A: Drone with a camera

Memory Limit : **256 MB**

Maggy got a camera-equipped drone as a present. She wants to take photos of two nearby straight infinite roads using a drone and best road photos are taken directly above the road. She would like to make drone's route as short as possible, as she is afraid that the batteries might not last long enough. Of course, the drone must start and finish at Maggy's house! Help her to plan the shortest possible route.

Input

The first line of the input contains three integer numbers a, b, c ($-1\,000\,000 \leqslant a, b, c \leqslant 1\,000\,000$, $a \neq 0$ or $b \neq 0$), each separated by a single space. They describe the first road, which satisfies the equation $ax + by = c$. The second and last line of the input contains a similar description of the second road.

The two roads are different, i.e. they do not coincide. Furthermore, they do not pass through Maggys' house, whose coordinates are $(0, 0)$.

Output

You should write a single real number in the first and only line of the output – the smallest possible length of drone's route. The answer is accepted, if its relative or absolute error is at most 10^{-6}.

Example

Input	Output
1 0 2 -1 1 1	5.099019513593

B: Fibonaccis' vouchers

Memory Limit : **256 MB**

The school organizes a fair to commemorate Fibonaccis. Johnny is responsible for a gift shop, in which you can pay only using special vouchers, whose face values are Fibonacci numbers. Johnny has difficulty with processing such strange values and has decided to accept only exact payments with exactly k vouchers, not necessarily of different face values. Now he needs to set the prices – there are n different items at the gift shop and Johhny wants to put a different price on each of them. Sometimes there are many ways to pay one price, in such a case Johnny counts this price only once. He calculated all the prices and now he wants to verify his calculations. To make it quick, it is enough to name the last, n-th price. Help Johnny – write a program that, given n and k, computes the n-th smallest price that can be paid using exactly k vouchers.

Input

The first and only line of the input contains two natural numbers k and n ($1 \leqslant k \leqslant 100, 1 \leqslant n \leqslant 10^{18}$), separated by a single space.

Output

You should write a single natural number in the first and only line of the output – the n-th smallest number that can be paid with k (not necessarily different) vouchers whose face values are Fibonacci numbers, assuming that this number is at most 10^{18}, or `NIE` (Polish for "no"), if it is larger than 10^{18}.

Example

Input	Output
`2 11`	`13`

Using exactly 2 vouchers it is not possible to pay the prices 1 and 12.

Input	Output
`1 100`	`NIE`

The 100-th Fibonacci number is (much) greater than 10^{18}.

C: Chinese remainder theorem

Memory Limit : **256 MB**

Johnny is a computer science student. This semester he became well versed with the Chinese Remainder Theorem. While waiting for a next lecture he heard that Maggie complained that she cannot solve her homework; as he heard the familiar words "modulo" and "system of equations" he immediately offered his help to the damsel in distress. It turns out that Maggie's task is much different than those that Johnny is accustomed to solve, it is of the following form:

$$\begin{cases} a_1 & \equiv & b_1 & \pmod m \\ a_2 & \equiv & b_2 & \pmod m \\ \vdots & \vdots & & \vdots \\ a_n & \equiv & b_n & \pmod m \end{cases}$$

(where $\equiv$ means equivalence modulo) and for the given $a_1, b_1, \ldots, a_n, b_n$ Maggie should compute the largest m such that all of the equations hold. Maggie already started processing the equations and she ensured that $a_i \geqslant b_i$ for each i. Johhny cannot fail and lose his face – help him to solve the task.

Input

The first line of the input contains a single natural number n ($1 \leqslant n \leqslant 100\,000$), denoting the number of equations.

The second line contains n integers $a_1, a_2, \ldots, a_n$, each separated by a single space, these are the numbers on the left-hand sides of consecutive equations.

The third and last line contains n integers $b_1, b_2, \ldots, b_n$, each separated by a single space, these are the numbers on the right-hand sides of consecutive equations.

The inequality $0 \leqslant b_i \leqslant a_i \leqslant 10^{18}$ holds for each i ($1 \leqslant i \leqslant n$). The system of equations is nontrivial: $a_i \neq b_i$ holds for some i ($1 \leqslant i \leqslant n$).

Output

You should write a single natural number in the first and only line of the output – the largest m for which the given system of equations is satisfied.

Example

Input	Output
3 7 17 9 3 5 1	4

System of equations

$$\begin{cases} 7 & \equiv & 3 & \pmod 4 \\ 17 & \equiv & 5 & \pmod 4 \\ 9 & \equiv & 1 & \pmod 4 \end{cases}$$

is satisfied and it is easy to verify that it is not satisfied for $m > 4$.

Input	Output
3 4 6 5 2 2 2	1

System of equations

$$\begin{cases} 4 & \equiv & 2 & \pmod 1 \\ 6 & \equiv & 2 & \pmod 1 \\ 5 & \equiv & 2 & \pmod 1 \end{cases}$$

is satisfied and it is easy to verify that it is not satisfied for $m > 1$.

D: Road

Memory Limit : **512 MB**

Johnny became a controller at the Road Center. His task is to investigate the effectiveness of snow removal on a certain road during a series of blizzards. The road is divided into consecutive one kilometer-long segments, numbered with consecutive natural numbers from 1 to n. Johnny quickly got to work and gathered the information on events:

- Meteorological Center provided him with information about the blizzards; the intensity of a blizzard is determined by two parameters f, g: in i-th minute ($i \geqslant 1$) of such a blizzard $f \cdot i + g$ millimeters of snow fall everywhere on the road. Each blizzard ends in the minute preceding the first minute of the next blizzard. The time is calibrated so that the first blizzard begins at a positive minute, and in minute 0 there is no snow on the road.
- Snow Removal Center provided Johnny with information about the plows and sanders. Each route of a plow or a sander is always a fragment of a road consisting of road segments numbered with consecutive numbers. If a plow clears some road segments in t-th minute then at the end of t-th minute there is no snow on the cleared road segments. Likewise, spreading salt of quality s on some road segments in t-th minute ensures that at the end of minutes $t, t+1, \ldots, t+s$ there is no snow on those road segments. Different salts, even of the same quality, act independently and have no effect on each other; likewise, plows do not remove salt from the road.
- Road Center has sent its queries – give the thickness in millimeters of the largest snow cover at the end of a given minute at the given fragment of a road consisting of segments numbered with consecutive numbers.

Johnny pre-processed and sorted the collected data. Unfortunately, computing the answer to queries is too difficult for him. Help him! Write a program that computes the answers to queries of the Road Center.

Input

The first line of the input contains two natural numbers n and q ($1 \leqslant n \leqslant 10^9, 1 \leqslant q \leqslant 300\,000$) separated by a single space and denoting, respectively, the number of kilometer segments of the road and the number of events. In each of the following q lines there is a description of an event of one of the following four types:

- `t L a b`, which means that the plow cleared in t-th minute a road fragment consisting of segments numbered from a to b.
- `t S a b s`, meaning that a sander spreads salt of quality s in t-th minute on a road fragment consisting of segments numbered from a to b.
- `t ? a b`, meaning that the Road Center wants to know the maximum thickness of snow cover at the end of t-th minute on a road fragment consisting of segments numbered from a to b.
- `t B f g`, meaning that t-th minute is the last minute of the previous blizzard (if it exists), and $(t+1)$-th minute is the first minute of the blizzard with intensity f, g.

In all events the following conditions are met: $1 \leqslant t \leqslant 10^9, 1 \leqslant a \leqslant b \leqslant n, 1 \leqslant s, f, g \leqslant 10^9$.
In addition, the values of t for consecutive events are increasing, and the first event is always of type `B`.

Output

For each event `?` print in a separate line the remainder modulo $10^9 + 7$ of the largest thickness of snow cover at the end of the given minute at the specified road segment.

Example

Input	Output
3 4 2 B 1 2 3 ? 2 2 4 L 1 3 5 ? 1 3	3 5

The table below shows the snow coverage on each of the road segment at the end of each minute. It also shows the amount of snow that fell in each minute. The numbers in bold correspond to queries.

minute	1	2	3	snowfall
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	3	**3**	3	3
4	0	0	0	4
5	**5**	**5**	**5**	5

Till the first query $1 \cdot 1 + 2 = 3$ milimeters of snow fell on the whole road. Between the plow passage in minute 4 and the second `?` query additional $3 \cdot 1 + 2 = 5$ milimeters of snow fell.

Input	**Output**
1 3 1 B 1 1 2 B 3 3 3 ? 1 1	8

minute	1	snowfall
0	0	0
1	0	0
2	2	2
3	**8**	6

Till the `?` event the first blizzard took place for one minute and the second also took place for one minute.

Input	**Output**
5 5 1 B 1 2 2 S 1 3 5 3 ? 3 4 4 ? 1 1 10 ? 1 1	7 0 30

minute	1	2	3	4	5	snowfall
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	3	3	3
3	0	0	0	**7**	**7**	4
4	**0**	0	0	12	12	5
5	0	0	0	18	18	6
6	0	0	0	25	25	7
7	0	0	0	33	33	8
8	9	9	9	42	42	9
9	19	19	19	52	52	10
10	**30**	30	30	63	63	11

E: Evaluation

Memory Limit : **256 MB**

Octagon is in charge of many facilities of strategic importance, such as rocket silos, radars, canteens, veteran offices, uniform warehouses, ... It is of utter importance to secure the network of connections between those facilities, but on the other hand, the threat to this network should be minimized. For each bidirectional road linking different facilities, the Attack Threat Coefficient (colloquially called the coefficient) was determined. Johnny, recently employed by the Octagon, even in the middle of the night can compute a subset of roads that minimizes the sum of coefficients and still allows moving between any pair of facilities; we call a road that is in at least one such a subset a *key road*.

However, nothing can be taken for granted – coefficients may change. As part of the annual evaluation, the Octagon decided to compute for each road the highest value of x such that when its coefficient is set to x (and all other coefficients are left as they were), this road is a key road. This task was assigned to Johnny, who cannot err in such an important matter.

Input

The first line of the input contains two integers n and m ($1 \leqslant n \leqslant 100\,000$, $n - 1 \leqslant m \leqslant 1\,000\,000$), separated by a single space, denoting the number of facilities under Octagon's supervision and the number of bidirectional roads between them, respectively. The facilities are numbered with consecutive natural numbers from 1 to n.

In each of the following m lines there are three integers: a, b and c ($1 \leqslant a, b \leqslant n$, $a \neq b$, $0 \leqslant c \leqslant 10^9$), each separated by a single space, describing a bidirectional road between facilities a and b with coefficient c. Any unordered pair $\{a, b\}$ occurs at most once.

Existing roads always allow moving between any two facilities.

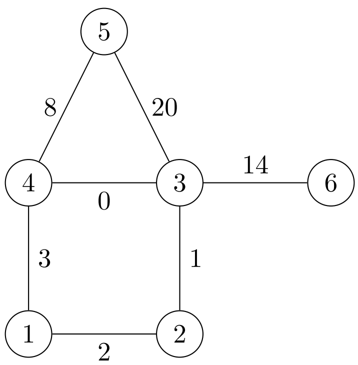
Output

You should write m lines, one for each road from the input, in the same order in which the roads are given in the input. In the line corresponding to a particular road you should write one natural number – the maximum value such that after changing the road coefficient to this value (and leaving other coefficients unchanged) this road is a key road. If this value is greater than 10^9 or can be arbitrarily large, you should write 10^9.

Example

Input	Output
6 7 1 2 2 2 3 1 3 4 0 1 4 3 3 5 20 4 5 8 3 6 14	3 3 3 2 8 20 1000000000

The road network from the example looks as follows:



F: Baking pans

Memory Limit : **256 MB**

Maggy loves to bake. She has recently bought three round baking pans of known base area. She baked cakes in the first two of them and intends to take them to a party at a friend's house. However, as carrying two baking pans is quite tricky, it is much more preferable to put both cakes into the third baking pan and carry only it. Maggy cannot really tell whether the cakes will fit. Moreover, she needs to leave some slack between the cakes, as otherwise the icings would mix. Help Maggy out – compute whether the two cakes fit into the third baking pan.

Tests' description

In the first line of the input there is a single natural number $1 \leqslant t \leqslant 1\,000$, which denotes the number of data sets that are described in the following lines, one data set per line. Each of data sets complies with the specification in *One data set*. For an individual data set you should write the result in one line, they should comply to the specification in *Result for one data set* and they should be written in the order in which data sets appear in the input.

One data set

One data set consists of three natural numbers p, d, t ($1 \leqslant p, d, t \leqslant 8 \cdot 10^9$), each separated by a single space, denoting the areas of the first, second and third baking pan, respectively.

Result for one data set

For each data set you should print a single line containing exactly one word: `TAK` (Polish for "yes") if the cakes from the first and second baking pans fit into the third baking pan together, with a non-zero slack, or `NIE` (Polish for "no") in the other case.

Example

Input	Output
3 3 9 25 4 9 25 5 9 25	TAK NIE NIE

G: Taste in art

Memory Limit : **256 MB**

Maggie has a large collection of modern paintings. She is particularly proud of those depicting colourful squares – each painting having a different number of squares. She intends to finally make an exhibition of the paintings for her friends, but their sublime tastes require special handling. Maggie knows that if among the presented paintings there are three depicting $k, 2k$ and $3k$ squares, for some k, then the whole exhibition will be perceived as *predictable*, thus dull, thus a failure. If no such a k exists, the event will be a great success. On the other hand, Maggie wants to show (off) as many paintings as possible. She spends hours on trying to choose the largest possible number of paintings, yet avoiding a predictability. Help her in this task.

Input

The first line of the input contains a single natural number n ($1 \leqslant n \leqslant 50\,000$), denoting the number of paintings with squares in Maggie's collection. In the second and last line of the input there is a sequence of n pairwise distinct natural numbers a_i ($1 \leqslant a_i \leqslant 10^9$), each separated by a single space, denoting the number of squares on each of paintings in order.

Output

You should write a single integer number in the first and only line of the output – the largest possible number of Maggie's paintings with squares that can form a non-predictable exhibition.

Example

Input	Output
3 6 9 3	2

The three paintings form a predictable exhibition, but any two of them do not.

Input	Output
6 1 2 3 4 5 6	5

The largest set of paintings that form an unpredictable exhibition includes all paintings except the second one.

H: Hobby

Memory Limit : **256 MB**

Maggy's frustration is reaching its peak lately. Johnny does not leave the house, it is hard to talk to him – all day he sits with his books with logical puzzles. Sudoku, KenKen, Kakuro, Kuromasu, ... Maggy is no longer even able to remember those bizarre names. She finally came up with a bold idea. To demonstrate the futility and repetitiveness of filling charts with numbers, she will write a program that instantly solves such puzzles.

Johnny's current favourite puzzle is Suko. In this game you fill in a 3×3 chart with numbers from 1 to 9. Chart's fields are numbered from 1 to 9: the fields in the i-th row from the top are $3i-2$, $3i-1$ and $3i$, in the left-to-right order. In addition, each field has a colour: red, green or blue. The completed chart must meet the following conditions on subsquares 2×2 and fields of each color:

- In each field there is a number between 1 and 9, no number is repeated.
- The sum of numbers in fields 1, 2, 4 and 5 is equal to v_1.
- The sum of numbers in fields 2, 3, 5 and 6 is equal to v_2.
- The sum of numbers in fields 4, 5, 7 and 8 is equal to v_3.
- The sum of numbers in fields 5, 6, 8 and 9 is equal to v_4.
- The sum of numbers in the red fields is equal to $v_{\mathtt{A}}$.
- The sum of numbers in the green fields is equal to $v_{\mathtt{B}}$.
- The sum of numbers in the blue fields is equal to $v_{\mathtt{C}}$.

Help Maggy prove to Johnny the childishness of his hobby and write a program solving Suko puzzles.

Input

The first line of the input contains three natural numbers $v_{\mathtt{A}}, v_{\mathtt{B}}$ and $v_{\mathtt{C}}$ ($1 \leqslant v_{\mathtt{A}}, v_{\mathtt{B}}, v_{\mathtt{C}} \leqslant 42$), each separated by a single space. In the second row there are four natural numbers v_1, v_2, v_3, v_4 ($10 \leqslant v_1, v_2, v_3, v_4 \leqslant 30$), each separated by a single space. The third line contains a description of the colours of the first row from the top. It is a sequence of three characters from the set {`A`, `B`, `C`}, denoting the colours red, green and blue, and describes colours of the consecutive fields in the first row of the chart. In the fourth and fifth lines there are similar descriptions of the second and third row of the chart.

For each colour there is a field of this colour in the chart.

Output

You should write three lines describing the solution of the given puzzle – the i-th line should contain a description of i-th row of the chart in the form of three digits placed in the consecutive fields, from left to right.

If there are many solutions, you can write an arbitrarily chosen one.

If there is no chart meeting the conditions, one word `NIE` (Polish for "no") should be written in the first and only line.

Example

Input	Output
8 19 18 18 18 20 25 BBB BAA CCC	537 462 198

I: Grade book

Memory Limit : **256 MB**

Maggy is a die hard – she still has a grade book and collects hand written grades from her lecturers. The lecturers' offices are numbered with consecutive natural numbers, starting from 1, and are located along an infinite corridor. The grade from each lecture can be picked up daily, but only in a specific office and only for one minute during the day. Receiving a grade takes a negligible amount of time, but moving between adjacent offices, in any direction, takes exactly 1 minute. A single lecturer can read several different lectures and then they may, although does not have to, give the grades to some of them at the same time; in such a case, receiving any number of grades still takes negligible amount of time.

Maggy attended n lectures and for each of them she knows in which office and in which minute of the day she can get the grade. Every day Maggy gets up early, so that in minute 1 she can be in any office. Help her determine the minimum number of days she needs to collect all the grades.

Input

The first line of the input contains one integer n ($1 \leqslant n \leqslant 500\,000$) denoting the number of lectures Maggy attended. In each of the next n lines there is a description of one lecture. One description consists of two integers p, t ($1 \leqslant p, t \leqslant 10^9$), separated by a single space, meaning that a grade from this lecture can be obtained daily in office p in t-th minute counted from the beginning of each day.

Output

You should write one integer number in the first and only line of the output – the minimal number of days Maggy needs to pick up all grades.

Example

Input	Output
7 2 1 1 4 3 2 1 1 4 2 5 3 1 1	3

On the first day Maggy can collect all grades from office number 1. On the second day she is able to collect grades in offices 2 and 3, and on the third day – in offices 4 and 5.

J: Identical scarves

Memory Limit : **256 MB**

Maggy makes some extra money by knitting scarves. Today she got lucky – a merchant made an offer for as many scarves as possible, under one condition, though – he wants only scarves of the same length, counted in the number of rows (otherwise they look bad in the store). He announced that he would return shortly, after exactly k moments. Maggy knows the current lengths of all scarves and both stitching and unravelling a row takes her one moment. Help Maggy – compute how many scarves of equal length she can produce.

Input

The first line of the input contains two integers n and k ($1 \leqslant n \leqslant 100\,000$, $0 \leqslant k \leqslant 10^9$), separated by a single space, denoting the number of scarves and the number of moments after which the merchant will return. In the second and last line of the input there are n natural numbers a_i ($1 \leqslant a_i \leqslant 10^9$), each separated by a single space. These are the lengths of consecutive scarves, counted in the number of rows.

Output

You should write one integer number in the first and only line of the output – the maximum number of scarves of equal length that Maggy can produce before the merchant returns.

Example

Input	Output
5 6 1 2 3 4 4	5

In 6 moments Maggy can make the lengths of all scarves equal to 2.

K: Pocket money

Memory Limit : **256 MB**

Johnny wants to become a professional road cyclist. He has even found the perfect bike, but he does not have the funds to buy it. He asked his mother for a daily pocket money allowance, to which she agreed under some terms: the initial amount of the pocket money is 0 and then every day mother will give Johnny the current amount of pocket money and then check Johnny's grades – if he brings more fives and sixes than the twos and ones, she will raise the pocket money by one, if less – she will decrease it by 1, and in the remaining case the value does not change. If the value gets negative, mother will stop paying Johnny at all and he will never buy the bike.

Years later, John recalls those events with great affection. He still remembers many details: he collected exactly the bike's cost and the final amount of his pocket money was 0. One thing he cannot recall is the actual price of the bike. He even found his grade sheets, but they are old and tattered, so sometimes he cannot read what grades he got each day. Can you help John and compute the best upper and lower bounds on the price of the bike? Unfortunately, it is possible that John made some mistakes when reading the grades and it is not possible to write a correct sequence of pocket money amounts consistent with the data that John provided.

Input

The first and only line of the input contains one string of length at most 1 000 000, consisting of symbols `+`, `-`, `0` and `_`. The characters represent changes of the pocket money in the following days: `+` means that on this day Johnny's mother raised his pocket money (by 1), `-` means that on this day she decreased it (by 1), `0` means that the pocket money amount has not changed, and `_` means that John is unable to say what happened on this particular day.

Output

You should write in the first and only line of the output two natural numbers, separated by a single space, denoting the minimal and maximal possible price of the bike, respectively.

If the given string cannot be turned into a valid sequence of changes of the pocket money amount, then you should write a single word `NIE` instead (Polish for "no").

Example

Input	Output
`+_+-0_0_+-`	`3 13`

The lowest bike price is achieved for the sequence `+-+-0000+-`, the largest for `+++-0-0-+-`.

Input	Output
`__---_`	`NIE`

Regardless of the grades in the first two days, the amount of the pocket money after the fifth day would be negative.

L: Lines

Memory Limit : **256 MB**

Johnny has learned to play tic-tac-toe. He loves the game so much that he has decided to generalize it to larger boards. Obviously he does not know gomoku, as in his generalization to win you need to fill a whole line (vertical, horizontal or one of two main diagonals) with your symbol. Designing a strategy for his game looks difficult and so Johnny has decided to deal with a simpler problem – how many tie boards of size $n \times n$ are there, that is, boards that are completely filled with noughts and crosses such that no line (vertical, horizontal or one of the main diagonals) is filled with one symbol. Johnny wants to solve the most basic variant of the problem, so he allows the proportion of noughts and crosses to be arbitrary and *does not identify* boards that can be obtained by rotation or symmetry. Even in this basic variant the problem seems to be difficult, help Johnny solve it!

Input

The first and only line of the input contains two natural numbers n and p ($1 \leqslant n \leqslant 300$, $2 \leqslant p \leqslant 10^9 + 9$) separated by a single space, where n is the size of Johnny's board and p is a prime number.

Output

You should write one natural number in the first and only line of the output – the remainder modulo p of the number of all $n \times n$ boards completely filled with noughts and crosses in which there is no line (horizontal, vertical or one of the two main diagonals) filled with just one symbol.

Example

Input	Output
3 101	32

Input	Output
4 3	2

M: Magical maze

Memory Limit : **512 MB**

Maggy's class went on an excursion to a maze! The maze has the shape of a rectangle of height n and width m meters and consists of $n \cdot m$ square chambers, each of size 1×1 meter. There is one-directional passage between every two chambers that share a side. Due to repairs some of those passages are closed. In fact, it is not even known whether it is possible to reach the exit from the entrance.

Maggy was given a map of the maze before entering it. It contains the directions of the passages as well as the information about the closed passages. The map also shows that the entrance to the maze leads to the chamber in the upper left corner of the map and the only exit is in the chamber in the bottom right corner of the map. There is also an information that you cannot loop forever inside the maze: if you leave any chamber with any passage, it is not possible to get back to this chamber.

Maggy wants to start at the entrance, go through the maze and leave by the exit. She also wants to write down the numbers of two favourite chambers of the maze that she has visited, in the order in which she has seen them; perhaps she will like one of the chambers so much that she will write it down twice. If Maggy fails to leave the maze she will be upset and will not write down anything. Given the map of the maze, compute in how many different ways Maggy can write down those two numbers.

Input

The first line of the input contains two natural numbers n and m ($1 \leqslant n \cdot m \leqslant 500\,000$) separated by a single space, denoting the height and width of the maze, respectively. The following $2n - 1$ lines contain the map of the maze.

In the $(2i)$-th line ($1 \leqslant i \leqslant n$) there is a string of $m - 1$ characters from the set {`>`, `<`, `*`}, describing the passages between consecutive chambers in the i-th row of the map: if the j-th character in the string is `>` then there is a passage from the j-th to the $(j+1)$-st chamber in the i-th row; `<` means that there is a passage from the $(j+1)$-st to the j-th chamber in the i-th row; while `*` denotes that there is no passage between these two chambers, in either direction.

Similarly, in the line $2i + 1$ ($1 \leqslant i \leqslant n - 1$) there is a string of m characters from the set {`v`, `^`, `*`}, describing the passages between the chambers in the i-th and $(i+1)$-st rows of the map: if the j-th character is `v` then there is a passage from the j-th chamber in the i-th row to the j-th chamber in $(i+1)$-st row, `^` means that there is a passage from the j-th chamber in $(i+1)$-st row to the j-th chamber in the i-th row, and `*` denotes that there is no passage between the j-th chambers in the i-th and $(i+1)$-st row, in either direction.

The entrance to the maze leads to the first chamber in the first row, and the exit is in the last chamber in the last row.

Output

You should write one natural number in the first and only line of the output – the number of possible ways in which Maggy can write down the numbers of two (not necessarily different) visited chambers, in the order of visiting them.

If it is not possible to reach the exit, then you should write 0.

Example

Input	Output
2 3 >> *^v <>	10

There is only one way to reach the exit: first go right twice and then once down.