Yet Satisfiability Again!

CPU TIME LIMIT

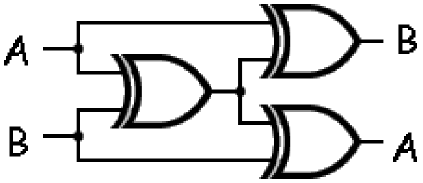
3 seconds

MEMORY LIMIT

1024 MB

DIFFICULTY

medium (3.2)



Picture from Wikimedia Commons

Alice recently started to work for a hardware design company and as a part of her job, she needs to identify defects in fabricated integrated circuits. An approach for identifying these defects boils down to solving a satisfiability instance. She needs your help to write a program to do this task.

Input

The first line of input contains a single integer, not more than 5, indicating the number of test cases to follow. The first line of each test case contains two integers n and m where $1 \leq n \leq 20$ indicates the number of variables and $1 \leq m \leq 100$ indicates the number of clauses. Then, m lines follow corresponding to each clause. Each clause is a disjunction of literals in the form Xi or ~Xi for some $1 \leq i \leq n$, where ~Xi indicates the negation of the literal Xi. The "or" operator is denoted by a 'v' character and is seperated from literals with a single space.

Output

For each test case, display `satisfiable` on a single line if there is a satisfiable assignment; otherwise display `unsatisfiable`.

Sample Input 1

```
2
3 3
X1 v X2
~X1
~X2 v X3
3 5
X1 v X2 v X3
X1 v ~X2
X2 v ~X3
X3 v ~X1
~X1 v ~X2 v ~X3
```

Sample Output 1

```
satisfiable
unsatisfiable
```