

CSE101 – Spring 2020

Programming Assignment #4

(12 points, Submission due date: 28 April 2020)

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 “Assign4Answer1.py” and for problem 2 as “Assign4Answer2.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.
- Write 2 test cases to test your code for all problems that do not take user input.

Problems

Problem 1:

(3 points)

Given a number n , what is the largest gap between successive primes which are less than number n ? Write a Python program to solve this problem.

```
>>>largestGap(100)
8
>>>largestGap(1000)
20
>>>largestGap(10000)
36
```

Problem 2:

(3 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 3:

(2 points)

A random walk is a particular kind of probabilistic simulation that models a number of random steps in space. You can think of a one-dimensional random walk in terms of coin flipping. Suppose you are standing on a straight sidewalk that extends both in front of and behind you. You flip a coin. If it comes up heads, you take a step forward; tails means to take a step backward.

Suppose you take a one-dimensional random walk of n steps. On average if you take a random walk 100 times, how many steps away from the starting point will you end up? Write a program to help you investigate this question. Note that the values it prints will naturally vary each time the program is run.

```
>>>randomWalk(10)
Average position change after 100 walks of 10 steps = 0.46
>>>randomWalk(50)
Average position change after 100 walks of 50 steps = 0.9
>>>randomWalk(100)
Average position change after 100 walks of 1000 steps = 0.84
```

Problem 4:

(4 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.

Create an array of at least 10 potential words and have your program pick one of these words at random each time the game is run.

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'
```