Problem A. Appeals

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

In Formula-1 racing, judges often give penalties to drivers in the form of adding 5 seconds to the final race time for various violations. However, after the race, the team can appeal and if the appeal is successful, the penalty will be removed. So the final race protocol can change even after the finish.

...During the preparation of the protocol, the name of the winning driver was already entered into the protocol, but at the moment of completing the text entry, a message arrived that as a result of the appeal consideration, the winner had changed.

The following actions are available in the text editor used to prepare the protocol:

- Move the cursor 1 position to the left if the cursor is not at the beginning of a word (for example, `word|` becomes `wor|d`; here and below, '`|`' indicates the cursor position).
- Move the cursor 1 position to the right if the cursor is not at the end of a word (for example, `wo|rd` becomes `w|ord`).
- Delete the character in the position **before** the cursor if the cursor is not at the beginning of a word (for example, `wor|d` becomes `wo|d`).
- Insert a character in the position **before** the cursor. The cursor and the characters to the right of the cursor are shifted to the right (for example, `wor|d` becomes `worl|d` when the character '`l`' is inserted).

Given the names of the winner before and after the appeal consideration, it is required to determine the minimum number of actions required to make the corrections. At the beginning of the process, the cursor is positioned after the last character of the original winner's name, and at the end of the process, the winner's name should be corrected and the cursor should be positioned after the last character of the corrected name.

Input

The first line of the input contains the name of the winner before the appeal consideration — the word w_1. The second line contains the word w_2 — the name of the new winner of the race. Both words are non-empty, consist of lowercase Latin letters, and each word has a length of no more than 100 characters.

Output

Output a single line — the minimum number of actions required to change the winner's name based on the appeal consideration.

Examples

standard input	standard output
`hamilton` `hakkinen`	12
`schumacher` `schumacher`	0

Problem B. Boxing Notes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

In the professional boxing, the score is determined by judges' notes. In each of the 12 rounds, the judge gives 10 points to the winner of the round and 9 points to the loser of the round. The boxer's score after N rounds is the sum of the points he has earned.

If a round ends in a knockdown, the fight is stopped and the winner is declared prematurely.

You are given two integers — the score on the judge's notes for the boxer in the blue corner and for the boxer in the red corner.

Calculate how many rounds the fight lasted.

Input

The first line of the input contains a single integer — the number of points scored by the boxer in the blue corner at the time the fight was stopped ($0 \le b \le 120$). The second line contains a single integer - the number of points scored by the boxer in the red corner at the time the fight was stopped ($0 \le r \le 120$). It is guaranteed that the judge gave either 9:10 or 10:9 for each round.

Output

Output a single integer - the number of rounds in the fight.

Examples

standard input	standard output
10 9	1
120 108	12
19 19	2

Problem C. Collisions

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Given an ideal rubber ball and a track on which it rolls. We will consider the track as a straight line, where the ball starts at position 0, moving with speed 1 to the right, towards positive positions (so, after one second it will be at position 1, after two at position 2 and so on).

There are N walls installed, i-th wall is installed at the position X_i and have the toughness T_i. When the ball hits the wall, it bounces off and starts moving in the opposite direction. As the ball is ideal, it will continue to move at the same speed 1. After the collision, the toughness of the wall decreases by 1, and if it decreased to 0, that wall disappears.

Write the program that, for given positions and initial toughness of the walls, will determine how much time will pass from the start of the ball's movement to the last collision with the wall. Because the answer may be too big, print it modulo 998 244 353.

Input

The first line of the input contains one integer N ($1 \le N \le 50\,000$) — the number of the walls.

Each of the following N rows contains two integers X_i and T_i ($-10^9 \le X_i \le 10^9$, $1 \le T_i \le 10^9$) — position of the i-th wall and the initial value of its toughness, recpectively.

You may assume that all X_i are pairwise distinct, that there are no wall at position 0, and that the ball is guaranteed to hit at least one wall.

Output

Print one integer — the amount of time in seconds from the start of the movement to the last collision with the wall modulo 998 244 353.

Examples

standard input	standard output
3 2 7 -2 1 -4 1	22
2 10000000 10836006 -10000000 87654321	535438694

Problem D. Distinct Roman Reorderings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Given the string S, consisting of letters '`I`', '`V`', '`X`', '`L`', '`C`', '`D`' and '`M`'. Your task is to count the Roman anargams of the string, i.e. the number of different ways to reorder the letters in the string in a way that the resulting string is correctly written Roman numerals (degenerated reordering is counted as well).

Two reorderings are considered distinct, if the corresponding strings are not equal.

The following table from Wikipedia displays how the Roman Numerals are written:

Individual decimal places	Thousands	Hundreds	Tens	Units
1	M	C	X	I
2	MM	CC	XX	II
3	MMM	CCC	XXX	III
4		CD	XL	IV
5		D	L	V
6		DC	LX	VI
7		DCC	LXX	VII
8		DCCC	LXXX	VIII
9		CM	XC	IX

Note that:

- The numerals for 4, 9, 40, 90, 400 and 900 are written using "subtractive notation" where the first symbol is subtracted from the larger one (for example, for 40 ("`XL`") '`X`' (10) is subtracted from '`L`' (50)). These are **the only** subtractive forms in standard use.
- A number containing several decimal digits is built by appending the Roman numeral equivalent for each, from highest to lowest.
- Any missing place (represented by a zero in the place-value equivalent) is omitted.
- The largest number that can be represented in the Roman notation is 3,999 (MMMCMXCIX).

Input

The first line of the input contains one integer T — the number of the test cases ($1 \le T \le 77\,777$). Each of the following T lines contains one test case — the string s, consisting of letters '`I`', '`V`', '`X`', '`L`', '`C`', '`D`' and '`M`' ($1 \le |s| \le 15$).

Output

For each test case, print one integer — the number of the Roman anagrams of the given string.

Example

standard input	standard output
2 IC XV	1 1

Problem E. Engineer and Boat

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

The famous engineer Leonardo is building the boat in the form of a rectangular parallelepiped. The width, length and height of the boat shall be integers (measured in centimeters), and Leonardo found that the width shall be exactly 1/3 of length, and height shall be exactly 1/2 of width.

The maximal weight of the cargo loaded on the board is W kilograms. The boat itself weights 50 kilograms (regardless of size). A loaded boat floats when the combined weight of the boat and cargo equals exactly the weight of the water displaced by the boat. The top side of the boat and the surface of the water are parallel. Leonardo wants the freeboard (distance from the top of the boat to the surface of the water) to be 30 cm or more.

Write a program to find the minimum length, width, and height of the boat given the weight of the cargo on the boat. Note that 1000 cubic centimeters of water weighs 1 kg.

Input

Input consists of one integer W ($1 \le W \le 1000$) — the weight of the cargo, in kilograms.

Output

Print three integers in one line — the minimum length, width and height of the boat.

Examples

standard input	standard output
42	216 72 36
128	240 80 40

Problem F. Friendly Figures

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

The robot is drawing the picture on the paper.

The robot draws a picture in the following way:

- Draws one line segment per one step.
- If the new line segment have the common point with any of existing segments, this point is the endpoint for both segments.

Consider two line segments as the parts of the same *figure*, if the segments are connected directly or indirectly, i.e. there exists the sequence of segments with one of those segments as first element, another one as last, and the property that any two neighboring segments in the sequence share the endpoint.

So after each step, the paper contains several figures.

Consider the figure *friendly*, if it can be drawn with a single stroke, i.e. one can draw complete figure without removing the writing instrument from the paper even once and without tracing the same line twice. However, it **does not matter** how many times the same end point is passed.

Your task is to print number of the friendly figures on the paper after each drawing step of the robot.

Input

The first line of the input contains one integer N — the number of the steps ($1 \le N \le 2 \cdot 10^5$).

i-th of the following N lines describe i-th step and contains four integers $x_{i,1}$, $y_{i,1}$, $x_{i,2}$, $y_{i,2}$, in order — the coordinates of the endpoints of the segment drawn by robot on the i-th step ($0 \le x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2} \le 10^9$, the length of the segment is strictly greater than zero).

You may assume that for any two segments their intersection is either empty or is exactly one point, that is endpoint for both segments.

Output

Print N lines. In i-th line print one integer — the number of the friendly figures on the paper after step i.

Example

standard input	standard output
8 5 1 8 4 5 7 2 4 10 6 10 2 2 4 5 1 5 1 5 7 8 4 5 7 8 4 10 2 10 6 8 4	1 2 3 2 2 2 0 1

Problem G. Guess The Integer

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

This is an interactive problem

Your task is to guess an integer N between 1 and 10^6, secretly written by the jury program. To do that, you may operate with the special register X. At the beginning of the game $X = N$.

You may ask the queries in the form `?` d, where d is the integer between 0 and 10^6. The jury program **replaces** current value of X with the $X + d$. If $X + d > 2^{20}$, you lose. Otherwise the jury program answers 1, if $X + d$ is the square of some integer, and 0 if there is no such integer. You can ask no more than 2023 queries.

If you decide that you know enough to tell the value of N, you tell the answer in the form `!` N. If it is correct, the solution is accepted. This action is **not** counted as a query.

You may assume that the interactor is **not** adaptive, i.e. value of N does not changes at the time of interaction.

Interaction Protocol

The interaction is started by your program to printing the query. The query have the format `?` d ($0 \le d \le 10^6$). The jury program answers 0 or 1 depending of new value of the register X (1 if for some integer q $q \cdot q = X$, and 0 otherwise). Then you ask the next query and so on. If you want to print the answer, use format `!` N.

Do not forget to flush the output after each query (or after printing the answer), otherwise you can get the Idleness Limit error.

Example

standard input	standard output
1 0 1 1	? 2 ? 8 ? 8 ? 75 ! 7

Problem H. How Many Cars?

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

In Byteland, the license plates on the cars have the following format:

`letter letter digit digit digit letter letter digit digit`

where letter denotes lowercase English letter, and digit — the decimal digit. Any combination of the letters and digits is allowed.

The taxi cars in the Byteland are using the license plates with the following property: both digital and alphabetic parts of the plate are palindromes. For example, the plate with number `ab123ba21` is used by the taxi car, because both `abba` and `12321` are palindromes, and the plate `aa111aa22` is used for other type of car, because 11122 is not a palindrome.

The road policy received the picture of some taxi car. The picture information have same format as the Byteland license plate, but there may be a wildcard (`*`) character in some places, meaning that the digit or letter at this place was not recognized.

Given the information from the picture, find the number of distinct taxi cars that may correspond to the car on the picture.

Input

Input contains one string of length 9. Characters at positions 0,1,5,6 are either lowercase English letters or '`*`'; characters at other positions are either decimal digits or '`*`'.

Output

Print one integer — the number of taxi car license plates corresponding to the input string.

Examples

standard input	standard output
a*****b**	0
*********	676000
aa777aa77	1

Problem I. Incredible Quick Sort

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Bytica invented the new way to sort the integers, and she called it the XORting.

Given a list of integers $a_1, a_2, \ldots, a_N$, she can quickly find an integer X so that

$$a_1 \oplus X \le a_2 \oplus X \le \cdots \le a_N \oplus X,$$

where $\oplus$ means bitwise XOR, so all she needs to do then is replace a_i with $a_i \oplus x$ and voila, the list is sorted!

Your task is the implement multiple XORting queries.

You are given a list of non-negative integers $a_1, \ldots, a_N$ and Q queries of the form p_i, v_i, which means that the number a_{p_i} changes to v_i in i-th query.

Your task is to find $Q+1$ integers $c_0, c_1, \ldots, c_Q$, where c_i is the **smallest non-negative** integer X such that the list $a_1 \oplus X, \ldots, a_N \oplus X$ is sorted, after the first i changes have been performed. If there is no such X, print -1 instead of that value of X.

Input

The first line of the input contains an integer N ($1 \le N \le 10^6$) the number of elements of the array a.

The second line contains N integers $a_1, a_2, \ldots, a_N$ ($0 \le a_i < 2^{30}$), the initial values of the elements of the array.

The third line contains an integer Q ($0 \le Q \le 10^6$), the number of queries.

The following Q lines contain two integers p_i ($1 \le p_i \le N$), and v_i ($0 \le v_i < 2^{30}$) each, which means that the number a_{p_i} changes to v_i.

Output

Print $Q+1$ lines, i-th line containing one integer c_{i-1}.

The number c_i should be the smallest possible integer X that XORts the list after the changes $1, 2, \ldots, i$ have been performed (or -1 if there is no such X).

Example

standard input	standard output
3 0 1 4 3 2 7 3 3 1 4	0 2 -1 4

Problem J. Jams

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

An application is being developed that allows drivers to choose the optimal speed to avoid traffic jams.

In the process of working on this application, data collection on the operation of traffic lights was carried out as follows.

Suppose there is a traffic light installed at the intersection that works according to the following rules:

if at time t the traffic light changes from red to green, then at time $t + g$ the traffic light will switch from green to yellow (that is, the green light will be on from time t to time $t + g$, excluding time $t + g$), at time $t + g + y$ it will switch to red (that is, the yellow light will be on from time $t + g$ to time $t + g + y$, excluding the last one), and at time $t + g + y + r$ it will switch from red to green (that is, the red light will be on from time $t + g + y$ to time $t + g + y + r$, excluding the last one).

The signal color sensor installed in the car records the time at which the car crosses the intersection and the color of the traffic light at that moment.

Based on the collected data, your task is to determine the probability that at the given time T the traffic light will have the specified color c.

Input

The first line of the input contains three integers g, y, and r - the duration of the green, yellow, and red signals, respectively ($0 \le g, y, r \le 10^8$).

The second line of the input contains a single integer n - the number of records collected by the sensor ($1 \le n \le 1000$).

Each of the following n lines contains one record. The record consists of two fields: the time t ($0 \le t \le 10^9$) and a single letter that specifies the color of the traffic light ('`g`' for green, '`y`' for yellow, and '`r`' for red).

The last line contains, in a similar format, the time T and the color of the traffic light c for which a request is made.

Output

Output a single number - the probability that at time T the traffic light will have the color c with an absolute or relative error of 10^{-3}.

Examples

standard input	standard output
4 4 4 3 2 g 18 y 34 r 5 g	0.250000000000000
4 4 4 4 2 g 6 y 10 r 14 g 4 r	0.000000000000000
6 6 6 6 5 g 6 g 9 y 12 y 15 r 19 r 7 g	1.000000000000000

Problem K. Keyboards Production

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

The Thinkbox company plans to build the new factory in jungle for their new product — the wireless keyboard with the pointstick, illuminated keys and half-of-year duration per one charge. The success of this project will guarantee the Thinkbox long time domination on the market.

The factory will be placed under the forest and can be shaped as a square. The forest can be represented as $R \times C$ matrix, each cell representing the type of trees that is dominating in that cell. The tree types are denoted by lowercase English letters. The factory terrain then shall be the square $F \times F$ cells, where $1 \le F \le \min(R, C)$, that is aligned with the cells of the forest.

The CEO if the Thinkbox wants the factory to be as secure as it possible. Nowadays is too easy for other big hardware companies to take the picture of the forest, where the factory shall be placed, using the drones. So CEO of Thinkbox wants to choose the tree types placement in the square such that there will be at least K squares with the same side in the forest that have exactly the same distribution of tree types (i.e. for any two of K places there exists an parallel shift that turns one of those squares into another considering the tree types). Note that those squares may overlap.

The CEO of the Thinkbox wants factory to be as cheap as possible. The tax for the factory does not depend on the length size, so the higher value of F means less relative price per cell.

Given the map of the forest and the value of K, find the maximal length of side F for the underforest factory, or print -1, if the CEO's requirement can not be held at all.

Input

The first line of the input contains three integers R, C, and K ($1 \le R, C \le 600$, $2 \le K \le R \cdot C$) — the number of rows and columns in the matrix representing the forest, and the number of the similar squares, respectively.

Then R lines follow, each consisting of C lowercase English letters — the map of the forest. The cells with the same dominating type of the trees are denoted with the same letters, with distinct dominating types — with the distinct letters.

Output

Print one integer — the maximal side of the square F. If it is impossible to find K squares with the same distribution of the tree types, print -1 instead.

Examples

standard input	standard output
4 5 2 baacd bbaab bbbaa bbbbb	3
3 3 5 cbb dxb xxx	-1

Note

In the Sample 1, the squares 3×3 with the upper left corner (0,0) and the upper left corner (1,1) have the same shape

```
baa
bba
bbb
```