CSE101 – Fall 2019

Programming Assignment #3

Due November 5 2019 by 11:59pm, KST. The assignment is worth 25 points.

Instructions

For each of the following problems, create an error-free Python program.

- Each program should be submitted in a separate Python file that follows a particular naming convention: Submit the answer for problem 1 as “Assign3Answer1.py” and for problem 2 as “Assign3Answer2.py” and so on.
- These programs should execute properly in PyCharm using the setup we created in lab.
- At the top of every file add your name and Stony Brook email address in a comment.
- Please provide at least 2 test cases for each problem, except for Problems 4 and 6. This means calling the function you have created with an example input, so that when I run the entire program, I see the output of your test cases in the console.
- Please continue to use the naming conventions in Python and programming style that was mentioned in Assignment 1.

Problems

Problem 1: Merging two sorted lists (4 points)

Write a merge function that given two sorted lists of numbers or strings merges them to create a single sorted list. Write your own logic for the merge function. Do NOT use the built-in heapq.merge function provided by Python.

```
>>> from PythonLabs.RecursionLab import *

>>> a = RandomList(5, 'elements', sorted=True)

>>> a

['In', 'Md', 'Th', 'Tl', 'Zr']

>>> b = RandomList(5, 'elements', sorted=True)

>>> b

['Be', 'Cf', 'Cl', 'Np', 'Zn']

>>> merge(a,b)

['Be', 'Cf', 'Cl', 'In', 'Md', 'Np', 'Th', 'Tl', 'Zn', 'Zr']

>>> a = RandomList(5, sorted=True)

>>> a

[1, 9, 22, 75, 87]
```

```
>>> b = RandomList(5, sorted=True)

>>> b

[25, 26, 59, 64, 69]

>>> merge(a, b)

[1, 9, 22, 25, 26, 59, 64, 69, 75, 87]
```

Problem 2: Scrabble sort (4 points)

Write a Python program containing a function named `scrabble_sort` that will sort a list of strings according to the length of the string, so that shortest strings appear at the front of the list. Words that have the same number of letters should be arranged in alphabetical order. Write your own logic for the sort function. Do NOT use the built-in sort function provided by Python. To test your function, create a list of random words by using the RandomList function in PythonLabs and passing `'words'` as an argument. For example:

```
>>> from PythonLabs.RecursionLab import *

>>> a = RandomList(20, 'words')

>>> scrabble_sort(a)

>>> a

['mum', 'gawk', 'tree', 'forgo', 'caring', ... 'unquestioned']
```

Problem 3: Minimum perimeter problem (3 points)

Given an integer named `area` which represents the area of a rectangle, write a function `minPerimeter` that calculates the minimum possible perimeter of the given rectangle, and prints the perimeter and the side lengths needed to achieve that minimum. The length of the sides of the rectangle are integer numbers.

For example, given the integer `area = 30`, possible perimeters and side-lengths are:

(1, 30), with a perimeter of 62
(2, 15), with a perimeter of 34
(3, 10), with a perimeter of 26
(5, 6), with a perimeter of 22

Thus, your function should print only the last line, since that is the *minimum* perimeter. Some example outputs:

```
>>>minPerimeter(30)

22 where rectangle sides are 5 and 6.

>>>minPerimeter(101)

204 where rectangle sides are 1 and 101.

>>>minPerimeter(4564320)

8552 where rectangle sides are 2056 and 2220.
```

Problem 4: Fuel efficiency problem (3 points)

Write a function fuelEfficiency that computes the fuel efficiency of a multi-leg journey. The function will first prompt for input for the starting odometer reading and then get information about a series of legs. For each leg, the user enters the current odometer reading and the amount of gas used in liters (separated by a space). The user signals the end of the trip with a blank line. The program should print out the kilometers per liter achieved on each leg (to 1 decimal precision). The program should also print out for the trip the total distance, total fuel consumed, and fuel efficiency for the entire trip. See the example below:

```
>>>fuelEfficiency()

>>>Enter initial odometer reading: 1000

>>>Enter leg information: 1040 2

20.0 km per liter

>>>Enter leg information: 1080 3

13.3 km per liter

>>>Enter leg information: 1190 10

11.0 km per liter

>>>Enter leg information: 1250 5

12.0 km per liter

>>>Enter leg information:

Total distance = 250 km, Fuel consumed = 20 liters.

Total Fuel Efficiency = 12.5 km per liter.
```

Problem 5: The Goldbach's conjecture problem (4 points)

The Goldbach conjecture asserts that every even number is the sum of two prime numbers. Write a function goldbachConjecture that gets an even number as an input and then finds two prime numbers that add up to the number.

```
>>> goldbachConjecture(38)

38 = 7 + 31 (both are primes)

>>> goldbachConjecture(44)

44 = 3 + 41 (both are primes)

>>> goldbachConjecture(56)

56 = 3 + 53 (both are primes)
```

Problem 6: Word guessing game (7 points)

Write a Python game that requires the user to guess the characters in a hidden word following the description and example below.

Whenever the user guesses a character correctly, all occurrences of that character are shown. If you guess all the characters in the word, print out the word and let the user know they won. If you fail to guess any character in the word a total of 6 times, print out the word and tell the user they have lost.

Any guesses should be case-insensitive, meaning they should match a letter whether the letter is capitalized or lowercase. If the user types in more than one letter, only the first letter should be used as input. Additionally, if the user repeats the same letter, the program should remind them that they already guessed that letter and not count it as a guess.

Pick a random word for the user to guess using PythonLabs RandomList.

An example of the game running is below:

```
Welcome to the word guessing game. You have 6 attempts to guess the
letters in _ _ _ _ _

>>>Enter a letter to guess: s
's' is in the word
_ _ _ s _    Number of attempts left:  6

>>>Enter a letter to guess: R
There is no 'r' in the word
_ _ _ s _    Number of attempts left:  5

>>>Enter a letter to guess: o
'o' is in the word
_ o _ s _    Number of attempts left:  5

>>>Enter a letter to guess: o
You've already guessed o
_ o _ s _    Number of attempts left:  5

>>>Enter a letter to guess: h
'h' is in the word
h o _ s _    Number of attempts left:  5

>>>Enter a letter to guess: u
'u' is in the word
h o u s _    Number of attempts left:  5

>>>Enter a letter to guess: e
'e' is in the word
You won! The word is 'house'
```