Question 1.

A simple word search puzzle can be represented as a list of lists as follows.

```
puzzle = [['b', 'o', 'g', 'c'],
          ['g', 'd', 'c', 'a'],
          ['v', 'o', 'o', 't'],
          ['c', 'g', 'w', 'g']]
```


Here is a code that prints out the word “dog”, which is found in the puzzle:

```python
def get_word_diag(puzzle, row, column, length):
    word = ''
    for i in range(length):
        word += puzzle[row][column]
        row += 1
        column += 1
    return word

print(get_word_diag(puzzle, 1, 1, 3))  # print the word that starts at row 1, column 1, and goes diagonally to the bottom right, and is of length 3
```

Note that since `puzzle` is a list of lists, `puzzle[row]` is the row-th row of puzzle (a list), and so `puzzle[row][column]` is just the column-th element of the row `puzzle[row]`.

Write a function that takes in a puzzle in the format specified above and a list of words, and returns all the words in the list that appear in the word search puzzle. Start by writing functions (or expanding `get_word_diag`) to retrieve words of a given length that start at coordinate `(row, column)`, but go vertically, horizontally, diagonally in a different direction, etc. Then try to retrieve all the possible words from the puzzle and compare them to the words in the list.