рассмотрим выпуклый N -угольник, такой, что никакие три его диагонали не пересекаются в одной точке внутри N -угольника. Найдите количество точек пересечения диагоналей для такого N -угольника.

Input

Входной файл содержит одно число N , $3 \leq N \leq 100$.

Output

Выведите одно число — количество точек пересечения диагоналей.

Examples

polygon.in	polygon.out
3	0
4	1
6	15

Problem P. Race

Input file: `race.in`
Output file: `race.out`
Time limit: 2 seconds
Memory limit: 64 Mebibytes

В некоторой стране проводится велогонка. В стране есть N городов (занумерованных от 1 до N). Между этими городами есть M дорог с односторонним движением. Гонка начинается в городе с номером 1 и заканчивается в городе с номером 2. Сколько существует различных маршрутов гонки? (Два маршрута считаются различными, если последовательности дорог, из которых состоит маршрут, различны).

Input

Первая строка входного файла содержит два целых числа N и M ($1 \leq N \leq 10^4, 1 \leq M \leq 10^5$) — количество городов и дорог соответственно. Каждая из последующих M строк содержит два различных целых числа A и B , задающих дорогу между городами A и B . Два города могут быть соединены более, чем одной дорогой.

Output

Выведите количество различных маршрутов гонки. Если число содержит более 9 знаков, выведите последние 9 цифр ответа. Если маршрутов бесконечно много, выведите `'inf'`.

Example

<code>race.in</code>	<code>race.out</code>
6 7 1 3 1 4 3 2 4 2 5 6 6 5 3 4	3
6 8 1 3 1 4 3 2 4 2 5 6 6 5 3 4 4 3	inf

Problem Q. Square

Input file: `square.in`
Output file: `square.out`
Time limit: 2 seconds
Memory limit: 64 Mebibytes

Задана матрица N , состоящая из R строк и C столбцов. Элементами матрицы являются только числа 0 и 1. Найдите размер стороны такой наибольшей квадратной подматрицы матрицы N , что она содержит более одного числа и переходит в себя, будучи повернута на 180 градусов.

Input

Во входном файле заданы два целых числа R и C — соответственно количество строк и столбцов в матрице $1 \leq R, C \leq 300$. В последующих R строках по C символов в каждой задаётся сама матрица N .

Output

Выведите размер стороны наибольшей подматрицы с описанными в условии свойствами, или -1 , если такой подматрицы не существует.

Example

<code>square.in</code>	<code>square.out</code>
<code>3 6 101010 111001 101001</code>	<code>3</code>
<code>3 3 101 111 100</code>	<code>-1</code>

Problem R. Table

Input file: table.in
Output file: table.out
Time limit: 2 seconds
Memory limit: 64 Mebibytes

Артур собирается отпраздновать новоселье и хочет пригласить на празднование как можно больше гостей. Для этого он заказал большой прямоугольный стол, за которым он сможет сидеть вместе с гостями.

Количество человек, которые могут сидеть за столом, равно его периметру. Артур собирается выбрать стол так, чтобы он помещался в его комнате и чтобы он мог пригласить с собой на обед максимальное количество гостей. Стороны стола при этом должны быть параллельны стенам комнаты.

По заданному плану комнаты найдите максимальное количество гостей, которых Артур сможет позвать на новоселье.

Input

Первая строка входного файла содержит два целых числа R и C ($1 \leq R, S \leq 400$) — размеры комнаты, в которой будет размещён стол. Далее следует план комнаты, разделенной на единичные квадраты. Каждая из последующих R строк содержит ровно S символов. Если единичный квадрат свободен, он обозначается точкой ('.'), если занят, обозначается буквой 'X'.

Артур должен поставить стол (прямоугольник с целыми сторонами) так, чтобы он закрывал только свободные квадраты.

Output

Выведите количество гостей, которых Артур может пригласить на новоселье и усадить с собой за новый стол.

Examples

table.in	table.out
2 2 . .	7
4 4 X.XX X..X ..X. ..XX	9
.X. X.X .X.	3