Consider the following mini-version of chess: We have a 4×4 chessboard, with four white pawns on the first rank (bottom line in the input) and four black pawns on the last rank. The goal is for the player to get one of his pawns to the other end of the board (last rank for the white player, first rank for the black player), *or* to stalemate his opponent. A player is stalemated if he has no legal moves when it's his turn to move (this includes having no pawns left).

The pawns move as in ordinary chess, except that they can never move two steps. That is, a pawn may either move one step forward (toward the opposite rank), assuming that this square is empty. A pawn can also capture a pawn of the opposite color if it's one step ahead and to the left or right. Captured pieces are removed from the game.

Given the position of the pawns on a chessboard, determine who will win the game assuming both players play optimally. You should also determine the number of moves the game will last before it's decided (assuming the player that will win tries to win as fast as possible and the player to lose will lose as slow as possible). It will always be white's turn to move from the given position.

Input

The first line in the input contains the number of test cases (at most 50). Each case contains four lines describing the chessboard, preceeded by a blank line. The first of the four lines will be the last rank of the chessboard (the starting point for the black pawns). Black pawns will be denoted with a '`p`', white pawns with a '`P`', and empty squares with a '`.`'. There will be between one and four pawns of each color. The inital position will not be a final game position, and white will always have at least one legal move. Note that the input position may not necessarily be one that could have arisen from legal play from the games starting position.

Output

For each test case, output a line containing the text `white (`xx`)` if white will win, or `black (`xx`)` if black will win. Replace xx with the number of moves (which will always be an odd number if white wins and an even number if black wins).

Sample Input

```
2

.ppp
....
.PPP
....

...p
...p
pP.P
...P
```

Sample Output

```
white (7)
black (2)
```