ACM-ICPC Thailand Southern Programming Contest 2013

Hosted by
Department of Computer Engineering
Prince of Songkla University Hatyai Campus

10 August 2013

Contest Problems

- There are **8** problems (A-H) to solve within 3 hours 30 minutes.
- Solve as many problems as you can, in an order of your choice.
- Use C or C++ or Java to program at your convenience for any problems.
- Input and output of each program are **standard input** and **outpu**t.

Problem A	Unlock My Safe
Problem B	Two Mysterious Alphabets from a Tree
Problem C	Max Volume
Problem D	Birthday Statistics
Problem E	Nonogram
Problem F	Jane's First Words
Problem G	Range Sum Query
Problem H	Sum of Distinct Numbers ผลรวมเลขไม่ซ้ำ

Problem E. Nonogram

Time Limit: 3s

Nonogram is a logic puzzle with simple rules and challenging solutions. The rules are simple. You have a grid of squares, which must be either filled in black (we denote this with 1) or marked with X/blank (we denote this with 0). Beside each row of the grid are listed the lengths of the runs of black squares on that row. Above each column are listed the lengths of the runs of black squares in that column. Your aim is to find all black squares.

Input

The first line of input contains a positive integer **TC**, the number of nonogram puzzles that you have to solve. Each test case starts with a blank line, followed by two positive integers **W** and **H** in the next line that denote the width and the height of the puzzle. We guarantee that **1 ≤ W*H ≤ 20**.

Then there will be two blocks of puzzle description that started with a blank line: one for rows (that contains **H** lines) and another for columns (that contains **W** lines). In those two blocks, you will be given comma separated values that denote the lengths of the runs of black squares in each row/columns (note: it can be blank).

Output

For each test case (puzzle), determine if the answer is unique. If it is unique, output a 2D binary matrix of **H** lin es containing **W** characters. Put 1 if the corresponding cell is black, or 0 otherwise. If it is not unique, just print "*not unique*" in one line (without "). Separate the output of two test cases with a blank line.

Sample Input

```
3

5 4

3
1,2
1,1
2

3
1
1
1,1
3

7 2

1,1,1,1
1,1,1

1
1
1
1
1
1
1
```

```
7 1

1,3,1

1

1
1
1

1
```

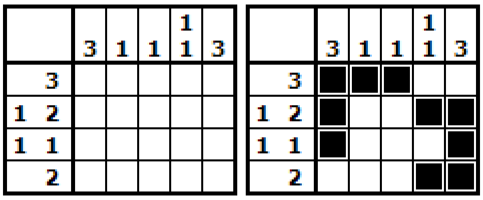
Sample Output

```
11100
10011
10001
00011

1010101
0101010

1011101
```

Visualization of the first Sample Input & Sample Output



Problem Author

Dr Steven Halim, School of Computing, National University of Singapore