[E] Binary Grammar

Batas waktu: 1 detik per *test case*

Batas *Memory*: 32 MB

Deskripsi Masalah

Mr. Arzaki is currently trying to develop a new language called Binary Language. It is a language that contains either **0** or **1**. However, there is another entity such as *****. Given a Binary Grammar **1*01***. Is **0111** considered a valid language?

Based on that question, we know that ***** is an identifier as such that it is either an empty character or an infinitely numbers of characters. Rest assured, we are only doing a test for finitely numbers of characters. For an example, given a Binary Grammar **1***. A valid language for this grammar is `` (empty character, it is actually blank but we use backtick just for display), **1**, **11**, **111**, **111...**, and so on.

Then, does **0111** considered as a valid language for **1*01*** Binary Grammar? The answer is... yes!

Now, given a string **S** as our Binary Grammar, you are trying to check whether it is a valid language or not by using "OK" or "NOT OK" (without double quotes) answers for Q queries of language. This looks like a pattern matching, right? Well, let's see...

Format Masukan dan Keluaran

First line contains S $(1 \leq |S| \leq 30)$ that denotes Binary Grammar. Second line contains Q $(3 \leq Q \leq 20)$, a positive integer that denotes the number of queries. Q lines contains T_i $(1 \leq |T_i| \leq 50)$, the language that we are trying to validate.

For each Q, print whether it is a valid language or not by using "OK" or "NOT OK" (without double quotes).

Contoh Masukan/Keluaran

Masukan	**Keluaran**
`1*0`	`NOT OK`
`16`	`NOT OK`
`0000`	`NOT OK`
`0001`	`NOT OK`
`0010`	`NOT OK`
`0011`	`NOT OK`
`0100`	`NOT OK`
`0101`	`NOT OK`
`0110`	`NOT OK`
`0111`	`NOT OK`
`1000`	`NOT OK`
`1001`	`NOT OK`

1010 1011 1100 1101 1110 1111	NOT OK NOT OK OK NOT OK
1*01* 4 111111110 011111111 111101111 111000111	OK OK OK NOT OK
*** 3 000000000000000 111111111111111 011010101001010	NOT OK NOT OK NOT OK

Penjelasan Contoh Masukan/Keluaran

First Example

The only one language that valid is **1110** since **1*** covers **111** and then we match the last character which is **0**.

Second Example

See the fourth input.
First, **1*** covers **111**.
Second, **0** covers a single **0**.
Third, there is no more characters that would fits since there are **00** leftover although the last **1*** covers **111**.

Third Example

It is also possible to add ***** without any **0** or **1** behind it. There will be no match though.