CMPSC-132: Programming and Computation II
Spring 2020

Homework 1

Due Date: 01/31/2020, 11:59PM ET
100 pts

Read the instructions carefully before starting the assignment. Make sure your code follows the stated guidelines to ensure full credit for your work.

Instructions:

- The work in this lab must be completed alone and must be your own.
- **Download the starter code file from the HW1 Assignment on Canvas. Do not change the function names or given started code on your script.**
- A doctest is provided as an example of code functionality. Getting the same result as the doctest does not guarantee full credit. You are responsible for debugging and testing your code with enough data, you can share ideas and testing code during your recitation class. As a reminder**, Gradescope should not be used to debug and test code**!
- Each function must return the output (Do not use print in your final submission, otherwise your submissions will receive a -10 point deduction)
- Do not include test code outside any function in the upload. Printing unwanted or ill-formatted data to output will cause the test cases to fail. Remove all your testing code before uploading your file (You can also remove the doctest). Do not include the input() function in your submission.
- **Examples of functionality are given in the starter code. Your code should return None when parameters don't follow the given specifications.**

Goal:

[20 pts] Write the function *rectangle*(*perimeter*, *area*), which takes two positive integers, *perimeter* and *area*. It **returns** an integer which is the length of the longest side of a rectangle with integer side lengths *w* and *h* which has the given *perimeter* and *area*. If no such rectangle exists, it returns None. As a reminder the perimeter of a rectangle with sides *w* and *h* is 2*w* + 2*h*. The area is *w* * *h*. **Hint:** The built-in function round takes a number as its argument and returns the nearest integer. For example, round(4.0) returns 4, round(4.3) returns 4, and round(4.7) returns 5. Similarly, to do integer division and get an integer result, discarding any fractional result, there is another operator, //, which performs floor division

Example:
rectangle(14, 10) will return 5 because a 2 x 5 rectangle 5 is the longest side of a rectangle with perimeter 14 and area 10
rectangle(25, 25) will return None because a 2.5 x 10 rectangle does not have integer side lengths

[20 pts] Write the method *translate*(*translationDict*, *txt*) that takes in a dictionary and a string. The dictionary contains keys of strings that have a value of another string. You will be translating the input string using the key-value pairs in the dictionary to a new string that has replaced all the words in the input string with the words' value in the dictionary. Your function should convert txt

to all lowercase letters and remove punctuation before making the translations. If a word in the string is not in the translation dictionary, the word will retain its original form. You can assume that all the keys in the dictionary are lowercase strings. **Hint:** str.lower() method returns a new string with all lowercase letters, str.split() splits a string into a list, default separator is any whitespace, and str.isalnum() returns True if all characters in a string are alphanumeric.

Example:
```
>>> myDict = {'up': 'down', 'down': 'up', 'left': 'right', 'right': 'left', '1': 'one'}
>>> text = 'UP down left right forward 1 time'
>>> translate(myDict, text)
'down up right left forward one time'
```

[10 pts] Write the function *onlyTwo*(*x, y, z*) which takes in three positive integers *x*, *y* and *z*. The function returns an integer with the sum of the squares of the two largest numbers. **Hint:** the *max()* method could be useful

Example:
onlyTwo(1, 2, 3) will return 13 since 2*2 + 3*3 = 13
onlyTwo(3, 3, 2) will return 18 since 3*3 + 3*3 = 18

[10 pts] Write the function *sumDigits*(*n*) which takes in a positive integer and sums its digits. **Hint**: Using floor division (//) and modulo(%) could be helpful here.

Example:
sumDigits(1001) will return 2 since 1+0+0+1 = 2
sumDigits(59872) will return 31 since 5+9+8+7+2 = 31

[20 pts] Write the function *largeFactor*(*num*) which takes in an integer *num* that is greater than 1. The function returns the largest integer that is smaller than *num* and evenly divides *num*.

Example:
largeFactor(15) will return 5 since the factors of 15 are 1, 3 and 5
largeFactor(7) will return 1 since 7 is prime

[20 pts] The Hailstone sequence of numbers can be generated from a starting positive integer *n* by:

- If *n* is even then the next *n* of the sequence = *n*/2
- If *n* is odd then the next *n* of the sequence = (3 * *n*) + 1
- Continue this process until *n* is 1.

The (unproven) Collatz conjecture is that the hailstone sequence for any starting number always terminates. Write the function hailstone(num) that takes in an integer *num* and returns a list with the hailstone sequence starting at *num*.

Example:
hailstone(5) will return [5, 16, 8, 4, 2, 1]
hailstone(6) will return [6, 3, 10, 5, 16, 8, 4, 2 , 1]

Deliverables:

- Submit your code with the file name HW1.py to the HW1 GradeScope assignment before the due date. If you were unable to complete a function, comment out the function definition.