#Assignment 8: Implement isPalindrome and isAnagram methods

#Assignment 8: Test for both positive and negative cases

# Panlindrome: See https://en.wikipedia.org/wiki/Palindrome#Characters,\_words,\_or\_lines

# This function returns "TRUE" if the given word is a palindrome

# else it returns "FALSE"

def isPalindrome(word):

print(word)

print(word[::-1])

if (word == word[::-1]):

return "TRUE"

else:

return "FALSE"

# Anagram: See https://en.wikipedia.org/wiki/Anagram

# This function returns "TRUE" if the two words are Anagrams

# else it returns "FALSE"

def areAnagrams(word\_1, word\_2):

if (sorted(word\_1) ) == (sorted(word\_2) ):

return "TRUE"

else:

return "FALSE"

# Test cases for the two functions

# Palindrome positive test cases (should return TRUE)

# radar, level, rotor, kayak, racecar, madam

x = isPalindrome("radar")

print("is radar palindrome? = ", x)

# Palindrome positive test cases (should return FALSE)

# python, java, silc

x = isPalindrome("python")

print("is python palindrome? = ", x)

# Anagram positive test cases (should return TRUE)

# evil=vile, silent=listen, eleven plus two=twelve plus one

word\_x = "evil"

word\_y = "vile"

result = areAnagrams(word\_x, word\_y)

print("Are ", word\_x, " and ", word\_y," anagrams? = ", result)

# Anagram positive test cases (should return FALSE)

# python = pxthon, java = lava, a = abcdefghijklmnopqrstuvwxyz

word\_x = "python"

word\_y = "pxthon"

result = areAnagrams(word\_x, word\_y)

print("Are ", word\_x, " and ", word\_y," anagrams? = ", result)