

Lab Assignment 8: CS2233

29th October, 2025

A **boolean** formula is called **CNF** (conjunctive normal form) if it is represented via **conjunction** (**boolean-AND**) of several clauses such that each clause is a **disjunction** (**boolean-OR**) of literals (variables or negation of variables). Following is an example of the CNF formula

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3) \wedge (x_4 \vee x_5 \vee \neg x_2 \vee x_1).$$

Further, a **boolean** formula is called **2-CNF** if each clause consists of exactly two literals.

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_3) \wedge (\neg x_1 \vee x_2).$$

A **boolean** formula is called **satisfiable** if there exists an assignment of the boolean variables that makes the formula evaluate to **True**.

Problem statement: Suppose a 2-CNF formula over n variables and m clauses is given as input, write a program that determines whether the formula is **satisfiable** or not. If the formula is **satisfiable**, then output the assignments of the variables that satisfy the formula. The running time of the algorithm should be polynomial in n, m .

Hint: Search 2-CNF is in P.

Input format

- The first line of input consists of number of testcases.
- For each testcase, first line contains n and m , which indicates the number of variables and clauses respectively.
- next m line contain m clauses as shown in sample test cases.

Output format

- Output contains one line corresponding to each testcase.
- For each testcase it contains the variable values for which given boolean formula evaluated as True.
- Otherwise print "Unsatifiable".

Example:

Input:

```
3
3 3
1 -2  # x1  ~x2
-1 -3  # ~x1 V ~x3
-1 2   # ~x1 V  x2
3 4
1 2   # x1  V  x2
-1 -2  # ~x1 V ~x2
1 -3  # x1  V ~x3
-1 3   # ~x1 V x3
2 2
1 2   # x1 V  x2
-1 -2  # ~x1 V ~x2
```

Output:

```
0 0 1      # x1 = False, x2 = False, x3 = True
Unsatifiable
0 1      # x1 = True, x2 = False
```