Problem A. Intersect Until You're Sick of It

Input file:	input.txt
Output file:	output.txt
Time limit:	0.5 seconds
Memory limit:	64 mebibytes

Ural contests usually contain a lot of geometry problems. Many participants do not conceal their discontent with such disbalance. Still, we have decided not to break the tradition and give you an unbalanced contest. Let's start!

Consider an iterative process for a set of points on a plane. Every iteration consists of three steps:

1. Draw a line through every pair of different points.
2. Find all intersections of all pairs of different non-parallel lines.
3. Merge the initial set of points with the set of intersection points and go to step one.

After each iteration, the number of points either increases or stays the same.

You are given a set of points. Iterations repeat while the number of points increases. How many points will be in the set after the end of this iterative process?

Input

The first line contains an integer n ($1 \le n \le 300$). Further input describes n different points. For every point, you are given a pair of integer coordinates whose absolute value does not exceed 10^8.

Output

If the process is infinite, print "oo" (two lowercase Latin letters 'o'), otherwise print the number of points in the set after the end of the process.

Example

input.txt	output.txt
4 0 0 0 1 1 0 1 1	5

Problem B. Binary Palindromes

Input file:	input.txt
Output file:	output.txt
Time limit:	0.5 seconds
Memory limit:	64 mebibytes

For each integer i from 1 to n, you must print a string s_i of length n consisting of letters 'a' and 'b' only. The string s_i must contain exactly i distinct palindrome substrings. Two substrings are considered distinct if they are different as strings.

Input

The input contains one integer n ($1 \le n \le 43$).

Output

You must print n lines. If for some i, the answer exists, print it in the form "i : s_i" where s_i is one of possible strings. Otherwise, print "i : NO".

Example

input.txt	output.txt
4	1 : NO 2 : NO 3 : NO 4 : aaaa

Problem C. Chess

Input file:	input.txt
Output file:	output.txt
Time limit:	2 seconds
Memory limit:	256 mebibytes

Your task is to determine for a given chess position if white are in check, white king is checkmated, there is a stalemate or nothing from this list is true.

It is guaranteed that:

1. There are no pawns on the chessboard.
2. There is exactly one white and exactly one black king.
3. White king has already made some moves and black king is not in check.
4. White is next to move.
5. Existence of sequence of moves which leads from initial chess position to the given one is **not guaranteed**.

Input

The first line contains one integer k ($1 \le k \le 50$), the number of game positions.

Then k descriptions of the positions follow. Each description starts with a blank line. After that follow 8 lines of 8 characters each, where '`.`' denotes an empty square, '`N`' a white knight, '`B`' a white bishop, '`R`' a white rook, '`Q`' a white queen, and '`K`' a white king. Letters '`n`', '`b`', '`r`', '`q`' and '`k`' denote black knight, bishop, rook, queen and king respectively.

Output

For each position, print the answer on a separate line. If the white king is in checkmate or stalemate, print "`checkmate`" or "`stalemate`" respectively. If white is in check, but not in checkmate, print "`check`". Otherwise, print "`ok`".

Example

input.txt	output.txt
2 RNBKQBNR .... rnbkqbnr K......r ......r. ....k	ok checkmate

Problem D. Minimum Cover

Input file:	input.txt
Output file:	output.txt
Time limit:	1 second
Memory limit:	256 mebibytes

A vertex cover of a graph is a set of vertices such that each edge of the graph is incident to at least one vertex of the set. A minimum vertex cover is a vertex cover with minimal cardinality.

Consider a set of all minimum vertex covers of a given bipartite graph. Your task is to divide all vertices of the graph into three sets. A vertex is in set N ("Never") if there is no minimum vertex cover containing this vertex. A vertex is in set A ("Always") if it is a part of every minimum vertex cover of the given graph. If a vertex belongs neither to N nor to A, it goes to the set E ("Exists").

Input

The first line of input contains three integers n, m, k: the size of the first vertex set of the bipartite graph, the size of the second vertex set and the number of edges ($1 \le n, m \le 1000$; $0 \le k \le 10^6$). Next k lines contain pairs of numbers of vertices, connected by an edge. First number denotes a vertex from the first set, second — from the second set. Vertices in each set are numbered starting from one. No pair of vertices is connected by more than one edge.

Output

On the first line, print a sequence of n letters 'N', 'E', 'A' without spaces. The letter on position i corresponds to the set containing i-th vertex of the first set. The second line must contain the answer for the second vertex set in the same format.

Example

input.txt	output.txt
11 9 22 1 1 1 2 1 3 1 8 1 9 2 1 2 3 3 2 3 4 4 3 4 5 5 2 5 4 5 6 6 6 6 7 7 5 7 7 8 7 9 7 10 7 11 7	AEEEEEENNNN EEEEEEANN

Problem E. Black-and-white Square

Input file:	input.txt
Output file:	output.txt
Time limit:	2 seconds
Memory limit:	256 mebibytes

Encouraged by success of his previous paintings, Kazimir Malevich prepares to create a new masterpiece. In order to reach a whole new level of abstraction, he decided to divide the process of creation into several steps.

He started by taking a white canvas and tiling it into a grid of $n \times m$ identical square cells. After that, he painted some of the cells black. Now Malevich wants to cut out some square area along the grid lines, and he wants it to be striped. Contemporary artists define a striped area as follows: it has exactly k distinct black rows, exactly k distinct black columns and all other cells are white. Now Malevich is startled by the following question: "How many distinct ways to find a striped square are there?". Help him understand how many distinct striped square areas can be found on the canvas. Two square areas are considered distinct if they have a different side length or different coordinates of the upper left corner.

Input

The first line contains integers n, m and k ($1 \le n, m, k \le 1000$). Next n lines describe the canvas. Each line consists of m characters, each character is either '1' or '0'. A zero means that the corresponding cell on the canvas is white, a one means that it is black.

Output

A single line must contain the answer to the problem.

Example

input.txt	output.txt
5 6 2 010100 111111 010100 111111 010100	7

Note

We will denote a square area as (R, C, L) where R and C denote the row and column of the upper left corner and L denotes its side length. In the given example, there are seven different ways to choose a squared striped area: $(2, 2, 3)$, $(1, 1, 4)$, $(1, 2, 4)$, $(2, 1, 4)$, $(2, 2, 4)$, $(1, 1, 5)$, $(1, 2, 5)$.

Problem F. Palindromes and Superpower — 2

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	**1.2 seconds**
Memory limit:	**150 mebibytes**

Dima adds letters s_1, ..., s_n one by one to the end of a word. After each letter, he asks Misha to tell him how many new palindrome substrings appeared when he added that letter. Two substrings are considered distinct if they are different as strings. Which n numbers will be said by Misha if it is known that he is never wrong?

Oh, by the way, don't forget to carefully consider **TL**, **ML** and maximum string length. You might have already seen this problem with different limitations. We have done everything to make your old solution unacceptable :)

Input

The input contains a string $s_1 \dots s_n$ consisting of letters '`a`' and '`b`' ($1 \le n \le 5\,000\,000$).

Output

Print n numbers without spaces: i-th number must be the number of palindrome substrings of the prefix $s_1 \dots s_i$ minus the number of palindrome substrings of the prefix $s_1 \dots s_{i-1}$. The first number in the output should be one.

Example

input.txt	output.txt
abbbba	111111

Note

The only Java solution we have for this problem handles input and output very carefully. We can guarantee that it meets **ML** and **TL**/2.

We are against separate **TL** and **ML** for different programming languages. We also do not consider lack of a Java solution for a problem as normal. Anyway, if you choose to write your code in Java, you must surely know particular ways to deal with memory, input and output in this language.

Just in case, here is a tip from us: we handled input and output using **FileReader** and **FileWriter** only.

```
FileReader in = new FileReader("input.txt");
FileWriter out = new FileWriter("output.txt");
s = new char[SIZE];
result = new char[SIZE];
int slen = in.read(s, 0, SIZE);
...
out.write(result, 0, slen);
out.flush();
```

Problem G. Nanomatryoshkas

Input file:	input.txt
Output file:	output.txt
Time limit:	3 seconds
Memory limit:	256 mebibytes

Matryoshka is a traditional Russian recursive doll. But everything changes, and even matryoshka needs a little innovation. Due to the use of new materials, it became possible to make a matryoshka arbitrarily thin without decreasing its durability. Soon, these new nanomatryoshkas filled the market. Now, salesman Alexander has a problem: he needs to place all nanomatryoshkas on a shelf in his shop.

Each nanomatryoshka has an internal volume and an external volume. One nanomatryoshka fits into another if the external volume of the first one does not exceed the internal volume of the second one. Alexander is sure that nanomatryoshkas should be placed in a row so that **no** nanomatryoshka (except the last one) fits into the next one in the row. Help Alexander, and he might give you a discount for a couple of nanomatryoshkas!

Input

The first line contains an integer n ($2 \le n \le 10^5$) which is the number of nanomatryoshkas. Next n lines contain two integers each: internal and external volumes of a corresponding nanomatryoshka. It is guaranteed that the internal volume of each nanomatryoshka never exceeds the external volume, but they can be equal. Both numbers are in range from 1 to 10^6.

Output

If it is impossible to place nanomatryoshkas in the described order, print "No". Otherwise, on the first line, print "Yes", and on the second line, print n integers: the numbers of nanomatryoshkas in their order on the shelf. Nanomatryoshkas are numbered starting from one in the order of their appearance in the input file. If there are several solutions, print any of them.

Examples

input.txt	output.txt
3 1 5 2 2 6 7	Yes 3 1 2
3 2 2 2 2 3 4	No

Problem H. Nikita

Input file:	input.txt
Output file:	output.txt
Time limit:	5 seconds
Memory limit:	256 mebibytes

After Nikita failed to solve a problem about queries on a segment at IOI, he decided to please the participants of Petrozavodsk training camp by another problem of the same nature.

You are given a string s, an integer k and queries.

There are two types of queries:

1. For given numbers l and r, fill a substring $s[l..r]$ with character c.
2. For given l and r, determine the number of pairs i, j such that $l \le i \le j \le r$, $j - i + 1 \le k$ and $s[i..j]$ is a palindrome.

Characters in the string are numbered starting from one.

Input

The first line contains a string s and an integer k ($1 \le k \le 50$). The length of the given string does not exceed 10^5. On the second line, you are given an integer m ($1 \le m \le 10^5$) which is the number of queries. Next m lines describe the queries. Each line starts with a query type (1 or 2), then follow integers l, r ($1 \le l \le r \le |s|$) and a lowercase Latin letter c (for type 1 queries only).

Output

For every type 2 query, print an integer on a separate line.

Example

input.txt	output.txt
abacaba 4 3 2 1 3 1 1 3 c 2 1 4	4 10

Problem I. Kirill the Gardener — 2

Input file:	input.txt
Output file:	output.txt
Time limit:	5 seconds
Memory limit:	256 mebibytes

Today is the third moon day, so it's time to plant potato! Gardener Kirill stands on a huge rectangle potato field which has $n \times m$ sockets. Distance between two adjacent sockets equals one step. Luckily, they are already dug up. Also, Kirill has a bucket big enough to hold k potatoes.

Kirill is a very organized young man, so he plants potatoes using a strict algorithm. He starts from south-west corner and moves to the north until he reaches the end of the field. Then he takes one step east and moves to the south border of the field. After that, he takes one more step east. He repeats these steps while there are still unprocessed sockets. However, the volume of his potato bucket is limited, so after each k sockets, Kirill has to go to the south border of the field and then one step out of the field to fill the bucket again (there are infinitely many potatoes outside the south border of the field). After that, he makes one step north to return to the socket from which he left the field, and then goes to the next unprocessed socket using the shortest path (Kirill comes from Manhattan and can move only parallel to coordinate axes). Surely, if the bucket becomes empty after Kirill plants a potato in the last socket, he won't go back for more potatoes. How many steps will our young gardener make?

Input

The first line contains the number of test cases: an integer t from 1 to 10. Next t lines contain test cases, one test case per line. Each test case line contains three integers n, m, k ($1 \le n, m \le 10^{12}$; $1 \le k \le 10^{18}$; $\min(n, m) \le 10^6$) which are the number of sockets on the field's south border, on its west border and the capacity of the bucket.

Output

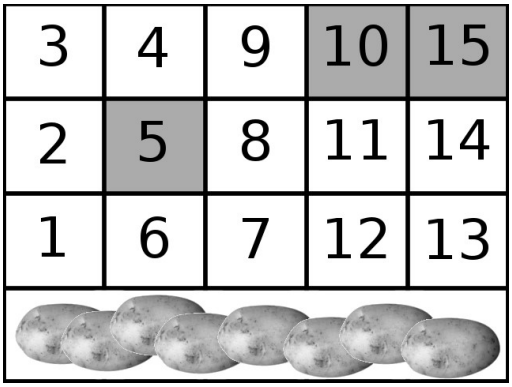
For each test case, print one line containing the answer modulo $10^9 + 7$.

Example

input.txt	output.txt
10 5 3 5 10 1 2 1 110 1 3 999 17 879 12 7 765 97 345 333 333 333 101 107 9999 100 100 5 557 139 78	20 17 12099 179524 28745 95142 221776 10924 210097 215376

Note

The picture and comment are on the next page.



The sequence of visited squares will be as follows: 1, 2, 3, 4, 5, 6, out, 6, 7, 8, 9, 10, 11, 12, out, 12, 11, 12, 13, 14, 15. So, the total number of steps is 20.

Problem J. 31 Palindromes

Input file:	input.txt
Output file:	output.txt
Time limit:	1 second
Memory limit:	256 mebibytes

For every prefix of the given string, determine whether it is possible to split it into $1, 2, 3, 4, 5, \ldots, 31$ non-empty palindromes.

Input

The input contains a line of n lowercase Latin letters ($1 \le n \le 3 \cdot 10^5$).

Output

Print n non-negative integer numbers separated by line breaks. The i-th line should contain a decimal number. If you consider the binary representation of this decimal number, its digit on position $(j-1)$ must be equal to one if the prefix of length i can be split into j palindromes, and zero otherwise.

Example

input.txt	output.txt
abaa	1 2 5 14

Note

$1_{10} = 1_2$; $2_{10} = 10_2$; $5_{10} = 101_2$; $14_{10} = 1110_2$; $abaa = aba|a = a|b|aa = a|b|a|a$.

Problem K. Non-binary Palindromes

Input file:	input.txt
Output file:	output.txt
Time limit:	0.5 seconds
Memory limit:	64 mebibytes

For each integer i from 1 to n, you must print a string s_i of length n consisting of lowercase Latin letters. The string s_i must contain exactly i distinct palindrome substrings. Two substrings are considered distinct if they are different as strings.

Input

The input contains one integer n ($1 \le n \le 25$).

Output

You must print n lines. If for some i, the answer exists, print it in the form "i : s_i" where s_i is one of possible strings. Otherwise, print "i : NO".

Example

input.txt	output.txt
4	1 : NO 2 : NO 3 : abca 4 : bbca