Problem A. Allowed Swaps

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

You are given a permutation of length N and a list of *allowed swaps*. Each swap is defined by two distinct integers between 1 and N, inclusive — positions in the permutation that can be swapped.

Your task is to sort the permutation using no more than $5 \cdot 10^5$ allowed swaps or determine that it is impossible. Note that each allowed swap can be used more than once.

Input

The first line contains a single integer N — the length of the permutation ($3 \le N \le 1000$).

The second line contains N distinct integers between 1 and N, inclusive — the permutation itself.

The third line contains a single integer S ($1 \le S \le 2 \cdot 10^5$) — the number of allowed swaps. Then S lines follow, each containing two integers p_i and q_i ($1 \le p_i, q_i \le N$; $p_i \ne q_i$), denoting that elements on positions p_i and q_i can be swapped.

You may assume that no two allowed swaps coincide.

Output

If it is impossible to sort the array using no more than $5 \cdot 10^5$ swaps from the given list, print -1.

Otherwise, print the number of swaps used, K, followed by K pairs of integers x_j and y_j ($1 \le x_j, y_j \le N$; $x_j \ne y_j$) describing the sequence of swaps. Each pair must belong to the list of allowed swaps (formally, for each j, there must exist i such that either $x_j = p_i$ and $y_j = q_i$, or $x_j = q_i$ and $y_j = p_i$).

If there is more than one solution, print any of them. The number of swaps does not have to be minimized.

Examples

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

Problem B. Banned Words

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

You are given a list of words consisting of lowercase English letters. Calculate the number of words of length L that do not contain any of the words from the list as a substring.

Since the answer may be too big, print it modulo 998 244 353.

Input

The first line contains two integers b and L ($1 \le b \le 100$; $1 \le L \le 10^9 + 7$) — the number of words in the list and word length you are interested in.

Each of the following b lines contains a non-empty word consisting of lowercase English letters. The words are distinct. The total length of the words is not greater than 100.

Output

Print the number of words of length L that do not contain any word from the list as a substring, modulo 998 244 353.

Examples

standard input	standard output
3 1 a c m	23
4 2 i cp pc cc	622
4 12 china russia bytedance mw	558811200

Problem C. Cost Of Subtree

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

The valuable tree with n vertices grows near Byteazar's house. Edge i has cost v_i assigned to it.

A subtree of the tree is defined as a non-empty **connected** subset of its edges.

The *cost* of a subtree is defined as the number of edges in the subtree multiplied by the **lowest** value of v_i in it.

Byteazar wants to make some money by selling subtrees, so he wants to know the maximum cost of a subtree of his tree.

Input

The first line contains a single integer n ($2 \le n \le 10^5$) — the number of vertices in the tree. Each of the following $n-1$ lines contains three integers a_i, b_i and v_i ($1 \le a_i, b_i \le n$; $a_i \ne b_i$; $1 \le v_i \le 10^9$) — the vertices connected by the edge and its cost.

Output

Print one integer — the maximum cost of a subtree of the given tree.

Examples

standard input	standard output
10 6 4 8 5 6 7 2 3 5 3 1 2 2 7 3 9 7 4 8 2 6 8 10 7 6 2 4	24
6 1 3 12 5 3 4 3 4 2 2 4 5 6 2 6	12

Problem D. Disk Troubles

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

The Special String Storage (SSS) has a large bank of strings. For each string, its cyclic redundancy check code (CRC) is also stored. The hard disks at SSS are so reliable that each string has at most one corrupted bit.

Byteazar has to write a program that will repair such errors. To do that, he must solve the following problem.

Consider polynomials in one variable over the field $GF(2)$ (it means that the coefficients can be just 0 and 1, and all calculations for them are performed modulo 2). Let $P(x) = x^{32} + x^{26} + x^{15} + x^7 + 1$. The polynomial $P(x)$ has an amazing property: for any two distinct integers i and j such that $0 \le i, j < 2^{32}$, the polynomials $x^i \bmod P(x)$ and $x^j \bmod P(x)$ are also distinct.

Here, $A(x) \bmod B(x)$ is the remainder of polynomial division of $A(x)$ by $B(x)$. Formally, we have $A(x) \bmod B(x) = R(x)$ where the highest degree of x in $R(x)$ is strictly lower than the highest degree of x in $B(x)$ and there exists a polynomial $Q(x)$ such that $A(x) = Q(x) \times B(x) + R(x)$. As with integer division, there exists exactly one such $Q(x)$ and exactly one $R(x)$ for any polynomial $A(x)$ and any non-zero polynomial $B(x)$. For example, $x^{32} \bmod P(x) = x^{26} + x^{15} + x^7 + 1$; here, $Q(x) = 1$.

For each given polynomial $Q(x)$, Byteazar should find the minimal nonnegative k such that $x^k \bmod P(x)$ is equal to $Q(x)$. Help him do it.

Input

The input consists of one or more test cases.

Each test case consists of single line containing a polynomial $Q(x)$. Each polynomial consists of one or more terms; consecutive terms are separated by character '`+`'. Each term is written as `x^`k where the power k is an integer such that $0 \le k < 32$. All terms are distinct and given in the order of decreasing k. There are no spaces in the input.

There are at most 200 test cases. The input will be terminated by a line containing a single zero which should not be considered as a test case.

Output

For each test case, write a single line with the answer to the problem.

Example

standard input	standard output
`x^0`	`0`
`x^1`	`1`
`x^26+x^15+x^7+x^0`	`32`
`x^26+x^21+x^15+x^13+x^7+x^6+x^0`	`38`
`x^31+x^25+x^14+x^6`	`4294967294`
`0`	

Problem E. Embeddings

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Given a string A of length n. Consider palindromic substrings of this string. Each palindromic substring is defined by its starting position s and its end e ($1 \le s \le e \le n$) such that letters in A starting at position s and ending at position e, inclusive, form a palindrome (i.e. $A[s+i] = A[e-i]$ for any integer i between 0 and $e-s$, inclusive).

Let's define an *embedding* of depth $k \ge 1$ as a sequence of k palindromic substrings of A with the following property: $s_1 < \ldots < s_k$ and $e_1 > \ldots > e_k$, i.e. palindromes in the embedding are strictly contained in each other like the Russian dolls.

Given A, count the number of possible embeddings. Since this number can be too large, calculate it modulo 998 244 353.

Input

The input consists of a single line containing the string A. The string is non-empty and consists of no more than 10^6 lowercase English letters.

Output

Print the number of possible embeddings modulo 998 244 353.

Examples

standard input	standard output
bonobo	16
banana	18

Note

For the sample input 1, we have nine embeddings of depth 1 (1-1, 2-2, 3-3, 4-4, 5-5, 6-6, 2-4, 4-6, 1-5), six embeddings of depth 2 (3-3 in 2-4, 5-5 in 4-6, 2-2 in 1-5, 3-3 in 1-5, 4-4 in 1-5, 2-4 in 1-5), and one embedding of depth 3 (3-3 in 2-4 in 1-5), with $9 + 6 + 1 = 16$ embeddings in total.

Problem F. Fox Labeling

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Bytica works in the biological expedition. Now she got n colored foxes for experiments. There's a problem, however: foxes of the same color are absolutely indistinguishable.

Now Bytica wants to mark the foxes with colored labels. She repeats the following steps until each pair of foxes becomes distinguishable:

- select some color that has not yet been used (not even as the natural color of some fox),
- get k random foxes from the cage (each subset of size k is equally likely to be chosen),
- mark each of them with a label of the selected color.

Each procedure (three steps) takes exactly one minute.

Two foxes are considered to be distinguishable if their colors differ or sets of colors of their labels differ.

The only thing Bytica is unsure about now is the expected time before all foxes will be pairwise distinguishable. So she asked you to write a program to compute this value.

Input

Input consists of two lines. The first line contains two integers n and k ($1 \le k < n \le 30$). The second line contains n integers, the i-th integer describes the initial color of the i-th fox.

Colors are denoted by positive integers not exceeding 1000.

Output

Print one real number: the expected time in minutes before each pair of foxes becomes distinguishable. Your answer will be considered correct if its relative or absolute error is within 10^{-6}.

Examples

standard input	standard output
2 1 20 20	1.000000
2 1 15 1000	0.000000

Problem G. Game With Permutations

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

This is an interactive problem.

A game with permutations has the following rules. The judge program first generates some permutation p of length N and tells you N. This permutation is never changed throughout the game.

Your task is to guess the permutation. For that purpose, you may use permutations of length N as queries. Let's see how the jury program answers them.

- When the jury program receives a permutation q, it checks positions Q_i of each integer from 1 to N in this permutation. For example, for permutation $q = (2, 3, 1, 4, 5)$, we get $Q = (3, 1, 2, 4, 5)$.
- Same calculation is applied to the permutation p, and integers $P_1 \dots P_N$ are calculated. For example, for permutation $p = (5, 2, 1, 4, 3)$, we get $P = (3, 2, 5, 4, 1)$.
- Finally, the jury program calculates an array D such that $D_i = |P_i - Q_i|$ and returns it to your program **sorted in ascending order**. For the example above, $D = (0, 1, 3, 0, 4)$, and you will receive these integers in sorted order: $(0, 0, 1, 3, 4)$.

Note that you can ask no more than 240 queries before you tell the answer.

Interaction Protocol

First your program reads a single integer N ($1 \le N \le 400$) — the length of the permutation you need to guess.

After that you can make queries. Each query consists of $n + 1$ space-separated tokens. The first token is '`?`', then the permutation you are using in the query follows. Don't forget to terminate the line by the end-of-line character and to flush the output after sending the query. Then you read the jury program's answer — a sorted array of n integers.

When you are ready to guess the permutation, print $n + 1$ space-separated tokens: the first token is '`!`' and then the permutation you guess follows.

Examples

standard input	standard output
`5`	
	`? 2 3 1 4 5`
`0 0 1 3 4`	
	`? 1 2 3 4 5`
`0 0 2 2 4`	
	`? 5 4 3 2 1`
`0 2 2 2 2`	
	`? 5 2 1 4 3`
`0 0 0 0 0`	
	`! 5 2 1 4 3`

Problem H. Histogram and Blue Rectangles

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Consider an integer array a of length n where all a_i are positive. Such an array may be represented as *histogram*. To draw the histogram, for each i from 0 to $n-1$ we draw a blue rectangle with vertices $(i, 0)$, (i, a_i), $(i+1, a_i)$, $(i+1, 0)$.

The rectangle with the vertices at integer points is called *covered* by the histogram, if the sides are parallel to the coordinate axes, and all the internal points of the rectangle are blue.

For each j between 1 and n calculate the maximal area of the rectangle covered by first j columns of the given histogram.

Input

The first line of the input contains one integer n ($1 \le n \le 2 \cdot 10^5$), the length of the array a. The second line contains n integers; the i-th of those integers represents a_i ($1 \le a_i \le 10^7$).

Output

Print n integers, each on the new line. i-th of those integers is the maximal area of the rectangle covered by first i columns of the given histogram.

Examples

standard input	standard output
5 1 7 2 3 9	1 7 7 7 9
5 4 5 3 8 7	4 8 9 12 15

Problem I. Integer Equation Checker

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

In this task you need to create a checker for integer equations with error correction.

The integer equation is given in the form:

`<number1><operation><number2>=<number3>`

wherein "`<numberI>`" denotes any positive integer less than or equal to 10^9 without leading zeros, and "`<operation>`" is one of the signs '`+`', '`-`', '`*`' or '`/`' that represents one of the four basic math operations (note that unary plus and unary minus are **not allowed**). An expression that satisfies these properties is called *well-formatted*.

The checker shall read the equations and give the outcome in the next way:

- `Correct` — a well-formatted mathematically correct equation is given.
- `Format Error` — the given equation does not match the default format.
- `Typo: <equation1> instead of <equation2>` — the given equation (`<equation1>`) is well-formatted but not mathematically correct and is is enough to replace up to two characters in it to obtain a mathematically correct and well-formatted equation (`<equation2>`).
- `Math Error` — the given equation is well-formatted, mathematically incorrect, and it is impossible to correct two or less characters and obtain a well-formatted and mathematically correct equation.

Note that characters can be only replaced, they cannot be inserted or deleted.

Input

The first line of the input contains a non-empty string of up to 30 characters that represents the equation to be checked. This word consists only of decimal digits ('`0`' - '`9`'), arithmetic operators ('`+`', '`-`', '`*`, and '`/`) and equality signs ('`=`').

Output

Print the result of evaluation. Refer to the samples for clarity.

Examples

standard input	standard output
2-2=0	Format Error
7/2=3	Typo: 7/2=3 instead of 6/2=3
2*2=47	Math Error
239*2=237	Typo: 239*2=237 instead of 239-2=237
112*11=13	Math Error
1000000000+3=1000000003	Format Error
2*2=4	Correct

Problem J. Joy With Cookies

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

Joseph and Carl are playing the next funny game. There are n different types of rectangular cookies, each cookie have side X and Y of length x_i and y_i respectively. Both players have access to infinite number of each cookie.

The game consists of two stages: setup and play. At the first stage, Joseph sets up the playfield.

- He choose exactly k cookies and puts them on the plane such as sides of each cookie are parallel to the coordinate axis and no two cookies share a common point.
- During the setup stage (and only then), Joseph can **freely** choose orientation of any of placed cookies: place the side X horizontally or or rotate the cookie by 90 degrees.
- At the end of the stage, each the chosen k cookies becomes the bottom for the some stack, so play stage starts with k empty stacks,.

AFter the setup stage, the play starts. The players put cookies in turns, Carl starts first. For one turn player chooses a cookie and puts it in some stack at the top. The following requirements must be held:

- The side X of the cookie at this stage of game **must be horizontal**.
- The cookie A can be placed to the top of stack b, if vertical side of A is strictly shorter than vertical side of the cookie from the top, and horizontal side of A is strictly shorter than horizontal side of the cookie from the top.

The player, who cannot do the correct move at his turn, lose.

You know the cookies that were chosen by Joseph. Your task is to find one of possible initial rotations of cookies which ensure that Joseph at the play stage wins when both players playing optimally, or tell that it is impossible for Joseph to win regardless of his decisions about starting cookies' orientation.

Input

The first line of the input contains two integers n and k ($1 \le n, k \le 10^5$) — number of cookie types and number of cookies used by Joseph at the setup stage.

i-th of the following n lines contain two numbers x_i and y_i ($1 \le x_i, y_i \le 10^5$) — lengths of x-parallel and y-parallel sides of the i-th cookie type. You may assume that no two cookie types same x_i size and no two cookie types have same y_i size.

The last line of the input contains k integers a_i ($1 \le a_i \le n$) — cookie types used by Joseph while setting the game up.

Output

.

If it is impossible for Joseph to put the chosen k cookies to ensure victory, print "`impossible`. Otherwise print one binary string of length k. The i-th from the left bit is equal to 0, the i-th cookie in order they are listed in the input is not rotated (i.e. x_i is horizontal), if it is equal to 1, cookie is rotated by 90 degrees and y_i is horizontal. IF there are more than one solution, you may print any of them.

Examples

standard input	standard output
5 3 1 1 4 5 3 4 5 3 6 6 1 2 4	010
5 3 1 1 3 4 5 3 4 5 7 7 5 1 2	impossible

Problem K. Knights of Round Table

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

$2N$ knights sit around a round table, they are numbered by King Arthur in clockwise order with integers from 1 to $2N$. Knight j is a neighbor of knights $j-1$ and $j+1$, knights 1 and $2N$ are neighbors too. The knights represent N orders, each order is represented by exactly two knights and each knight represents exactly one order.

There are two types of magical potion today prepared by Merlin. Let's call them X and Y. Merlin wants to distribute drinks between the knights satisfying the following conditions:

- Each knight will drink exactly one of the potions, X or Y.
- Knights from the same order must drink different potions.
- No three consecutive knights at the table drink the same potion.

Find any distribution which conforms with Merlin's requirements or determine that no such distribution exists

Input

The first line contains a single integer N ($2 \le N \le 5 \cdot 10^5$) — the number of orders represented at the table. Each of the next N lines describes one order and contains two integers a and b ($1 \le a, b \le 2N$), denoting the numbers of knights of that order at the table. All numbers in those N lines are pairwise distinct.

Output

If there is no solution, print a single word "`impossible`".

Otherwise, print a single string with $2N$ characters '`X`' and '`Y`' describing the distribution of potions. The i-th character must represent the selection for knight i.

If there is more than one possible distribution, print any of them.

Examples

standard input	standard output
4 1 5 2 3 6 7 4 8	XYXYYXYX

Problem L. Lands of Infinistan

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

The country of Infinistan can be represented as an infinite plane. In the country there are n straight walls and a river which has the form of a quadratic curve (an ellipse, a parabola or a hyperbola). The walls and the river divide the country into some lands of non-zero area. Your task is, given all the equations, to count the number of those lands.

Input

The first line contains a single integer n — the number of straight walls ($1 \le n \le 20$).

The second line contains six integers q_0, q_1, q_2, q_3, q_4 and q_5 — coefficients of the river that is a quadratic curve $q_0 \cdot x^2 + q_1 \cdot xy + q_2 \cdot y^2 + q_3 \cdot x + q_4 \cdot y + q_5 = 0$.

Each of the following n lines contains three integers a_i, b_i and c_i — coefficients of the straight line $a_i \cdot x + b_i \cdot y + c_i = 0$.

All q_i, a_i, b_i and c_i do not exceed 100 by absolute value. The straight lines may coincide. You may assume that the quadratic curve is not degenerate, and that for any i either $a_i \neq 0$ or $b_i \neq 0$.

Output

Print one integer — the number of lands.

Examples

standard input	standard output
1 1 0 1 0 0 -1 1 -1 0	4
2 1 0 0 0 -1 0 2 -1 -1 6 9 1	7
2 1 0 -1 0 0 -1 3 0 6 -5 0 -10	6