

Problem A. Forgotten Numbers

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

There's Pooh. Pooh hasn't much Brain, but he never comes to any harm. He does silly things and they turn out right. This time Pooh proudly pronounced an arithmetic \langle Expression \rangle .

There's Owl. Owl hasn't exactly got Brain, but he Knows Things. He would know right BNF to parse Pooh's $\langle \text{Expression} \rangle$.

$$\begin{aligned} \langle \text{Expression} \rangle &\longrightarrow \langle \text{MultTerm} \rangle \mid \langle \text{Expression} \rangle + \langle \text{MultTerm} \rangle \mid \langle \text{Expression} \rangle - \langle \text{MultTerm} \rangle \\ \langle \text{MultTerm} \rangle &\longrightarrow \langle \text{Term} \rangle \mid \langle \text{MultTerm} \rangle * \langle \text{Term} \rangle \\ \langle \text{Term} \rangle &\longrightarrow \langle \text{Number} \rangle \mid (\langle \text{Expression} \rangle) \mid \langle \text{MinTerm} \rangle \mid \langle \text{MaxTerm} \rangle \mid \langle \text{DivTerm} \rangle \\ \langle \text{MinTerm} \rangle &\longrightarrow \min (\langle \text{Expression} \rangle, \langle \text{Expression} \rangle) \\ \langle \text{MaxTerm} \rangle &\longrightarrow \max (\langle \text{Expression} \rangle, \langle \text{Expression} \rangle) \\ \langle \text{DivTerm} \rangle &\longrightarrow \text{div2} (\langle \text{Expression} \rangle) \end{aligned}$$

Also he knows that $\langle \text{Number} \rangle$ is some integer from 1 to 100 inclusive, min and max are minimum and maximum operations respectively, and $\text{div2}(x)$ is the greatest integer that doesn't exceed $\frac{x}{2}$.

And then there's Eeyore. And Eeyore is so miserable anyhow that he wouldn't mind about the fact that some numbers were forgotten and substituted with "?". Eeyore will patiently try to evaluate an $\langle \text{Expression} \rangle$ even when some numbers are unknown.

Input

The only line contains an `<Expression>` with forgotten numbers substituted with “?”. Its length does not exceed 100 characters. It is guaranteed that the expression can be parsed with the given BNF.

Output

Output the number — result of Eeyore’s computations if there is exactly one possible answer. Output “?” otherwise. Please note that Eeyore knows that numbers were integers ranging from 1 to 100.

Examples

[illegible]

Problem B. Colored Ring

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

There's a colored ring on the table. The ring has k sectors, and i -th sector has color i (all k colors are pairwise distinct). Sectors are numbered consecutively in clockwise direction. Kanga allowed Baby Roo to write n different integers from 1 to n on the ring. Number i should be written either on sector with color x_i , said Kanga, or on any other that is at most a_i sectors counterclockwise far from x_i or at most b_i sectors clockwise far.

Kanga said that the numbers are written correctly if the above restrictions are met, each sector has at most one number written on it and the numbers $1, 2, \dots, n$ are ordered in clockwise direction: if we start from the sector where we wrote 1 and go clockwise, we meet the other numbers in order $2, 3, \dots, n$ and then 1 again.

Now Baby Roo wants to know how many different ways are there to spoil the colored ring by writing the numbers correctly. Please note that it doesn't matter what number you write on the sector to get it spoiled.

Input

First line contains two integers n and k ($1 \leq n \leq 15$, $1 \leq k \leq 60$, $n \leq k$). Each of the next n lines contains three integers x_i, a_i, b_i ($1 \leq x_i \leq k$; $0 \leq a_i, b_i$; $a_i + b_i < k$). The numbers x_i are sorted in strictly increasing order.

Output

Output the number of different ways to spoil the ring.

Examples

<code>input.txt</code>	<code>output.txt</code>
1 5 1 2 1	4
3 8 4 0 3 5 0 3 6 0 0	3
3 7 2 2 4 4 1 3 6 1 5	35

Problem C. Christopher Robin is Learning Sorting Permutations

Input file: input.txt
Output file: output.txt
Time limit: 3 seconds (*4 seconds for Java*)
Memory limit: 256 MiB

```
// WRITTEN BY ME AND MR CORMEN HELPD
int psort(int L, int R) {
    if (L >= R)
        return 0;

    int i = L;
    int j = R;
    int x = Random.next();
    int x0 = A[x];

    assert(L <= x && x <= R);

    while (i < j) {
        while (A[i] < A[x]) i++;
        while (A[j] > A[x]) j--;
        if (i <= j) {
            if (i == x)
                x = j;
            else if (j == x)
                x = i;
            swap(A[i], A[j]);
            i++;
            j--;
        }
    }

    swap(A[x], A[x0]);

    return R - L + psort(L, x0 - 1) + psort(x0 + 1, R);
}
```

In the last line, function calls are made from left to right: `psort(L, x0 - 1)` is called first, `psort(x0 + 1, R)` is called second.

Given n and the sequence returned by `Random.next()`, find an array A such that `psort(1, n)` will return the maximal possible value. Array A should be a permutation of n integers from 1 to n . Please note that sorting may fail due to assertion statement.

Input

There is a single integer n on the first line ($1 \leq n \leq 50$). The second line contains n integers ranging from 1 to n separated by spaces. These are the numbers returned by n successive calls of `Random.next()` in the order they were made.

Output

If there is no such array **A**, output “No solution”. Otherwise, output “Solution exists” on the first line and the array **A** itself on the second line. If there are several possible arrays such that `psort(1, n)` returns the maximal possible value, output any one of them.

Examples

input.txt	output.txt
3 1 2 3	Solution exists 1 2 3
7 1 7 1 7 1 7 1	No solution

Problem D. Clever Plan

Input file: `input.txt`
Output file: `output.txt`
Time limit: 2 seconds (*4 seconds for Java*)
Memory limit: 256 MiB

There is Rabbit. He hasn't Learnt in Books, but he can always think of a Clever Plan. This time he prepared the plan for Pooh on how to fool a Heffalump and get a lot of Hunny. Or is it a plan on how to fool themselves?

Consider n pots of honey. Pooh randomly distributes pots between Heffalump and himself in such a way that both of them get at least one pot and all such distributions are equiprobable (the pots are indistinguishable, so there are $n - 1$ such distributions). Then they start to play the game.

The game consists of rounds, and it is over when either Heffalump or Pooh has no pots anymore. Each round is played using a fairly complex Rabbit's plan, so all we can say about a round is just three numbers: p, q, r ($p + q + r = 1$). One of the following three events happens: either Heffalump takes one of Pooh's pots with probability p , Pooh takes one of Heffalump's pots with probability q , or nothing happens with probability r . And so on, every round.

Consider i rounds were played, the game did not yet end, and distribution of pots between Heffalump and Pooh became exactly the same as it was before the first round. What is the probability of such an event?

Input

The first line contains the only integer T — the number of test cases ($1 \leq T \leq 200$).

Each test case consists of three lines. The first line of each test case contains an integer n — the number of pots ($3 \leq n \leq 26$). The second line of t -th test case contains three numbers: p_t, q_t, r_t ($0.3 \leq p_t, q_t, r_t < 1$, $p_t + q_t + r_t = 1$). The probabilities are given with at most two digits after the decimal point. The third line contains an integer k_t ($1 \leq k_t \leq 1600$) — the maximal number of rounds to play.

Output

For each test case, output k_t lines; i -th of them should contain the probability of the event described above after i rounds have passed. Answers should be given with relative error not exceeding 10^{-3} .

Example

input.txt	output.txt
1	4.000000e-01
3	2.500000e-01
0.3 0.3 0.4	1.720000e-01
5	1.201000e-01
	8.404000e-02

Problem E. Visiting

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

Winnie-the-Pooh loves visiting. There are several trackways in the Forest connecting his friends' houses. He wants to be able to go from every house to every other one.

But Pooh goes through each trackway in one direction only. That's because Pooh is always humming a song while walking, and he cannot hum it in reverse order, so he denies himself to go in opposite direction. That's why he needs some new trackways!

Input

The first line contains two integers n and m ($1 \leq n \leq 500$, $0 \leq m \leq 800$) — number of houses and number of existing trackways, correspondingly.

Next m lines contain descriptions of trackways. Each description consists of two numbers a_i and b_i ($a_i \neq b_i$) which means that Pooh goes through this trackway from house a_i to house b_i . Houses are numbered from 1 to n . All (a_i, b_i) pairs are different.

Output

On the first line output k — the minimal possible number of new trackways.

On the next k lines, output descriptions of these new trackways. Each description should follow the same format as the input. Your answer should be lexicographically the least possible.

Examples

<code>input.txt</code>	<code>output.txt</code>
3 2 1 2 1 3	2 2 1 3 1
4 2 1 2 3 4	2 2 3 4 1

Note

Sequence of pairs u_1, u_2, \dots, u_k is lexicographically smaller than sequence of pairs v_1, v_2, \dots, v_k if there exists an index i , $1 \leq i \leq k$, such that $u_i < v_i$ and $u_j = v_j$ for all $j < i$; here, pairs are also compared lexicographically.

Pair (a_1, b_1) is lexicographically smaller than pair (a_2, b_2) if either $a_1 < a_2$ or both $a_1 = a_2$ and $b_1 < b_2$ hold.

Problem F. Repeating b -ary

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

Trying to get out of Rabbit's house through the ground hole, Pooh got stuck in that hole. Christopher Robin said they should wait for Pooh to get thin. In order to sustain Pooh, Christopher Robin suggested him to write down b -ary representation of $\frac{1}{x}$. By the time Winnie-the-Pooh finishes his writings, he will surely get thin. How long will it take depends on the number of digits in *non-periodic* part and the number of digits in *periodic* part of b -ary representation. Given x , find these numbers.

Input

The only line contains two integers x and b ($2 \leq x \leq 10^{12}$, $2 \leq b \leq 10^{18}$) where b is the base of desired representation.

Output

Output the minimal possible lengths of non-periodic part and periodic part.

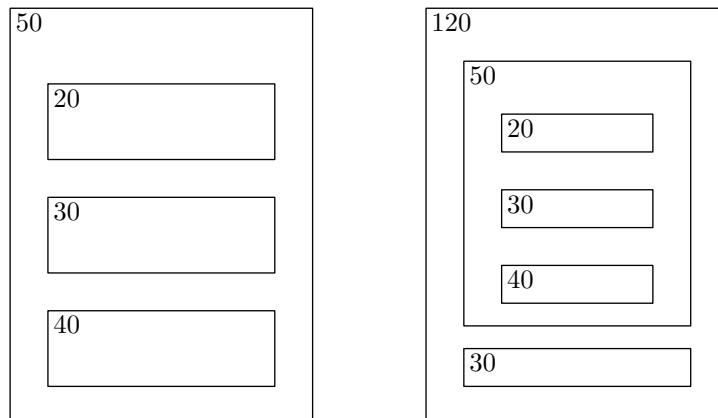
Examples

<code>input.txt</code>	<code>output.txt</code>
2 10	1 0
3 10	0 1
2 2	1 0
3 2	0 2

Problem G. Christopher Robin is Learning Object-oriented Programming

Input file: `input.txt`
Output file: `output.txt`
Time limit: 2 seconds (*4 seconds for Java*)
Memory limit: 256 MiB

Every object has its scalar value and its references list, possibly empty. The order of references in the list **does** matter. One can illustrate objects as follows:



Let us explain the picture. The left object has 50 as its scalar value. Its references list has three objects with scalar values 20, 30 and 40 (their references lists are empty); the right object has 120 as its scalar value and has references to the left object (illustrated as an embedded object) and an object with scalar value 30.

Two objects are equal if their illustrations are indistinguishable. Please note that objects may have circular references causing inability to illustrate the object on paper due to infinite embedding, but such illustrations are still virtually comparable.

Input

The first line of the input contains a single integer n ($2 \leq n \leq 10^5$) — the number of objects.

The following n lines describe objects. Each description starts with an integer s ($0 \leq s \leq 10^5$) — the scalar value of the object. Number s is followed by an integer m ($0 \leq m \leq 10$) and m numbers forming the list of references. Each reference is a number of some object. Objects are numbered from 1 to n in the order they are given in the input.

Output

Output exactly n lines.

On the i -th line output a positive integer — class number of the i -th object. Equal objects should belong to the same class, different objects should have different class numbers. The class numbers can be any integers from 1 to $2^{63} - 1$, inclusive.

Example

input.txt	output.txt
8	10
50 3 3 4 5	12
120 2 1 4	3
20 0	4
30 0	5
40 0	10
50 3 3 4 5	12
120 2 6 4	11
120 2 4 6	

Problem H. Expotition

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

One fine day Pooh had stumped up to the top of the Forest to see if his friend Christopher Robin was interested in Bears at all. You know, that was the day when Christopher Robin was preparing an Expedition to the North Pole. Christopher Robin asked Pooh to tell the others to get ready and to bring Provisions. Pooh had already alerted Rabbit and now hurried down to Piglet's house.

"Oh! Piglet," said Pooh excitedly, "we're going on an Expotition, all of us, with things to eat. To discover something."

For one moment Piglet was scared by *Something*, but then he said "if Christopher Robin is coming I don't mind anything." Of course, in case Christopher Robin weren't coming, Piglet won't ever go.

Expotition started at the top of the Forest, where everyone saw all Rabbit's friends-and-relations.

"I didn't ask them," explained Rabbit carelessly. "They just came. They always do. They can march at the end, after Eeyore."

Eeyore began to complain about Rabbit's relations, because in case he wants to sit down for a little rest, he have to brush away some of Rabbit's smaller friends first, which is annoying. Eeyore was really close to refuse the whole Expo.

Now Christopher Robin faced the problem. There are n persons including himself that are ready to go. And Christopher Robin wants k of them to go on an Expotition regardless of anything (he knows who are these k persons). And, trying to please everyone, Christopher Robin wants an Expotition group to be complete: there should be no way to add one or two or more persons in such a way that no one will complain. So, who will finally go to discover the North Pole?

Input

There are two integers on the first line: n and m ($1 \leq n \leq 150\,000$, $0 \leq m \leq 150\,000$). Next m lines contain complaints of two different kinds. Each complaint consists of three integers t_i , a_i and b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$). If $t_i = 1$ then a_i won't go on an Expotition if annoying person b_i is coming. (Christopher Robin failed to remember names of all Rabbit's friends-and-relations, so he just gave everyone a number instead of a name.) And if $t_i = 2$ then a_i won't ever go on any adventure without his friend b_i .

Persons are quite friendly, so the number of complaints with $t_i = 1$ won't exceed 40. All pairs (a_i, b_i) are unique.

Next line contains an integer k ($0 \leq k \leq n$). And, if k is positive, the last line contains k integers — the persons whom Christopher Robin will surely take on an Expotition.

Output

On the first line output a single integer s — the number of persons going to discover the North Pole. On the second line output s integers (separated by spaces, pairwise different) — who are these persons. Feel free to output any answer that is complete. It is guaranteed that at least one answer exists.

Examples

input.txt	output.txt
3 2 1 1 2 1 2 3 1 3	2 1 3
3 3 2 1 2 2 1 3 1 2 3 0	1 2

Problem I. Wrong Sort of Bees

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

Trying to get some honey from the top of large oak-tree, Pooh was holding a balloon in his paws and singing a song:

*How sweet to be a Cloud
Floating in the Blue!
Every little cloud
Always sings aloud.*

*How sweet to be a Cloud
Floating in the Blue!
It makes him very proud
To be a little cloud.*

There was a wind to blow him nearer to the tree, and Pooh moved along a straight line with constant height and constant speed V_{Pooh} meters per minute. Unfortunately, direction wasn't accurate enough and he missed the tree.

Pooh suddenly noticed a bee directly at distance R meters in front of him in the direction of his movement. It then began to fly around him, also maintaining the same height as Pooh. Pooh noticed (from his point of view) that the bee moves in a Definitely Suspicious way. It flies around him with constant angular velocity Ω_{bee} revolutions per minute. Furthermore, the distance between the bee and the Bear is reducing as fast as V_{bee} meters per minute!

Pooh had just been thinking and faced a very important problem when the bee touched him. How long was the trajectory of the bee? It's so important because Pooh knows that a bee cannot fly too far, and if it flies too far indeed, then it belongs to the wrong sort of bees that make the wrong sort of honey in which Pooh isn't interested at all.

For the purpose of this problem, the sizes of Pooh and the bee are negligible.

Input

The first line contains the only integer T ($1 \leq T \leq 100$) — number of test cases.

Each test case is a single line that contains four integers: R , V_{Pooh} , V_{bee} , Ω_{bee} . It is guaranteed that $1 \leq R, V_{Pooh}, V_{bee}, \Omega_{bee} \leq 100$.

Output

Output T lines each of which contains an answer for the corresponding test case — the length of the bee's trajectory. Numbers should be given with absolute or relative error of at most 10^{-4} .

Example

<code>input.txt</code>	<code>output.txt</code>
1 1 1 1 1	3.36757

Problem J. Maximal Sum

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

“I think Christopher Robin is a pretty cool guy,” said Pooh carelessly, “eh sums integers without carrying and doesn’t afraid of anything.”

“Oh, Pooh! Do you mean that Christopher Robin writes something like $5 + 5 = 0$, $19 + 3 = 12$ and $75 + 25 = 90$?” asked Piglet anxiously.

“Yes,” said Bear, “And he calls that *digitwise sum*. He just sums pairs of corresponding digits modulo 10. But what maximal sum can Christopher Robin make if I ask him to use numbers from *my very own list* only? Do you wonder, Piglet?”

Input

On the first line Winnie-the-Pooh wrote a single integer n ($1 \leq n \leq 10^5$). After a little break he wrote n more lines: on line number $i + 1$ he wrote an integer x_i ($0 < x_i < 10^6$). When Winnie-the-Pooh had finished his writings, he brought them right to Christopher Robin.

Output

On the first line output a single integer a — maximal digitwise sum Christopher Robin can make. On the second line output integer number m — how many numbers Christopher Robin have used to make sum a . But $0 \leq m \leq 108$ because Winnie-the-Pooh is a Bear of Very Little Brain and he won’t check the answer if m is greater than one hundred eight. On the third line output m numbers used to make sum a separated by spaces. Remember that each number in the third line should be one of x_i numbers from the Pooh’s list.

If there is no answer Winnie-The-Pooh can check, output just a on the first line and -1 on the second line instead.

Examples

<code>input.txt</code>	<code>output.txt</code>
1 1	9 9 1 1 1 1 1 1 1 1 1
2 12 13	99 9 12 12 12 12 12 12 12 12 13
2 15 5	95 9 15 15 15 15 15 15 15 15 15

Problem K. Leapfrog

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

All Rabbit's friends-and-relations play a leapfrog. Today, n of them have stood in a straight line and begun to play. How do they play a leapfrog? They do it as follows:

1. One of friends-and-relations (let's call him X) chooses another one (Y) to jump over his back.
2. X jumps over Y 's back in such a way that he still stands on the same line as others do, and the distance between X and Y remains the same as before the jump.

Rabbit's friends-and-relations are quite small, so any of them can easily fit into pretty the same place.

Input

First line contains the only integer n ($1 \leq n \leq 100$).

Second line contains n numbers a_i separated by spaces — initial positions of Rabbit's friends-and-relations on the line. So i -th person stands in position a_i .

Third line contains n numbers separated by spaces — desired positions of Rabbit's friends-and-relations. Current positions are considered desired iff it is possible to permute numbers of current positions in such a way that the list of positions' numbers becomes equal to the list of desired positions' numbers.

All the numbers are integers ranging from 1 to 100 inclusive.

Output

Output "No" if Rabbit's friends-and-relations will never come into desired positions.

Otherwise, output "Yes" and then output the performed leapfrogs (you can output none, if it's the case). Each next line should contain a pair of numbers X and Y for each time when X jumps over Y 's back (Rabbit's friends-and-relations' names are 1, 2, ..., n).

The number of jumps shouldn't exceed 5×10^5 , and no one should ever stand in a position that is more than 10^8 units far from position 0.

Examples

<code>input.txt</code>	<code>output.txt</code>
2 1 2 5 6	Yes 1 2 2 1 1 2 2 1
2 1 2 1 3	No