



## Problem B. Colored Ring (Division 1 Only!)

Input file:            Output file:           output .txt  
Time limit:            3 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, ...,  $n$  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, ...,  $n$  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

input.txt	output.txt
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 (Division 1 Only!)

Input file:           input.txt  
Output file:          output.txt  
Time limit:           4 seconds (*7 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:            .txt  
Output file:           output.txt  
Time limit:            2 seconds (*7 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 (Division 1 Only!)

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

input.txt	output.txt
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:            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.

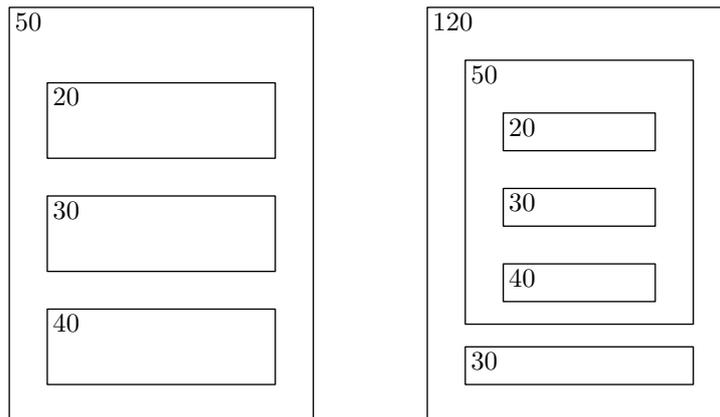
### Examples

input .txt	output .txt
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 (Division 1 Only!)

Input file: `input.txt`  
Output file: `output.txt`  
Time limit: 3 seconds (*6 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 (Division 1 Only!)

Input file:            .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:            Output file:           output.txt  
Time limit:           2 seconds (*5 seconds for Java*)  
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  $\Omega_{bee}$  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:  $R, V_{Pooh}, V_{bee}, \Omega_{bee}$ . 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

input.txt	output.txt
1 1 1 1 1	3.36757

## Problem J. Maximal Sum (Division 1 Only!)

Input file:            .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

input.txt	output.txt
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 (Division 1 Only!)

Input file:            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  $5 \times 10^5$ , and no one should ever stand in a position that is more than  $10^8$  units far from position 0.

### Examples

input.txt	output.txt
2 1 2 5 6	Yes 1 2 2 1 1 2 2 1
2 1 2 1 3	No

## Problem L. Shops and Balloons (Division 2 Only!)

Input file:            .txt  
Output file:           output.txt  
Time limit:           12 seconds  
Memory limit:         256 MiB

In country, where Forest is placed, there are  $N$  cities ( $N \leq 5,000$ ), with each city having various trade routes to other cities. In total, there are  $T$  two-sided trade routes ( $0 \leq T \leq 25 \cdot 10^6$ ). For each trade route between two cities  $x$  and  $y$ , there is a transportation cost  $C(x, y)$  to ship between the cities, where  $C(x, y) \geq 0$ ,  $C(x, y) \leq 10^4$  and  $C(x, y) = C(y, x)$ . Out of the  $N$  cities,  $K$  ( $1 \leq K \leq N$ ) of these cities have stores with really nice balloons that can be purchased on-line.

The price for each balloon in city  $x$  is  $P_x$  ( $0 \leq P_x \leq 10^4$ ).

Find the minimal price to purchase one balloon on-line and have it shipped to a particular city  $D$ , located near Forest ( $1 \leq D \leq N$ ) using the cheapest possible trade-route sequence. Notice that it is possible to purchase the balloon in city  $D$  and thus require no shipping charges.

### Input

The first line of input contains integer  $N$  — the number of cities. You can assume the cities are numbered from 1 to  $N$ . The second line of input contains  $T$ , the number of trade routes. The next  $T$  lines each contain 3 integers,  $x$   $y$   $C(x, y)$ , to denote the cost of using the trade route between cities  $x$  and  $y$  is  $C(x, y)$ . The next line contains the integer  $K$ , the number of cities with a store that sells really nice balloons on-line. The next  $K$  lines contains two integers,  $z$  and  $P_z$ , to denote that the cost of a balloon in city  $z$  is  $P_z$ . The last line contains the integer  $D$ , the destination city.

### Output

Output the minimal total cost of purchasing a balloon on-line and shipping it to city  $D$ .

### Example

input.txt	output.txt
3	6
3	
1 2 4	
2 3 2	
1 3 3	
3	
1 14	
2 8	
3 3	
1	

## Problem M. Christopher Robin is Learning Bar Graphs (Division 2 Only!)

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

A bar graph is a graph consisting of vertical stacks of blocks ('#') of different height.

You will have to draw the graph given the values it has to represent. You will be given the desired height of the graph in number of blocks,  $H$ , and the values of the  $N$  bars of the graph  $X_1, \dots, X_N$ .

The bar with the maximum value should have height  $H$  and the other bars should be scaled proportionally to that a bar has  $n$  blocks if and only if the proportional value is more or equal than  $n$ , but less than  $n + 1$ .

### Input

Input file consists of one line, beginning with two integers  $H$  and  $N$  (graph height in lines and number of values to be represented, respectively). Then follow these values —  $N$  integers  $X_1, \dots, X_N$ , at least one between them is non-zero ( $1 \leq H \leq 100$ ,  $1 \leq N \leq 100$ ,  $0 \leq X_i \leq 10^4$ ).

### Output

Print requested bar graph (see example for reference).

### Example

input.txt	output.txt
16 7 30 21 39 1 17 16 62	<pre># # # # # # # # # # # # # # # # # ### # ### ### ### ### ### ### ### ###</pre>

## Problem N. Encryption (Division 2 Only!)

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

For security reasons, Pooh and Rabbit have decided to encode their messages using a Caesar Cipher.

Each message will consist of upper and lower case Latin characters, along with some punctuation. For each character in the message, sender increases its value by the shift amount, preserving case and wrapping around in case of overflow. For example, with a shift of 1 'a' becomes 'b', 'b' becomes 'c', 'z' becomes 'a', 'K' becomes 'L' etc. Non-letter characters should be output unchanged.

Your task is to encrypt given message with given shift amount.

### Input

Input will consist of the int integer  $n$  — length of shift ( $1 \leq n \leq 10^9$ ), followed by the contents of the message. Each message will be no longer than 1024 characters, but may contain multiple lines. Each line will consist of characters with codes between 32 and 126, inclusively.

### Output

Print encoded message.

### Example

input.txt	output.txt
27	Uijt jt @ Uftu Nfttbhf!
This is @ Test Message!	Op QjHmFu dbo sfbe ju!
No PiGlEt can read it!	Bsf zpv tvsf uibu
Are you sure that	Dpef xpslt?
Code works?	

## Problem O. Cool numbers (Division 2 Only!)

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

Piglet likes interesting numbers like 64. It turns out that 64 is both a square and a cube, since  $64 = 8^2$  and  $64 = 4^3$ . Piglet calls these numbers *cool*. More formally, number  $n$  is called *cool*, if  $n = k^p$  and  $n = l^q$ , where  $k, p, l, q$  are positive integers,  $\gcd(p, q) = 1$  and both  $p$  and  $q$  are greater, than 1.

Write a program that helps Piglet figure out how many integers in a given range are cool.

### Input

On the first line of input, you are given an integer  $a$  ( $1 \leq a \leq 10^8$ ). On the second line of input, you are given an integer  $b$  such that  $a \leq b \leq 10^8$  — first and last integer in a given range.

### Output

The output should be the number of cool numbers in the range  $a$  to  $b$  inclusively.

### Example

input.txt	output.txt
1 82	2
101 1001	1

## Problem P. Lights (Division 2 Only!)

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

For his birthday, Eeyore has been given a grid of  $R$  rows of lights, with each row containing  $L$  lights. Lights can be in one of two states: on or off. For this question, the topmost row is row  $R$ , and the bottom-most row is row 1. Also, beside all rows except the topmost row (row  $R$ ), there is a button which can be pushed.

Pushing the button beside row  $k$  ( $1 \leq k < R$ ) will perform an «exclusive-or» operation on each light of row  $k$ , which is described below. Consider column  $i$  in row  $k$ , where  $1 \leq i \leq L$ . If the lights in column  $i$  of row  $k$  and column  $i$  of row  $k + 1$  are both the same (i.e., both on, or both off), then pushing the button beside row  $k$  will cause the light in column  $i$  of row  $k$  to be off. If the lights in column  $i$  of row  $k$  and column  $i$  of row  $k + 1$  are different (i.e., one is on, and the other is off), then pushing the button beside row  $k$  will cause the light in column  $i$  of row  $k$  to be on. An example is shown below, for  $L = 4$ :

Column numbers	1	2	3	4
Row $k + 1$	on	on	off	off
Row $k$ before button pushed	on	off	on	off
Row $k$ after button pushed	off	on	on	off

You are told which lights are initially on and which are initially off. You must calculate how many different light patterns are possible for the bottom row by any sequence of button pushes.

### Input

The first line of input will contain the integer  $R$ , the number of rows ( $1 < R < 30$ ). The second line of input will contain the integer  $L$  ( $1 \leq L < 8$ ), the number of lights per row. The next  $R$  lines of input will contain  $L$  integers, where the integer will either be 0 (to indicate «off») or 1 (to indicate «on»). Pairs of consecutive integers will be separated by one space character. These  $R$  lines will be given in top- down order: that is, the third line of input will be the description of row  $R$ , the fourth line will be  $R - 1$ , and so on, until the last line describes the bottom row.

### Output

Output the number of possible light patterns of the bottom row.

### Example

input.txt	output.txt
4 3 0 0 1 0 1 1 1 0 1 0 0 1	4