FR. CONCEICAO RODRIGUES COLLEGE OF ENGG.Fr. Agnel Ashram, Bandstand, Bandra (W) Mumbai 400 050.

Class:	S.E. I.T.
Subject Name:	Python Lab
Tutorial No:	**1**
Mapped to CO	LO1. Describe the Numbers, Math functions, Strings, List, Tuples and Dictionaries in Python LO2. Express different Decision Making statements and Functions
Deadline:	**25/02/2018**
Date of Submission:	
Roll No:	
Name of the Student:	

Evaluation:

Sr. No	Rubric	Grade
1.	**On time completion and submission (2)**	
2.	**Completeness (2)**	
3.	**Accuracy (4)**	
4.	**Plagiarism check (2)**	

Signature of the Teacher

Rubrics for assessment of Tutorial:

Indicator	**Poor**	**Average**	**Good**
On time completion and submission (2)	Late submission(0)	Two sessions late (0.5)	Submits on time(2)
Completeness(2)	Not able to soleve any problem(0)	Able to solve single ptoblem(1)	Able to solve all the problems(4)
Accuracy(4)	No output(0)	Partial output or program works for very few test cases(1)	Correct output. Program works for all test cases(4)
Plagiarism check(2)	Identified that all codes are copied from other students(0)	Identified that upto 25% of codes are self written and for rest of the codes due to credit is given to classmate from which help is taken(0.5)	Identified that all the codes are self written(2)

1. An airline has assigned each city that it serves a unique numeric code. It has collected information about all the direct flights it operates, represented as a list of pairs of the form (i,j), where i is the code of the starting city and j is the code of the destination.
   It now wants to compute all pairs of cities connected by one intermediate hope --- city i is connected to city j by one intermediate hop if there are direct flights of the form (i,k) and (k,j) for some other city k. The airline is only interested in one hop flights between different cities --- pairs of the form (i,i) are not useful.

   Write a Python function onehop(l) that takes as input a list of pairs representing direct flights, as described above, and returns a list of all pairs (i,j), where i != j, such that i and j are connected by one hop. Note that it may already be the case that there is a direct flight from i to j. So long as there is an intermediate k with a flight from i to k and from k to j, the list returned by the function should include (i,j).
   The input list may be in any order. The pairs in the output list should be in lexicographic (dictionary) order. Each pair should be listed exactly once.

   For instance

   ```
   >>> onehop([(2,3),(1,2)])
   [(1, 3)]

   >>> onehop([(2,3),(1,2),(3,1),(1,3),(3,2),(2,4),(4,1)])
   [(1, 2), (1, 3), (1, 4), (2, 1), (3, 2), (3, 4), (4, 2), (4, 3)]

   >>> onehop([(1,2),(3,4),(5,6)])
   []
   ```
2. A list of integers is said to be a valley if it consists of a sequence of *strictly* decreasing values followed by a sequence of *strictly*increasing values. The decreasing and increasing sequences must be of length at least 2. The last value of the decreasing sequence is the first value of the increasing sequence.

   Write a Python function valley(l) that takes a list of integers and returns True if l is a valley and False otherwise.

   Here are some examples to show how your function should work.

   ```
   >>> valley([3,2,1,2,3])
   True
   >>> valley([3,2,1])
   False
   >>> valley([3,3,2,1,2])
   False
   ```

3. Write a Python function maxcount(l) that takes a list of immutable values as argument. The list could have repeated values. The function should return the number of times the most frequent value is repeated.

   For instance, maxcount([1,17,31,17,22,17]) should return 3 because the most frequent value, 17, occurs 3 times. Likewise maxcount(["the","higher","you","climb","the","further","you","fall"]) is 2 becaues the most frequent values, "the" and "you" both occur 2 times.

   Sample Test Cases
   Input   Output
   Test Case 1
   maxcount([1,17,31,17,22,17])
   Output
   3
   Case 2
   maxcount(["the","higher","you","climb","the","further","you","fall"])
   2
   Case 6
   maxcount(["the","higher","you","climb","the","further","you","fall","you","climb","the", "further","you","fall"])
   4

4. Write a Python function sublist(l1,l2) that takes two sorted lists as arguments and returns True if the the first list is a sublist of the second list, and returns False otherwise.

   A sublist of a list is a segment consisting of contiguous values, without a gap. Thus, [2,3,4] is a sublist of [2,2,3,4,5], but [2,2,4] and [2,4,5] are not.

   Sample Test Cases
   Input   Output
   Test Case 1
   sublist([2,2,3],[2,2,3,4,5])
   True
   Test Case 2
   sublist([2,2,4],[2,2,3,4,5])
   False
   Test Case 3
   sublist([2,2,3],[2,2,3,4,5])
   True

Test Case 4
sublist([2,2,4],[2,2,3,4,5])
False
Test Case 5
sublist([1],[3,4,1,5,6])
True
Test Case 6
sublist([],[7,8,9])
True
Test Case 7
sublist([13],[])
False

5. Write a Python program that reads input from the keyboard (standard input). The input will consist of an even number of lines of text. The input will be terminated by a blank line. Suppose there are 2n lines of input. Your program should print out the last n lines of the input, i.e., the second half of the input, followed by the first n lines, i.e., the first half of the input.

   E.g., if the input is the following:

   our dear friend,
   let's eat

   then the output should be:

   let's eat
   our dear friend,

   Sample Test Cases
   Input
   Test Case 1
   our dear friend,
   let's eat

   Output
   let's eat
   our dear friend,

Test Case 2
Input
Now is the time
For all good men
To come to the aid
Of the party

Output
To come to the aid
Of the party
Now is the time
For all good men

6. Write a short program that prints each number from 1 to 100 on a new line.

For each multiple of 3, print "Fizz" instead of the number.

For each multiple of 5, print "Buzz" instead of the number.

For numbers which are multiples of both 3 and 5, print "FizzBuzz" instead of the number.

Write a solution (or reduce an existing one) so it has as few characters as possible.