NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
ISLAMABAD CAMPUS
INTRODUCTION TO COMPUTING (CS101) - FALL 2016
ASSIGNMENT # 2 – Part I

Due Date: October 30, 2016 (10:00pm)
Instructions

1. Write the Python programs.
2. Solution to all the problems should be written in a separate (.py) file.
3. Submit the source code (i.e. python code in .py file) via slate. Submissions via email will not be accepted.
4. Moreover, you must submit your notebook file (.ipynb) as well with all the codes and their outputs in different cells.
5. Use proper naming convention to name the file containing source code.

   For example, the file containing the source code for first question of the first assignment should be named as **i16xxxx_assignment_2_q1.py**, replace i16xxxx with your student number.

   Similarly, the third question of the second assignment should be named as **i16xxxx_assignment_2_q3.py**, etc.

3. Please write your name and roll number at the beginning of the each program using comments.

Plagiarism: Plagiarism is not allowed. If found doing plagiarism you will be awarded zero marks in the assignment.

Note:

- **Follow the given instructions to the letter, failing to do to so will result in a zero.**

[Recall that we can process the string as list, one character at a time.]

1. **Checking for Solution (or Span):** Write a program that takes four column vectors (lists) as inputs and tell whether the last vector is spanned by the first three vectors or not.

   For example, If we enter following four lists (or column vectors),

   v1=[1,3,5]

   v2=[2,4,-3]

   v3=[3,6,7]

   v4=[5,12,-6]

   then we know that v4= 2*v1+3*v2+(-1*v3). Thus v4 (a column vector) lies in the space spanned by first three vectors. If it is indeed spanned by the first three vector print the span coefficients, otherwise print "Does not lie in the space spanned by the column vectors".

   Note: You need to only check for integer coefficients, as there will be no fractional part.

2. **Filtering Negative Elements** Write a program that filters negative elements out of a list. The program should build a new filtered list while the original list should remain unchanged. For example, if a list containing the elements 2, –16, 2, –5, 0, 1, –2, –3 is used in the program, the program should build a new list containing 2, 2, 0, 1. Note the original ordering of the non-negative values is unchanged in the result.

3. **Rotate The List** Write a program that shifts all the elements of a list backward one place. The last element that gets shifted off the back end of the list is copied into the first (0th) position. For example, if a list containing the elements 2, 1, 10, 4, 3, 6, 7, 9, 8, 5 is used in the program, it would be transformed into 5, 2, 1, 10, 4, 3, 6, 7, 9, 8. Note that your function must physically rearrange the elements within the list, not just print the elements in the shifted order.

4. **Balanced List** Write a program that determines if the number of even and odd values in an integer list is the same. The program should print true if the list contains 5, 1, 0, 2 (two evens and two odds), but it should print false for the list containing 5, 1, 0, 2, 11 (too many odds). The function should print true if the list is empty, since an empty list contains the same number of evens and odds (0 for both).

The program must not affect the contents of the list.

5. **Unique Elements** Write a program that prints true if a list contains duplicate elements; it should print false if all the elements in the list are unique. For example, the list [2, 3, 2, 1, 9] contains duplicates (2 appears more than once), but the list [2, 1, 0, 3, 8, 4] does not (none of the elements appear more than once). An empty list has no duplicates. The program must not affect the contents of the list.
6. **Longest Substrings** Write a program that finds the longest substring in a given string. For example, if string contains following value = 'My Village is Vill', your program should print "Village", since this is the longest substring.
7. **Counting Vowels** Write a program that counts up the number of vowels contained in the a string. Valid vowels are: 'a', 'e', 'i', 'o', and 'u'. For example, if string contains following value = 'azcbobobegghakl', your program should print:

   Number of vowels: 5

   You can use raw_input function to take input from the user.
8. **Counting Bob** Write a program that prints the number of times the string 'bob' occurs in a user input. For example, if string contains following value = 'azcbobobegghakl', then your program should print

   Number of times bob occurs is: 2

   You can use raw_input function to take input from the user.
9. **Equality of lists/Strings** Write a piece of code that examines two lists of integers and reports whether the two lists contain the same elements. By definition, an list is always equal to itself. Note that lists of different lengths cannot be equal, so the method should return false in such cases.
10. **Counting of Elements** Write a code that prints the count of each element of a list. For example, if the list passed contains the values [27, 15, 15, 11, 27,9,13], your method should print.

    27 occurred 2 times.

    15 occurred 2 times.

    11 occurred 1 time.

    9 occurred 1 time.

    13 occurred 1 time.

11. **Frequently Occurring Element** Write a code that prints the most frequently occurring element of a list of integers. Break ties by choosing the lower value. For example, if the list passed contains the values [27, 15, 15, 11, 27], your method should return 15. [Hint: Use the solution of above program]
12. **Permutation [Challenging]** Write a program that determines if two lists contain the same elements, but not necessarily in the same order. The program should print true if the first list contains 5, 1, 0, 2 and the second list contains 0, 5, 2, 1. The program should print false if one list contains elements the other does not or if the number of elements differ. This program could be used to determine if one list is a permutation of another list. The program does not affect the contents of either list.