Python Coding Assignment #1

Ques.1

MAKE A PROGRAM TO MERGE TWO LIST INTO A SINGLE DICTIONARIES

● Take inputs from the user

● Any one list must contain unique elements

● both the list should be of the same size

● both the list should be a combination of numbers and names

● Name of dictionary you can take it accordingly

● file name should be in lnbmerge.py

Ques 2.

Write a program that converts Paragraph to List

● Take input from the user in string form(a sentence or para)

● All the words in the string having more than 4 letters should be stored in a list

● file name must be lnbstring_list.py in which you are writing code.

Sample Input : A paragraph is defined as “a group of sentences or a single sentence that forms a unit”. Length and appearance do not determine whether a section in a paper is a paragraph.

*Sample Outpu*t : [‘paragraph’ , ‘defined’, ‘group’ , ‘sentences’ , ‘forms’ , ‘Length’,‘appearance’ , ‘determine’ , ‘whether’ , ‘section’ , ‘paper’ , ‘paragraph’]

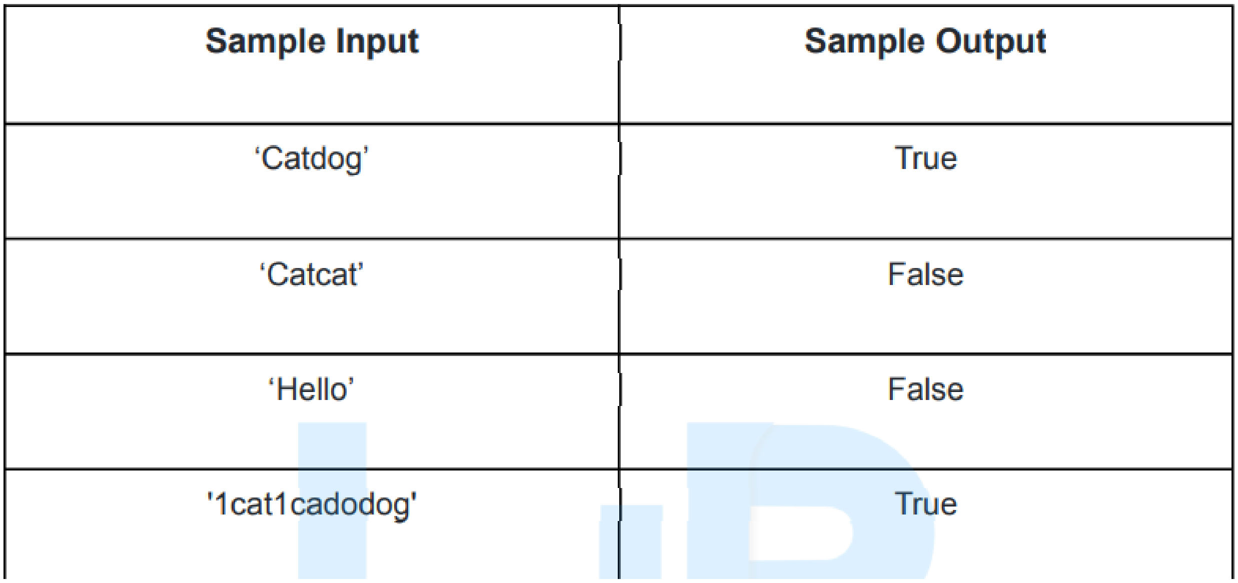
Ques 3.

Raining like cats and dogs

● Take user input in the form of a string.

● Return True if in the string "cat" and "dog" appear the same number of times in the given string

● file name must be lnbcatndog.py in which you are writing code



Sample Input	Sample Output
‘Catdog’	True
‘Catcat’	False
‘Hello’	False
'1cat1cadodog'	True

Ques 4.

Calendar

● Take input from the user in form of three integers
● The three integers represent a year, month and day
● Print the season of that month and day
● Now check whether the given year is a leap year or not

● If the year is a leap year then print the number of days in that year else print the next leap year
● file name must be lnbcalendercoditional.py in which you are writing the code

Ques 5.

Trekking with friends
● Three friends Suman, Amit and Ravi have gone for trekking and there they decided to plan a racing among themselves
● Amit beats Suman by A meters, Ravi by B meters and Suman beats Ravi by C meters
● Find the total length of trek they all have traveled
● File name must be in lnbassingmnet2.py in which you are writing code

Sample Input: A= 10 m, B= 20 m, C= 12m
Sample Output: Total length of the Track= 60 m

Ques 6.

Filter
● Create a function even_filter with one parameter. This function will return a list which will contain only odd values.
● Create a function odd_filter with one parameter. This function will return a list which will contain only even values.
● Call the function even_filter and pass a list of numbers as an argument.
● Call the function odd_filter and pass a dictionary as an argument.
● NOTE: The list and dictionary must be user define

Ques 7.

Greatest Common Divisor
● Take a user input in a list consisting of 2 or more than 2 numeric values.
● Find the GCD of the numbers of the list using a user defined function.
● File name must be lnbGCD.py in which you are writing code

Ques 8.

Write a program to print twin primes less than 1000. If two consecutive odd numbers are both prime then they are known as twin primes.

Define two functions checkPrime(): which checks whether the number is prime or not. If yes then return it.
The second function twinPrime(): Generates the list of twin primes and returns them.

Ques 9.

Write a program using functions to implement these formulae of permutations and combinations.
Number of permutations of n objects taken r at a time: $p(n, r) = n! / (n-r)!$.
Number of combinations of n objects taken r at a time is: $c(n, r) = n! / (r!*(n-r)!) = p(n,r)/r!$

Ques 10.

If all digits of a number n are multiplied by each other repeating with the product, the one digit number obtained at last is called the multiplicative digital root of n. The number of times digits need to be multiplied to reach one digit is called the multiplicative persistence of n.

Example: 86 -> 48 -> 32 -> 6 (MDR 6, MPersistence 3)
341 -> 12->2 (MDR 2, MPersistence 2)

Define three functions : prodDigits(): that finds the product of digits of a number, function MDR() and MPersistence() that input a number and return its multiplicative digital root and multiplicative persistence respectively.

Ques 11.

Given a list of integer numbers and an integer target, return indices of the two numbers such that they add up to target.

You may assume that each input would have exactly one solution, and you may not use the same element twice. You can return the answer in any order.

Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].

Ques 12.
‘Nand’ is an operator that is false only when both of its operands are true. Python doesn’t have a nand operator, so how can we write an expression that is equivalent to A nand B?

Ques 13.

Using books, the Web, or any other resource, find out what a logic gate is, and, in particular, what a half adder is. An output of the work should be a table showing the inputs and outputs for a half adder. Implement a half adder as a few lines of Python code. Write a test program that shows that your implementation works as expected. You should be able to test that your code works for all possible input values.

Ques 14.

This question is based on a function for stepping through a representation of a pack of cards. The exact nature of the cards isn’t important.
We use a list to represent pack and in our example we use numbers between 1 and 5 as cards. They could equally be numbers and suits as tuples, strings of character names, etc.

Here’s the function:

```
# A hand of cards
cards = [ 1 , 5 , 3 , 4 , 2 , 3 , 2 ]
def nextCard ( cards ) :
next = cards[0]
newHand = cards[1:] + [ cards[0] ]
return next , newHand
```

- What is the type of values returned by this function?
- Describe in words what the function does.

- It would be a simple matter for this function to alter the input list so that the first element becomes the last. This would simplify the return value, which is currently two items. Why might this be considered a poorer solution than the function as it is now?
- Write a loop that calls this function repeatedly, printing out the card from the top of the deck each time.
- Using the given function and your answer to the previous question as a starting point, write a program that checks a deck of cards for consecutive identical pairs (a 2 followed by a 2, for example). If a pair is found, a message should be displayed.
- What happens if the input list is empty? Correct the function so that it returns (None , [ ]) when an empty list is used as input.