CSE101 – Introduction to Computers
Python Programming Assignment # 3
(25 points, Submission due date: 25 April 2019)

Instructions

For each of the following problems, create an error free efficient Python program. Each program should be submitted in a separate Python file respectively that follows a particular naming convention. (E.g. The Python program for Question 1 should be in .py file with name Assign3Answer1.py. The Python program for question 2 should be in .py file with name Assign3Answer2.py. Include one or two input cases in your program. The program should execute properly in PyCharm).

Problems

Problem 1: Average number of comparisons (6 points)

Write a Python program that contains function named ***def test_search(size, ntests)*** that will compute the average number of comparisons made by a successful linear search. Your function should take two arguments, size and ntests, which determine how long the test list should be and how many tests to run. For example, to search a list of 100 numbers 25 times the call to test_search would be

```
test_search(100,25)
```

Your function should create an empty list to hold the results of the tests. Use a loop to call isearch the specified number of times, making sure it searches for a random value known to be in the list. Figure out how many comparisons were made by each test and append the count to the list of results. Finally, use your mean function to compute the average number of comparisons.

Here is an example for 250 tests on a list of 1,000 numbers:

```
>>> test_search(1000,250)
```

512.384

The results show that on average it took 512 comparisons to search a list of 1,000 items.

Problem 2: Scrabble sort (7 points)

Write a Python program containing function named ***def scrabble_sort(a)*** 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 (but words that all have the same number of letters will be arranged in alphabetical order). To test your function make a list of random words by passing ’words’ to RandomList in PythonLabs, e.g.

```
>>> a = RandomList(20, ’words’)

>>> scrabble_sort(a)

>>> a

[’mum’, ’gawk’, ’wist’, ’forgo’, ’caring’, ... ’unquestioned’]
```

Problem 3: Triangular triplet (6 points)

An list A consisting of N integers is given. A triplet (P, Q, R) is triangular if $0 \leq P < Q < R < N$ and:

A[P] + A[Q] > A[R],

A[Q] + A[R] > A[P],

A[R] + A[P] > A[Q].

Where P, Q and R are distinct elements. For example, consider list A such that:

A[0] = 10 A[1] = 2 A[2] = 5

A[3] = 1 A[4] = 8 A[5] = 20

Triplet (0, 2, 4) is triangular.

Write a function ***def triangular(A)*** that, given a list A consisting of N integers, if there exists a triangular triplet returns a list containing triangular triplet or returns None if there is no triangular triplet. If there is more than one triangular triplet, the first found triangular triplet is returned as output.

```
>>> triangular([10, 2, 5, 1, 8, 20])

[0, 2, 4]

>>> triangular([10, 50, 5, 1])

None
```

Problem 4: Frog jump (6 points)

A small frog wants to get from one point to another point in a giant plane. The frog is currently located at position (X_1, Y_1) and wants to get to a position (X_2, Y_2). The small frog always jumps a fixed distance, D. Count the minimal number of jumps that the small frog must perform to reach its target.

Write a function ***def MinFrogJumps (P1, P2, D)*** that, given two lists P1 and P2 representing coordinates of starting position and ending position respectively and D representing jump distance, shows distance between two points (in a straight line) and returns the minimal number of jumps from position P1 to a position P2.

```
>>>MinFrogJumps([10, 15], [-2, -5], 5)

Distance between P1 and P2: 23.32

Minimum number of jumps: 5
```