Problem Description

You are given a circuit C, you must convert it to a CNF formula F such that C is satisfiable if and only if F is satisfiable.

(You must follow the conversion method given in class. Please refer to Tutorial 7, question D1(b)).

There will be N variables and M gates. The variables will be numbered x_1, x_2, ..., x_N. The gate will be numbered g_1, g_2, ..., g_M. For every gate g_i, you must create a temporary variable x_{i+N}.

You may output the clause in any order. For each clause, you can output the literals in any order.

Input Format (Same as input format for C)

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 M and N separated by a space, the number of logic gates and the number of variables respectively. The gates are numbered from N+1...N+M. The output of the gate #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 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 represents the value of x_i (if $1 \leq I \leq N$), or the output of gate i (if $N + 1 \leq i \leq N + M$)

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

Output Format (Same as input format for A)

For each case, you have to output in the following format :

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 v x_3 v $-x_4$) will be written as `3 1 3 -4.`

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

Input Sample

```
1
2 3
OR 1 2
AND 4 3
```

Output Sample

```
7
2 4 -1
2 4 -2
3 -4 1 2
2 -5 4
2 -5 3
3 5 -4 -3
1 5
```

Constraint

Time Limit : 2s
1 ≤ T ≤ 15
1 ≤ N ≤ 15
1 ≤ M ≤ 100

Score – (67 points)

There is only one test file for this problem.

Note

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