Problem A. Tritwise Mex

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	256 mebibytes

Let us denote $\text{mex}(a, b)$ (minimum excludant) as the minimum non-negative integer which is neither equal to a nor equal to b. It always holds that $\text{mex}(a, b) < 3$, thus we can define tritwise mex. If we write a and b in ternary notation:

$$a = \sum_{i=0}^{k-1} a_i \cdot 3^i, \quad b = \sum_{i=0}^{k-1} b_i \cdot 3^i,$$

where a_i and b_i are integers from 0 to 2, we define mex_3 as follows:

$$\text{mex}_3(a, b) = \sum_{i=0}^{k-1} \text{mex}(a_i, b_i) \cdot 3^i.$$

You are given two sequences $a_0, \ldots, a_{3^k-1}$ and $b_0, \ldots, b_{3^k-1}$. You have to compute the sequence $c_0, \ldots, c_{3^k-1}$:

$$c_k = \sum_{\text{mex}_3(i,j)=k} a_i \cdot b_j.$$

Input

The first line of input contains a single integer k ($1 \le k \le 12$).

The second line of input contains 3^k integers $a_0, \ldots, a_{3^k-1}$ ($0 \le a_i \le 10^3$).

The third line of input contains 3^k integers $b_0, \ldots, b_{3^k-1}$ ($0 \le b_i \le 10^3$).

Output

Output 3^k integers $c_0, \ldots, c_{3^k-1}$ separated by spaces.

Examples

standard input	standard output

1
1 1 1
1 1 1
	4 3 2

2
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
	16 12 8 12 9 6 8 6 4

Note

For reference: $3^{12} = 531\,441$.

Problem B. Associativity Degree

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 mebibytes

Consider a binary operation $\circ$ defined on numbers 1 through n:

$$\circ : \{1, \ldots, n\} \times \{1, \ldots, n\} \to \{1, \ldots, n\}.$$

Let us define its *associativity degree* as the number of triplets $i, j, k \in \{1, \ldots, n\}$ for which $\circ$ is associative:

$$i \circ (j \circ k) = (i \circ j) \circ k.$$

Your task is, given n and k, to construct an operation $\circ$ such that its associativity degree is exactly k.

Input

The first line of input contains two integers n and q ($1 \le n \le 64$, $1 \le q \cdot n^2 \le 10^6$).

The i-th of the next q lines contains a single integer k_i ($0 \le k_i \le n^3$).

It is guaranteed that all k_i are distinct.

Output

For each given value of k_i, do the following:

If there is no operation $\circ$ with associativity degree k_i for given n, output "`NO`" on a single line.

Otherwise, output "`YES`" on the first line, followed by n lines containing n integers each. The j-th integer in the i-th line must be equal to $i \circ j$.

Examples

standard input	standard output
3 1 27	YES 1 1 1 1 2 3 1 3 2

1 2
0
1
	NO
	YES
	1

Note

The operation from the first sample can be concisely described as $i \circ j = 1 + ((i-1) \cdot (j-1)) \bmod 3$, and it is fully associative.

Problem C. Medium Hadron Collider

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

Okarin Kyouma is a brilliant scientist working on Medium Hadron Collider (MHC) in Socit Europenne de Recherche Nuclaire (SERN). His current experiments aim to unravel the mysterious Stein's gate effect.

MHC consists of a sufficiently large amount of sections arranged on a line and numbered $1, 2, 3, \ldots$ along the line. Particle beams traverse through sections in ascending order of section numbers, it takes one second for a beam to pass any section. There are two types of beams which may be present in MHC: electronic beams e^- and positronic beams e^+.

Initially, there is a single electronic beam in the beginning of the first section of MHC. Each section is affected by the Stein's gate effect. When a particle beam goes through the center of section $i > 1$, the gate is triggered and creates another beam in the center of section $i - 1$. This new beam is either the same or inverted (that is, e^+ instead of e^-, and vice versa). The new beam moves in the same direction and with the same speed as the initial beam. Beam recreation is coherent: at each step, either all sections produce the same beams or all sections produce inverted beams.

Then created beam interferes with the one already present in the section, if any. Beams of the same type merge, while beams of opposite types annihilate. Formally, we may say that a single electronic beam has charge -1, single positronic beam has charge $+1$, and an "empty" beam has charge 0. The result of collision of beams with charges α and β will be a beam with charge $\gamma = \alpha + \beta$. Here are a few examples:

- A single electronic beam colliding with a single positronic beam leave an "empty" beam.
- Two single positronic beams become a "double" positronic beam with total charge $+2$.
- A positronic beam with total charge $+5$ colliding with an electronic beam with total charge -8 become an electronic beam with total charge -3.

The end of the N-th second of experiment approaches. So, the Stein's gate effect was activated exactly $N - 1$ times, but the initial electronic beam is yet in the section numbered N. Okarin wants to measure the total charge of beams traversing through some sections but, unfortunately, measurement tools for sections 1 through 128 are out of reach. Okarin needs these values urgently, but he only has time to check the measurements in at most 10 sections. Moreover, he does not have proper permission to get measurements in sections beyond number 512, thus he may only ask about sections from 129 to 512 inclusively.

You need to help Okarin recover measures in sections from 1 to 128. Note that detectors in sections are not perfect, and can only show 7 digits and the sign. Thus, for example, if a detector approaches $+10\,000\,000$, it will show $-9\,999\,999$ instead. After that, $+10\,000\,001$ will be shown as $-9\,999\,998$, and so on.

Interaction Protocol

The first line of input contains a single integer N which is the number of seconds passed since the start of the experiment. All actual test cases will have $N = 10^7$.

After reading N, you may make queries in the form of "`?` x" on a single line, where the integer x ($129 \le x \le 512$) is the number of section you are interested in. In return, you will get a single integer c ($-10^7 < c < 10^7$) which is equal to the total charge of beams currently located in section x, as shown by the detector. You may make at most 10 queries.

When you are ready, output the answer in the form of "`!` c_1 c_2 $\ldots$ c_{128}" on a single line, where c_i is the total charge of the beam in the i-th section. Numbers should be formatted in the same way as detector measures (that is, $-10^7 < c_i < 10^7$).

Example

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

Note

The example for this problem has $N = 4$ and does not coincide with the actual first test case. In the example, the Stein's gate effect was activated three times. Let us assume that its first and second activations produced the same beams, and the third activation produced inverted beams. Then:

1. Initially, there is a single beam with charge -1 in the first section.
2. After 1.5 seconds, the initial beam goes through the center of the second section and is recreated in the center of the first section. At this moment, we have beams with charges -1 in the centers of first and second sections.
3. After 2.5 seconds, the initial beam goes through the center of the third section, while its copy goes through the center of the second section. Both beams are recreated the same in preceding sections. Then the first copy of the initial beam in the second section interferes with the new copy. After that, we have beams with total charges -1, -2 and -1 in the centers of first, second, and third sections correspondingly.
4. After 3.5 seconds, beams once again pass centers of sections. But this time, they are recreated with inverted signs. After all collisions, we will have a beam with charge $+1$ in the first section, a beam with charge $+1$ in the second section, a beam with charge -1 in the third section, and a beam with charge -1 in the fourth section.

In the example, the solution asks what are the charges in sections from 1 to 5, and then outputs the charges in sections from 1 to 4. Recall that, in the actual test cases, the solution can ask only about sections 129 to 512, and must then output the measurements in sections 1 to 128.

Problem D. Basis Change

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

You are given sequences $\{a_i\}_{i=1}^k$ and $\{b_i\}_{i=1}^k$. Consider all sequences $\{F_i\}_{i=1}^\infty$ which satisfy the following linear recurrence for all $n > k$:

$$F_n = \sum_{i=1}^k a_i F_{n-i}.$$

You have to find a sequence $\{c_i\}_{i=1}^k$ such that, for all such $\{F_i\}_{i=1}^\infty$, the following linear recurrence is satisfied for all $n > b_k$:

$$F_n = \sum_{i=1}^k c_i F_{n-b_i}.$$

Input

The first line of input contains a single integer k $(1 \le k \le 128)$.

The second line of input contains k integers $a_1, \dots, a_k$ $(1 \le a_i \le 10^9)$.

The third line of input contains k integers $b_1, \dots, b_k$ $(1 \le b_1 < b_2 < \dots < b_k \le 10^9)$.

It is guaranteed that the solution exists and is unique. Moreover, it is guaranteed that sequences a_i and b_i were uniformly randomly chosen among possible ones with some fixed number k for all test cases except the example.

Output

Output k integers $c_1, \dots, c_k$ on a single line. If $c_k = \frac{P}{Q}$ such that P and Q are coprime, output $(P \cdot Q^{-1}) \bmod (10^9 + 7)$. It is guaranteed that $Q \not\equiv 0 \pmod{10^9 + 7}$.

Example

standard input	standard output

2
1 1
1 3
	2 1000000006

Note

In the example, we have $F_n = F_{n-1} + F_{n-2}$. We can write $F_n - F_{n-1} = (F_{n-1} + F_{n-2}) - (F_{n-2} + F_{n-3})$. Thus $F_n = 2F_{n-1} - F_{n-3}$.

Problem E. Scored Nim

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

Alice and Bob want to play Nim. Well, some kind of it.

Initially, they have n heaps of stones, i-th heap containing a_i stones. Players take alternating turns, Alice moves first. On their turn, a player chooses any heap with at least two stones and splits it into two new heaps with at least one stone in each. After that, the player colors all stones in one of the new heaps white and all stones in the other one black. The game lasts until every heap contains only one stone.

After the game, Alice's score is the number of white stones on the board, and Bob's score is the number of black stones. Both players want to maximize their score and play optimally. What will be Alice's score?

Input

The first line of input contains a single integer n ($1 \le n \le 128$).

The second line of input contains n integers $a_1, a_2, \ldots, a_n$ ($2 \le a_i \le 128$).

Note that the initial colors of the stones are irrelevant.

Output

Output a single integer: Alice's score if both players play optimally.

Example

standard input	standard output
2 2 2	2

Note

In the example, no matter how players move, both heaps will be split between them equally.

Problem F. Milliarium Aureum

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

Roman Empire is vast and, as everybody knows, all roads lead to Rome!

Following the trail of the Last Centurion, professor Melody Song found an ancient map of Roman roads. The exact position of Rome was forgotten long ago, and Melody wants to recover it from the map. There are n cities on the map, numbered from 1 to n, and m roads. Each road is marked as either *minor* or *major*.

Major roads were used to travel to Rome and formed a spanning tree, and minor roads were used as alternatives or to travel between other cities. Each road had some fixed width. It is known that major roads were very wide: if we consider two paths from Rome to any other city, an arbitrary path P and a simple path Q using the major roads, the thinnest road on path P could not be wider than the thinnest road on path Q.

The map found by Melody contains information on every road in Roman Empire, namely its type t and width w. Your task is to help her determine which cities may correspond to Rome according to the map.

Input

The first line of input contains two integers n and m ($1 \le n, m \le 10^5$).

Each of the next m lines contains four integers, t_i, u_i, v_i, and w_i, which describe a bidirectional road from city u_i to city v_i with type t_i and width w_i ($1 \le u_i, v_i \le n$, $1 \le w_i \le 10^9$). The type is $t_i = 0$ for minor roads and $t_i = 1$ for major roads.

There may be multiple roads between the same pair cities, and there may be roads connecting a city to itself. It is guaranteed though that there is exactly one simple path via major roads between each pair of cities.

Output

On the first line, output an integer k which is number of cities which may be Rome.

On the second line, output k integers **in ascending order** which are the numbers of those cities.

It is guaranteed that there is at least one such city.

Examples

standard input	standard output

4 6
1 1 2 2
1 1 3 2
1 1 4 2
0 2 3 3
0 3 4 3
0 4 2 3
	1
	1

3 3
0 2 3 1
1 1 2 2
1 1 3 2
	3
	1 2 3

Problem G. Permutant

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

Consider an $n \times n$ matrix A. Let us denote element in i-th row and j-th column as $A_j^{(i)}$.

You are given a sequence $a_1, \dots, a_n$ and a permutation $\pi_1, \dots, \pi_n$ such that the first row is formed by sequence a_k:

$$A_k^{(1)} = a_k$$

And any consequent row is formed by applying permutation π_k to the previous one:

$$A_k^{(i)} = A_{\pi_k}^{(i-1)}$$

Your task is to calculate $\det A$: the determinant of matrix A. Since it may be very large, output it modulo $10^9 + 7$.

Input

The first line of input contains a single integer n ($1 \le n \le 5000$).

The second line of input contains n integers $a_1, \dots, a_n$ ($1 \le a_i \le 10^9$).

The third line of input contains n distinct integers $\pi_1, \dots, \pi_n$ ($1 \le \pi_i \le n$).

Output

Output a single number which is the answer to the problem.

Examples

standard input	standard output

4
1 1 1 1
4 2 3 1
	0

2
2 1
2 1
	3

Note

Recall that the determinant can be defined as follows:

$$\det A = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} A_{\sigma_i}^{(i)}$$

Here, S_n is the set of all permutations of $\{1, \dots, n\}$, and $\operatorname{sgn}(\sigma)$ is the sign of permutation σ: -1 if it has an odd number of inversions and $+1$ if this number is even. An inversion is a pair of indices (i, j) such that $i < j$ but $\sigma_i > \sigma_j$.

Problem H. Game Of Chance

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

Billionaires Robin McBobin and Ronald Dump are playing the Game of Chance.

The game takes n turns. On each turn, one of the players has the right of choice, Robin gets it for the first move. On each turn, an integer chosen uniformly and independently from 1 to m appears on the screen. The player with the right of choice has to choose whether to take this number and pass the right of choice to the opponent, or to give this number to the opponent but keep the right of choice for themselves.

Both Robin and Ronald are more interested in dominating their opponent than in gaining scores, so both will choose the option which would maximize the expected difference between their and the opponent's sum of numbers. Both players play optimally.

Let d_n be the expected value of difference between Robin's sum and Ronald's sum after n turns. It may be proven that for $m \geq 3$, there exists a rational number d such that $\lim\limits_{n\to\infty} d_n = d$. You have to find this number.

Input

The first line of input contains a single integer t which is the number of test cases ($1 \leq t \leq 5 \cdot 10^5$).

Each test case is given on a single line containing a single integer m ($3 \leq m \leq 10^9$).

Output

For each test case, if $d = \frac{P}{Q}$ such that P and Q are coprime, output $(P \cdot Q^{-1}) \bmod (10^9 + 7)$. It is guaranteed that $Q \not\equiv 0 \pmod{10^9 + 7}$.

Example

standard input	standard output
2	1
3	333333337
4	

Note

For $m = 3$, the answer is $d = 1$. For $m = 4$, the answer is $d = 1.333... = \frac{4}{3}$.

Problem I. Incomparable Pairs

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

You are given a string $s = s_1 s_2 \ldots s_n$. Consider an unordered pair of its substrings $\{a, b\}$. Let us call such pair *incomparable* if neither a is a substring of b nor b is a substring of a. You have to compute the number of incomparable pairs of substrings of s.

Input

The first line of input contains a single string s consisting of lowercase English letters ($1 \le |s| \le 10^5$).

Output

Output a single integer which is the answer to the problem.

Examples

standard input	standard output
abba	8
abacaba	64

Problem J. The Zong of the Zee

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

Imperturbable Captain of the Unterzee chases the Pirate-Poet. Their rivalry was set long ago and since then knows no limits. Every time the Captain arrives at the port where the Pirate-Poet was expected to be, they find nothing but the message from the Pirate-Poet acknowledging that she departed the port a few days ago and leaving the Captain with another line of never-ending masterpiece: the Zong of the Zee.

Both the Captain and the Pirate-Poet know that the end of the Zong will put an ultimate end to their rivalry. Once the Captain retrieves it, they will find the Pirate-Poet and the final clash will begin.

Port after port, line after line... Perhaps, the answer was there all along. The Captain knows they can't possibly last long enough to witness the end of the Zong, thus the only thing they can do is to outplay the Pirate-Poet.

So far, every single line of the poem was an anagram of the first one. Moreover, the Captain suspects that all lines are obtained in the same way, namely by applying some fixed permutation $\pi_1, \ldots, \pi_n$ to the previous line. That is, the character on position i on a new line was on position π_i on the previous one.

The Captain wants to recover the whole poem by guessing the intended permutation. But firstly they want to know how many variants they have to check. The Captain thoroughly copied every single line of the poem, m lines of n Unterzian letters each. Unfortunately, it is sometimes hard to understand the Captain, thus some of letters (but at most one per line) may be replaced by question marks ("?"), meaning that you are uncertain what letter exactly it is. You have to calculate the number of permutations $\pi_1, \ldots, \pi_n$ which are consistent with the given data: in other words, such permutations that there is a way to replace all "?" with some Unterzian letters so that each line of the poem is obtained from the previous one by applying this permutation.

Output the answer modulo $10^9 + 7$. The Captain knows what to do with this information. Probably.

Input

The first line of input contains two integers m and n ($2 \leq n, m \leq 3000$).

The i-th of the following m lines contains the i-th line of the poem. Each such line contains exactly n characters, each character being either an Unterzian letter or a question mark ("?"). Each line contains at most one question mark.

Each Unterzian letter has ASCII code between 33 and 126, except for "?" which has code 63.

Output

Output a single integer which is the answer to the problem.

Examples

standard input	standard output

`3 3`
`ib.`
`.?b`
`b.?`
	`1`

`2 4`
`abb?`
`ba?a`
	`4`

Problem K. Expected Value

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	**16 mebibytes**

Here is a game played with sequence $a_1, \ldots, a_n$. On each turn, the player chooses some position $i < n$ uniformly at random, replaces the element a_i with $a_i - a_{i+1}$, and then removes the element a_{i+1} from the sequence. This continues until there is only one element left. What is the expected value of the last element?

Input

The first line of input contains a single integer n ($2 \le n \le 4000$).

The second line of input contains n integers $a_1, \ldots, a_n$ ($1 \le a_i \le 4000$).

Output

If the answer is $\frac{P}{Q}$ such that P and Q are coprime, output a single integer which is $(P \cdot Q^{-1}) \bmod (10^9 + 7)$. It is guaranteed that $Q \not\equiv 0 \pmod{10^9 + 7}$.

Example

standard input	standard output
2 2 1	1

Note

Pay attention to the non-standard memory limit.