

# Chessboard State 1: National Chess Tournament

**Note: This is an easier version of the problems CSTATE2 and CSTATE3. A solution to those problems (except the I/O format) will pass here.**

The inhabitants of Slovakistan decided to organize a national chess tournament.

Since the first place prize is 1 point on SPOJ, everyone played as if their life was at stake.

However, since time was not measured in this chess tournament (unlike at most chess tournaments in the rest of the world), a problem occurred - players took way too long to admit defeat.

They always just kept thinking and thinking, trying to find some move which would prevent their king from being taken. Just one more moment... And perhaps one more...

The inhabitants of Slovakistan are planning to hold the tournament next year as well. It would be great if a skilled programmer lent them a hand and got rid of this nuisance!

For a given description of a chessboard, decide its state - which player's king is under threat, and whether it is a Check or a Checkmate.

For the purposes of this problem only take into account basic moves of every chesspiece; complicated moves such as Castling, En passant, moving a pawn 2 squares or Promotion are not accepted in Slovakistan.

## Input

The first line of the input contains the integer  $1 \leq t \leq 2000$  - the number of chessboards.  $t$  descriptions of a chessboard follow.

Each description of a chessboard consists of **8** rows, each consisting of **8** characters (the chessboard is of the traditional dimensions **8x8**).

These characters are from the set **{.KQRBHPkqrbhp}**, representing in this order an empty square, king, queen, rook, bishop, knight (horse), and pawn.

The pieces belonging to the White player are marked by lowercase characters; the upper side of the chessboard (first in the input) belongs to him, hence white pawns move downwards.

The pieces belonging to the Black player are marked by uppercase characters; the lower side of the chessboard belongs to him, hence black pawns move upwards.

A blank line follows after every chessboard.

The number or placement of the pieces may by all means be impossible to reach in a standard game of chess, however you may assume that on every chessboard there is exactly one white

king ('k') and one black king ('K').

## Output

For each chessboard output one line with one of the following messages:

- **"Safe"**, if neither players' kings are being threatened
- **"Impossible"**, if both players' kings are being threatened
- **"{colour} Check"**, where **{colour}** is either **"Black"** or **"White"**, if the respective player's king is being threatened, and there exists at least one valid move by any of his pieces after which his king is no longer threatened
- **"{colour} Checkmate"**, where **{colour}** is either **"Black"** or **"White"**, if the respective player's king is being threatened, and there exists no valid move by any of his pieces after which his king is no longer threatened

## Additional note

There are 13 input files "0" through "12".

In order to make finding mistakes easier and/or let you implement various chesspieces one by one, with the exception of the example (file #0), the following holds:

Chesspiece	Appears first in file #
King	1
Knight	1
Rook	3
Bishop	5
Queen	5
Pawn	8

Additionally, after submission you can view extra information about the result of each file by clicking the result text ("accepted", "wrong answer", ...).

As an example, if you have received WA, the extra information looks like so:

File #5: Result is WA, message is 'Test Case 47: Your answer: Black Checkmate; Correct answer: Black Check', time = 0.100000, mem = 123456

Note that the message can go haywire if your program does something unexpected, such as end before reading all input.

## Example

**Input:**

7

....k...  
.....  
...H...  
.....  
...h...  
.....  
...K...  
.....

.....  
...h...  
..k.....  
.....H..  
...K...  
.....  
.....  
.....

k..H...  
...H...  
.HH....  
.....  
.....  
.....K.  
.....  
.....

k..H...  
...H...  
.HH....  
.....  
.....  
.....K.  
.r.....  
.....

K.....Rr  
R.h...r  
.....  
.....  
rr.....  
....k..  
.....  
.....

k.....  
.....R  
.....  
...B...  
.R.....  
.....K.  
..q.....  
.....

.....  
..PPP...  
..PkP...  
...Pp...

...K....

.....

.....

.....

**Output:**

Impossible

Safe

White Checkmate

White Check

Black Checkmate

White Check

Black Check

This is my first time using my own judge/master judge; although this problem is 'interactive' in order to get the info message, the input is all given first before your output is read, so hopefully it should have no effect. If you have a suspicion it is doing something funky, do let me know :)