# Problem D
## Christmas Gifts
## Input: gift.in

A pleasant headache that parents have every year is preparing Christmas gifts for their children. Because each child envies each other having seen what others received, the parents have to prepare gifts so that no child should envy any other child.

Before a husband and wife go for Christmas shopping, they announce to their children a fixed set of gift candidates and declare that each child will receive a subset of the candidates. The children then respond to the parents with conditions for them to be happy. Each child's condition is relatively defined in terms of the gifts that other siblings would have. The goal of the parents is to satisfy all their children's conditions with the smallest such gift set for each child.

Each child's condition is always expressed as a conjunction:

$$\text{I need at least } (\alpha_1 \text{ and } \alpha_2 \text{ and } \cdots \text{ and } \alpha_k)$$

where each $\alpha_i$ is either
  [type 1] a constant subset of the gift candidates, or
  [type 2] one sibling's name (meaning his/her gifts), or
  [type 3] "common-things-of" two $\alpha_i$'s of type 1 or 2 (meaning the gifts that are common among the two), or
  [type 4] one sibling's name followed by "except-for" a constant subset of the gift candidates (meaning the sibling's
       gifts excluding those in the constant subset).

For example, for three children $\{X, Y, Z\}$ and for three candidates $\{a, b, c\}$ for gifts, let the condition for each child be:

$$X \quad \text{needs at least} \quad (\{a, b\} \text{ and common-things-of } (Y, Z))$$
$$Y \quad \text{needs at least} \quad (\text{common-things-of } (Z, \{b, c\}))$$
$$Z \quad \text{needs at least} \quad (\{a\} \text{ and } (X \text{ except-for } \{c\}))$$

Then the smallest gifts for each child that satisfy his/her condition are $\{a, b\}$ for X, $\{b\}$ for Y, and $\{a, b\}$ for Z.

## Input
The input consists of $T$ test cases. The number of test cases ($T$) is given in the first line of the input file.
Gift candidates are numbered from 1 to $n$ ($1 \le n \le 1000$). Children are numbered from 1 to $m$ ($1 \le m \le 100$).

The input is a sequence of non-empty lines:

The first line has the number ($T$) of test cases. The sequence of test cases is from the second line.
The first line of each test case has two integers, the number ($n$) of gift candidates followed by the number ($m$) of children.

The next lines show the sequence of children's conditions in the order of child 1's, child 2's, ⋯, child $m$'s. Each child's condition is expressed as follows:

— The first line has two integers, the child's identification and the number of $\alpha_i$'s.
— From the second line, one line represents one part (the $\alpha_i$) of the child's condition. Each line for $\alpha_i$ starts with its type: "-1" for type 1, "-2" for type 2, "-3" for type 3, and "-4" for type 4.

[type 1] "-1" is followed by a number $k$ and a sequence of integers for the gift elements. The $k$ is the number of integers in the sequence. It represents a constant set of gift candidates. For example,
```
-1 2 1 2
```
represents a set {1, 2} of gifts.

[type 2] "-2" is followed by a single integer for a sibling's identification. For example,
```
-2 2
```
represents sibling 2's gifts.

[type 3] "-3" is followed by a sequence of two formats of type 1 or 2. For example,
```
-3 -2 3 -1 2 2 3
```
represents the common-things-of sibling 3's gifts and set {2, 3}. As another example,
```
-3 -2 2 -2 3
```
represents the common-things-of sibling 2's and 3's gifts.

[type 4] "-4" is followed by type 2 followed by type 1. For example,
```
-4 -2 1 -1 1 3
```
represents sibling 1's gifts except-for {3}.

## Output

The output is the sequence of solutions for the given test cases. The solutions are printed in the order of the test cases. Each solution is the sequence of lines, one line per child, in the increasing order of children's identification numbers. Each line starts with the child's identification number followed by his/her gift identification numbers in the increasing order. For an empty gift, only the child number is printed.

| Sample Input (gift.in) | Output for the Sample Input |
|---|---|
| 3<br>2 2<br>1 1<br>-1 1 1<br>2 1<br>-4 -2 1 -1 1 1<br>1 1<br>1 1<br>-3 -1 1 1 -1 1 1<br>3 3<br>1 2<br>-1 2 1 2<br>-3 -2 2 -2 3<br>2 1<br>-3 -2 3 -1 2 2 3<br>3 2<br>-1 1 1<br>-4 -2 1 -1 1 3 | 1 1<br>2<br>1 1<br>1 1 2<br>2 2<br>3 1 2 |