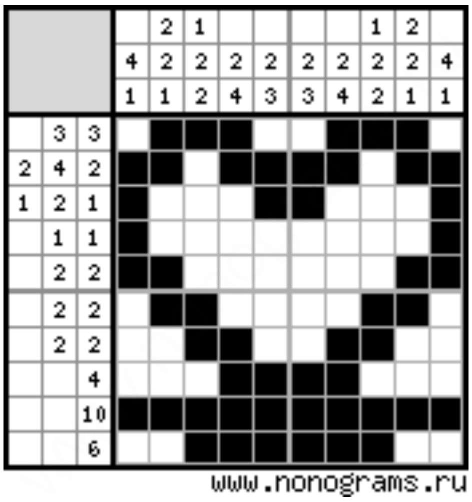
Nonogram

The Academy of Young Europeans (AYE) is organizing a Nonogram competition. Nonogram is a simple puzzle in which you are given a rectangular grid with **R** rows and **C** columns and a set of rules used to fill the grid. The cells inside the grid must be colored using either black or white according to the rules given. A sample of a (solved) nonogram puzzle can be seen below:



To solve the puzzle, you need to color a cell in either black or white, according to the rules given. Each rule corresponds to a single row or column, and the colored cells must match this rule. Each rule consists of a set of numbers. Each number represents the number of consecutive black cells, separated by a number of white cell(s) in between.

For example, the rule "3 3" (refer to the first row of the sample above) means that there are two sets of 3 consecutive black squares in that specific row. There can be preceding or succeeding white cells before the first and last set of blacks, and there must be at least one white cell in between the two set of blacks.

To help organize the competition, the AYE has employed you. They assign you one job: you need to write a program to quickly verify whether the given solutions are valid. This program must be able to tell whether a given nonogram instance is valid, given a set of rules and a colored grid. An answer is valid if it satisfies all the rules given in the nonogram puzzle.

Input

The first line of input consists of two space-separated integers, **R** (4 <= **R** <= 500) and **C** (4 <= **C** <= 500), the number of rows and columns in the puzzle respectively. The next line will be **blank.**

The next **R** lines will be the rules for each row of the puzzle. The first rule corresponds to the first row, and so on. The next line will be **blank**.

The next **C** lines will be the rules for each column of the puzzle. The first rule corresponds to the first column, and so on. The next line will be **blank**.

The next **R** lines contain **C** space-separated integers, denoting the color of the cell at that position respectively. The number '0' represents a white cell while the number '1' represents a black cell. This filled-grid represents the solution that needs to be checked using the rules given above.

Output

If the given filled grid is a valid solution based on the set of rules given, print "VALID" (without quotes). Otherwise, print "INVALID" (without quotes). Your output <u>must contain a newline character</u>.

Sample Input

```
4 4

0
1
1
4

1
2
1 1
1

0 0 0 0
0 0 1 0
0 1 0 0
1 1 1 1
```

Sample Output

```
VALID
```

Explanation

The sample input above shows a valid solution based on the given rule. For example, in the first row, the rule "0" means that there are no black cells in that particular row. In the third column of the grid, the rule "1 1" is applied, hence there are 2 black cells, separated by a number of white cells in between (which is one white cell in this case).

Skeleton

You are given the skeleton file NonogramChecker.java. You <u>should see a non-empty file</u> when opening it, otherwise you are in the wrong directory.

Notes:

1. You should develop your program in the subdirectory `ex1` and use the skeleton java file provided. You should not create a new file or rename the file provided.
2. If your algorithm is different from the given skeleton, you are free to write a solution according to your own algorithm.
3. You do not need to use OOP for this sit-in lab.
4. You are free to define your own helper methods. **Remember to use <u>private methods</u> whenever possible.**
5. Please be reminded that the marking scheme is:

   Input : 10%
   Output : 10%
   Correctness : 50%
   Programming Style : 30%, which consists of:
   - Meaningful comments (pre- and post- conditions, comments inside the code): 10%
   - Modularity (incremental programming, proper modifiers [public / private]): 10%
   - Proper Indentation: 5%
   - Meaningful Identifiers (for both method and variable names): 5%

   Compilation Error **:** Deduction of **50% of the total marks obtained**.