This semester, we will use Python Idle. Please submit your .py file on Moodle. Also submit this sheet on Moodle with the four submission boxes completed.

Programming Assignment #1 Anagram Checker-Jali Purcell

Exercise 10 in section 1.4 asks you to design an algorithm for checking whether or not two words are anagrams.

Step 1: Design your algorithm. Make sure your algorithm ignores cases and spaces.

Gather two words to test if they are anagrams or not.
Take out the spaces, and make the letters lowercase.
Sort each word into two different lists.
For each element in the lists, if they are not the same, then they are not anagrams.
If every element in the list matches, then they are anagrams.

Step 2: Copy the code from the box below into your Python compiler and complete it by implementing your algorithm and writing the header comments. (See the document Browning Coding Style Guide in the syllabus section of Moodle.) You must name your function *anagram* and it must have two string parameters. Your function must return either *True* or *False*. Make your code robust by checking to see that the values of the inputs are both strings. Write a pleasant message if they are not.

Step 3: Write additional test cases using my testPair function. Test your anagram function by calling test().

Step 4: When your function works correctly on these tests and on additional tests you write, submit your .py file on Moodle and complete the boxes below.

```
#The test function conducts a series of tests by calling testPair.

def test():
  testPair(1,'eat','tea',True)
  testPair(2,'Clint Eastwood','Old West Action',True)
  testPair(3,'arranged','deranged',False)
    YOUR ADDITIONAL TESTS GO HERE

#The testPair function takes a test number, two words, and the expected result (true if they are
anagrams and false otherwise). It prints a notice of whether or not the anagram function gave the
correct answer for these two words.

def testPair(testNumber,word1,word2,expected):
  if anagram(word1,word2)==expected:
    result = "passed"
  else:
    result = "failed"
```

```
  print("Test "+str(testNumber)+"  "+result+":"+ word1+" "+word2+" "+str(expected))


#The anagram function takes two words and returns true if they are anagrams and false otherwise.

def anagram(word1,word2):
   YOUR CODE GOES HERE
```

Your anagram function:

```
#The anagram function takes two words and returns true if they are anagrams and false otherwise.

def anagram(word1,word2):
   # initialize loop variables
   i=0
   j=0

   # make a connected string for each word
   # lowercase, no space
   word3=word1.replace(" " , "").lower()
   word4=word2.replace(" " , "").lower()

   # sort alphabetically into a list
   string1=[sorted(word3)]
   string2=[sorted(word4)]

   # iterate through lists
   for i in string1:
      for j in string2:

         # if elements do not match, they are not anagrams
         if i!=j:
            return False
         else:

            # if every element matches, return true
            return True
```

Transcript showing your code passes my initial tests:

```
>>> test()
Test 1  passed:eat tea True
Test 2  passed:Clint Eastwood Old West Action True
Test 3  passed:arranged deranged False
>>>
```

Ln: 72 Col: 4

Your additional test cases:

testPair(4, 'listen', 'Silent', True)
testPair(5, 'anagram', 'manga', False)
testPair(6, ‘COSMIC’, 'comics', True)

Transcript showing your code passes your additional tests:

```
>>> test()
Test 1  passed:eat tea True
Test 2  passed:Clint Eastwood Old West Action True
Test 3  passed:arranged deranged False
Test 4  passed:listen Silent True
Test 5  passed:anagram manga False
Test 6  passed:COSMIC comics True
```