

4083 HTML Wrapper

In this world of information explosion, many websites use database to store information. WebPages are automatically generated by programs. These programs are usually written in PHP, JSP, ASP, and CGI. Webpage template and layout are also included in the programs. Information data is optionally loaded from database and is filled in the webpage at the time that the webpage is visited. The completed webpage will be displayed in the browser.

Automatically generated WebPages have some major differences with manually created ones.

1. WebPages created by the same program will have the same template.
2. Apart from the data loaded from database, WebPages created by the same program will have the same content.
3. WebPages created by the same program will have the same framework of background data.

It is desirable that the background data could be extracted from WebPages and reused. This is the so called “Information Extraction”.

HTML is composed of tags. HTML tags are always enclosed in angle-brackets (< >) and are case-insensitive.

Tags typically occur in begin-end pairs. These pairs are in the form

`<tag> ...</tag>`

where the `<tag>` indicates the beginning of a tag-pair, and the `</tag>` indicates the end. (The three dots indicate an arbitrary amount of content between the tags.) The usual way to refer to each tag is “tag” for the first and “slash-tag” for the second, where tag is the actual name of the tag being discussed.

One pair of tags define a container. Any content within a container has the rules of that container applied to it. For example, the text within a “boldface container” would be boldfaced. Similarly, paragraphs are defined using a “paragraph container.”

On another hand, thinking of tag-sets as containers will help you remember that tags should always be balanced. That is to say, you should keep containers nested within each other.

Since HTML is based on these structures, so it is often the case that the arrangement of text within a container is irrelevant. For example, within a paragraph container, all of the text can be in one long line, or in a series of separate lines, or with every word on its own line, or every word separated by nineteen spaces. These would all be displayed exactly the same. That is to say, whitespace doesn’t matter. (Whitespace is all of the blank areas in a text file-empty lines, extra spaces, and so on.)

We need to first define some basic functionalities. Given a HTML webpage, such as:

```
<HTML>
<BODY>
Author:
<B>Tom</B>
<UL>
<LI>
<I>Writings:</I>
The C++ Programming Language
</LI>
<LI>
<I>Writings:</I>
```

```

<B>Thinking in Java</B>
</LI>
<LI>
<I>Writings:</I>
Javascript
</LI>
</UL>
</BODY>
</HTML>

```

We want to know the exact position of a specific data such as: “Thinking in Java”. All the HTML tags enclose this data are:

```

<HTML>
<BODY>
<UL>
<LI>
<B>

```

You are chosen to be a member in the “Information Extraction” team. Your task is to output the HTML tags that contains the inquired data (It is WRAPPER), in original order.

Please note:

1. HTML tags are not case-sensitive. Each tag could be followed by other characters. We just ignore these following character. For example `` are just the same as `<A>`.
2. There could be redundant whitespaces (space, TAB, end of line) in HTML.
3. All the content except the tags are case-sensitive and not contained of ‘<’ or ‘>’.
4. All the whitespaces out of HTML tags are regarded as separators. In another word, any consecutive whitespaces are regarded as one space.
5. All given WebPages are syntactically correct. It means that each HTML tag is paired up (Actually there are some tags that are not paired up in actual HTML, but we just ignore that in this problem).
6. All the HTML tags we need to considered are `<HTML>`, `<BODY>`, `<HEAD>`, `<TITLE>`, ``, ``, ``, `<DIV>`, ``, `<I>`, `<P>` and their corresponding `</tag>`. We just consider other tags as whitespaces.

Input

Every input case begins with a HTML webpage (≤ 5000 words). The webpage begins with ‘<<<<’ in one line and ends with ‘>>>>’ in one line. The webpage is followed by an integer k in one line ($1 \leq k \leq 20$), indicating the number of the inquired data that need to be processed. Every line in the following k lines is an inquired data. Each inquired data contains at least one word and at most 20 words. Two adjacent words are separated by spaces.

Output

Output the HTML tags in lower cases, each in a line. If the inquired data cannot be found in the input, output ‘No Wrapper’. If the inquired data appears several times in the input, just output the tags that contain the first one.

Output an empty line after each query.

Sample Input

```
<<<
<HTML>
<BODY>
Author:
<B>Tom</B>
<UL>
<LI>
<I>Writings:</I>
The C++ Programming Language
</LI>
<LI>
<I>Writings:</I>
<B> Thinking in Java </B>
</LI>
<LI>
<I>Writings:</I>
Javascript
</LI>
</UL>
Java
<hello>
script
< / BODY>
< / HTML>
>>>
3
Javascript
java
Java script
```

Sample Output

```
<html>
<body>
<ul>
<li>
```

No Wrapper

```
<html>
<body>
```