When designing tables for a relational database, a functional dependency (FD) is used to express the relationship between the different fields. A functional dependency is concerned with the relationship of values of one set of fields to those of another set of fields.

The notation `X->Y` is used to denote that when supplied values to the field(s) in set `X`, the assigned value for each field in set `Y` can be determined. For example, if a database table is to contain fields for the *social security number* (`S`), *name* (`N`), *address* (`A`), and *phone* (`P`) and each person has been assigned a unique value for `S`, the `S` field functionally determines the `N`, `A` and `P` fields. This is written as `S->NAP`.

Develop a program that will identify each redundant FD in each input group of FDs. An FD is redundant if it can be derived using other FDs in the group.

For example, if the group contains the FDs `A->B`, `B->C`, and `A->C`, then the third FD is redundant since the field set `C` can be derived using the first two. (The `A` fields determine values for the `B` fields, which in turn determine values for the fields in `C`.) In the group `A->B`, `B->C`, `C->A`, `A->C`, `C->B`, and `B->A`, all the FDs are redundant.

Input

The input file contains an arbitrary number of groups of FDs. Each group is preceded by a line containing an integer no larger than 100 specifying the number of FDs in that group. A group with zero FDs indicates the end of the input.

Each FD in the group appears on a separate line containing two non-empty lists of field names separated by the characters `-` and `>`. The lists of field names contain only uppercase alphabetic characters. Functional dependency lines contain no blanks or tabs. There are no trivially redundant FDs (for example, `A->A`).

For identification purposes, groups are numbered sequentially, starting with 1; the FDs are also numbered sequentially, starting with 1 in each group.

Output

For each group, in order, your program must identify the group, each redundant FD in the group, and a sequence of the other FDs in the group which were used to determine the indicated FD is redundant. If more than one sequence of FDs can be used to show another FD is redundant, any such sequence is acceptable, even if it is not the shortest proof sequence. Each FD in an acceptable proof sequence must, however, be necessary.

If a group of FDs contains no redundancy, display `No redundant FDs.`

Sample Input

```
3
A->BD
BD->C
A->C
6
P->RST
VRT->SQP
PS->T
Q->TR
QS->P
SR->V
5
A->B
A->C
B->D
C->D
A->D
3
A->B
B->C
A->D
0
```

Sample Output

```
Set number 1
     FD 3 is redundant using FDs: 1 2

Set number 2
     FD 3 is redundant using FDs: 1
     FD 5 is redundant using FDs: 4 6 2

Set number 3
     FD 5 is redundant using FDs: 1 3
          --OR--
     FD 5 is redundant using FDs: 2 4

Set number 4
     No redundant FDs.
```