

Problem A. Agile permutation

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

You are given an integer n and a permutation of numbers p_1, p_2, \dots, p_n . You are also given two integers a, b . There are 2 types of operations. The first type: swap any two elements of the permutation for the price of a coins. The second type: shuffle the permutation randomly for the price of b coins. You are allowed to make any number of operations of each type in any order. Your goal is to get identity permutation $(1, 2, \dots, n)$ from the given one. You need to find the mathematical expectation of the number of spent coins on achieving the goal with the optimal strategy.

Input

The first line of the input file consists of numbers n, a, b , separated by spaces.

The second line contains the permutation p of length n , elements are separated by spaces.

$$1 \leq n \leq 20$$

$$1 \leq a, b \leq 1000$$

$$1 \leq p_i \leq n$$

Output

Output single number: the mathematical expectation of the number of spent coins. Your answer will be considered correct, if it's absolute or relative error don't exceed 10^{-9} .

Examples

standard input	standard output
2 5 5 1 2	0.00000000000000000000
2 5 1 2 1	2.00000000000000000000

Problem B. BSUIR Open X

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

At the anniversary BSUIR Open, there simply must be a task about creating tasks for BSUIR Open.

Let's start with the fact that the tasks for the anniversary BSUIR Open should be especially interesting. That is why n task sets were developed for such purposes. It was assumed that among these sets the only one, the best set will be selected to be used for the competition. But then the setters realized that solving problems from one set would not be so interesting. In general, each set in itself is very interesting and would be perfect for a regular competition, but not for the anniversary BSUIR Open.

To make the competition more interesting, it was decided to take tasks immediately **from two sets of tasks**. Each set has its own code name, and some sets **may have the same name**. In order for the competition to be perfect, it is necessary that the names of these sets could be used to make the string "BSUIROPENX" by appending one string to the end of the other. For example, if we selected two task sets "FOO" and "BAR", then you can make either the string "FOOBAR" or "BARFOO".

Now that you know how to make the perfect competition for the anniversary BSUIR Open, you need to determine the number of ways to do this. Two methods are considered different if they differ in at least one set of tasks, or their order is different.

Input

The first line contains a single integer n – the number of task sets.

The next n lines contain lines consisting only of capital English letters – code names of task sets.

$$1 \leq n \leq 10^5$$

The total length of the names of the task sets does not exceed 10^6 .

Output

In a single line print a single integer – the number of ideal competitions that can be made up of the indicated sets of tasks.

Examples

standard input	standard output
4 BSUIR BSU OPEN IROPENX	1
13 BSUIR OPENX BSUIR OPENX BSUIR OPENX BSUIR OPENX BSUIR OPENX BSUIR OPENX BSUIR	42

Note

In the first test case, you can use the set “BSU” and the set “IROPENX”.

Problem C. Crossed out letter

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Consider string s . Let's call s with some single character removed – s_0 , and without some, possibly another, character – s_1 . You are given s_0 and s_1 , find **any** suitable string s or determine, that there are none.

Input

The first line of the input data contains one string s_0 consisting of lowercase English letters.

The second line of the input data contains one string s_1 consisting of lowercase English letters.

$$1 \leq |s_0|, |s_1| \leq 3 \cdot 10^5$$

$$|s_0| = |s_1|$$

Output

Print a single line s consisting of lowercase English letters or “IMPOSSIBLE” (in capital letters, without quotes).

Examples

standard input	standard output
abacaa aacaba	abacaba
bsuir openx	IMPOSSIBLE

Note

In the first test case, removing from “abac**a**ba” second character “b” we get s_0 = “abacaa”, and removing “abac**a**ba” first character “b”, we get s_1 = “aacaba”.

In the second test case there isn't any string s .

Problem D. Dull game

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Nim is a game in which two players take turns removing objects from distinct heaps. On each turn, a player must remove at least one object, and may remove any number of objects provided they all come from the same heap. The player who takes the last item wins.

You are given heaps with sizes a_0, a_1, \dots, a_n . Find the heaps subsequence S which satisfies the following requirements:

- If we optimally play the “Nim” game using heaps from this subsequence, the first player loses.
- $a_0 \in S$.

You should also process m queries for modifying the sequence of heaps: the i -th query p_i, x_i means that from now on $a_{p_i} = x_i$. After each request, it is necessary to find the subsequence S_i that satisfies the requirements described above.

It is guaranteed that initially, as well as after each modification query, the number of matching subsequences S heaps will be **exactly one**.

Input

The first line of input data contains two space-separated integers n and m .

The second line of input data contains $n + 1$ space-separated integers a_0, a_1, \dots, a_n .

The following m lines contain modification queries descriptions, i -th line contains two space-separated integers t_i, x_i . To calculate p_i you have to know the answer for **the previous** question. Namely, if you printed numbers k, b_1, b_2, \dots, b_k as the answer of the previous question, then $p_i = t_i \oplus k \oplus b_1 \oplus b_2 \oplus \dots \oplus b_k$.

$$2 \leq n \leq 1000$$

$$0 \leq m \leq 1000$$

$$0 \leq a_0, a_1, \dots, a_n < 2^n$$

$$0 \leq p_i \leq n$$

$$0 \leq x_i < 2^n$$

Output

In the first line print integer k – the size of subsequence S .

In the second line print the sequence of space-separated integers $b_1 < b_2 < \dots < b_k$ where $S = a_{b_1}, a_{b_2}, \dots, a_{b_k}$.

After each i -th query print the subsequence S_i in the same format.

Examples

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

Problem E. Elegance in moves

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

Given a board of size $n \times m$ cells, you have to find the number of different pairs of cells between which the chess queen can walk in a single move and without crossing any of the given rectangles. Additionally, it is known that each cell belongs to no more than one rectangle. Recall that in a single move the queen can move any number of squares in a straight line — vertically, horizontally or diagonally.

Since the answer can be large, print it modulo $10^9 + 7$.

Input

The first line contains three integers $n \ m \ k$ — field size and the number of rectangles, respectively. The next k lines contain four integers $r1_i \ c1_i \ r2_i \ c2_i$ — the coordinates of the i -th rectangle. No two different rectangles share a common cell.

$$1 \leq n, m \leq 10^9$$

$$0 \leq k \leq 10^5$$

$$1 \leq r1_i \leq r2_i \leq n$$

$$1 \leq c1_i \leq c2_i \leq m$$

Output

Print a single integer, denoting the number of pairs of cells between which the chess queen can walk in a single move outside of the rectangles modulo $10^9 + 7$.

Examples

standard input	standard output
1 6 1 1 3 1 3	4
3 3 1 2 2 2 3	11
6 9 5 1 6 4 8 3 2 6 2 2 1 6 1 1 3 5 5 3 9 4 9	42

Problem F. Function analysis

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 256 megabytes

Let p be a sequence of numbers $(1, 2, 3, \dots, n-1, n)$, and q be a random sample of $m \leq n$ elements of p , such that i -th element of q is chosen equiprobable and independently.

Denote by $nth(a, b)$ the element that is in the b -th position if we order a in non-decreasing order. For example, $nth(a = (5, 2, 3, 2), b = 4) = 5$.

For each m , s.t. $d \leq m \leq n$ find the probability that $nth(p, k) < nth(q, d)$, modulo 998244353. In other words, if the desired probability is $\frac{P}{Q}$, print $P \cdot Q^{-1} \bmod 998244353$.

Input

A single line of input contains three integers separated by space n , d and k .

$$1 \leq k \leq n \leq 3 \cdot 10^5$$

$$1 \leq d \leq n$$

Output

Print $n - d + 1$ lines, each of them containing a single integer, the probability for m from d to n both including (modulo 998244353).

Example

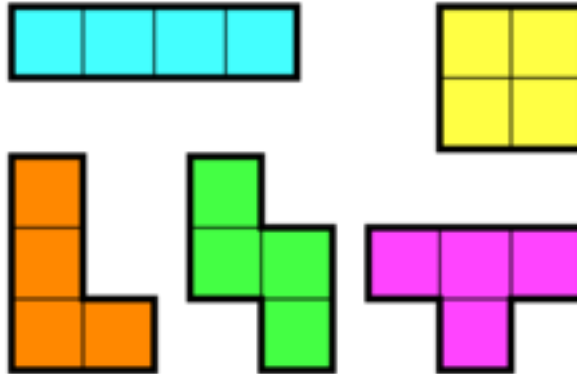
standard input	standard output
5 2 3	119789323 15971910 552628074 239898083

Problem G. Geometric shapes

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

You have to tile all cells of the grid $n \times m$ with shapes from tetris (tetromino) except for one cell with coordinates (r, c) .

There are the following tetromino shapes:



And also their turns and reflections.

Input

The first line contains a single integer t — the number of testcases. The following t lines contain four space-separated integers n_i m_i r_i c_i , denoting the size of the grid and coordinates of the cell, which you don't have to tile, respectively.

$$1 \leq r_i \leq n_i$$
$$1 \leq c_i \leq m_i$$
$$\sum n_i * m_i \leq 10^5$$

Output

For each test case print “YES” if tiling is possible. Next, print $n_i \times m_i$ numbers denoting the tiling. Each of the numbers correspond to the number of the figure to which the cell belongs. The cell (r_i, c_i) has to contain 0, and the remaining figures should be numbered sequentially starting with 1. If tiling is impossible, then print “NO” in a single line.

Example

standard input	standard output
2	YES
3 3 2 2	1 1 1
4 4 1 2	1 0 2
	2 2 2
	NO

Problem H. Hockey championship

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

The competition involves $2 \times n$ teams from m countries. Teams are randomly matched into n pairs. You know the expected value of the number of pairs in which both teams are from the same country. Find a possible country distribution of teams which has a given expected value.

Input

A single line contains two positive integers x and y . The expected value is equal to $\frac{x}{y}$.

$$1 \leq x, y \leq 1000$$

Output

If there is no suitable distribution of teams by country, print in a single line “-1”.

Otherwise, in the first line of the output file print one positive integer m — the number of countries in which there are teams participating in the competition. In the second line print m positive integer separated by a space — the number of teams in the corresponding country. The sum of the printed numbers has to be even and must not exceed 10^4 . It is guaranteed that if there is a suitable distribution, then there is a distribution that satisfies the given restrictions.

Examples

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

Problem I. Items in boxes

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

You have 2^n different boxes, each of them containing a different items. Find the number of ways to take exactly one item from each box modulo 2^{n+2} .

In other words, if the required number of ways is x , print the remainder of dividing x by 2^{n+2} .

Input

The only line of the input data contains two integers separated by a space n and a .

$$1 \leq a, n \leq 10^9$$

Output

Print a single number — the remainder of dividing the number of ways to choose one item from each box by 2^{n+2} .

Examples

standard input	standard output
5 10	0
10 5	1
1 2	4

Note

In the third example, $2^n = 2$ boxes, each with $a = 2$ items. It turns out that there are two ways to take an item from the first and two ways to take an item from the second, total $2 \cdot 2 = 4$ of the method. The remainder of the division by $2^{n+2} = 2^3 = 8$ is 4.

Problem J. Jenga

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Jenga — board game invented in the distant 1970th year. In this game, bars are used, where the length of each bar is exactly three times its width, and the height is approximately equal to half its width. A tower is built from these bars, where each floor consists of three bars, and the bars of the next floor are laid perpendicular to the bars of the previous floor. Then the two players take out one bar from the turret one by one and place it on the upper floor perpendicular to the bars of the upper floor. At some point, the tower ceases to be stable and collapses under the influence of its own gravity. The player after the move which the tower collapses loses. The figure below shows an example of a tower after a certain number of moves.



A stable tower is considered to be a tower in which all floors except the upper one contain either at least two bars or one bar located in the middle of the floor. You are asked to calculate how many stable towers there are, consisting of exactly n bars. Two towers are considered different if there is no such rotation about an axis perpendicular to the surface of the base combining these towers.

Input

The only line contains a single integer n — the number of Jenga bars.

$$1 \leq n \leq 10^{18}$$

Output

Print a single integer — the number of possible ways to get a stable Jenga tower. Since the answer may be too large, it must be printed modulo $10^9 + 7$.

Examples

standard input	standard output
1	1
42	729888649
2	4

Problem K. K-ones xor

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

You are given integers n, m and an array a_1, a_2, \dots, a_n of length n consisting of m -bit integers. You are also given an integer k . You need to find m -bit integer x , which has no more than k ones in binary representation. Out of all this numbers, you need to find such number, that after applying $a_i = \max(a_i, a_i \oplus x)$ to the array, sum of the array will be maximal. \oplus denotes the operation of bitwise *XOR*. If there are multiple such numbers, you need to find the minimal one.

Input

The first line of input file contains three integers n, m, k . The second line of input file contains the array a_1, a_2, \dots, a_n .

$$1 \leq n \leq 10^5$$

$$1 \leq m \leq 30$$

$$0 \leq k \leq m$$

$$0 \leq a_i < 2^m$$

Output

Output single number x - answer for the task.

$$0 \leq x < 2^m$$

Examples

standard input	standard output
3 2 2 3 2 2	1
2 1 1 0 0	1

Problem L. Long integer

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

You are given the positive integer x_0 in the decimal notation, and q queries, the i -th of them can be one of two types:

- Add some given digit d_i to the right of the number, i.e. $x_i = \overline{x_{i-1}d_i}$.
- Cross out the rightmost digit from the number, i.e. $x_{i-1} = \overline{x_i e_i}$.

After each query, print the remainder of dividing x_i by $10^9 + 7$.

It is guaranteed that after each query the number will be positive ($x_i \geq 1$).

Input

The first line of the input file contains a single integer x_0 .

The second line of the input file contains a single integer q .

The following q lines denote the queries.

If the i -th query is a query to add a digit, then the i -th line contains the character “+” (without quotes) and d_i .

If the i -th query is a query to cross out a digit, then the i -th line contains the character “-”.

$$1 \leq x_0 < 10^{100\,000}$$

$$1 \leq q \leq 10^5$$

$$0 \leq d_i \leq 9$$

Output

After i -th query, print the remainder of dividing x_i by $10^9 + 7$.

standard input	standard output
123	1235
3	12351
+ 5	1235
+ 1	
-	
42	420
23	4200
+ 0	42000
+ 0	420000
+ 0	4200000
+ 0	42000000
+ 0	420000002
+ 0	200000001
+ 2	0
+ 9	4
+ 4	42
+ 4	4
+ 2	0
-	200000001
-	420000002
-	42000000
-	4200000
-	420000
-	42000
-	4200
-	420
-	42
-	4
-	
-	