Unsatisfying

Usually a computer scientist tries to satisfy some constraints. This time you will try to make some logical statements unsatisfiable.

You will be given a list of logical statements of the following form:

$$p_1 \mid p_2$$
$$\sim p_2 \mid p_3$$
$$p_3 \mid \sim p_4$$

Here '**|**', the disjunctive statement, stands for logical **OR** (the result is **TRUE** if either proposition is **TRUE**, possibly both). '**~**' is negation, forcing the value to be the opposite truth value.

To satisfy a list of logical statements, you must assign truth values (**TRUE** or **FALSE**) to each variable such that all the given statements result as **TRUE**. Your task is to add disjunctive statements to the list to make the list of statements unsatisfiable. But you cannot use the negation symbol!

All disjunctive statements (both those given and ones you add) must have exactly 2 terms. The ones given can use negation, but the ones added cannot.

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. Each test case will begin with a line containing two integers ***n*** and ***m*** ($1 \le n,m \le 2{,}000$), where ***n*** is the number of variables and ***m*** is the number of disjunctions. The variables will be numbered 1..***n***.

Each of the next ***m*** lines will contain two integers ***a*** and ***b*** ($1 \le |a|,|b| \le n$), representing the subscript in the variable. A negative value is the negated version of that variable.

Output

Output a single integer, which is the minimum number of disjunctive clauses to add to make the list unsatisfiable. If it is not possible, output -**1**.

Sample Input	Sample Output
2 1 1 2	-1
4 5 1 2 -1 -3 -2 3 3 -4 -2 -3	1