MIS 6382
Object Oriented Programming in Python
Spring 2021
Homework Three

The following guidelines should be followed and will be used to grade your homework:

- The code for each question should be implemented using Jupiter notebooks.
- This is an individual homework assignment; no group submissions will be accepted.
- Testing samples shown in the question serve as the bottom line for debugging purpose, i.e. your code should work for all testing samples. You can definitely endeavor to handle more exceptions, but as long as your code is not too specific, is logically right, and works for testing samples, it will be considered as right.
- **All the code should be included in one single Jupyter Notebook file (.ipynb) and submitted to eLearning.** The file should be named using your name and the chars "hw3", e.g. firstname_lastname_hw3. You will be penalized 15% of the grade if your submission does not follow these requirements.
- You will get zero points if your program has syntax errors.

1. Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged. You need to use an input() function to take inputs. The testing samples come from separate cases so you don't need to use a loop to keep asking for inputs.

```
Please enter a string:sing
The new string is: singly

Please enter a string:song
The new string is: songing

Please enter a string:si
The new string is: si
```

2. Write a Python function called has_sublist (), that accepts a list as input and checks if any of its elements are themselves lists. You only need to check the following three testing cases and print out the results as shown below. You can directly assign values to the argument list and don't need to use an input() function.

```
It is True that list [1, 2, [3, 4]] has a sublist in it.
It is False that list ['a', 'b', 'c', 'd'] has a sublist in it.
It is False that list [1, 2, ('a', 'b')] has a sublist in it.
```

3. Write a Python function concat_list() that accepts a list containing string values and an integer as input and prints out a new list where each element in the list is concatenated with all integers from 1 to n. You only need to check the following testing case and print out the results as shown below. You can directly assign values to the input list and integer (i.e. no need to use an input() function).

```
The input list is ['a', 'b', 'c', 'd'].
The input integer is 5.
The new list is ['a1', 'a2', 'a3', 'a4', 'a5', 'b1', 'b2', 'b3', 'b4', 'b5', 'c1', 'c2', 'c3', 'c4', 'c5', 'd1', 'd2', 'd3', 'd
4', 'd5'].
```

4. Write a Python program that accepts a comma separated sequence of words as input and prints the **unique** words in sorted form (alphanumerically).

```
Please enter a sequence of words seperated by commas:good,no,yes,hello,day,good,hello
The new sorted list is: ['day', 'good', 'hello', 'no', 'yes']

Please enter a sequence of words seperated by commas:good no yes
The sequence is not seperated by commas
```

5. Step 1: Use list comprehension expression to create a list to include elements that satisfy the following criteria: (1) All elements are integers between 1 and 30 (inclusive); (2) All elements are not divisible by 2 nor by 5. Print out the list and move to the next step.
Step 2: Convert all elements that are divisible by 3 to their opposites (e.g. 3 to -3) and print out the new list (step-2 list).
Step 3: Sum up the value of all positive elements in the step-2 list and print out the result.
You can use the following format to print out your results.

```
My initial list is: [redacted]
My step-2 list is: [redacted]
The sum of positive elements in my step-2 list is: [redacted]
```