03/03/14 05:56:33 /Users/Clayton/Desktop/Repos/Senior-Sem-Projects/trie.py

```python
class Node:
    def __init__(self, cargo, end=False):
        self.cargo = cargo
        self.next = {}
        self.end = end

class Trie:
    def __init__(self):
        self.root = Node('.')

    def insert(self, word):
        current = self.root

        for letter in range(len(word)):
            if current.next.has_key(word[letter]):
                current = current.next[word[letter]]
            else:
                if letter == len(word)-1:
                    current.next[word[letter]] = Node(word[letter])
                    current.next[word[letter]].end = True
                else:
                    current.next[word[letter]] = Node(word[letter])
                    current = current.next[word[letter]]

    # go to end of substring typed in in trie
    # (move to new root)
    def traverse(self, word):
        current = self.root

        for letter in word:
            if current.next.has_key(letter):
                current = current.next[letter]
            else:
                return False
        return current

    # get all words starting at that new root
    def recommend(self, word, node, words=[]):

        if node.next.keys() == [] or node.end == True: # if at end of branch
or word
            words.append(word)

        for letter in node.next.keys(): # loop through dictionary values
            self.recommend(word + letter, node.next[letter], words) #
recursive call

        return words

    # wrapper method to minimize method calls in main
    def recommendations(self, word):
        startingNode = self.traverse(word)

        if not startingNode:
            return "no words found"

        words = self.recommend(word, startingNode)
        return words

def main():
    myTrie = Trie()

    myTrie.insert("dog")
    myTrie.insert("dad")
    myTrie.insert("cat")
    myTrie.insert("dogma")

    substring = raw_input("Enter string: ")

    for word in myTrie.recommendations(substring):
        print word

if __name__ == '__main__':
    main()
```