

HackerChess is a variant of chess played at HackerRank. It is a game played between two players who make moves in turns until one of them cannot make any move. The player who cannot make a move loses the game and the other player is declared the winner. The game is played on a board with n rows and n columns. The only pieces used in the game are rooks. A rook in HackerChess moves only vertically, which means that it never leaves a column to which it belongs. Moreover, in a single move, a rook moves through any number of unoccupied cells. Notice that there are no captures in HackerChess, two rooks cannot occupy the same cell, and a rook cannot jump over another rook. Each player has exactly one rook in each of the n columns of the board.

Given the initial position of the rooks and knowing that the second player makes the first move, decide who will win the game if both players play optimally.

Input Format

In the first line, there is a single integer t denoting the number of games to be played. After that, descriptions of t games follow:

In the first line, there is a single integer n denoting the size of the board. Next, n lines follow. In the i -th of them there is a single integer $r_{1,i}$ denoting the row of the rook belonging to the first player placed in the i -th column. After that, another n lines follow. In the i -th of them there is a single integer $r_{2,i}$ denoting the row of the rook belonging to the second player placed in the i -th column.

Constraints

- $1 \leq t \leq 10$
- $2 \leq n \leq 2000$
- $1 \leq r_{1,i}, r_{2,i} \leq n$
- $r_{1,i} \neq r_{2,i}$

Output Format

Print exactly t lines. In the i -th of them, print `player-1` if the first player will win the i -th game. Otherwise, print `player-2` in this line.

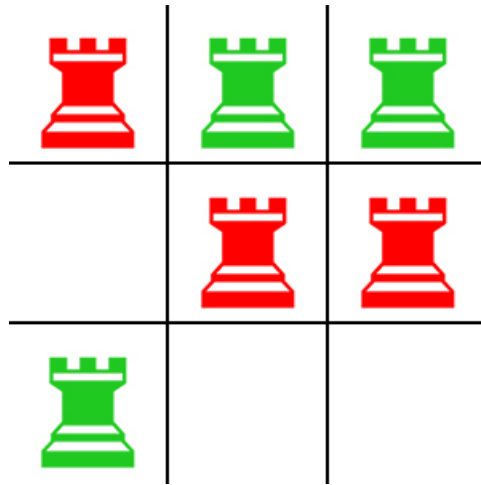
Sample Input 0

```
1
3
1
2
2
3
1
1
```

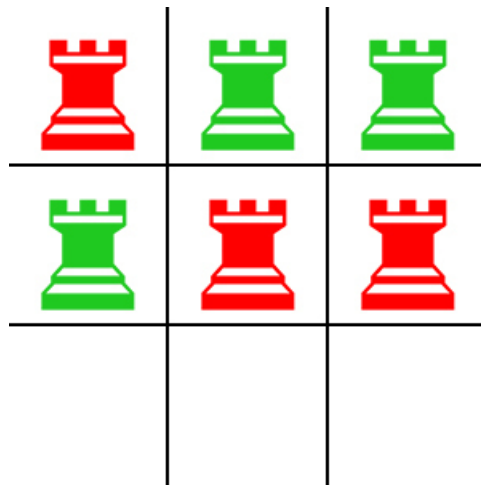
Sample Output 0

Explanation 0

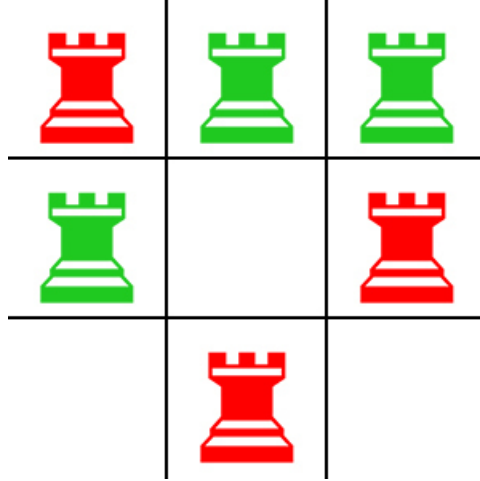
There is only one game player in the sample input. The game is played on the board with **3** rows and **3** columns. Let's denote the first player's rooks as red rooks and the second player's rooks as green ones. Then the initial position of the game looks like this:



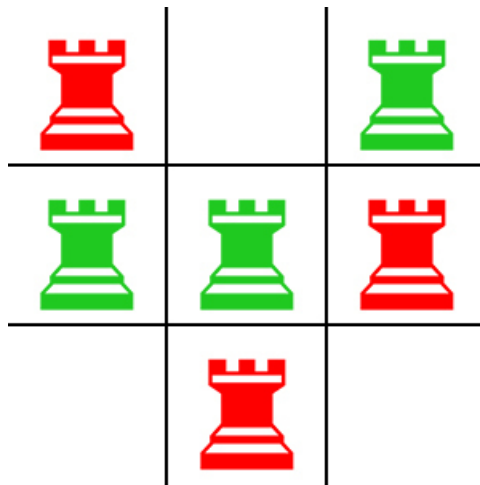
The second player moves first and he can move his rook in the first column to the second row. After this move, the position looks as follows:



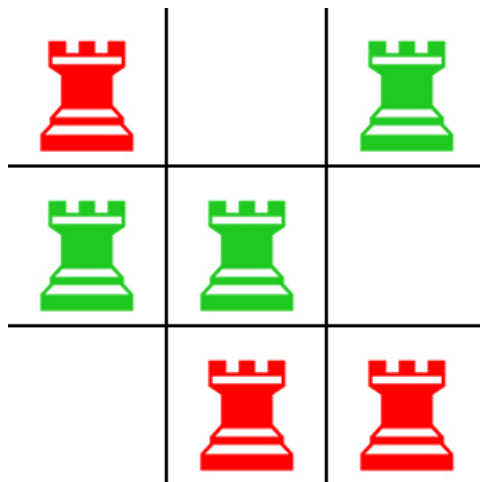
Next, it is the first player's turn. He cannot make any move with his rook in the first column, so he has to make a move in the second or the third column. Without the loss of generality, let's assume that he makes a move in the second column. He can only make one such move, i.e. move the rook from the second to the third row. This results in the following position:



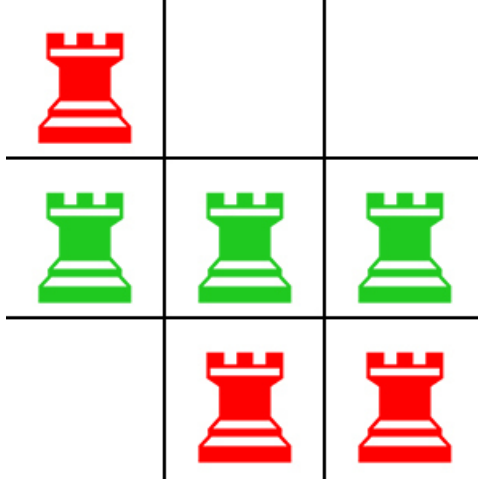
After that, the best move for the second player is to move his rook in the second column from the first to the second row. After this move, the position looks like this:



Next, it is again the first player's move. The only move he can make is to move his rook in the third column from the second to the third row. It results in the following position:



Then, the best move for the second player is to move his rook in the third column from the first to the second row. After that, the position looks as follows:



Next, it is the first player's move, but since he is unable to make any valid move, he loses and the second player is declared a winner.

It shows that regardless of the first player's moves, the second player has a strategy leading to his victory.

Sample Input 1

```
1
4
3
3
3
3
4
4
4
4
```

Sample Output 1

```
player-1
```

Explanation 1

Second player cannot make a move so *player* — 1 is the winner.