Problem Description

You are given a CNF-formula as input. Output a circuit (with AND, OR, NOT gates) so that the truth table of the circuit and the formula are the same.

As you may know, there may be more than one circuit that represents a CNF-formula. You may output any of them. Don't worry, we have a special checker to check whether your output circuit has the same truth table as that of the input CNF-formula.

Input Format

The first line will consist of an integer T, the number of test cases.
T cases follow. For each case:

The first line will consist of two integers C and N, separated by a space, which represent the number of clauses and variables respectively.

C lines follow. Each line describes each clause. Each clause has the following format:

The first integer K describes the number of literals in that clause. The next K integers describe the literals. The number i represents literal x_i if i>0, or literal $-x_{-i}$ if i<0.

For example, $(x_1 \vee x_3 \vee -x_4)$ will be written as `3 1 3 -4`.

Output Format

Output that doesn't follow this format may cause undefined behavior in our checker. Please follow this format!!!

For each case, you have to output in the following format:
The first line will consist of a single integer M, the number of logic gates you use. The gates are numbered from N+1...N+M. The output of the gate numbered N+M is the final output of the circuit.

M lines follow. Each line describes each logic gate. Each logic gate has the following format:

The first token will be either "AND", "OR", or "NOT", depending what is your the gate.

If the first token is "NOT", it will be followed by a single integer representing the input index.

Otherwise, it will be followed by two integers separated by a space representing the input index.

Input index is an integer i which represents the value of x_i (if $1 \le I \le N$), or the output of gate i (if $N + 1 \le i \le N + M$).

Input index of gate i must be between 1 and N+i-1 inclusive.

If the format above is too complicated to understand, see the I/O sample.

Input Sample

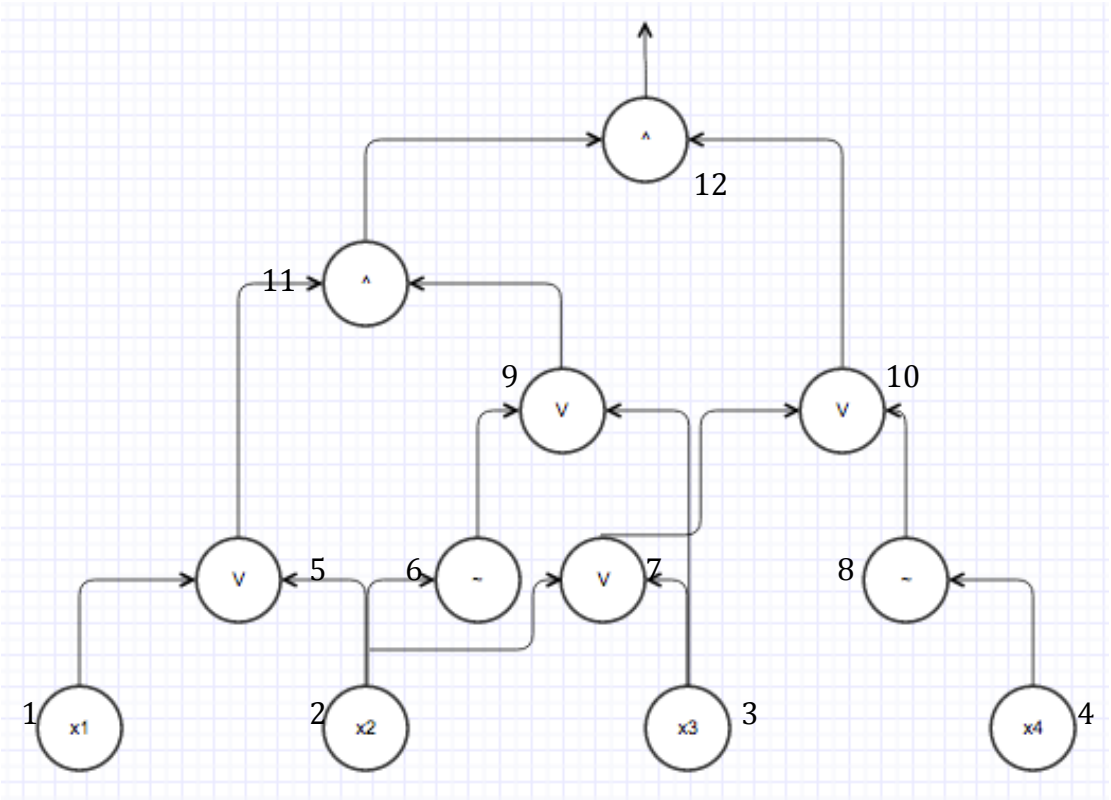
```
1
3 4
2 1 2
2 -2 3
3 2 3 -4
```

Output Sample

```
8
OR 1 2
NOT 2
OR 2 3
NOT 4
OR 6 3
OR 7 8
AND 5 9
AND 11 10
```

Explanation

The formula in the input sample can be written as.
$(x_1 \vee x_2) \wedge (-x_2 \vee x_3) \wedge (x_2 \vee x_3 \vee -x_4)$.
The formula can be represented by the following circuit. Note that the circuit below is not the only correct circuit. If your output doesn't match with the sample below, your output may also be correct.



Constraint

Time Limit: 2s
$1 \le T \le 10$
$1 \le N \le 15$
$1 \le C \le 100$
Each clause consists of at least 2 literals and at most 4 literals.
Your output should not consists of more than 1000 gates.

Score – (Max Score: 37)

There will be two test files for this problem.

Test file 1 (17 points)
There will be additional constraint for this test file: Each clause consist of **exactly** 2 literals.

Test file 2 (20 points)
No additional constraint.

Note

Java version used is "gcj-java-3.2.2".
C++ version used is "g++ 4.4.7".