Problem A. Template for Search

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

You are required to find a palindome string with a minimal length which matches a given template for search. Palindrome is a string which can be read in the same way in both directions (forward and backward). An empty string is also a palindrome. The template can contain lower case latin letters corresponding to the same letters in a string, symbol '?' corresponding to an arbitrary latin letter and symbol '*' corresponding to a zero or more arbitrary latin letters.

Input

First line contains a string s — a template string. This string contains only lower case latin letters, symbols '?' and '*'.

$$1 \le |s| \le 500$$

Output

You are required to print a single line containing a palindrome string with a minimal length which matches a given template. The palindrome should contain only lower case latin letters. If there is no such palindrome, you are required to output "-1". If there are multiple possible palindromes, you may output any of them.

Examples

standard input	standard output
`*ac?ba`	`abacaba`
`ac?ba`	`-1`

Problem B. Redistribution of Digits

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

You are required to form numbers based on a given set of digits. Each number is also provided with an upper bound. Each digit from the given set must be used exactly once. If the set does not contain a digit, you are not allowed to use it.

Input

First line contains an integer t — the number of tests.

Each of the next t lines contains a string s — a given set of digits, integer n — quantity of numbers which should be formed and n integers a_i — upper bounds for numbers. The digits for all numbers a_i are given in a non-increasing order starting from the most significant digit. The set of digits and the numbers a_i do not contain a digit 0.

$$1 \le t \le 1000$$

$$1 \le |s| \le 500$$

$$1 \le n \le 50$$

$$1 \le a_i \le 10^9$$

Output

You are required to output t lines containing the given quantity of numbers corresponding the conditions above. If a solution does not exist you should print `"-1"`. If there are multiple solutions, you may output any of them.

Example

standard input	standard output

3
1234 2 21 43
12534 2 21 43
42 1 42
	14 23
	-1
	42

Problem C. Partial Sums

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

You have a matrix A_0, consisting of n rows and m columns. Rows and columns are numbered with consecutive natural numbers starting from 1. The elements of the matrix are zeros and ones. Denote the element of this matrix at the intersection of the i row and j column as $A_0[i, j]$.

Consider an infinite sequence of matrices A_k. The matrix A_k ($k > 0$) also consists of n rows and m columns and it is a matrix of partial sums for the matrix A_{k-1} modulo 2. Formally, this means that

$$A_k[i, j] = \sum_{1 \le u \le i} \sum_{1 \le v \le j} A_{k-1}[u, v] \mod 2$$

It is required to find a minimum $k > 0$ such that the matrices A_k and A_0 are element-wise equal.

Input

The first line of the input data contains two integers n and m — the number of rows and columns in the matrix A_0. The following n lines contain descriptions of the rows of the matrix. Each line consists of m characters, each character is either 0 or 1.

$$1 \le n, m \le 10^6$$
$$n \times m \le 10^6$$

Output

Output the single number k — the answer to the problem.

Examples

standard input	standard output
1 1 1	1

4 2
00
01
10
11
	4

Problem D. Lis on Circle

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

There are n people at the round gaming table. Each of them has a set of cards. Every card contains some number x. Players make turns consecutively, one after another, starting from the player number 1. A player in his turn can either skip his turn (to pass), or put (and leave on the table) a card with a number that is strictly greater than the previously played last number. No more than k players in a row can pass the turn. All players know the numbers written on the other players cards and always play optimally. Help gamblers to assemble an increasing sequence of maximum length.

Input

The first line contains two numbers n and k — the number of players and the maximum possible amount of turn skipping in a row.

The next n lines contain the description of the cards players have in their hands. The first number in the m_i is the number of cards the current player has in his hand. The following space separated m_i numbers represent the written on the cards numbers x.

$$0 \leq \sum m_i \leq 10^5$$

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

$$0 \leq k < n$$

$$0 \leq x \leq 10^9$$

Output

In the first line print the single number — the length of the maximum sequence. In the next lines print two space separated numbers: the player index number and the number written on the card he played. If several solutions exist, output any of them.

Example

standard input	standard output
`3 1`	`9`
`4 1 10 12 20`	`1 1`
`2 11 21`	`3 3`
`4 3 5 15 22`	`1 10`
	`2 11`
	`1 12`
	`3 15`
	`1 20`
	`2 21`
	`3 22`

Problem E. Very Simple Sum

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

You are given two arrays a and b, each consisting of n integers. Your task is to calculate a very simple sum:

$$\sum_{1\le x\le n}\sum_{1\le y\le n}\sum_{1\le z\le n}\sum_{1\le w\le n}(a_x+a_y+a_z+a_w)^{(b_x\oplus b_y\oplus b_z\oplus b_w)} \mod 998244353$$

Input

First line contains a single integer n — the number of elements in arrays a and b.

Second line contains n space-separated integers a_i — the elements of the array a.

Third line contains n space-separated integers b_i — the elements of the array b.

$$1 \le n \le 10^5$$

$$1 \le a_i, b_i \le 500$$

Output

Output a single integer — the value of the very simple sum.

Examples

standard input	standard output
1 1 1	1

5
227 67 445 67 213
297 171 324 493 354
	42

Problem F. Random XOR

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

There is an array a containing n integers. Also, there is initially empty array b. Some elements of a are going to be added to b. Each element is added with probability P independently from others. Then the value of s is to be computed:

$$s = \oplus_{i=0}^{|b|} b_i$$

where $\oplus$ is bitwise exclusive OR (if the array b is empty, s equals to zero). You are required to compute the expected value of s^2.

Input

The first line of input contains three integers n, X and Y. The probability P is equal to $\frac{X}{Y}$.

The second line contains n integers a_i divided by spaces — elements of the array a.

$$1 \le n \le 10^5$$
$$0 \le X < 10^9 + 7$$
$$0 < Y < 10^9 + 7$$
$$X \le Y$$
$$0 \le a_i < 10^9 + 7$$

Output

The answer can be always represented as a fraction $\frac{u}{v}$ where u and v are co-prime numbers and $v \neq 0 \mod (10^9 + 7)$ You are required to output only one number — $u \times v^{-1} \mod (10^9 + 7)$

Example

standard input	standard output

`3 1 2`
`2 8 10`
	`42`

Problem G. Sum of Distances in Cactus

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

Find the sum of the distances between all pairs of vertices in a cactus graph. A cactus graph is a graph in which every edge belongs to at most one simple cycle. The distance between vertices is calculated as the number of edges in the shortest path connecting a given pair of vertices.

Input

First line contains two integers n and m — the number of vertices and the number of edges in the cactus.

Each of the following m lines contains two integers u_i v_i — the numeric labels of vertices connected by an edge.

It is guaranteed that the graph is connected and does not have self-loops and multiple edges.

$$1 \le n \le 10^5$$

$$n - 1 \le m \le 2 \times n$$

$$1 \le u_i, v_i \le n$$

Output

Output a single line containing the sum of the distances between all pairs of vertices.

Examples

standard input	standard output

`3 3`
`1 2`
`2 3`
`3 1`
	`3`

`7 8`
`2 1`
`3 1`
`5 1`
`3 2`
`4 3`
`5 7`
`6 3`
`4 6`
	`42`

Problem H. Not A + B

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

You are required to print a value of c which is not equal to the sum of two given numbers a and b.

Input

First line contains an integer t — number of tests. Next t lines contain two integers a and b each.

$$1 \le t \le 10^3$$

$$1 \le a, b \le 50$$

Output

You are required to output an integer c for each test in a separate line. If there are multiple solutions, you may output any of them.

$$1 \le c \le 50$$

Example

standard input	standard output

3
1 2
3 4
5 6
	12
	34
	42

Problem I. Cutting

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

A given number can be cut into two non-empty numbers and replaced with the absolute value of the difference between these two numbers. It is forbidden to obtain zero after such an operation. Such a cut can be repeated several times. It is required to get the minimum possible number in the end.

Input

The first line contains an integer t — the number of tests.

Each of the following t lines contains one integer n — the initial number for cutting.

$$1 \le t \le 10^3$$

$$1 \le n \le 10^{12}$$

Output

For each test, you need to output a path to get the minimum number. First print the integer number m — the amount of numbers in the path. Then output m integers. The first number is an initial number, and the last one is the minimum possible number after all cuttings. The cutting must be accomplishable between adjacent numbers. If there are several solutions, output any of them.

Example

standard input	standard output
3	1 7
7	2 100 1
100	2 42 2
42	

Problem J. Paternity Testing

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

You have a tree consisting of n nodes labeled from 1 to n. The tree is rooted at node 1. A function $cnt(v, l, r)$ is defined as the number of nodes in a subtree of node v, that have indices from l to r inclusive. You are required to answer q queries. The query is represented by a pair (l_i, r_i). The answer to the query is a sum $\sum_{l \le i \le r} cnt(i, l, r)$.

Input

First line contains an integer n — the number of nodes in the tree.

Next $n - 1$ lines indicate ancestors of the nodes in the tree. Each i-th line of those $n - 1$ lines contains the ancestor's index for the $i + 1$-th node in the tree.

The following line contains a single integer q — the number of queries to be answered.

Each of the next q lines contains two numbers u_i and v_i — encoded queries.

$$1 \le n \le 50000$$
$$1 \le q \le 50000$$
$$0 \le u_i, v_i \le 10^9$$

Let ans_i be the answer to the i-th query ($ans_0 = 0$). Then, the parameters of the i-th query are:

$$x_i = 1 + ((u_i \oplus ans_{i-1}) \mod n)$$
$$y_i = 1 + ((v_i \oplus ans_{i-1}) \mod n)$$
$$l_i = min(x_i, y_i)$$
$$r_i = max(x_i, y_i)$$

Output

Print q lines. The i-th line should contain the answer to the query (l_i, r_i).

Example

`standard input`	`standard output`

9
1
2
3
4
5
5
7
8
5
0 8
1 2
2 3
4 5
6 7
	42
	8
	3
	3
	3

Problem K. Chess Positions

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

You have an unlimited number of white and black chess pieces such as the queen, bishop, knight and rook. You are also given two numbers, w and b, the number of white and black pieces that must be under attack. It is required to create a position on the board 8×8 suitable for the restrictions described above.

The queen, the bishop and the rook move in the allowed direction before another figure is met and can attack if it is of a different color. The Knight also attacks a figure of a different color and and can jump over other pieces.

Input

First line contains integer t — number of tests. Next t lines contain two integers w and b each — number of white and black chess pieces under attack, correspondingly.

$$1 \le t \le 10^3$$

$$0 \le w, b \le 50$$

$$w + b \le 64$$

Output

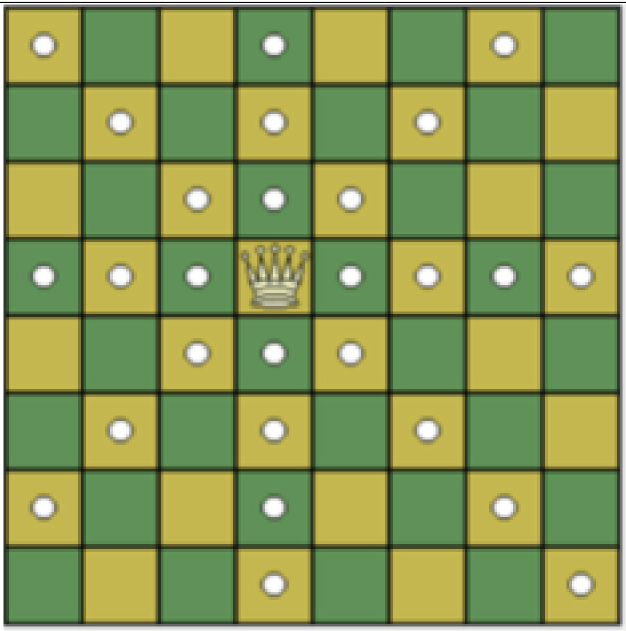
For each test, you are required to output a chess position which satisfies given conditions. The position should be represented by 8 lines containing 8 symbols each. The positions should be split by an empty line. Empty cells should be printed as '`.`', cells containing queen should be represented as '`q`', bishop cells – '`b`', knight cells – '`k`', and rook cells – '`r`'. White pieces should be printed in upper case and black ones in lower case. If there are multiple correct positions, you may print any of them. It is guaranteed that the correct answer always exists.

Example

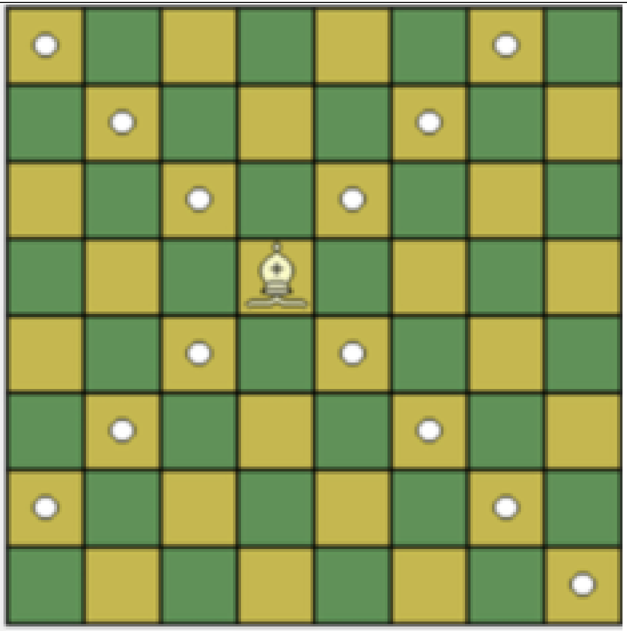
standard input	standard output

2
2 3
4 2
	Q.b...rR
	....K...
	....
	.....b..
	....
	..k.....
	.k......
	......Q.
	
	....
	Q.k...Q.
	....
	...k....
	..r..B..
	....BkK.
	.....K..
	..K.K...

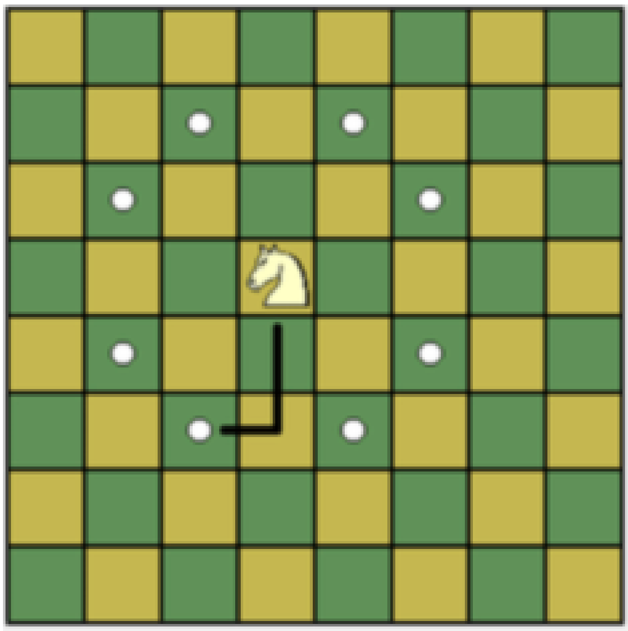
Note



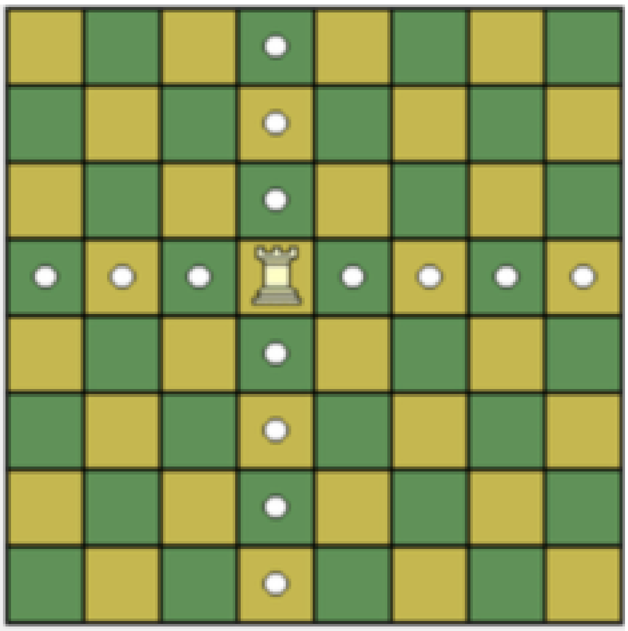
Queen



Bishop



Knight



Rook