HOMEWORK 12
CS 125
due at 11:45pm (classtime) on Tuesday, October 6

Write a Python *function* to solve each of the following problems. Plan each function on paper before you implement it in code.

Prepare your solutions in a single Python file. Use comments to clearly state the problem number for each of your solutions. Provide test cases to show that your functions produce the desired output. Upload your file to the [Homework 12 assignment on Moodle](#).

1. **Postal Codes**: Write a function `postalCode(state)` that takes a state name as a parameter and returns the postal code abbreviation for that state. (For example, the postal code for Minnesota is MN.) Your function must use a dictionary to store the state names and postal codes. At minimum, your function must work for the eight states whose names start with M; optionally, you may enter all fifty states in your dictionary.

2. **Reverse Lookup**: Write a function `reverseLookup(dictionary, value)` that finds all of the keys in a dictionary that map to a specific value. The dictionary and the value to search for are sent to the function as parameters. The function returns a (possibly empty) list of keys from the dictionary that map to the provided value.

   For example, if `d = {'a':5, 'b':3, 'c':5, 'a':2}`, then a call to `reverseLookup(d, 5)` returns `['a', 'c']`.

3. **Scrabble Scores:** In the game of Scrabble, each letter of the alphabet is worth a certain number of points. The points for each letter are given in the following table:

1 point	A, E, I, L, N, O, R S, T, U
2 points	D, G
3 points	B, C, M, P
4 points	F, H, V, W, Y
5 points	K
8 points	J, X
10 points	Q, Z

   The score of a word played in the game is the sum of the scores of its letters. Write a

function `scrabbleScore(word)` that accepts a word as a parameter and returns the Scrabble score of the word. In your function, create a dictionary that maps letters to scores, and use this when computing the score of each word.

4. **Number Words**: Write a function `numberWord(n)` that takes an integer `n` between 0 and 999 as a parameter and returns a string containing the English words for that number. For example, if the parameter is 253, then the function returns **"two hundred fifty three"**. Use one or more dictionaries to implement your solution, rather than a long sequence of if statements.