

Search in XML

The XML (eXtensible Markup Language) is gaining popularity as a new standard for data representation and exchange on the internet. XML provides a text-based means to describe and apply a tree-based structure to information. The XML document consists of nested elements, some of which usually have attributes and content. But for simplifying this problem, we needn't consider the attributes and content, i.e. only tags allowed. An element typically consists of two tags, a start tag and an end tag. The start tag consists of a name surrounded by angle brackets, like "<tag>"; the end tag consists of the same name surrounded by angle brackets, but with a slash preceding the name, like "</tag>". The element's content is empty or other sub-element (child) that appears between the start tag and the end tag. Specially, no XML element that has the same tag in its direct sub-elements (children), i.e. All sibling elements have different tag names. The following is an valid example for XML documents.

```
<THU>
<Team>
  <ACRush></ACRush>
  <Jelly></Jelly>
  <Cooly></Cooly>
</Team>
<JiaJia>
  <Team>
    <Ahyangyi></Ahyangyi>
    <Dragon></Dragon>
    <Cooly><Amber></Amber></Cooly>
  </Team>
</JiaJia>
</THU>
```

For identifying the elements in a document, we number the elements in according to the order that the start tags of the elements appear in the document. For instances, "THU" is numbered 1. The first "Team" is numbered 2. "ACRush" is numbered 3. "Ahyangyi" is numbered 8.

The problem of querying XML documents has been given much attention by researchers. Now we are given a querying pattern of XML documents and a text of XML documents. The following is an valid example for pattern.

```
<Team><Cooly></Cooly></Team>
```

And we are requested to find all occurrences of the pattern in the text of XML documents. Here, the pattern occurs at a particular text position if placing the pattern with root element at that text position leads to a situation in which each pattern element overlaps some text element with the same label. Because the sibling elements have different labels, there is only one way to put the pattern into the text.

Input

There are two parts in the input file. The first part is a valid XML documents with exactly one root element. The second part is a valid XML documents as querying pattern with exactly one root element. Please ignore all whitespaces (unvisiable characters) in the input file, i.e. only consider the uppercase and lowercase letter and "/", "<", ">". Assume XML documents is always strictly a

rooted tree. The input file is less than 100kb.

Output

Output all the occurrences of pattern in a text of XML documents. The first line consists of an integer n that denotes the number of the occurrences. Then the next n line, each line consists of an id number of an element that occurs the query pattern. Please print them in increasing order.

Example

Input:

```
<THU>
  <Team>
    <ACRush></ACRush>
    <Jelly></Jelly>
    <Cooly></Cooly>
  </Team>
  <JiaJia>
    <Team>
      <Ahyangyi></Ahyangyi>
      <Dragon></Dragon>
      <Cooly><Amber></Amber></Cooly>
    </Team>
  </JiaJia>
</THU>
<Team><Cooly></Cooly></Team>
```

Output:

```
2
2
7
```