A. Diverse Game

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

Petr, watching Sergey's stream, came up with a matrix a, consisting of n rows and m columns (the number in the i-th row and j-th column is denoted as $a_{i,j}$), which contains all integers from 1 to $n \cdot m$. But he didn't like the arrangement of the numbers, and now he wants to come up with a new matrix b, consisting of n rows and m columns, which will also contain all integers from 1 to $n \cdot m$, such that for any $1 \leq i \leq n, 1 \leq j \leq m$ it holds that $a_{i,j} \neq b_{i,j}$.

You are given the matrix a, construct **any** matrix b that meets Petr's requirements, or determine that it is impossible.

Hurry up! Otherwise, he will donate all his money to the stream in search of an answer to his question.

Input

Each test consists of multiple test cases. The first line contains an integer t ($1 \leq t \leq 10^3$) — the number of test cases. Then follows the description of the test cases.

The first line of each test case contains two integers n and m ($1 \leq n, m \leq 10$) — the number of rows and columns of matrix a.

The next n lines contain m integers each, describing matrix a. The i-th of these lines contains the elements of matrix $a_{i,1}, a_{i,2}, \ldots, a_{i,m}$.

It is guaranteed that all numbers in matrix a are distinct and $1 \leq a_{i,j} \leq n \cdot m$.

It is guaranteed that the sum of $n \cdot m$ over all test cases does not exceed $5 \cdot 10^4$.

Output

For each test case, output $n \cdot m$ integers — any suitable matrix b, or -1 if such a matrix does not exist.

Standard Input	Standard Output
5 1 1 1 2 1 2 1 1 5 2 4 5 3 1 2 4 1 2 3 4 5 6 7 8 3 3 4 2 1 9 8 3 6 7 5	-1 1 2 4 5 3 1 2 6 7 8 5 2 3 4 1 8 3 9 7 5 6 2 1 4

Note

In the first test case, there is only one element in the matrix, so matrix b is the only matrix and it does not fit.

In the second test case $a_{1,1} = 2 \neq 1 = b_{1,1}$, $a_{2,1} = 1 \neq 2 = b_{2,1}$.

,

B. Fun Game

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

Vova really loves the XOR operation (denoted as $\oplus$). Recently, when he was going to sleep, he came up with a fun game.

At the beginning of the game, Vova chooses two binary sequences s and t of length n and gives them to Vanya. A binary sequence is a sequence consisting only of the numbers 0 and 1. Vanya can choose integers l, r such that $1 \leq l \leq r \leq n$, and for all $l \leq i \leq r$ **simultaneously** replace s_i with $s_i \oplus s_{i-l+1}$, where s_i is the i-th element of the sequence s.

In order for the game to be *interesting*, there must be a possibility to win. Vanya wins if, with an **unlimited** number of actions, he can obtain the sequence t from the sequence s. Determine if the game will be *interesting* for the sequences s and t.

Input

Each test consists of multiple test cases. The first line contains an integer q ($1 \leq q \leq 10^4$) — the number of test cases. Then follows the description of the test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 2 \cdot 10^5$) — the length of the sequences s and t.

The second line of each test case contains a binary sequence s of length n.

The third line of each test case contains a binary sequence t of length n.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output "Yes" if the game will be *interesting*, otherwise output "No".

You can output each letter in any case (for example, the strings "yEs", "yes", "Yes", and "YES" will be recognized as a positive answer).

Standard Input	Standard Output
6 1 0 1 7 0110100 0110100 9 100101010 101111110 4 0011 1011	NO YES YES NO YES YES

4 0100 0001 8 10110111 01100000	

Note

In the first test case, Vanya will not be able to change the sequence s with the only possible action of choosing $l = r = 1$.

In the second test case, the sequences s and t are already equal.

In the third test case, Vanya can act as follows:

1. Choose $l = 3$ and $r = 5$, then s will become $\mathtt{101101010}$.
2. Choose $l = 5$ and $r = 6$, then s will become $\mathtt{101111010}$.
3. Choose $l = 7$ and $r = 7$, then s will become $\mathtt{101111110}$.

C. Hungry Games

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

Yaroslav is playing a computer game, and at one of the levels, he encountered n mushrooms arranged in a row. Each mushroom has its own level of toxicity; the i-th mushroom from the beginning has a toxicity level of a_i. Yaroslav can choose two integers $1 \le l \le r \le n$, and then his character will take turns from left to right to eat mushrooms from this subsegment one by one, i.e., the mushrooms with numbers $l, l+1, l+2, \ldots, r$.

The character has a toxicity level g, initially equal to 0. The computer game is defined by the number x — the maximum toxicity level at any given time. When eating a mushroom with toxicity level k, the following happens:

1. The toxicity level of the character is increased by k.
2. If $g \le x$, the process continues; otherwise, g becomes zero and the process continues.

Yaroslav became interested in how many ways there are to choose the values of l and r such that the final value of g is not zero. Help Yaroslav find this number!

Input

Each test consists of multiple test cases. The first line contains an integer t ($1 \le t \le 10^4$) — the number of test cases. Then follows the description of the test cases.

The first line of each test case contains two integers n, x ($1 \le n \le 2 \cdot 10^5, 1 \le x \le 10^9$) — the number of mushrooms and the maximum toxicity level.

The second line of each test case contains n numbers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single number — the number of subsegments such that the final value of g will not be zero.

Standard Input	Standard Output
5 4 2 1 1 1 1 3 2 1 2 3 1 6 10 6 3 1 2 1 4 3 8 5 999999999 999999999 999999998 1000000000 1000000000 500000000	8 2 0 10 7

Note

In the first test case, the subsegments $(1, 1)$, $(1, 2)$, $(1, 4)$, $(2, 2)$, $(2, 3)$, $(3, 3)$, $(3, 4)$ and $(4, 4)$ are suitable.

In the second test case, non-zero g will remain only on the subsegments $(1, 1)$ and $(2, 2)$.

In the third test case, on the only possible subsegment, g will be zero.

D. Funny Game

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

Vanya has a graph with n vertices (numbered from 1 to n) and an array a of n integers; initially, there are no edges in the graph. Vanya got bored, and to have fun, he decided to perform $n-1$ operations.

Operation number x (operations are numbered in order starting from 1) is as follows:

- Choose 2 different numbers $1 \leq u, v \leq n$, such that $|a_u - a_v|$ is divisible by x.
- Add an undirected edge between vertices u and v to the graph.

Help Vanya get a connected* graph using the $n-1$ operations, or determine that it is impossible.

---

* A graph is called connected if it is possible to reach any vertex from any other by moving along the edges.

Input

Each test consists of multiple test cases. The first line contains an integer t ($1 \leq t \leq 10^3$) — the number of test cases. Then follows the description of the test cases.

The first line of each test case contains the number n ($1 \leq n \leq 2000$) — the number of vertices in the graph.

The second line of each test case contains n numbers $a_1, a_2, \cdots a_n$ ($1 \leq a_i \leq 10^9$).

It is guaranteed that the sum of n over all test cases does not exceed 2000.

Output

For each test case, if there is no solution, then output "No" (without quotes).

Otherwise, output "Yes" (without quotes), and then output $n-1$ lines, where in the i-th line, output the numbers u and v that need to be chosen for operation i.

You can output each letter in any case (for example, the strings "yEs", "yes", "Yes", and "YES" will be recognized as a positive answer).

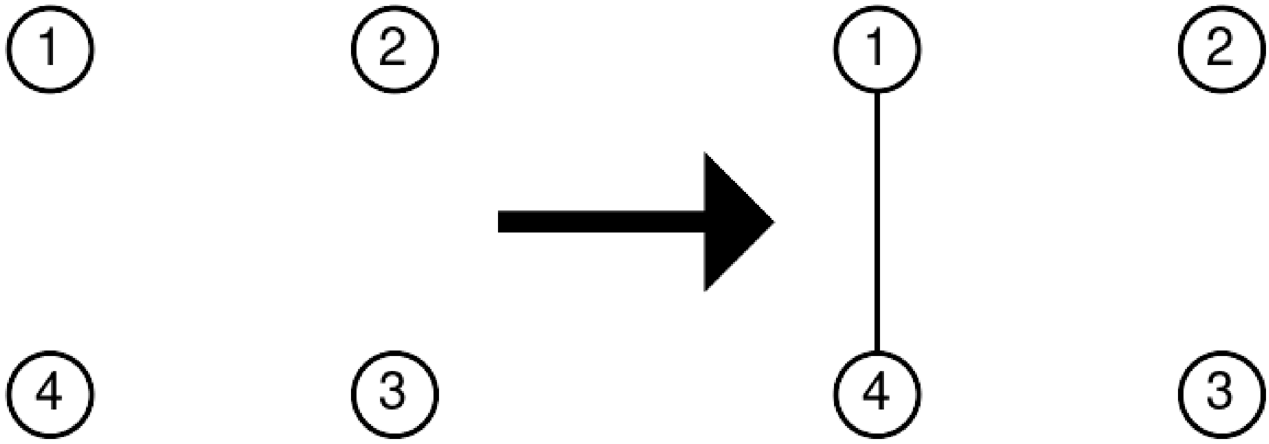
Standard Input	Standard Output
8 2 1 4 4 99 7 1 13 5 10 2 31 44 73 5 87 6 81 44 32 5 62 35 33 79 16 5 6 51 31 69 42	YES 2 1 YES 4 1 2 1 3 2 YES 5 1 4 1 3 1 2 1 YES 4 1

5 52 63 25 21 5 12 33 40 3 11 31 43 37 8 50 5 12 22	3 1 2 1 5 4 YES 3 1 5 1 2 1 4 2 YES 4 1 5 1 2 1 3 2 YES 2 1 5 2 3 1 4 3 YES 9 1 12 9 11 1 10 1 6 1 7 6 2 1 8 2 5 2 3 1 4 1

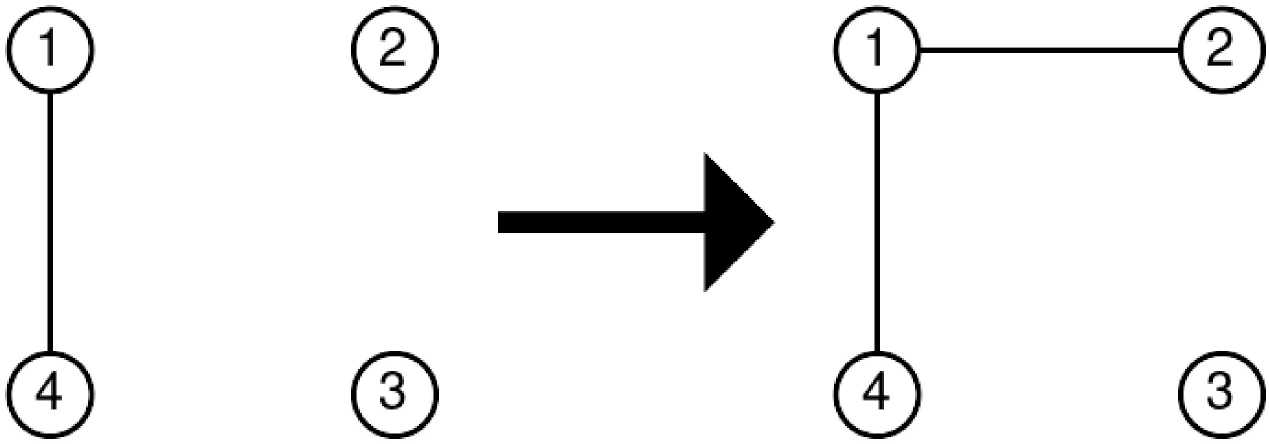
Note

Let's consider the second test case.

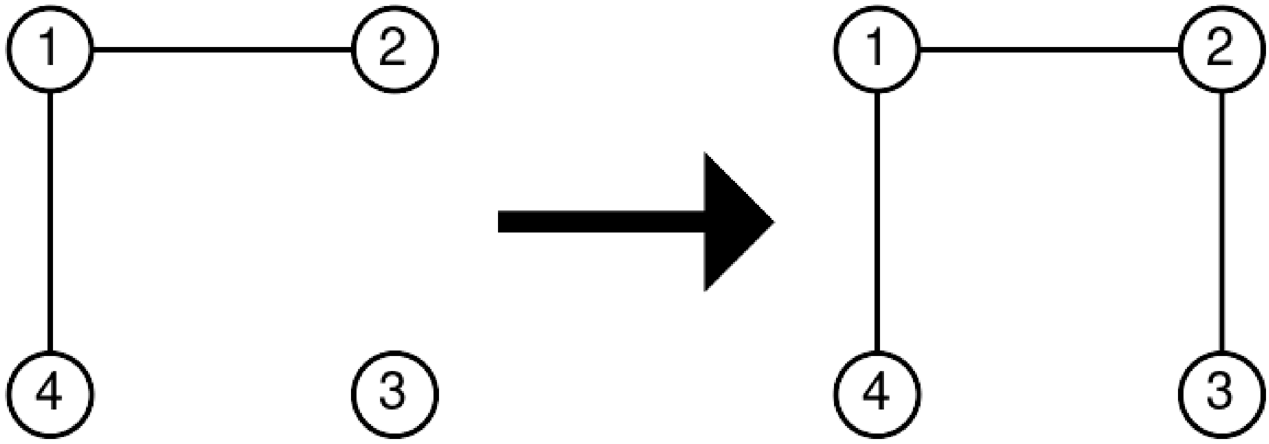
- First operation $(x = 1)$: we can connect vertices 4 and 1, since $|a_4 - a_1| = |13 - 99| = |-86| = 86$, and 86 is divisible by 1.



- Second operation $(x = 2)$: we can connect vertices 2 and 1, since $|a_2 - a_1| = |7 - 99| = |-92| = 92$, and 92 is divisible by 2.



- Third operation $(x = 3)$: we can connect vertices 3 and 2, since $|a_3 - a_2| = |1 - 7| = |-6| = 6$, and 6 is divisible by 3.



From the picture, it can be seen that a connected graph is obtained.

,

E. Wooden Game

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

You are given a forest of k rooted trees*. Lumberjack Timofey wants to cut down the entire forest by applying the following operation:

- Select a subtree† of any vertex of one of the trees and remove it from the tree.

Timofey loves bitwise operations, so he wants the bitwise OR of the sizes of the subtrees he removed to be maximum. Help him and find the maximum result he can obtain.

---

* A tree is a connected graph without cycles, loops, or multiple edges. In a rooted tree, a selected vertex is called a root. A forest is a collection of one or more trees.

† The subtree of a vertex v is the set of vertices for which v lies on the shortest path from this vertex to the root, including v itself.

Input

Each test consists of multiple test cases. The first line contains an integer t ($1 \le t \le 10^4$) — the number of test cases. Then follows the description of the test cases.

The first line of each test case contains a single integer k ($1 \le k \le 10^6$) — the number of trees in the forest.

This is followed by a description of each of the k trees:

The first line contains a single integer n ($1 \le n \le 10^6$) — the size of the tree. The vertices of the tree are numbered with integers from 1 to n. The root of the tree is vertex number 1.

The second line contains $n - 1$ integers $p_2, p_3, \ldots p_n$ ($1 \le p_i < i$), where p_i — the parent of vertex i.

It is guaranteed that the sum of k and n for all sets of input data does not exceed 10^6.

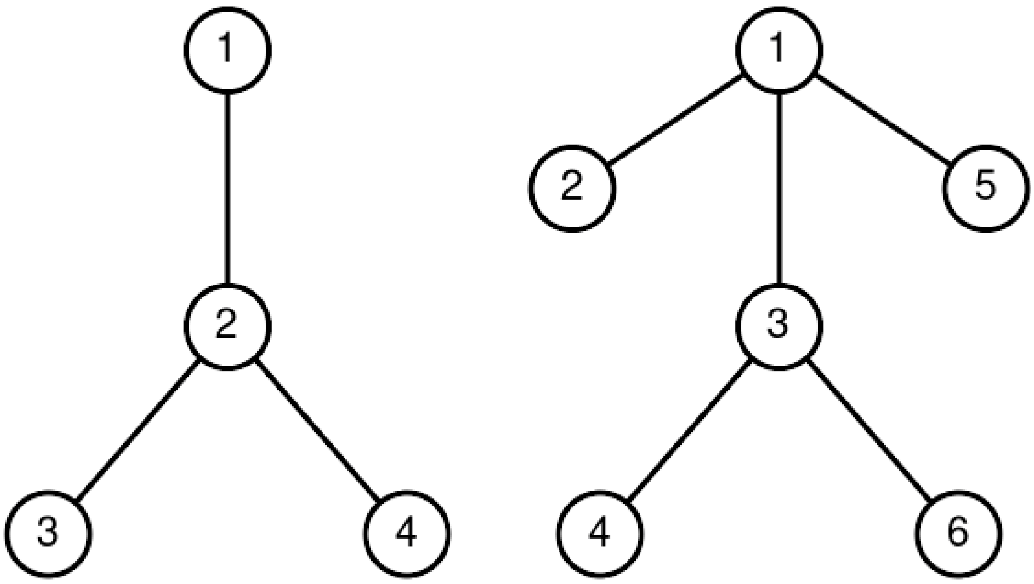
Output

For each test case, output a single integer — the maximum result that can be obtained.

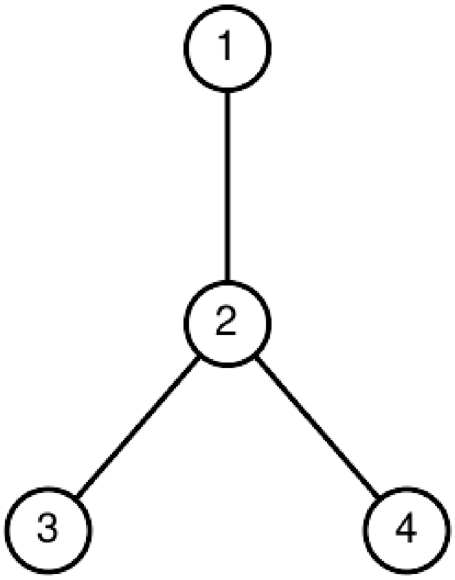
Standard Input	Standard Output
3 1 1 2 4 1 2 2 6 1 1 3 1 3 1 10 1 2 2 1 1 5 7 6 4	1 7 10

Note

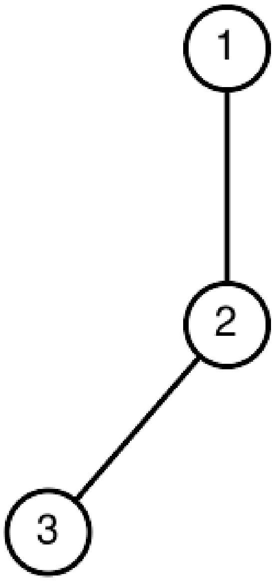
In the second test case, the trees look like this:



The first operation removes the entire second tree.



The second operation removes vertex 4 from the first tree.



The third operation removes the first tree. The result is $6|1|3 = 7$ ($|$ denotes bitwise OR).

In the third test case, the entire tree needs to be removed.

F. Stardew Valley

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

Pelican Town represents n houses connected by m bidirectional roads. Some roads have *NPCs* standing on them. Farmer Buba needs to walk on each road with an NPC and talk to them.

Help the farmer find a route satisfying the following properties:

- The route starts at some house, follows the roads, and ends at the same house.
- The route does not follow any road more than once (in both directions together).
- The route follows each road with an NPC exactly once.

Note that the route can follow roads without NPCs, and you do **not** need to minimize the length of the route.
It is **guaranteed** that you can reach any house from any other by walking on the roads with NPCs only.

Input

Each test consists of multiple test cases. The first line contains an integer t ($1 \leq t \leq 10^4$) — the number of test cases. Then follows the description of the test cases.

The first line of each test case contains two integers n and m ($2 \leq n \leq 5 \cdot 10^5, 1 \leq m \leq 5 \cdot 10^5$) — the number of houses and roads in `Pelican Town` respectively.

In each of the next m lines, three integers u, v, and c ($1 \leq u, v \leq n, c = 0/1$) are given — the ends of the road and whether an `NPC` is on this road. If $c = 1$, then the road has an NPC. If $c = 0$, then the road has no NPC.

The graph may contain multiple edges and loops, and if there are multiple edges with `NPCs` standing on them, the route must follow each of these roads.

It is guaranteed that you can reach any house from any other by walking on the roads with NPCs only.

It is guaranteed that the sum of n and m for all test cases does not exceed $5 \cdot 10^5$.

Output

For each test case, if there is no solution, then output "`No`" (without quotes).

Otherwise, output "`Yes`" (without quotes), and then output k — the number of roads in the route. In the next line, output $k + 1$ numbers — the houses of the route in the order of traversal. Note that the first house should match the last one, as the route is cyclic.

If there are multiple answers, you can print any of them.

You can output each letter in any case (for example, the strings "`yEs`", "`yes`", "`Yes`", and "YES" will be recognized as a positive answer).

Standard Input	Standard Output
3 3 2 1 2 1	NO YES 3

2 3 1 3 3 1 2 1 1 3 1 2 3 0 5 9 1 2 0 5 2 1 5 4 1 5 1 1 2 3 1 5 2 1 4 1 0 4 3 0 5 2 0	1 2 3 1 YES 7 1 2 5 4 3 2 5 1

Note

Note that in the third test case, there are multiple edges $(5, 2)$. You must walk on two of them.

,

G. Minecraft

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   512 megabytes
```

After winning another Bed Wars game, Masha and Olya wanted to relax and decided to play a new game. Masha gives Olya an array a of length n and a number s. Now Olya's task is to find a non-negative number x such that $\sum_{i=1}^{n} a_i \oplus x = s$. But she is very tired after a tight round, so please help her with this.

But this task seemed too simple to them, so they decided to make the numbers larger (up to 2^k) and provide you with their binary representation.

Input

Each test consists of several test cases. The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. Then follows the description of the test cases.

The first line of each test case contains two integers n and k ($1 \le n, k, n \cdot k \le 2 \cdot 10^6$) — the length of the array a and the length of the binary representation of all numbers.

The second line contains a string of length k, consisting of zeros and ones — the binary representation of the number s, starting from the most significant bits.

The next n lines also contain strings of length k, consisting of zeros and ones, the i-th of these strings contains the binary representation of the number a_i, starting from the most significant bits.

It is guaranteed that the sum of the values $n \cdot k$ for all test cases does not exceed $2 \cdot 10^6$.

Output

For each test case, output a string of length k on a separate line, consisting of zeros or ones — the binary representation of any suitable number x ($x \ge 0$), starting from the most significant bits, or -1 if such x does not exist.

Standard Input	Standard Output
4 4 5 01011 01110 00110 01100 01111 2 8 00101001 10111111 10011110 5 4 0101 0010 0000	01110 10011010 0010 -1

0000 0010 0011 6 5 00011 10110 11001 01010 11100 10011 10000	

Note

In the first test case, $s = 11, a = [14, 6, 12, 15]$, if $x = 14$, then $\sum_{i=1}^{n} a_i \oplus x = (14 \oplus 14) + (6 \oplus 14) + (12 \oplus 14) + (15 \oplus 14) = 0 + 8 + 2 + 1 = 11 = s$.

In the second test case, $s = 41, a = [191, 158]$, if $x = 154$, then $\sum_{i=1}^{n} a_i \oplus x = (191 \oplus 154) + (158 \oplus 154) = 37 + 4 = 41 = s$.

,

H. Fortnite

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

This is an interactive problem!

Timofey is writing a competition called Capture the Flag (or CTF for short). He has one task left, which involves hacking a security system. The entire system is based on polynomial hashes* .

Timofey can input a string consisting of lowercase Latin letters into the system, and the system will return its polynomial hash. To hack the system, Timofey needs to find the polynomial hash parameters (p and m) that the system uses.

Timofey doesn't have much time left, so he will only be able to make 3 queries. Help him solve the task.

---

* The polynomial hash of a string s, consisting of lowercase Latin letters of length n, based on p and modulo m is $(\mathrm{ord}(s_1) \cdot p^0 + \mathrm{ord}(s_2) \cdot p^1 + \mathrm{ord}(s_3) \cdot p^2 + \ldots + \mathrm{ord}(s_n) \cdot p^{n-1}) \bmod m$. Where s_i denotes the i-th character of the string s, $\mathrm{ord}(\mathrm{chr})$ denotes the ordinal number of the character chr in the English alphabet, and $x \bmod m$ is the remainder of x when divided by m.

Input

Each test consists of multiple test cases. The first line contains an integer t ($1 \leq t \leq 10^3$) — the number of test cases.

It is guaranteed that the p and m used by the system satisfy the conditions: $26 < p \leq 50$ and $p + 1 < m \leq 2 \cdot 10^9$.

Interaction

To make a query to the system, output `?` s, where s is a string of **no more than** 50 characters in length, the hash of which you want to know. In response to this query, you will receive the polynomial hash of the string s.

To output the answer, output `!` p m, where p is the base of the hash, and m is the modulus. After that, immediately proceed to the next test case.

You have to make not more than 3 queries `?`, otherwise you will get verdict `Wrong Answer`.

After outputting a query, do not forget to output a newline and flush the output buffer. Otherwise, you will receive the verdict `Idleness limit exceeded`. To flush the buffer, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see the documentation for other languages.

Standard Input	Standard Output
1	? aa
32	? yb

28	! 31 59

Note

Answer for the first query is $(ord(a) \cdot 31^0 + ord(a) \cdot 31^1) \mod 59 = (1 + 1 \cdot 31) \mod 59 = 32$.

Answer for the second query is $(ord(y) \cdot 31^0 + ord(b) \cdot 31^1) \mod 59 = (25 + 2 \cdot 31) \mod 59 = 28$.