Problem F. Majority

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 512 mebibytes

Little Cat learned Boolean circuits recently. Now he wants to construct a majority circuit.

A *circuit* over Boolean variables $x_1, \dots, x_n$ is a directed acyclic graph where each node (logical gate) is either an input node labeled by a variable x_i, or an operation node labeled by a logical operation $\vee$ or $\wedge$. There are exactly n input nodes, one for each of the input variables $x_1, \dots, x_n$. Additionally, a single node is **chosen** as the output of the circuit.

Each node computes an output: the input nodes (labeled by a variable) output exactly the variable written on them, and nodes labeled by $\vee$ (respectively, $\wedge$) output the logical OR (respectively, AND) of all incoming nodes. Note that logical NOT nodes are **forbidden**. See the example and notes for better understanding.

The in-degree of an input node is 0. The in-degree of an operation node is at least 1, and can be arbitrarily large. The out-degrees are arbitrary (possibly 0).

For convenience, there are two special constant nodes T (true) and F (false), which always output 1 and 0, respectively.

The majority circuit Maj_n has n inputs $x_1, \dots, x_n$, and it outputs 1 if at least half of inputs are 1, and outputs 0 otherwise. Formally, $Maj_n(x_1, \dots, x_n) = [2\sum_{i=1}^{n} x_i \geq n]$.

Define the *depth* of a circuit as the length of the longest (directed) path in the circuit, that is, the number of edges of the longest path.

Could you help Little Cat to construct a majority circuit over n inputs with depth at most 14?

Input

The input contains one line with an integer n ($2 \leq n \leq 64$) indicating the number of input nodes.

Output

The first line must contain an integer m ($1 \leq m \leq 5 \cdot 10^4$) representing the number of nodes labeled by $\vee$ or $\wedge$, so there are $n + m + 2$ nodes in the circuit in total. The input nodes $x_1, \dots, x_n$ are numbered by $1, \dots, n$. The constant true node T is numbered by -1, and the constant false node F is numbered by -2.

A total of m lines must follow. The i-th line must describe node $(n + i)$ in one of the following formats.

- "`OR` k_i a_1 a_2 $\dots$ a_{k_i}" (without quotes): node $(n + i)$ computes the logical OR of nodes a_j where $-2 \leq a_j < n + i$ and $a_j \neq 0$ for all $1 \leq j \leq k_i$.
- "`AND` k_i a_1 a_2 $\dots$ a_{k_i}" (without quotes): node $(n+i)$ computes the logical AND of nodes a_j where $-2 \leq a_j < n + i$ and $a_j \neq 0$ for all $1 \leq j \leq k_i$.

It is fine if $a_u = a_v$ for some $u \neq v$. You must guarantee that $\sum_{i=1}^{m} k_i \leq 2 \cdot 10^5$ and that the depth of the circuit does not exceed 14.

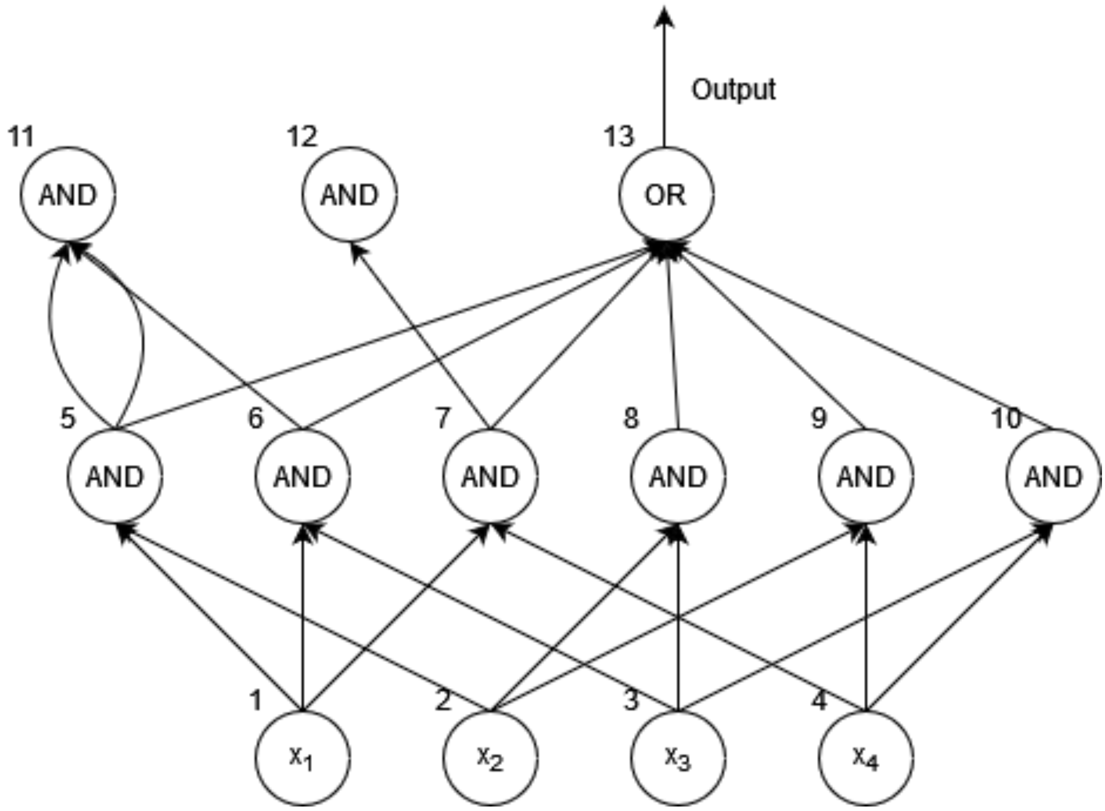
The output of the circuit **is chosen as** the output of node $n + m$.

To check the circuit you construct, Little Cat will test your circuit for 1500 rounds. In each round, Little Cat will generate an arbitrary input $x_1, \dots, x_n$ (he won't say how exactly) and test your circuit with that input. You pass this round if your circuit outputs the majority of the input $x_1, \dots, x_n$ correctly. You need to pass all the 1500 rounds.

Example

standard input	*standard output*
4	9 AND 2 1 2 AND 2 1 3 AND 2 1 4 AND 2 2 3 AND 2 2 4 AND 2 3 4 AND 3 5 5 6 AND 1 7 OR 6 5 6 7 8 9 10

Note



The sample output prints a depth-2 circuit computing Maj_4. The circuit $Maj_4(x_1, x_2, x_3, x_4)$ outputs 1 if and only if at least two input nodes are 1. Thus you can compute the logical AND of every pair of input nodes and output the logical OR of these ANDs.

Here are some notes on the circuit nodes:

- Nodes 1, 2, 3, and 4 are input nodes.
- Nodes 5, 6, 7, 8, 9, and 10 compute the logical AND of some input nodes.
- Nodes 11 and 12 are redundant.
- Node 13 is the output node.
- The constant nodes T and F are not drawn in the figure.

Here, the existence of redundant nodes will not affect the validity of the circuit as long as the constraints ($1 \leq m \leq 5 \cdot 10^4$, $\sum k_i \leq 2 \cdot 10^5$, $depth \leq 14$) are satisfied.

During the test, the following shows a possible scenario:

The input nodes are set to $x_1 = 1$, $x_2 = 0$, $x_3 = 0$, $x_4 = 1$.

Therefore, the outputs of nodes $x_5, \ldots, x_{13}$ are:

- $x_5 = x_1$ AND $x_2 = 1$ AND $0 = 0$
- $x_6 = x_1$ AND $x_3 = 1$ AND $0 = 0$
- $x_7 = x_1$ AND $x_4 = 1$ AND $1 = 1$
- $x_8 = x_2$ AND $x_3 = 0$ AND $0 = 0$
- $x_9 = x_2$ AND $x_4 = 0$ AND $1 = 0$
- $x_{10} = x_3$ AND $x_4 = 0$ AND $1 = 0$
- $x_{11} = x_5$ AND x_5 AND $x_6 = 0$ AND 0 AND $0 = 0$
- $x_{12} = x_7 = 1$
- $x_{13} = x_5$ OR x_6 OR x_7 OR x_8 OR x_9 OR $x_{10} = 0$ OR 0 OR 1 OR 0 OR 0 OR $0 = 1$.

The output of the circuit is $x_{13} = 1$, which is the majority of $\{1, 0, 0, 1\}$.