Vertical Rooks

In *HackerChess*, there is a special piece, known as the *VROOK*. A *VROOK* is like a normal Rook, except that it can move only along the column. A *VROOK* can't capture/kill another *VROOK*, and it also can't jump over another *VROOK*.

HackerChess is played as follows. The board is an *NxN* grid. It is played between two players. Each column of the board contains exactly *2 VROOKs*, one of *player-1* and the other of *player-2*. Each player has *N VROOKS*, arranged on the board. In a turn, a player can move any of his *VROOKs*, (i.e) if there are any possible *VROOKs* belonging to the player that has a position to which it can be moved. The player to make the last move wins.

Given an initial configuration of the board. If both *player-1* and *player-2* play optimally. Who will win the game, if *player-2* gets to move first?

Input
The first line of input contains an integer *T* denoting the number of test cases. Each case is as follows:

The first line contains an integer *N* the number of rows and columns in the board.

This is followed by *2N* lines containing a single integer each. Of the first *N* lines, the integer in the i^{th} line denotes the row index of the *VROOK* of *player-1* which is present in the i^{th} column. Similarly the next *N* lines denote the *VROOK* configuration for *player-2*. The row index is a 1-based index

Output
For each test case print a single line containing *"player-1"* or *"player-2"* depending on who wins the game (quotes for clarity).

Constraints
1 <= T <= 10
2 <= N <= 2000
1 <= X <= N

Sample Input

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

Sample Output

```
player-2
```

Explanation

If *player-2* moves the *VROOK* in column 1, from row 3 to 2, in all of the future moves *player-1* will have to move his *VROOKs* downwards and *player-2* will do the exact same move as *player-1*, (placing the *VROOK* adjacent to *player-1*), thus making **player-2** the winner.