

Advanced Problem
Incomplete Chess Boards

14 March 2009

Source File	<code>dominoes.{java,c,cc}</code>
Input File	<code>dominoes.in</code>
Output File	<i>standard output</i>

Tom gets a riddle from his teacher showing a number of chess boards, each with two squares removed. The teacher wants to know which boards can be completely covered by 31 dominoes. He promises ten bars of chocolate for the person who solves the problem correctly. Tom likes chocolate, but he cannot solve this problem on his own. So he asks his older brother John for help. John (who likes chocolate as well) agrees, provided that he will get half the prize.

John's abilities lie more in programming than in thinking and so decides to write a program. Can you help John? Unfortunately you will not win any bars of chocolate, but it might help you win this programming contest.

You are given are 31 dominoes and a standard 8x8 chess board, two distinct squares of which are removed. The square in row a and column b is denoted by (a, b) with $a, b \in \{1, \dots, 8\}$. A 2-by-1 domino can be placed horizontally or vertically onto the chess board, so it can cover two adjacent squares

$$(a, b) \text{ and } (a, b + 1)$$

or

$$(b, a) \text{ and } (b + 1, a)$$

with $a \in \{1, \dots, 8\}$ and $b \in \{1, \dots, 7\}$. The object is to determine if the so-modified chess board can be completely covered by 31 dominoes.

For example, it is possible to cover the board with 31 dominoes if the squares $(8, 4)$ and $(2, 5)$ are removed, as seen in Figure 1 below:

Input

The first input line contains the number of scenarios, N . Each of the following N lines contains four integers a, b, c , and d , separated by single blanks. These integers are in the range $\{1, \dots, 8\}$ and indicate that squares (a, b) and (c, d) are removed from the chess board. You may assume that $(a, b) \neq (c, d)$.

Output

For each scenario, output one line of the form:

Scenario n : R .

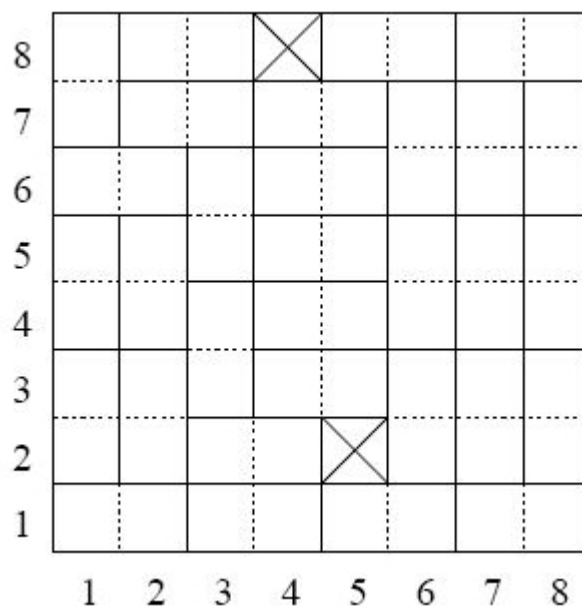


Figure 1: Sample covering of an incomplete chessboard.

where

- n is the scenario number, from 1 to N ,
- R is
`coverable.` – if 31 domines will cover the board, or
`not coverable.` – otherwise.

Output is emitted to standard output, with no leading or trailing spaces.

C, C++	<code>stdout</code>
C++	<code>cout</code>
Java	<code>System.out</code>

Example

Sample input and output are given in figures 2 and 3, respectively.

```
3
8_4_2_5
8_8_1_1
4_4_7_1
```

Figure 2: Input

```
Scenario_#1:_coverable.
Scenario_#2:_not_coverable.
Scenario_#3:_not_coverable.
```

Figure 3: Output