Challenge 15 - Take a corner

Othello is a two-player strategy game that is played on an 8x8 board. There are sixty-four identical game pieces called discs, which are white on one side and black on the other.

The following diagram shows the standard notation for Othello. Columns are labeled 'a' through 'h' from left to right, and rows are labeled '1' through '8' from top to bottom.

	a	b	c	d	e	f	g	h
1	a1	b1	c1	d1	e1	f1	g1	h1
2	a2	b2	c2	d2	e2	f2	g2	h2
3	a3	b3	c3	d3	e3	f3	g3	h3
4	a4	b4	c4	d4	e4	f4	g4	h4
5	a5	b5	c5	d5	e5	f5	g5	h5
6	a6	b6	c6	d6	e6	f6	g6	h6
7	a7	b7	c7	d7	e7	f7	g7	h7
8	a8	b8	c8	d8	e8	f8	g8	h8

The rules for Othello are very simple:

1. Players alternate taking turns, with Black moving first.
2. A legal move consists of placing a new disc on an empty square, and flipping **one or more** of the opponent's discs.
3. **All** of the opponent's pieces that 'sandwiched' between the disc that has just been placed on the board and a disc of the same color that

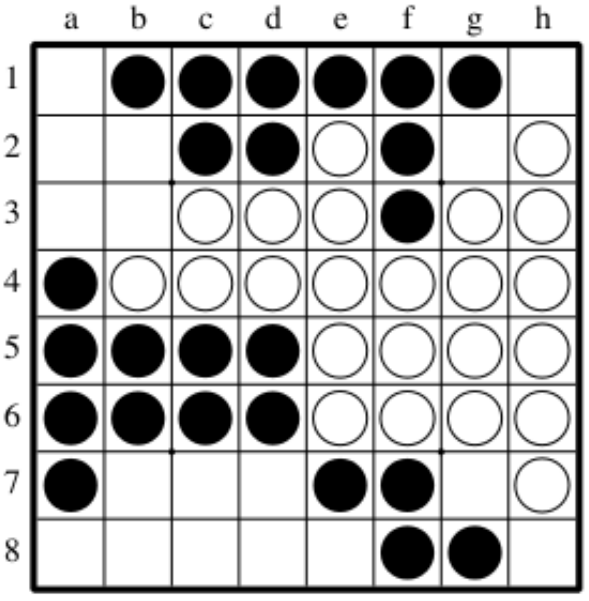
is already on the board should be flipped. Sandwiches can be formed **vertically, horizontally, or diagonally**. To create a sandwich, all the squares between the new disc and the disc of the same color that is already on the board must be occupied by the opponent's pieces, with no blank squares in between.

4. If a player is unable to make a legal move (in other words, regardless of where a new disc is placed, it is not possible to flip at least one of the opponent's pieces, the player must skip that turn.
5. If a player has at least one legal move available, the move must be made and the turn cannot be skipped.
6. The game continues until the board is completely filled or neither player has any legal moves available.

Perhaps the most basic strategy in Othello is to take the corners. According to the rules, a disc located in a corner cannot be flipped, so if you are able to take a corner, that disc will be yours for the rest of the game.

The following image shows an example of a "take a corner" puzzle:

White in 2



The expression "White in 2" indicates that it is White's turn, and you are asked to evolve a plan that will allow White to occupy one of the board's corners on White's second move. To solve the puzzle correctly, your plan must deal with every possible defense Black may carry out so that, no matter what move Black makes, it cannot stop White from occupying a

corner on White's second move. White does not have to prevent Black from taking a corner, as long as White is able to occupy a corner.

In the puzzle above, White's first move (and the solution to the puzzle) should be **b2**. This gives Black only three options: a2, a3, and b3. Regardless of the move that Black chooses, it will turn over the new disc at b2, allowing White to occupy a corner (**a1**) on its second move.

Input

The first line of the input contains an integer N. N puzzles follow. Each puzzle starts with a line indicating the player whose turn it is to move and the number of moves in which that player must occupy a corner. These lines have the format "[Black|White] in M", where M is an integer. Eight lines follow, each with eight characters, describing the current status of the board. A black disc is represented by an 'X', a white disc is represented by an 'O', and an empty cell is represented by a '.'.

Output

The output is the answer to each puzzle in a separate line. The answer to a puzzle is first move that the player should make in order to occupy a corner on his Mth move, or "Impossible" if it is not possible to force the opponent to give up a corner in the specified number of moves.

Notes

It is certain that the specified player will not be able to occupy a corner in less than M moves.

Limits

1 <= N <= 10
2 <= M <= 6

Example input

```
3
White in 2
.XXXXXX.
..XXOX.O
..OOOXOO
XOOOOOOO
```

```
XXXXOOOO
XXXXOOOO
X...XX.O
.....XX.
Black in 2
.OOOO...
O.OOX..O
OOXXXOOO
OOOXXOO.
OOOXXOO.
.OXXXXOO
.OXXXXXO
OOXOOO..
White in 2
.XXX.O..
..XXOO.X
XOOXXXX.
XXXXXXXO
XXXXOXX.
.XXXOXXX
...OX...
..XXX...
```

Example output

```
b2
a6
Impossible
```

Submit & test your code

To test and submit code we provide a set of tools to help you. Download contest tools if you haven't already done that. You will then be able to test and submit your solution to this challenge with the challenge token.

```
Challenge token: kQsZ9jQmcNVFo1-LRbwS
```

To test your program

```
./test_challenge kQsZ9jQmcNVFo1-LRbwS path/program
```

A nice output will tell you if your program got the right solution or not. You can try as many times as you need.

To submit your program to the challenge

```
./submit_challenge kQsZ9jQmcNVFo1-LRbwS
path/source_pkg.tgz path/program
```

Note that you first need to solve the test phase before submitting the code. During the submit phase, in some problems, we might give your program harder questions, so try to make your program failsafe.

Important: In this phase, you must provide the source code used to solve the challenge and, if necessary, a brief explanation of how you solved it.

Remember **you can only submit once!** Once your solution is submitted you won't be able to amend it to fix issues or make it faster, so please be sure your solution is finished before submitting it.

If you have any doubts, please check the info section.

Go ahead

I'm done! :)

Once you have submitted your code, hit refresh and continue to next challenge.

I'm stuck! :(

Be sure you follow the Tuenti Engineering twitter for updates and possible

hints during the contest.

If this challenge is too hard and you are blocked, you will be able to skip it after two hours. Note that **you won't be able to complete it later**, and you have a limited number of challenges to skip.

Finally, if you run out of skips but are still really stuck with one problem, you will be able to skip it after 24 hours.

Challenge status:

Test case	Not done
Solution submitted	Not done
Skip	You still have to wait 0h, 30m and 0s to be able to skip this challenge

Refresh status