Hash-shashin

Hashshashin, more commonly known as the *Order of Assassins*, was a prominent religious order best known for their, uh, *covert operations* against their opponents. In order to carry out such *covert operations* without alerting the outside world, the order members, known as *Assassins*, often communicate via rigorously encrypted messages. The sender first converts the message to a string s of length n consisting of only characters `0` and `1`. Then, the sender computes a sequence, aptly called *hash*, defined as follows:

- $hash_0 = 0$
- $hash_i = (2 \cdot hash_{i-1} + s_i) \bmod m, \forall 1 \leq i \leq n$

Here, s_i is the i-th character (numbered from 1) of the binary message and m is a positive integer agreed beforehand between the recipient and the sender. The sender then sends the sequence *hash* to the intended recipient.

It is easy to see that the message can be reconstructed with the knowledge of m, provided $m \geq 3$. Additionally, to ensure that the message has not been compromised, the sender will separately send another string *key* of length n that contains the comparison between each pair of consecutive elements in *hash*. In other words:

- key_i is “<” if $hash_{i-1} < hash_i$,
- key_i is “=” if $hash_{i-1} = hash_i$, and
- key_i is “>” if $hash_{i-1} > hash_i$.

Your task is to help Altaïr, a senior *Assassin*, on his upcoming *covert operation*. Altaïr expects to receive a message from his mentor Mualim outlining his next objectives. However, he only received the key; the message containing the *hash* has been intercepted by hostile spies. As it is obviously unsafe to send another message at the moment, Altaïr wants to recover as many characters of the original message as possible. More specifically, for each position i ($1 \leq i \leq n$), Altaïr wants to know whether s_i is constant for every possible message s that is consistent with the received key. It is possible that the key itself has been tampered and there exists no satisfying string s, in which case Altaïr also wants you to let him know. Please help him!

Input

The first line contains a single integer t ($1 \leq t \leq 10\,000$) — the number of test cases. The description of the test cases follows.

- The first line of each test case contains two integers n, m ($1 \leq n \leq 200\,000$, $3 \leq m \leq 10^9$) — the length of the original message and the predetermined modulo used for encryption.
- The second line contains a string key of length n, where:

- key_i is “<” if $hash_{i-1} < hash_i$,
- key_i is “=” if $hash_{i-1} = hash_i$, and
- key_i is “>” if $hash_{i-1} > hash_i$.

It is guaranteed that the sum of n over all test cases does not exceed $200\,000$.

Output

For each test case, let S be the set of all binary strings s that matches the given key. If S is empty, output "`impossible`" (without the quotes). Otherwise, output a string a of length n, where:

- a_i is "0" if $s_i = 0 \; \forall s \in S$,
- a_i is "1" if $s_i = 1 \; \forall s \in S$,
- a_i is "?" otherwise.

Sample Input 1

```
3
5 6
<<<>>
10 998244353
==<<<<<<<<
5 3
<>=<=
```

Sample Output 1

```
10???
001???????
impossible
```

Sample Explanation

In the first test case, consider string `s = 10101`, the corresponding hash and key are as follows:

i	0	1	2	3	4	5
s_i		1	0	1	0	1
$hash_i$	0	1	2	5	4	3
key_i		$<$	$<$	$<$	$>$	$>$

Thus, the string `10101` matches the given key. The remaining satisfying strings are `10010`, `10011`, and `10100`.

In the second test case, any string of length 10 and starts with `001` matches the given key.

In the third test case, it can be proven that no string of length 5 matches the given key.