Problem 3

Ask the user for a string and print out whether this string is a palindrome or not.

|  |
| --- |
| wrd = input("Escriba una palabra:")  wrd2 = wrd[::-1]  if wrd == wrd2:  print("Esta palabra es un Palindromo")  else:  print("Esta palabra no es un Palindromo") |

Problem 4

Given two .txt files that have lists of numbers in them, find the numbers that are overlapping. One .txt file has a list of all prime numbers under 1000, and the other .txt file has a list of happy numbers up to 1000. The output should be stored in a third file, named as output.txt.

|  |
| --- |
| primeslist = []  with open('primenumbers.txt') as primesfile:  line = primesfile.readline()  while line:  primeslist.append(int(line))  line = primesfile.readline()  happieslist = []  with open('happynumbers.txt') as happiesfile:  line = happiesfile.readline()  while line:  happieslist.append(int(line))  line = happiesfile.readline()  overlaplist = []  for elem in primeslist:  if elem in happieslist:  overlaplist.append(elem)    print(overlaplist) |

Unit Test 3.

|  |
| --- |
| import unittest  import palindrome  class TestPalindrome(unittest.TestCase):  def test\_is\_palindrome\_true(self):  value = palindrome.is\_palindrome('racecar')  self.assertEquals(value, True)  def test\_is\_palindrome\_false(self):  value = palindrome.is\_palindrome('dedent')  self.assertEquals(value, False)  def test\_reverse\_normal(self):  value = palindrome.reverse('hello')  self.assertEquals(value, 'olleh')  def test\_reverse\_error(self):  list\_of\_bad\_value = [  123,  None,  ]  for bad\_value in list\_of\_bad\_value:  self.assertRaises(  TypeError,  palindrome.reverse,  bad\_value  )  if \_\_name\_\_ == '\_\_main\_\_':  unittest.main() |

Unit test 4

|  |
| --- |
| import unittest  import overlap  class Testoverlap(unittest.TestCase):  """  Return a list containing the elements which are in both primernumbers and happynumbers  >>> overlap([2,3,5,7,11,13,17,19,23,29,31,37,41,43,47], [1,7,10,13,19,23,28,31,32,44,49,68,70,79,82])  [7,13,19,23,31]  def test\_overlap\_numbers(primenumbers, happynumbers):  result = []  for element in primenumbers:  if element in happynumbers:  result.append(element)  return result  if \_\_name\_\_ == '\_\_main\_\_':  unittest.main() |