

792 Program Modules

The relations \rightarrow and \Rightarrow are defined over the set of functions of a program.

- $A \rightarrow B$ means that the function A directly calls the function B ;
- $A \Rightarrow B$ means that the function A calls the function B (directly or indirectly).

In addition, it is known that:

- if $A \rightarrow B$ then $A \Rightarrow B$
- if $A \Rightarrow B$ and $B \Rightarrow C$ then $A \Rightarrow C$

A computer aided software engineering system is able to group the functions of a program into modules according to the following rules:

1. If $A \Rightarrow B$ and $B \Rightarrow A$ then the functions A and B are grouped in the same module.
2. Any two functions which do not satisfy (1) must be in different modules.

Write a program that is able to modularize a program according to the rules (1) and (2) given above. The program reads sets of data from a text file. Each set of data stands for a different program to be modularized and has the format:

$$(Fun_1 Fun_{11} \dots Fun_{1n_1}) (Fun_2 Fun_{21} \dots Fun_{2n_2}) \dots (Fun_m Fun_{m1} \dots Fun_{mn_m}).$$

where Fun_i and Fun_{ij} , $i = 1, m$, $j = 1, n_j$, $m, n_j \geq 0$, are letters (case sensitive) designating names of functions. The meaning of the term $(Fun_i Fun_{i1} \dots Fun_{in_i})$, $n_i \geq 0$, is: $Fun_i \rightarrow Fun_{ij}$, $j = 1, n_j$. The data set is ended by a dot. Spaces, tabs and line breaks are used freely in the input. Input data are correct, i.e. each data set contains exactly one term $(Fun Fun_1 \dots Fun_n)$, for each Fun of the program encoded by the data set, and $Fun_i \neq Fun_j$ for $i \neq j$, $i, j = 1, n$.

The result of the program is on standard output. For each data set the program lists the function names of each computed module, one module per line. The names of the functions of a module are listed in ascending lexicographic order and are separated by single spaces. The modules are listed ascendingly according to the lexicographic order of their first component (function name). The output for the data set is followed by an empty line.

Input and Output

The example below shows an input file, which contains three data sets, and the corresponding output. The first data set stands for an empty set of functions and the corresponding output is an empty line. For the second data set there are two singleton modules 'a' and 'b', whereas for the third data set there are three modules 'a b e', 'c' and 'd'.

Sample Input

```
.
(a a) (a b) (b).
(a b c) (b a e) (c d)
(d d) (e b).
```

Sample Output

a

b

a b e

c

d