Problem A. Bad Hashing

Developer:	Ivan Kazmenko
Input file:	`bad-hashing.in`
Output file:	`bad-hashing.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Little boy Petya is solving a problem involving strings. We won't go into details of the problem. All that we need to know that the problem reduces to the following: given some strings consisting entirely of lowercase English letters, we have to answer for some pairs of strings whether they are equal or not.

Petya invented a way to do it fast: first, he computed a *hash function* of each string, and after that, when asked to compare two strings, compares their hashes (values of hash function) instead. Obviously, if hashes are not equal, strings are also different. On the other hand, if hashes are equal, it does not guarantee that the strings themselves are equal.

We want to *hack* Petya's solution, that is, to come up with two different strings such that their hashes are equal. In order to do so, let's examine what Petya's hash function does.

On closer examination, it turned out that Petya implemented *polynomial hashing* of strings. A polynomial hashing function is defined by multiplier p and modulus q. For empty string ε, the value of hash function $h(\varepsilon) = 0$, and for any string S and character c it is defined recursively by $h(S + c) = (h(S) \cdot p + \mathrm{code}(c)) \bmod q$. Here, $\mathrm{code}(c)$ is the ASCII code of character c. Recall that codes of lowercase English letters are consecutive: code('a') = 97, code('b') = 98, ..., code('z') = 122. We can also obtain a general formula: if $S = s_1 s_2 \dots s_n$, we have $h(S) = (\mathrm{code}(s_1) \cdot p^{n-1} + \mathrm{code}(s_2) \cdot p^{n-2} + \dots + \mathrm{code}(s_n) \cdot p^0) \bmod q$.

This hashing method is widely used, but Petya didn't account for the opportunity of his solution being hacked, so he made two substantial mistakes in choice of p and q. First of all, modulus q is too small, it is equal to just 2^{32} (Petya just calculates the value of hash function in 32-bit unsigned integer data type and does not take overflow into account). Additionally, multiplier p is even.

Given multiplier p, hack Petya's solution.

Input

The first line of input contains an integer p, multiplier for polynomial hashing ($0 < p < 2^{32}$). It is guaranteed that p is even.

Output

On the first two lines, output two different strings S and T such that $h(S) = h(T)$. The strings should consist entirely of lowercase English letters (ASCII codes 97–122) and have lengths from 1 to 100 000 characters. Note that lenghts of the strings are not required to be equal. In case of multiple answers, output any one of them.

Examples

bad-hashing.in	bad-hashing.out
4	ae ba
1000	vabavydw budqhmng

Explanations

In the first example, $h(S) = (97 \cdot 4 + 101) \bmod 2^{32} = 489$ and

$$h(T) = (98 \cdot 4 + 97) \bmod 2^{32} = 489.$$

In the second example $h(S) = 118\,097\,098\,097\,118\,121\,100\,119 \bmod 2^{32} = 834\,470\,743$ and

$$h(T) = 98\,117\,100\,113\,104\,109\,110\,103 \bmod 2^{32} = 834\,470\,743.$$

Problem B. Bridges: The Final Battle (Division 1 Only!)

Developer:	Sergey Kopeliovich
Input file:	`bridges3.in`
Output file:	`bridges3.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

> Does your research work have any practical applications?
>
> — Frequently asked question

Foreword

Serezha has almost finished his thesis. The subject hasn't changed since December: "Dynamic 2-Edge-Connectivity Problem". The algorithm has been invented and tested, the bound of $O(K \log K)$ has been proved. The only thing left is to write about "practical applications".

Well, does the problem "Dynamic 2-Edge-Connectivity Problem" indeed have any practical applications? That's not a simple question. Probably, it's even harder than the problem itself. Whatever, the thesis has to be done.

So, the first practical application: let us create a contest problem about it!

Problem

Given an undirected graph with no more than 10^5 vertices. Initially it does not contain any edges. You have to process requests `ADD x y` and `DEL x y` — to add and to remove edge from x to y, respectively.

After each request, you should find **the number of bridges** in the graph.

There are no multiedges and loops.

For every request to remove an edge, the corresponding edge exists.

Solution

The thesis has to be pretty hard to be written in five hours. So you are given five hints.

1. Requests to add or remove edge make the edge "alive" during some intervals of time.
2. Use "Divide and conquer" idea.
3. Compress components of biconnectivity.
4. Even having compressed biconnectivity components, the graph can be reduced provided there are few requests.
5. The solution in $O(K \log K)$ exists.

Input

The first line of input contains two integers N and K: the number of vertices and requests, respectively. $1 \le N \le 10^5$, $1 \le K \le 10^5$.

The following K lines contain requests, one per line. Each request starts with a word "`ADD`" or "`DEL`", depending on the type of the request. Two integers a_i and b_i follow, describing the edge to add or to remove. $1 \le a_i, b_i \le N$, $a_i \ne b_i$.

Output

Write K integers: the number of bridges in the graph after each request.

Example

bridges3.in	bridges3.out
4 8 ADD 1 2 ADD 2 3 ADD 1 3 DEL 2 3 DEL 1 2 ADD 2 4 ADD 1 4 ADD 2 3	1 2 0 2 1 2 3 0

Problem C. A Hatred Against Literature

Developer:	Sergey Kopeliovich
Input file:	`covertexts.in`
Output file:	`covertexts.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

> Fear leads to anger, anger leads to hate, hate leads to suffering.
>
> Yoda

Since childhood, Kolya didn't like classic literature. Especially, he didn't like K works, most remembered since school times.

Kolya has grown up. Nowadays he works at a leading position in the Ministry of Education. It's time for a child's dream to come true. However, one can't just tell to the society: "The "War and Peace" is awful!". Colleagues don't support Kolya's radical statements. So he had to develop a sly plan. Using the common fight against foul language, Kolya can produce a law against several words to consider them "bad". Every text containing "bad" words obviously can't be used in school education. And, probably, one shouldn't publish it at all.

The remaining part is to choose words to be forbidden. Nickolay's political talent is high enough to add any word to the law project. However, it takes money and energy for every new word, so Kolya asks you for help.

Write a program that, given several texts, finds a set of words such that:

1. In every text there is at least one word from the set
2. The number of words is minimal possible

Input

The first line of input contains the number of texts K. $1 \le K \le 12$. The following K lines of length not exceeding 10^5 contain works disliked by Kolya. Every text contains nothing but spaces and lowercase English letters. Every substring of the text, containing only letters and surrounded by spaces or ends of line, is said to be a word. Every text contains at least one word.

Output

In the first line write the minimal possible number of words to be "bad". In the following lines write those words, one per line.

Words can be listed in arbitrary order. If there are several solutions with minimal number of words, write any of them.

Example

<table>
<tr><th>covertexts.in</th><th>covertexts.out</th></tr>
<tr><td><pre>4
a b c sideli na trube
c   kakogo perepuga
na nana la lala vot takaya pesnya
etot salata iz kakogo to m</pre></td><td><pre>2
kakogo
na</pre></td></tr>
</table>

Note

The testset is not required to contain only literature works.

Problem D. Dictionary

Developer:	Natalya Ginzburg
Input file:	`dictionary.in`
Output file:	`dictionary.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Duke of Buckingham has got into an awkward situation.

A week ago he was taking rest in his library and enjoying a lavishly decorated twelve-volumed dictionary, a treasure that in fact belonged to the French crown. The dictionary was secretly granted to Buckingham by the queen of France, Anne of Austria, as a pledge of her indulgence. Buckingham had to time to glance through some volumes, and he came to a conclusion that the dictionary is purely decorative: the records inside seemed meaningless, as if someone unfamiliar with English language tried to imitate words by putting random letters in random places. The dictionary contained nothing but those "words" — there were no translations or interpretations. Still, it was a great piece of work: each letter was traced out painstakingly, each word lied on a single line, no word was written there twice, and all the words were ordered lexicographically. Besides, all the volumes were numbered, and each of them contained the same number of words.

The trouble arised yesterday: after the ball in Buckingham Palace, one of the volumes of precious dictionary has disappeared (it was neither the first nor the last volume).

And today Buckingham had to recieve the mostly unexpected visitors: some French musketeers, who asked him to return the dictionary to the queen as soon as possible (all the twelve volumes, please). It turned out that the enemies of the queen have somehow found out that she made such a present to duke of Buckingham. In order to discredit poor Anne of Austria in the eyes of her spouse, the king of France, they suggested him to organize in a couple of weeks a literary soiree in Paris City Hall, — and surely it couldn't go without the dictionary (yes, that very dictionary, the king insists on it).

The dictionary is to be returned to the queen urgently, but it seems that the stolen volume is not going to be found, so Buckingham is ready to make an exact copy of it. Well, maybe not that exact, but at least, no one should suspect the replacement.

Otherwise, the honour of the queen will be stained, and it cannot be tolerated.

Fortunately, at Buckingham's service are the best bookbinder of England (he'll bind the new volume in exactly the same way as the other eleven are binded), the best jeweller (he'll decorate it), the best scribe (he'll fill in the words), — and you (you'll estimate the time needed to finish the work). You only need to estimate the time it takes the scribe to fill in the words, the others are all ready with their estimations.

The duke has informed you and the scribe of the number of words in each volume, the last word of the previous volume and the first word of the next volume (both words consisted only of English letters — and the scribe has to use only English letters too). Yet the scribe has some freedom to choose the words to fill in — we cannot recover lost words anyway, so we only need to keep up the lexicographic order in the whole dictionary.

Besides, Buckingham reminds you and the scribe that no word can be written twice in the whole dictionary (the duke considers equal the words that differ only by case of some letters — so you can assume all the words are written in lower case).

The copy is to be done as soon as possible, so the scribe is going to use the least possible number of letters. Output this minimum number of letters to fill in, or -1 in case it's impossible for the scribe to solve the problem at all.

Input

The first line of input contains an integer n ($1 \le n \le 10^9$), the number of words in a single volume of a dictionary. The second line contains the last word in a previous volume. The third line contains the first word in the next volume. Both of words consist entirely of lowercase English letters and have lengths from 1 to 100 000 characters.

Output

Output a single integer number, the answer to the duke's question, or -1 in case there is no solution to the problem.

Examples

dictionary.in	dictionary.out
2 ab ba	3
3 ba ab	-1
5 abcdabcd abcdabcz	40

Explanations

In the first example, one of solutions giving the answer 3 is to fill in the words "ac" and "b".

In the second example, there exist no words that can be written after the last word of a previous volume and before the first word of the next volume (it seems that duke has made some mistake in this case, so his problem cannot be solved).

Problem E. A Game on a Graph

Developer:	Sergey Kopeliovich
Input file:	`graphgame.in`
Output file:	`graphgame.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Consider a directed graph and a token on one of the vertices.

Two persons, Sasha and Alexandra, play a game. During every move, the token is moved along some edge.

Every move is done collectively in the following way. Let there be K edges from the current vertex. First, integers from 1 through K are written down. After it, players strike out numbers in turns one by one. Sasha always moves first. Once there is only one number left, the token should be moved along the corresponding edge.

Some vertices of the graph are initially said to be winning for Sasha. They have type 1. Similarly, vertices of type 2 are said to be winning for Alexandra.

Whenever the token reaches a vertex of type 1 or 2, the corresponding player is declared winner and the game ends. If the game does not end, it is said to end in a draw.

Your task is to determine the winner, or that the game ends in a draw, given both players play optimally.

Input

The first line of input contains integers N, M, S, K_1, K_2: the counts of vertices and edges in the graph, the initial position of the token, the count of vertices of the first type and the count of vertices of the second type. $1 \le N, M \le 100\,000$. The second line contains K_1 integers from 1 through N: the vertices of the first type. The third line contains K_2 integers from 1 through N: the vertices of the second type. These $K_1 + K_2$ numbers are all distinct.

The following M lines describe edges. The edge from vertex a to vertex b is given as a line with integers a and b.

Every vertex, except vertices of types 1 and 2, has at least one edge from itself.

Output

Write one line with the result of the game, provided both players play optimally: either `Sasha wins`, or `Alexandra wins` or `Draw`.

Examples

graphgame.in	graphgame.out
2 4 1 1 0 2 1 1 1 2 2 1 2 2	Sasha wins
3 9 1 1 1 2 3 1 1 1 2 1 3 2 1 2 2 2 3 3 1 3 2 3 3	Draw

Problem F. Meeting "For the Fair Elections" (Division 1 Only!)

Developer:	Oleg Davydov
Input file:	`meeting.in`
Output file:	`meeting.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

December 21-st, 2012. The midst of yet another, 54-th, oppositionists meeting. Thousands of people came to Bolotnaya square. People shout slogans, that we won't list in this statement due to political reasons.

Eventually, alarums and excursions started. The head of police decided to end up with that, arresting some of the participants. The police knows, that the crowd contains not only oppositionists, but some members of "Nazhi" movement. One of the higher SWAT officers ordered you, as the head of IT department, to develop the plan of operation in a way to arrest all the oppositionists and to arrest the members of "Nazhi" with the minimal possible sum of value.

The layout is as follows: there are n oppositionists and m "nazhists" on the square. Oppositionists are marked on the plan with red points and "nazhists" with blue ones. The police worked hard to get coordinates of all the people on the square (to make the meetings handling easier they created a cartesian coordinate system beforehand). The police considers everyone not belonging to "Nazhi" an oppositionist, so he or she should be arrested.

The plan follows. The SWAT troopers will form a chain along a straight line. After it, every person on the one side from the line will be arrested. Every person exactly on the line might be arrested or not arrested to the police decision separately.

The operation should result in the arrest of all oppositionists. Unfortunately, one can't always separate them from "Nazhi" members. To make the proper decision, for every "nashist" you are given some secret information: his or her political value (shorter, the value). Now you should plan the operation to make the sum of values of arrested "nazhists" minimal possible.

Formally, you should select a straight line, that:

- All oppositionists are to one side from it (not strictly).
- The sum of values of all "nazhists" from the same side (strictly) is minimal possible.

Input

The first line of input file contains two integer numbers n and m. The following n lines contain coordinates of oppositionists, two integers per line. The following m lines contain three integers per line: coordinates and value of the corresponding "nazhist".

Constraints:

- There are no more than 200 000 people (including "nazhists").
- Coordinates are given in special units, so they are all integers not exceeding 10^6 by absolute value.
- The value of each "nazhist" is an integer from 0 through 1 000.

Output

In the first line of output write the only integer: the sum of values of **not arrested** "nazhists". This number is needed for the report. In the second line write four integers: coordinates of two distinct points to create the line between them. The coordinates should not exceed 10^9 by absolute value.

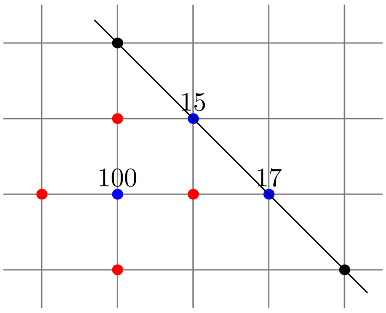

Рис. 1: The plan for the square from the first sample

Example

meeting.in	meeting.out
4 3 1 0 2 1 0 1 1 2 2 2 15 1 1 100 3 1 17	32 1 3 4 0

Note

Once your political points of view don't let you write a program to arrest oppositionists, remember that the program is flexible enough to swap red and blue points.

Problem G. Polygon (Division 1 Only!)

Developer:	Olga Bursian
Input file:	`polygon.in`
Output file:	`polygon.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Some time ago Vasya had some polygon made of cardboard. Vasya put it on the checked piece of paper, outlined the boundaries and got the first polygon on paper. Then he took his polygon of cardboard and put it on the piece of paper again, probably on some other place and probably turned over. Vasya outlined the boundaries again and got the second polygon on paper. Thus second polygon was obtained from the first by some isometric transformation, that is, a transformation which preserves distances between points.

Later, Vasya lost the cardboard polygon, but he still has the picture. He wants to obtain the second polygon from the first one again, but now he can do only several transformations of the same kind: draw a straight line, reflect the first polygon through the line and obtain another polygon, then draw a new line, reflect the obtained polygon through the new line etc.

As a result of this transformation sequence, Vasya hopes to get the second polygon from the first one. He does not want to crumple the picture much, so he wants to make no more than three transformations. Help him to do it.

A reflection axis is defined by equation $ax + by + c = 0$. Write the coefficients a, b and c of the reflection axes which will help to get the second polygon from the first one.

Input

The first line of input contains an integer n, the number of vertices in polygons ($3 \le n \le 100\,000$). The second line contains $2n$ real numbers $x_1^{(1)}$, $y_1^{(1)}$, $x_2^{(1)}$, $y_2^{(1)}$, …, $x_n^{(1)}$, $y_n^{(1)}$ — coordinates of the first polygon in counterclockwise order. The third line contains $2n$ real numbers $x_1^{(2)}$, $y_1^{(2)}$, $x_2^{(2)}$, $y_2^{(2)}$, …, $x_n^{(2)}$, $y_n^{(2)}$ — coordinates of the second polygon in counterclockwise order. All coordinates do not exceed 100 by absolute value and have no more than ten digits after the decimal point.

It is guaranteed that polygons in input have no self-intersections and no self-touchings, and also that the first polygon can be transformed into the second one by an isometric transformation. Note however that this transformation does not necessarily preserve numbering of vertices.

Input data is constructed with the aim to avoid most precision problems. The following formal conditions are met:

1. Polygon sides and polygon angles measured in radians are considered to be equal if they differ by no more then 10^{-8}.
2. It is guaranteed that sides and angles that are not considered to be equal differ by at least 10^{-6}.
3. Any two vertices of the same polygon are distinct; more precisely, pairwise distances between vertices are at least 10^{-6}.
4. Each edge has length at least 10^{-1}.
5. No three consecutive polygon vertices lie on the same line; more precisely, each polygon angle measured in radians differs from angles that are multiples of π by at least 10^{-3}.

Output

On the first line of output, write an integer m, the number of reflection axes ($0 \le m \le 3$; note that you don't need to minimize it). On the following m lines, write three real numbers on each: coefficients a, b and c of a reflection axis. Output reflection axes in the order in which Vasya should use them. The coefficients should satisfy the inequality $a^2 + b^2 > 0$. Write real numbers as precisely as possible! When checking the coincidence of the polygon obtained from the first one by reflections with the second polygon, coordinates of vertices are compared with precision 10^{-4}. If there are several answers, you can output any of them. It is guaranteed that for every possible test, there exists at least one answer.

Example

polygon.in
3 1.0 5.0 1.0 1.0 3.0 1.0 4.0 1.0 8.0 1.0 8.0 3.0
polygon.out
2 -7.000000 0.000000 31.500000 -2.000000 -2.000000 18.000000

polygon.in
5 1.0 1.0 4.5 1.0 3.0 5.5 1.0 4.0 2.0 3.0 8.0 2.0 9.0 1.0 10.5 3.0 6.0 4.5 6.0 1.0
polygon.out
3 -8.000000000000 3.000000000000 32.500000000000 -0.410958904110 -1.095890410959 4.794520547945 1.000000000000 1.000000000000 -10.000000000000

Problem H. Recognize it!

Developer:	Sergey Kopeliovich
Input file:	`segments.in`
Output file:	`segments.out`
Time limit:	5 seconds
Memory limit:	256 mebibytes

You have many things to learn, young padawan

Obi-Wan Kenobi

One drew N segments with integer endpoints on a plane. Their lengths were greater than or equal to 50. For every pair of segments, the angle between the corresponding lines was more than $\frac{\pi}{8} - 10^{-6}$. That implies, for example, that $N \le 8$.

Consder a grid with columns numbered 0 through $W-1$ and rows numbered 0 through $H-1$. The cells of the grid correspond to points of the plane with the same coordinates. Initially all cells are white.

The N segments were drawn on the grid in black. A cell was painted black if there is at least one segment such that the distance from it to the corresponding point does not exceed 1.

The segments fit into the grid, i. e. if the grid contained all points from $-\infty$ to ∞, there would be no more points to paint.

Input

The first line of input contains two integers H and W: the height and width of the grid. $1 \le H \le 300$, $1 \le W \le 300$.

The following H lines contain W characters each. These lines describe the grid. Character "`*`" stands for a black cell, character "`.`" stands for a white one.

Output

Write a single integer: the number of segments in the initial set.

Example

segments.in	segments.out
25 30 ....*........................*.. .***......................***. ..***.....................**.. ..*****..................**... ...*****.................**... ...**.***...............**.... ...***.***.............***.... ....**..****..........****.... ....**....***......******..... .....**....***..********...... .....**.....*******..**....... .....***...******...***....... ......**.*****.***..**........ ......******....*****......... ...******........****......... ..****.**.........****........ ...*...***.......******....... ........**.......**..***...... ........**......**....***..... .........**.....**.....****... .........**....**........***.. .........***..***.........***. ..........*....*...........*..	4

Note

The first test from judges' testset is the test from the sample.

The sample does not formally fit the problem statement, as the segments are too short. However, we hope that it might help you understand the statement, test your solution and it won't prevent correct solutions from being accepted.

Problem I. Yet Another Problem About Substrings (Division 1 Only!)

Developer:	Sergey Kopeliovich
Input file:	`substr7.in`
Output file:	`substr7.out`
Time limit:	2.5 seconds
Memory limit:	256 mebibytes

Oh, really... it is also bayan?

Conversation of two experienced contestants

Two sets of strings are given. Strings consist of small letters of English alphabet. Summary length of all strings in each set is not more than 10^5.

For each string from the first set you have to find the maximal by the length substring such, that it's a substring of one (or some) string of the second set.

Input

The first line contains description of the first set. Format of the description is following: integer N ($1 \le N \le 10^5$) — number of strings, and strings of the set itself. Strings are separated by one space.

The first line contains description of the second set in the same format.

Output

For each string from the first set output four numbers — length of the substring, position in the string of the first set, number of the pair string in the second set. position in the string of the second set.

Strings and positions in strings are numbered from one. You should output your answers for string of the first set in the same order, which it's given to you in the input data.

If there are several maximal substrings, you may output any one.

Example

substr7.in	substr7.out

`6 aba mama abacaba x opa z`
`2 cabama xopa`
	`3 1 1 2`
	`3 2 1 4`
	`4 4 1 1`
	`1 1 2 1`
	`3 1 2 2`
	`0 1 1 1`

Problem J. Level of Acidity (Division 2 Only!)

Developer:	*not stated*
Input file:	`acid.in`
Output file:	`acid.out`
Time limit:	3 seconds
Memory limit:	256 mebibytes

You are gathering readings of acidity level in a very long river in order to determine the health of the river. You have placed N sensors in the river, and each sensor gives an integer reading R. For the purposes of your research, you would like to know the frequency of each reading, and find the absolute difference between the two most frequent readings.

If there are more than two readings that have the highest frequency, the difference computed should be the largest such absolute difference between two readings with this frequency. If there is only one reading with the largest frequency, but more than one reading with the second largest fre- quency, the difference computed should be the largest absolute difference between the most fre- quently occurring reading and any of the readings which occur with second-highest frequency.

Input

The first line of input will be the integer N ($2 \le N \le 2 \cdot 10^6$), the number of sensors. The next N lines each contain the reading for that sensor, which is an integer R ($1 \le R \le 1000$). You should assume that there are at least two different readings in the input.

Output

Output the positive integer value representing the absolute difference between the two most fre- quently occurring readings, subject to the tie-breaking rules outlined above.

Examples

acid.in	acid.out
5 1 1 1 4 3	3
4 10 6 1 8	9

Problem K. Roman and Arabic (Division 2 Only!)

Developer:	*not stated*
Input file:	`aromatic.in`
Output file:	`aromatic.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

This question involves calculating the value of «aromatic» numbers which are a combination of Arabic digits and Roman numerals.

An aromatic number is of the form $A_1R_1A_2R_2 \dots A_nR_n$, where each A_i is an Arabic digit, and each R_i is a Roman numeral. Each pair A_iR_i contributes a value described below, and by adding or subtracting these values together we get the value of the entire aromatic number. An Arabic digit A can be 0, 1, 2, 3, 4, 5, 6, 7, 8 or 9. A Roman numeral R is one of the seven letters I, V, X, L, C, D, or M. Each Roman numeral has a base value: 1, 5, 10, 50, 100, 500 or 1000, respectively.

The value of a pair AR is A times the base value of R. Normally, you add up the values of the pairs to get the overall value. However, wherever there are consecutive symbols $ARA'R'$ with R' having a strictly bigger base value than R, the value of pair AR must be subtracted from the total, instead of being added.

For example, the number $3M1D2C$ has the value $3 \cdot 1000 + 1 \cdot 500 + 2 \cdot 100 = 3700$ and $3X2I4X$ has the value $3 \cdot 10 - 2 \cdot 1 + 4 \cdot 10 = 68$.

Write a program that computes the values of aromatic numbers.

Input

The input is a valid aromatic number consisting of between 2 and 20 symbols.

Output

The output is the decimal value of the given aromatic number.

Examples

aromatic.in	aromatic.out
3M1D2C	3700
2I3I2X9V1X	-16

Problem L. Giant Soccer (Division 2 Only!)

Developer:	*not stated*
Input file:	`giantsoccer.in`
Output file:	`giantsoccer.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

An «giant soccer» game operates under slightly different soccer rules. Each team consists of 99 players. A goal is only counted if the 4 players, in order, who touched the ball prior to the goal have jersey numbers that are in strictly increasing numeric order with the highest number being the goal scorer. Players have jerseys numbered from 1 to 99 (and each jersey number is worn by exactly one player). Given a jersey number of the goal scorer, indicate how many possible combinations of players can produce a valid goal.

Input

The input will be the positive integer J ($1 \le J \le 99$), which is the jersey number of the goal scorer.

Output

The output will be one line containing the number of possible scoring combinations that could have J as the goal scoring jersey number.

Examples

giantsoccer.in	giantsoccer.out
4	1
2	0
90	113564

Problem M. Sticks (Division 2 Only!)

Developer:	*not stated*
Input file:	`sticks.in`
Output file:	`sticks.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Let E be a set consisting of elements {/, \, -,|, .}, i. e., the elements of this set are: the diagonals of a unit square, the segments connecting the middle points of the opposite sides (horizontal and vertical) of unit square, and the center of the unit square, respectively.

Consider a rectangle $n \times m$ consisting of unit squares, each of which is labeled by exactly one of the elements of the set E. Let us call «a segment in the rectangle» a set of unit squares in the specified rectangle such that the labels on elements of the set form a continuous segment, and this set can not be extended to any other set, labels on the elements of which also form a continuous segment.

Your are given such a rectangle. Find the total number of the segments in this rectangle.

Input

First line of input contains two integers n and m ($1 \le n, m \le 100$) — height and width of the box, respectively.

Each of next n lines contain m characters from set «/», «\», «-», «|», «.». i-th character in j-th of those lines denotes the label on unit square (i, j).

Output

Print one integer — number of segments in the given rectangle.

Examples

sticks.in	sticks.out
4 5 ..|/. ..|.. ./|.. -----	4