Problem K. Ants

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

When moving, ants form rows so that each ant except the first is behind another ant. It is not widely known what happens when two rows of ants moving in opposite directions run into each other in a passage too narrow for both rows to pass through. One theory says that, in that situation, ants will jump over each other. From the moment the rows meet, each second every ant jumps over (or gets jumped over, as they agree upon) the ant in front of himself so that the two ants swap places, but only if the other ant is moving in the opposite direction. Find the order of the ants after T seconds.

Input

The first line contains two integers N_1 and N_2, the numbers of ants in the first and second rows, respectively. The next two rows contain the orders of ants in the first and second row (first to last). Each ant is uniquely determined by an uppercase letter of the English alphabet (this letter is unique between both rows). The last line of input contains the integer T ($0 \leq T \leq 50$).

Output

Output the order of the ants after T seconds on a single line. Our viewpoint is such that the first row of ants comes from our left side and the other one from our right side.

Example

ants.in	ants.out
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

Every evening villagers in a small village gather around a big fire and sing songs. A prominent member of the community is the bard. Every evening, if the bard is present, he sings a brand new song that no villager has heard before, and no other song is sung that night. In the event that the bard is not present, other villagers sing without him and exchange all songs that they know. Given the list of villagers present for E consecutive evenings, output all villagers that know all songs sung during that period.

Input

The first line of input contains an integer N, $1 \leq N \leq 100$, the number of villagers. The villagers are numbered from 1 to N. Villager number 1 is the bard. The second line contains an integer E, $1 \leq E \leq 50$, the number of evenings. The next E lines contain the list of villagers present on each of the E evenings. Each line begins with a positive integer K, $2 \leq K \leq N$, the number of villagers present that evening, followed by K positive integers separated by spaces representing the villagers. No villager will appear twice in one night and the bard will appear at least once across all nights.

Output

Output all villagers that know all songs, including the bard, one integer per line in ascending order.

Example

`bard.in`	`bard.out`

`4`
`3`
`2 1 2`
`3 2 3 4`
`3 4 2 1`
	`1`
	`2`
	`4`

`8`
`5`
`4 1 3 5 4`
`2 5 6`
`3 6 7 8`
`2 6 2`
`4 2 6 8 1`
	`1`
	`2`
	`6`
	`8`

Problem M. Circles

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

The 19th century German mathematician Hermann Minkowski investigated a non-Euclidian geometry. In this geometry the distance between two points A (x_1, y_1) and B (x_2, y_2) is defined as:

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

All other definitions are the same as in Euclidian geometry, including that of a circle:

A circle is the set of all points in a plane at a fixed distance (the radius) from a fixed point (the centre of the circle).

We are interested in the difference of the areas of two circles with radius R, one of which is in normal (Euclidian) geometry, and the other in Minkovsky geometry. The burden of solving this difficult problem has fallen onto you.

Input

The first and only line of input will contain the radius R, an integer smaller than or equal to 10000.

Output

On the first line you should output the area of a circle with radius R in normal (Euclidian) geometry. On the second line you should output the area of a circle with radius R in taxicab geometry. Outputs within 0.0001 of the official solution will be accepted.

Examples

circles.in	circles.out
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

Near a military base there is a system of trenches, modeled as line segments on a plane. During nighttime, when most soldiers are fast asleep, three guards stand watch of the trenches. Two guards can see each other if there is a trench (or a row of trenches) along the entire straight line segment between them and there is no third guard on that line segment. For security reasons, the guards must be placed so that each guard sees the other two. How many ways can they be placed?

Input

The first line contains the integer N ($1 \le N \le 20$) — the number of trenches. Each of the next N lines contains the description of one trench: four positive integers X_1, Y_1, X_2, Y_2 (all less than or equal to 1000), where X_1 and Y_1 are coordinates of one end, while X_2 and Y_2 are coordinates of the other end of the trench.

Trenches in the input may overlap and share endpoints.

Output

Output the number of ways the guards can be placed on a single line.

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

Consider a convex polygon with N vertices, with the additional property that no three diagonals intersect in a single point. Find the number of intersections between pairs of diagonals in such a polygon.

Input

The first and only line of input contains a single integer N, $3 \le N \le 100$.

Output

Output the number of intersections on a single line.

Example

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

A bicycle race is being organized in a land far, far away. There are N town in the land, numbered 1 through N. There are also M one-way roads between the towns. The race will start in town 1 and end in town 2. How many different ways can the route be set? Two routes are considered different if they do not use the exact same roads.

Input

The first line of input contains two integers N and M ($1 \le N \le 10^4, 1 \le M \le 10^5$), the number of towns and roads. Each of the next M lines contains two different integers A and B, representing a road between towns A and B. Towns may be connected by more than one road.

Output

Output the number of distinct routes that can be set on a single line. If that number has more than nine digits, output only the last nine digits of the number. If there are infinitely many routes, output `'inf'`.

Example

race.in	race.out

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

The program memory is a matrix composed of R rows and C columns consisting only of zeroes and ones. A virus pattern is a square submatrix in memory, consisting of more than one character, that, when rotated 180 degrees looks exactly the same.

Write a program that, given the layout of the memory, outputs the size of the virus pattern. The size of the virus pattern is the number of rows (or columns) that the pattern consists of.

Input

The first will contain two integers, R and C, smaller than or equal to 300. The next R lines will each contain C characters (zero or one) with no spaces.

Output

Output the size of the largest virus pattern on a single line, or output -1 if there are no virus patterns.

Example

square.in	square.out

3 6
101010
111001
101001
	3

3 3
101
111
100
	-1

Problem R. Table

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

Arhtur has bought a house and wants to invite to dinner as many people as possible to celebrate with him. For this he needs a large rectangular wooden table for which he will sit down with his guests.

The number of people a table can accommodate is equal to its perimeter (the sum of the lengths of all four sides). Arthur wants to buy a table such that it fits in his house and that as many people as possible can sit down with him for dinner. The table must be placed so that its edges are parallel to the edges of the house.

Given the layout of the house, find the number of people Arhtur can invite to dinner.

Input

The first line contains two integers R and C ($1 \leq R, S \leq 400$) — the dimensions of the house. Each of the following R rows contains exactly S characters (without spaces), whether a square is free '`.`' or blocked '`X`'. Arhtur can put his table only in free squares.

Output

Output the number of guests Arhtur can invite to dinner after he buys his table on a single line.

Examples

table.in	table.out

`2 2`
`. .`
`. .`
	`7`

`4 4`
`X.XX`
`X..X`
`..X.`
`..XX`
	`9`

`.X.`
`X.X`
`.X.`
	`3`