```
add keyword to All.words
for Entry in thesaurus
____ if Entry.Word = keyword
________ then if Entry.Synonyms is not empty
___________ then for synonym in Entry.Synonyms
_______________ add synonym to All.words
________ else if All.words is not empty
___________ then for searchWord in All.words
_______________ counter <- 0
____________________ if Corpus is not empty
_____________________________ then for Document in Corpus
_____________________________________ if Document is not empty
_____________________________________ then for word in Document
 _____________________________________ if word = searchWord
_____________________________________________ then counter <-- counter + 1
output: searchWord, counter

Submission:
add keyword to All.words
 for each Entry in thesaurus
 ____ if Entry.Word = keyword
________ then if Entry.Synonyms is not empty
 ___________ then for each synonym in Entry.Synonyms
 _______________ add synonym to All.words

for each searchWord in All.words
 _______ counter <-- 0
 _______ if Corpus is not empty
 __________ then for each Document in Corpus
_______________ if Document is not empty
____________________ then for each word in Document
_____________________________ if word = searchWord
 _____________________________________ then counter <-- counter + 1
 _______________ output: searchWord, counter


Function Programming Assignment:
def verify(number) : # do not change this line!
  if int(number[0]) == 4:
          if int(number[3]) - int(number[5]) == 1:
          sum = 0
          for num in number:
          if num != '-':
```

```
        sum += int(num)
        if sum % 4 == 0:
        if int(number[0] + number[1]) + int(number[7] + number[8]) == 100:
        return True
        else:
        return 4
        else:
        return 3
        else:
        return 2
  else:
        return 1

  # write your code here so that it verifies the card number

  # be sure to indent your code!

  #return False # modify this line as needed

input = "4094-3460-2754" # change this as you test your function
output = verify(input) # invoke the method using a test input
print(output) # prints the output of the function
# do not remove this line!

Final Code:
class Entry():
  def __init__(self, input_word, input_synonyms):
        self.word = input_word
        self.synonyms = input_synonyms

"""
        Thesaurus = List of Entry Objects
        Corpus = List of list of strings
"""

def search(keyword) :
  result = []
  All_words = [keyword]
  #All_words.append(keyword)
  for entry in Thesaurus:
        if entry.word == keyword:
        for synonym in entry.synonyms:
        All_words.append(synonym)
```

```
  for search_word in All_words:
          counter = 0
          for document in Corpus:
          for each_word in document:
          if search_word == each_word:
          counter += 1
          result.append((search_word, counter))


   # implement the function here

  return result # modify to return a list of tuples

input = "sad"
e1 = Entry("happy", ["excited", "smiling", "joyfull"])
e2 = Entry("sad", ["unhappy", "sorrowful", "dejected", "depressed"])
Thesaurus = [e1, e2]
doc1 = ["happy", "sad", "excited", "smiling", "happy"]
doc2 = ["depressed", "sorrowful", "sad", "excited", "happy"]
Corpus = [doc1, doc2]

output = search(input) # invoke the method using a test input
print(output) # prints the output of the function
# do not remove this line!
```