XML is becoming more and more the designated format for information exchange. As most of the currently stored information resides in databases, a way to convert it from the relational data model to a XML format is required.

In this problem we present a transformation rule suitable for a restricted subset of relational databases, namely

- a database is defined by a set of table headers like `R(#A, B, C)` where `R` is the name of the table, and `#A`, `B` and `C` the names of the columns;
- a column with a name starting by "`#`" is the key of the corresponding table, which is different in every table line and is thus used to identify the lines;
- a column `B` in a table `R`, other than the key, can either be a simple attribute or a reference to a table `S` if the name of the column `B` is mentioned in the key `#B` of table `S`, meaning that the table `R` depends on table `S`0;
- there is exactly one table that does not depend on any other;
- each table contains at most one reference;
- a table may reference itself;
- the lines in each table are represented in the same form of the table header, but with the specific values for the line instead of the names of the columns.

An example is of a database with three tables, `R`, `S` and `T`, and the corresponding XML translation is as follows:

```
------------------     ------------------------------
|  S(#C,A,D)     |     |  <DB>                      |
|  R(#A,B)       |     |  <R #A="a1" B="b1">        |
|  T(E,A)        |     |  <S #C="c2" D="d3">        |
|  data          |     |  </S>                      |
|  T(e1,a2)      |     |  <S #C="c3" D="d1">        |
|  S(c3,a1,d1)   |     |  </S>                      |
|  S(c1,a2,d2)   |     |  </R>                      |
|  S(c2,a1,d3)   |     |  <R #A="a2" B="b2">        |
|  R(a1,b1)      |     |  <S #C="c1" D="d2">        |
|  R(a2,b2)      |     |  </S>                      |
------------------     |  <T E="e1">                |
                       |  </T>                      |
                       |  </R>                      |
                       |  </DB>                     |
                       ------------------------------
```

The translation to XML is done according to the following rules:

- there is a main element called `DB`;
- an element called X starts with a tag called <X> and ends with a tag called </X>;
- each line in a table `R(#A, B)` is represented by an element called `R` and the columns which are not references to tables are represented by a pair '*column_name*=*value*' inside the opening tag `<R #A="a1" B= "b1"> ...</R>`;
- columns that are references are not explicitly represented but only implicitly by including the element of the corresponding line inside the element of the referred line, that is, between the opening and closing tags of the element as shown in the example.

Given a database dump, produce a XML translation for it according to the rules above. In case there are several direct sub-elements for the same element, order them first by the element name itself then by the first argument, second argument, etc.

In the example above, both `T` and `S` include a reference to `R`, via the column `A`. The translations of the `S` and `T` data-lines are embedded in the translation for the corresponding `R` line, the `S` data-lines before the `T` data-lines.

Input

The input will contain several test cases, each of them as described below. Consecutive test cases are separated by a single blank line.

The input is a sequence of text lines. The first lines contain table headers, in no particular order, followed by a line with the word '`data`' and then the remainin lines, one for each table data-line. The input is guaranteed to obey the rules mentioned above, namely, there is exactly one table header with no references to other tables and the other tables have at most one reference.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

The output is the corresponding XML translation following the rules defined above. It should have just one tag in each line, left justified. Use single spaces to separate attributes inside tags and no extra spaces, for instance around '`=`'.

Sample Input

```
S(#C,A,D)
R(#A,B)
T(E,A)
data
T(e1,a2)
S(c3,a1,d1)
S(c1,a2,d2)
S(c2,a1,d3)
R(a1,b1)
R(a2,b2)
```

Sample Output

```
<DB>
<R #A="a1" B="b1">
<S #C="c2" D="d3">
</S>
<S #C="c3" D="d1">
</S>
</R>
<R #A="a2" B="b2">
<S #C="c1" D="d2">
</S>
<T E="e1">
</T>
</R>
</DB>
```