

Attribute Parser

This challenge works with a custom-designed markup language *HRML*. In *HRML*, each element consists of a starting and ending tag, and there are attributes associated with each tag. Only starting tags can have attributes. We can call an attribute by referencing the tag, followed by a tilde, '~' and the name of the attribute. The tags may also be nested.

The *opening tags* follow the format:

```
<tag-name attribute1-name = "value1" attribute2-name = "value2" ...>
```

The *closing tags* follow the format:

```
</tag-name>
```

The attributes are referenced as:

```
tag1~value  
tag1.tag2~name
```

Given the source code in HRML format consisting of N lines, answer Q queries. For each query, print the value of the attribute specified. Print *"Not Found!"* if the attribute does not exist.

Example

```
HRML listing  
<tag1 value = "value">  
<tag2 name = "name">  
<tag3 another="another" final="final">  
</tag3>  
</tag2>  
</tag1>  
  
Queries  
tag1~value  
tag1.tag2.tag3~name  
tag1.tag2~value
```

Here, tag2 is nested within tag1, so attributes of tag2 are accessed as `tag1.tag2~<attribute>`. Results of the queries are:

Query	Value
tag1~value	"value"
tag1.tag2.tag3~name	"Not Found!"
tag1.tag2.tag3~final	"final"

Input Format

The first line consists of two space separated integers, N and Q . N specifies the number of lines in the HRML source program. Q specifies the number of queries.

The following N lines consist of either an opening tag with zero or more attributes or a closing tag. There is a space after the tag-name, attribute-name, '=' and value. There is no space after the last value. *If there are no attributes there is no space after tag name.*

Q queries follow. Each query consists of string that references an attribute in the source program. More formally, each query is of the form $tag_{i_1}.tag_{i_2}.tag_{i_3} \dots .tag_{i_m} \sim attr - name$ where $m \geq 1$ and $tag_{i_1}, tag_{i_2} \dots .tag_{i_m}$ are valid tags in the input.

Constraints

- $1 \leq N \leq 20$
- $1 \leq Q \leq 20$
- Each line in the source program contains, at most, **200** characters.
- Every reference to the attributes in the Q queries contains at most **200** characters.
- All tag names are unique and the HTML source program is logically correct, i.e. valid nesting.
- A tag can may have no attributes.

Output Format

Print the value of the attribute for each query. Print "Not Found!" without quotes if the attribute does not exist.

Sample Input

```
4 3
<tag1 value = "HelloWorld">
<tag2 name = "Name1">
</tag2>
</tag1>
tag1.tag2~name
tag1~name
tag1~value
```

Sample Output

```
Name1
Not Found!
HelloWorld
```