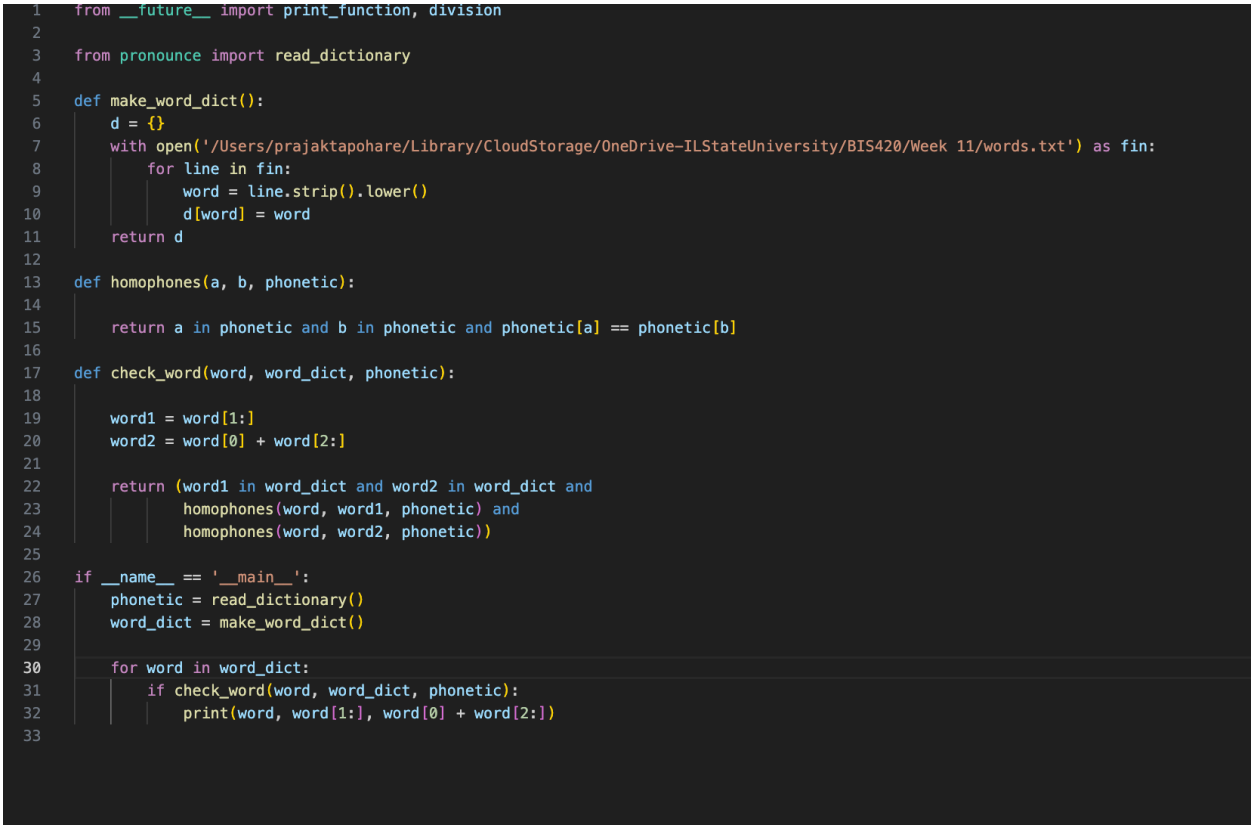
BIS 420 PROGRAMMING FOR DATA SCIENCE

PRAJAKTA POHARE
CHAPTER 10 EXERCISE 11.11
ILLINOIS STATE UNIVERSITY

Here's another Puzzler from Car Talk (http: // www. cartalk. com/ content/puzzlers): This was sent in by a fellow named Dan O'Leary. He came upon a common one-syllable, five-letter word recently that has the following unique property. When you remove the first letter, the remaining letters form a homophone of the original word, that is a word that sounds exactly the same. Replace the first letter, that is, put it back and remove the second letter and the result is yet another homophone of the original word. And the question is, what's the word?

Now I'm going to give you an example that doesn't work. Let's look at the five-letter word, 'wrack.' W-R-A-C-K, you know like to 'wrack with pain.' If I remove the first letter, I am left with a four-letter word, 'R-A-C-K.' As in, 'Holy cow, did you see the rack on that buck! It must have been a nine-pointer!' It's a perfect homophone. If you put the 'w' back, and remove the 'r,' instead, you're left with the word, 'wack,' which is a real word, it's just not a homophone of the other two words. But there is, however, at least one word that Dan and we know of, which will yield two homophones if you remove either of the first two letters to make two, new four-letter words. The question is, what's the word? You can use the dictionary from Exercise 11.1 to check whether a string is in the word list. To check whether two words are homophones, you can use the CMU Pronouncing Dictionary. You can download it from http: // www. speech. cs. cmu. edu/ cgi-bin/ cmudict or from http: // thinkpython. com/ code/ c06d and you can also download http: // thinkpython.com/ code/ pronounce. py , which provides a function named read_dictionary that reads the pronouncing dictionary and returns a Python dictionary that maps from each word to a string that describes its primary pronunciation.

Write a program that lists all the words that solve the Puzzler. Solution: http: // thinkpython.com/ code/ homophone. py .



```
from __future__ import print_function, division

from pronounce import read_dictionary

def make_word_dict():
    d = {}
    with open('/Users/prajaktapohare/Library/CloudStorage/OneDrive-ILStateUniversity/BIS420/Week 11/words.txt') as fin:
        for line in fin:
            word = line.strip().lower()
            d[word] = word
    return d

def homophones(a, b, phonetic):

    return a in phonetic and b in phonetic and phonetic[a] == phonetic[b]

def check_word(word, word_dict, phonetic):

    word1 = word[1:]
    word2 = word[0] + word[2:]

    return (word1 in word_dict and word2 in word_dict and
            homophones(word, word1, phonetic) and
            homophones(word, word2, phonetic))

if __name__ == '__main__':
    phonetic = read_dictionary()
    word_dict = make_word_dict()

    for word in word_dict:
        if check_word(word, word_dict, phonetic):
            print(word, word[1:], word[0] + word[2:])

```

from __future__ import print_function, division

from pronounce import read_dictionary

def make_word_dict():

d = {}

with open('/Users/prajaktapohare/Library/CloudStorage/OneDrive-ILStateUniversity/BIS420/Week 11/words.txt') as fin:

for line in fin:

word = line.strip().lower()

d[word] = word

```
    return d

def homophones(a, b, phonetic):

    return a in phonetic and b in phonetic and phonetic[a] == phonetic[b]

def check_word(word, word_dict, phonetic):

    word1 = word[1:]
    word2 = word[0] + word[2:]

    return (word1 in word_dict and word2 in word_dict and
            homophones(word, word1, phonetic) and
            homophones(word, word2, phonetic))

if __name__ == '__main__':
    phonetic = read_dictionary()
    word_dict = make_word_dict()

    for word in word_dict:
        if check_word(word, word_dict, phonetic):
            print(word, word[1:], word[0] + word[2:])
```