PYTHON ASSIGNMENT 1

1. Take User input and process

- Take 5 integer input from the user.
- Remove all numbers less than 9.
- Calculate the sum of remaining numbers
- File name must be lnbsum.py in which you are writing code
- Commit this code on your github link under pythonbasic branch

2. Take User input and decode logic

- Take 2 integer input from user and print their products(their multiplication)
- IF product is greater than 500 then return their sum
- If the product is smaller than 500 then return "Hello LNB code is running fine !!"
- file name must be lnbcal.py in which you are writing code
- commit this code on your github link under pythonbasic branch

3. Take User input in string form and perform operation

- Take input from user in string form only and calculate the length of string
- IF length is greater than 7 then display only those character which are present in even index number
- if length is less than or equals to 7 then display only those character which are present in odd index number
- file name must be lnbstrindex.py in which you are writing code
- commit this code on your github link under pythonbasic branch

4. List operations and decoding

- two lists are given below L1, L2.
- create a new list called L3 containing items in below given pattern
- From L1 it must take only odd index items
- From L2 it must take only even index items
- file name must be lnblist.py in which you are going to write the code
- commit this code on your github link under pythonbasic branch

Inputs : L1 = [11, 21, 24, 12, 18]
L2 = [14, 44, 25, 37, 13]

Expected output : L3=[11,24,18,44,37]

5. Attendance of the Employees

- A Company's CEO wants the attendance of all the employees
- There are e no.of employees and w no.of working days
- Find the maximum number of consecutive days on which all the employees are present
- File name must be lnbattendance.py in which you are writing the code
- Commit this code on your github link under pythonbasic branch

*** Constraints:**

1<=e<=10

1<=w<=31

Each data[i][j]={'P','A'}

Sample Input: e= 5 , w= 7

Data = [PPPPP, PPAPP, AAAPP, PAPAP, AAAAA, PAAAP, PPPPP]

Sample output: 1,7

6. Operating the tuples

- Take input from the user in form of integers (minimum10 & maximum 20 integers)
- Convert the integers into binary tuples
- Use the operators AND & OR for the last two binary tuples and then display the result
- Now convert the resultant tuple into integer and display the value
- File name must be lnbtuple.py in which you are writing the code
- Commit this code on your github link under pythonbasic branch

7. Registration File

- User have to create a CSV file with 3 columns i.e. ID, Amount and DOB
- Import the csv file and check if the ID and DOB are valid
- ID format:

  Length - 10

  First 5 must be uppercase letter

  Last 5 must be digits

- DOB must be rejected if that date is invalid
- If the same ID with the same DOB occurs, merge the amount of both and write it in one index
- Also add one more column with Label- Verification and write ‘Verified’ corresponding to each valid entry. Else write ‘Invalid’
- File name must be lnbregister.py in which you are writing code
- Save the CSV file as lnbregister.csv
- commit this code on your github link under python list branch

8. Splitting the list

- Create a list ranging from 1-20
- Create a new list that contains first and the last five elements of the existing list
- Now create and display another list that contains square of the elements of the new list
- Now split the new list into three parts where the length of splitted parts should be two, three and five respectively
- File name must be in lnblist.py in which you are writing code
- Commit this code on your github link under the python string branch

9. Raining like cats and dogs

- Take user input in the form of a string.
- Return True if 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

- commit this code on your github link under pythonstring branch

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

10. Convert tuple to list using given string

- Take input from the user in the form of tuple
- Take another input from the user in the form of string
- Convert the tuple to list
- Add string to the list after each element
- Convert list to tuple again
- File name must be lnbconvtuple.py in which you are writing the code
- Commit this code on your github link under pythonbasic branch

Sample Input: (21,37,18), K= “Age”

Sample output: (21,”Age”,37,”Age”,18,”Age”)

11. Break the tuple

- Take an input from the user in the form of a tuple consisting of only numeric values. The tuple must consist of at least 3 values.
- Divide the tuple into two tuples such that the sum of elements of each tuple have the least difference
- File name must be lnbbreaktuple.py in which you are writing code
- Commit this code on your github link under the python tuple branch.

Sample Input: (1,3,4,2,9,2,2)

Sample Output: (1, 4, 9)
(3, 2, 2, 2)

12. Rearranging the Lists

- Take numeric input from the user in a form of list of tuples
- Rearrange the tuples according to the number of elements in ascending order of tuples to obtain the new list
- If the list consists of tuples with equal elements then arrange them according to their sum of elements
- If sum of their elements is equal then arrange them according to their first least number
- Display the new list
- File name must be in lnblist.py in which you are writing code
- Commit this code on your github link under the python list branch

Sample Input: [(2,1,2), (2,5), (4,5,3,7), (3,1), (1,6)]

Sample Output: [(1,6), (2,5), (3,1, (2,1,2), (4,5,3,7)]

13. 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.
- commit this code on your github link under pythonbasic branch

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 Output : [‘paragraph’ , ‘defined’, ‘group’ , ‘sentences’ , ‘forms’ , ‘Length’ , ‘appearance’ , ‘determine’ , ‘whether’ , ‘section’ , ‘paper’ , ‘paragraph’]

14. 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
- commit the code on github link under pythonbasic branch

15. Create a set with following data

- Take user input and search it on google
- You can use any python module to do the job
- Take top 10 URL from google search
- Create a list of URL
- Now create a set with any name from the list
- Set items must be string and can't be more than 4 char long
- Set item must not contain URL starting protocolo like http or https
- Save code in a file named lnbsets.py
- Commit code in github python module1 branch