Checkmate

Intro

Bob and his wife just started watching **The Queen's Gambit** on **Netflix**. Bob wants to impress her, but he doesn't have any intuition or skill when playing chess. What he does have is access to some of the world's smartest students who happened to apply for this year's **PSI:ML** seminar. He's taken a couple of pictures of the chessboard in some interesting positions, but he can't figure out what's happening. Will you help him?

Description

In a nutshel

You need to find a chessboard on an image. Once you do you need to figure out where all the pieces are. After that you need to figure out if either king is in check. For maximum points figure out if that check is a checkmate.

More details

You are provided with a path to a folder in a form of `<prefix>/<test_case_num>` where `<test_case_num>` is the current test case. The folder's structure is as follows:

```
<test_case_num>:
│   <test_case_num>.png
│
├───pieces
│   ├───black
│   │       bishop.png
│   │       king.png
│   │       knight.png
│   │       pawn.png
│   │       queen.png
│   │       rook.png
│   │
│   └───white
│           bishop.png
│           king.png
│           knight.png
│           pawn.png
│           queen.png
│           rook.png
│
└───tiles
        black.png
        white.png
```

The chessboard is in the `<test_case_num>.png`. You need to find the following: - The coordinates of the top left pixel of the board. The output needs to be comma separated height and width, i.e. `x,y` on a single line. - The pieces that are on the chessboard. The output needs to be in the [FEN Notation](). Place it on the second line. - If the black or white player is giving the check, or none of them. The output should be `B`, `W` or `-` respectivley. Place it on the third line. If you're not sure leave the line empty. - If the check is checkmate. The output should be `1` if it is, and `0` if not or if there's no check at all. Place it on the fourth line. If you're not sure leave the line empty.

The other folders and files in them are provided for you to use as you see fit. If you think you can do it without them then you don't have to use them.

Example input
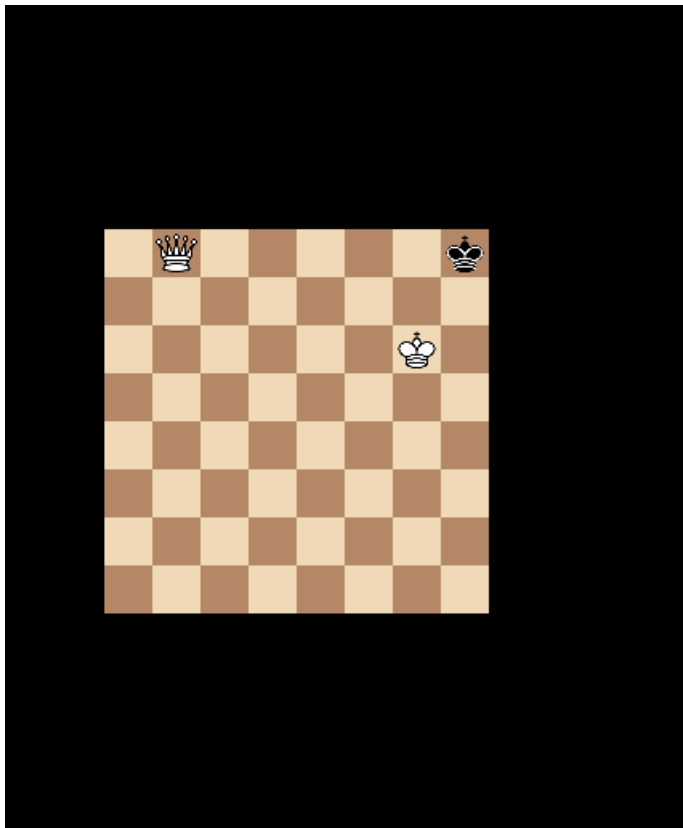
Let's say you were given the path to the folder containing this png:

0,0 is at the top left corner of the larger picture. From this point if you move down you increase the first coordinate, and if you move to the right you increase the second coordinate.

- The chess board's top left pixel is therefore `140,62` .
- On the board are the following figures:
  - White Queen on b8
  - Black King on h8
  - White King on g6
- White player is giving check
- There are no squares for black king to run so it is a checkmate

Example output

```
140,62
1Q5k/8/6K1/8/8/8/8/8
W
1
```

Scoring

- 20% for finding the coordinates of the top left pixel of the chessboard. It needs to be exact.
- 10%, 20%, or 30% for FEN string, depending on how much you get right.
- 20% for correctly finding the player giving the check. Only gets counted if you have any points in the previous two tasks. You get -20% penalty if you guess wrong, and the checkmate part of the task is automatically 0 points.
- 30% for correctly finding if the check is a checkmate. You get -30% penalty if you guess wrong.
- Minimum number of points for each test case is 0. This means that negative points apply only inside the test case.
- Maximum number of points for each test case is 26.
- You can have multiple submissions. We'll count only the best score across them.

Data

There are two data sets:

- *Public data set* is used for developing your solution. After you submit your solution, you will be able to see how well your solution performs against this data set. *Public data set* is not used for calculating the final score. Public data set is available **here**.
- *Private data set* is used for testing your solution. The final score will be measured against this data set. *Private data set* and the final score will be available after the homework finishes. During the competition you'll see `?` on those cases. *Private data set* contains different data than the *public data set*, but the distribution should be the same. *Private data set* also contains more examples.

Important notes

- There is no *en passant* move in the given test cases. You don't have to worry about that.
- There is no *castling* in the given test cases. You don't have to worry about that either.

- Other than the two exceptions mentioned above, standard chess rules apply.
- The most important file in the folder is the top level png containing the chessboard. The other folders and files in them are provided for you to use as you see fit. If you think you can solve it without them then you don't have to use them.
- The board doesn't rotate, so the top left tile will always be a8.

Limitations

- 7 [s] execution time
- 128 [MB] memory
- You can use only the libraries explicitly allowed by the competition. Check the announcements to see what those libraries are.