ACM-ICPC Live Archive

6427 Chess

There are two white rooks (♖) and one black king (♚) on a chess board. *White* starts. You're playing for *White* and need to checkmate the black king as soon as possible.

Write a program that will determine the minimum number of moves *White* needs to perform to checkmate black king, assuming *Black* follows the best possible strategy for prolonging the game.

Some information about the chess rules:

1. The position described above is not legal in chess since there is no white king on the board. Apart from that the game is according to the chess rules.

2. The board size is 8×8 cells. Columns of the board are labeled by the letters 'a' to 'h', and the rows by the digits 1 to 8.

3. The players (*White* and *Black*) alternately move one piece of their own color at a time. In the context of this problem we're only counting moves made by *White*.

4. The rook moves horizontally or vertically, through any number of unoccupied squares. It cannot jump over or stay in the same cell as another piece of the same color, namely the other white rook.

5. A check is a threat to capture the king on the next move turn, i.e. a position in which one of the rooks is on the same line as the king.

6. A king can move one square in any direction (horizontally, vertically, or diagonally) unless the move would place the king in check. If this condition holds, it's not prohibited for king to take over a cell already occupied by one of the rooks. In this case the rook is removed from play altogether.

7. Checkmate is a position in which a king is threatened with capture (i.e. is in check) and there is no legal move to escape the threat.

8. Stalemate is a situation where the player whose turn it is to move is not in check but has no legal move. The rules of chess provide that when stalemate occurs, the game ends as a draw (which is not acceptable for *White*).

Input

The input file contains several test cases, each of them as described below.

The first line of the input contains the number of positions (which is at least 1), and each of the following lines contain the description of a single position. All positions in the input are different. A position is represented by the location of the three pieces on the board: the black king first and then the two white rooks. It's guaranteed that no two pieces share a cell and there is no check at the initial position.

Sample Output

For each test case, write to the output one line for each position in the input. The line should contain the minimum number of moves which *White* needs to make to guarantee the checkmate, starting from the corresponding position. In case it's not possible to reach the goal, output '0' for relevant position.

Sample Input

```
2
c7 f1 g6
h6 c3 g8
```

Sample Output

```
2
1
```