1. MODULO

Given two integers A and B, A modulo B is the remainder when dividing A by B. For example, the numbers 7, 14, 27 and 38 become 1, 2, 0 and 2, modulo 3. Write a program that accepts 10 numbers as input and outputs the number of distinct numbers in the input, if the numbers are considered modulo 42.

Input

The input will contain 10 non-negative integers, each smaller than 1000, one per line.

Output

Output the number of distinct values when considered modulo 42 on a single line.

Sample tests

input 1 2 3 4 5 6 7 8 9 10 **output** 10	**input** 42 84 252 420 840 126 42 84 420 126 **output** 1	**input** 39 40 41 42 43 44 82 83 84 85 **output** 6

Clarification:
In the first example, the numbers modulo 42 are 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10.
In the second example all numbers modulo 42 are 0.
In the third example, the numbers modulo 42 are 39, 40, 41, 0, 1, 2, 40, 41, 0 and 1. There are 6 distinct numbers.

The 19th century German mathematician Hermann Minkowski investigated a non-Euclidian geometry, called the taxicab geometry. In taxicab geometry the distance between two points T1(x1, y1) and T2(x2, y2) is defined as:

$$D(T1,T2) = |x1 - x2| + |y1 - y2|$$

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 taxicab 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.

Note: Outputs within ±0.0001 of the official solution will be accepted.

Sample tests

input 1 **output** 3.141593 2.000000	**input** 21 **output** 1385.442360 882.000000	**input** 42 **output** 5541.769441 3528.000000

"Peter Pan frames" are a way of decorating text in which every character is framed by a diamond-shaped frame, with frames of neigbhouring characters interleaving. A Peter Pan frame for one letter looks like this ('X' is the letter we are framing):

```
..#..
.#.#.
#.X.#
.#.#.
..#..
```

However, such a framing would be somewhat dull so we'll frame every third letter using a "Wendy frame". A Wendy frame looks like this:

```
..*..
.*.*.
*.X.*
.*.*.
..*..
```

When a Wendy frame interleaves with a Peter Pan frame, the Wendy frame (being much nicer) is put on top. For an example of the interleaving check the sample cases.

Input

The first and only line of input will contain at most 15 capital letters of the English alphabet.

Output

Output the word written using Peter Pan and Wendy frames on 5 lines.

Sample tests

```
input
A

output
..#..
.#.#.
#.A.#
.#.#.
..#..
```

```
input
DOG

output
..#...#...*..
.#.#.#.#.*.*.
#.D.#.O.*.G.*
.#.#.#.#.*.*.
..#...#...*..
```

```
input
ABCD

output
..#...#...*...#..
.#.#.#.#.*.*.#.#.
#.A.#.B.*.C.*.D.#
.#.#.#.#.*.*.#.#.
..#...#...*...#..
```

The evil emperor Cactus has in his possession the Magic Keg and has flooded the Enchanted Forest! The Painter and the three little hedgehogs now have to return to the Beaver's den where they will be safe from the water as quickly as possible!

The map of the Enchanted Forest consists of R rows and C columns. Empty fields are represented by '.' characters, flooded fields by '*' and rocks by 'X'. Additionally, the Beaver's den is represented by 'D' and the Painter and the three little hedgehogs are shown as 'S'.

Every minute the Painter and the three little hedgehogs can move to 4 neighbouring fields (up, down, left or right). Every minute the flood expands as well so that all empty fields that have at least one common side with a flooded field become flooded as well. Neither water nor the Painter and the three little hedgehogs can pass through rocks. Naturally, the Painter and the three little hedgehogs cannot pass through flooded fields, and water cannot flood the Beaver's den.

Write a program that will, given a map of the Enchanted Forest, output the **shortest** time needed for the Painter and the three little hedgehogs to safely reach the Beaver's den.

Note: The Painter and the three little hedgehogs cannot move into a field that is about to be flooded (in the same minute).

Input

The first line of input will contain two integers, R and C, smaller than or equal to 50.
The following R lines will each contain C characters ('.', '*', 'X', 'D' or 'S'). The map will contain exactly one 'D' character and exactly one 'S' character.

Output

Output the **shortest** possible time needed for the Painter and the three little hedgehogs to safely reach the Beaver's den. If this is impossible output the word “KAKTUS” on a line by itself.

Sample tests

input	input	input
3 3 D.* ... .S. **output** 3	3 3 D.* ... ..S **output** KAKTUS	3 6 D...*. .X.X.. ....S. **output** 6

Clarification of the second sample test: The best they can do is to go along the lower border and then the left border, and get flooded one minute before reaching the den.

5. BOND

Everyone knows of the secret agent double-oh-seven, the popular Bond (James Bond). A lesser known fact is that he actually did not perform most of his missions by himself; they were instead done by his cousins, Jimmy Bonds. Bond (James Bond) has grown weary of having to distribute assign missions to Jimmy Bonds every time he gets new missions so he has asked you to help him out.

Every month Bond (James Bond) receives a list of missions. Using his detailed intelligence from past missions, for every mission and for every Jimmy Bond he calculates the probability of that particular mission being successfully completed by that particular Jimmy Bond. Your program should process that data and find the arrangement that will result in the **greatest** probability that **all** missions are completed successfully.

Note: the probability of all missions being completed successfully is equal to the product of the probabilities of the single missions being completed successfully.

Input

The first line will contain an integer N, the number of Jimmy Bonds and missions ($1 \le N \le 20$).
The following N lines will contain N integers between 0 and 100, inclusive. The j-th integer on the i-th line is the probability that Jimmy Bond i would successfully complete mission j, given as a percentage.

Output

Output the maximum probability of Jimmy Bonds successfully completing all the missions, as a percentage.

Note: Outputs within ±0.000001 of the official solution will be accepted.

Sample tests

input 2 100 100 50 50 **output** 50.000000	**input** 2 0 50 50 0 **output** 25.00000	**input** 3 25 60 100 13 0 50 12 70 90 **output** 9.10000

Clarification of the third example: If Jimmy bond 1 is assigned the 3rd mission, Jimmy Bond 2 the 1st mission and Jimmy Bond 3 the 2nd mission the probability is: 1.0 * 0.13 * 0.7 = 0.091 = 9.1%. All other arrangements give a smaller probability of success.

6. DEBUG

While debugging a program Mirko noticed that a bug in the program may be linked with the existence of so called square killers in the program memory. The program memory is a matrix composed of R rows and C columns consisting only of zeroes and ones. A square killer is a square submatrix in memory, consisting of more than one character, that, when rotated 180 degrees looks exactly the same. For example, the following matrix contains 3 square killers:

101010 111001 101001 memory	....10 ....01 ...... killer	....00. ...00. killer	101... 111... 101... killer

Mirko is wondering if there is a connection between the size of the largest square killer and the bug in the program. Help Mirko by writing a program that, given the layout of the memory, outputs the size of the largest square killer. The size of the square killer is the number of rows (or columns) that the killer consists of. In the example above the killer sizes are 2, 2 and 3, respectively.

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 ('0' or '1') with no spaces.

Output

Output the size of the largest killer on a single line, or output -1 if there are no square killers.

Sample tests

input 3 6 101010 111001 101001 **output** 3	**input** 4 5 10010 01010 10101 01001 **output** 3	**input** 3 3 101 111 100 **output** -1

1. PATULJCI

Every day, while the dwarves are busy in the mines, Snow White prepares dinner for them; seven chairs, seven plates, seven forks and seven knives for seven hungry dwarves.

One day nine dwarves came from the mines instead of seven (nobody knows how or why), each of them claiming to be one of Snow White's seven dwarves.

Luckily, each dwarf wears a hat with a positive integer less than 100 written on it. Snow White, a famous mathematician, realised long ago that the sum of numbers on the hats of her seven dwarves was exactly 100.

Write a program which determines which dwarves are legit, i.e. pick seven of nine numbers that add to 100.

Input

There are 9 lines of input. Each contains an integer between 1 and 99 (inclusive). All of the numbers will be distinct.

Note: The test data will be such that the solution is unique.

Output

Your program must produce exactly seven lines of output – the numbers on the hats of Snow White's seven dwarves. Output the numbers in any order.

Sample test data

input

```
7
8
10
13
15
19
20
23
25
```

output

```
7
8
10
13
19
20
23
```

input

```
8
6
5
1
37
30
28
22
36
```

output

```
8
6
5
1
30
28
22
```

N-puzzle is a puzzle that goes by many names and has many variants. In this problem we will use the 15-puzzle. It consists of a 4-by-4 grid of sliding squares where one square is missing. The squares are labeled with uppercase letters 'A' through 'O', with the desired layout as follows:

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	.

It can be useful (for example, when solving the puzzle using a computer) to define the "scatter" of a puzzle as the sum of distances between each square's current position and its position in the desired layout. The distance between two squares is their Manhattan distance (the absolute value of the sum of differences between the two rows and the two columns).

Write a program that calculates the scatter of the given puzzle.

Input

Four lines of input contain four characters each, representing the state of the puzzle.

Output

Output the scatter of the puzzle on a single line.

Sample test data

input

```
ABCD
EFGH
IJKL
M.NO
```

output

```
2
```

input

```
.BCD
EAGH
IJFL
MNOK
```

output

```
6
```

3. TROJKE

Mirko and Slavko are playing a new game, "Trojke" (Triplets). First they use a chalk to draw an N-by-N square grid on the road. Then they write letters into some of the squares. No letter is written more than once in the grid.

The game consists of trying to find three letters on a line as fast as possible. Three letters are considered to be on the same line if there is a line going through the centre of each of the three squares.

After a while it gets harder to find new triplets. Mirko and Slavko need a program that counts all the triplets, so that they know if the game is over or they need to search further.

Input

The first line contains an integer N ($3 \le N \le 100$), the dimension of the grid.

Each of the N following lines contains N characters describing the grid – uppercase letters and the character '.', which marks an empty square.

Output

Output the number of triplets on a single line.

Sample test data

Input

```
4
...D
..C.
.B..
A...
```

output

```
4
```

input

```
5
..T..
A....
.FE.R
....X
S....
```

output

```
3
```

input

```
10
....AB....
..C....D..
.E......F.
...G..H...
I........J
K........L
...M..N...
.O......P.
..Q....R..
....ST....
```

output

```
0
```

Mirko found a collection of N toy tanks dating back to the Second World War on his grandfather's attic. He promptly called his friend Slavko to play with him. They made a battlefield – a wooden board consisting of squares in N rows and N columns.

Each tank can be moved to one of the four neighbouring squares in a single move. A tank can shoot at any square in the same row and column. The tank is said to be guarding the row and column it is in.

Additionally, no two tanks can be in the same square at any time.

After many hours of play and two previous attempts, Mirko's mom yelled at them to come down for lunch again, and they decided to rearrange the tanks so that each tank guards a different row and column (meaning also that each row and column contains only one tank).

However, they want to do this using the minimum number of moves.

Write a program that finds the minimum number of moves required to rearrange the tanks so that each row and each column contains a single tank, and one such shortest sequence of moves.

Input

The first line of input contains the integer N ($5 \le N \le 500$).

Each of the following N lines contains two integers R and C ($1 \le R, S \le N$), the row and column of a single tank at the moment of mom's call. No two tanks are on the same square.

Rows and columns are marked 1 through N, top-down and left-to-right.

Output

Output the minimum number of moves (call this number K) on the first line.

Each of the next K lines should contain the tank being moved and the direction it is moved in, separated by a single space.

Tanks are numbered 1 through N, in the order in which they are given in the input.

The direction can be one of four uppercase letters: 'L' for left, 'R' for right, 'U' for up and 'D' for down.

Note: The sequence need not be unique.

Scoring

If both the number K and the sequence of moves are correct, your program will score full points on the test case.

If your program outputs the correct number K and does not output the sequence of moves, or the sequence of moves is incorrect, you will get 60% of the points for that test case.

Sample test data

input

```
5
1 1
1 2
1 3
1 4
1 5
```

output

```
10
1 D
2 D
3 D
4 D
1 D
2 D
3 D
1 D
2 D
1 D
```

input

```
5
2 3
3 2
3 3
3 4
4 3
```

output

```
8
1 R
1 R
2 U
2 U
4 D
4 D
5 L
5 L
```

input

```
6
1 1
1 2
2 1
5 6
6 5
6 6
```

output

```
8
2 R
2 D
3 D
3 R
4 U
4 L
5 L
5 U
```

5. BICIKLI

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 ($1 \leq N \leq 10\ 000$, $1 \leq M \leq 100\ 000$), 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".

Sample test data

input

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

output

```
3
```

input

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

output

```
inf
```

input

```
31 60
1 3
1 3
3 4
3 4
4 5
4 5
5 6
5 6
6 7
6 7
…
…
…
28 29
28 29
29 30
29 30
30 31
30 31
31 2
31 2
```

output

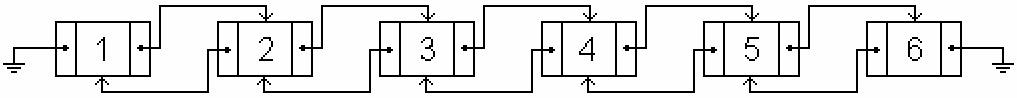
```
073741824
```

6. LISTA

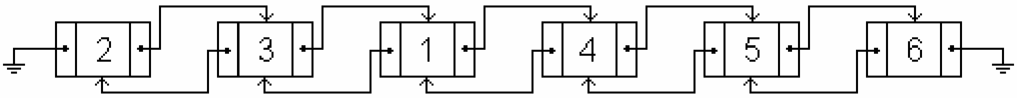
Mirko received a birthday present from his aunt in the US – a brand-new doubly-linked list (an example of which is shown in the figure below). The list contains N nodes numbered 1 through N. Two types of moves can be done on the list:

A) Move node X **in front of** node Y.

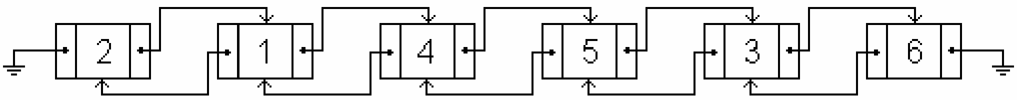
B) Move node X **after** node Y.



An example of a list with 6 nodes.



The list after the move "A 1 4".



The list after another move, "B 3 5".

Mirko played with his new toy for hours, writing down each move on a piece of paper so that he can reconstruct the list's initial state (nodes 1 through N in order from left to right).

When he decided to reconstruct the list, Mirko was astonished to find that there is no easy way to invert the moves and restore the list's initial state. Mirko cannot know where node X was prior to each move, only where it ended up.

Seeing how Mirko is still recovering from the shock, write a program that finds a minimal sequence of moves that restored the list's initial state from Mirko's logs.

Input

The first line of input contains two integers N and K ($2 \leq N \leq 500\ 000$, $0 \leq M \leq 100\ 000$), the number of nodes and the number of moves made by Mirko.

Each of the next M rows contains a description of a single move made by Mirko – the type of move ('A' or 'B') and two integers X and Y.

Output

Output the minimum number of moves (call this number K) on the first line.

Each of the next K lines should contain a description of a single move in the same format as in the input.

Note: The sequence need not be unique.

Scoring

If both the number K and the sequence of moves are correct, your program will score full points on the test case.

If your program outputs the correct number K and does not output the sequence of moves, or the sequence of moves is incorrect, you will get 60% of the points for that test case.

Sample test data

input

```
2 1
A 2 1
```

output

```
1
A 1 2
```

input

```
4 3
B 1 2
A 4 3
B 1 4
```

output

```
2
A 1 2
B 4 3
```

input

```
6 5
A 1 4
B 2 5
B 4 2
B 6 3
A 3 5
```

output

```
3
A 4 5
B 6 5
A 2 3
```