001_Python_Final_Assignment_01(1)

July 12, 2021

JLUFE Spring 2021 (Feb-July)

Final Assignment Report

JILIN UNIVERSITY OF FINANCE AND ECONOMICS

Department of College of Managment Science and Information Engineering

Specialty Of Information Management and Information System

(2021)

Final Assignment: Part 01

12/07/2021

MODULE: Data Mining

Submitted by: Sehun() 0314021805421 (1854)
QQ: 2112440859 GitHub ID:sehun123456

1 RULES:

1. I have added tips and required learning resources for each question, which helps you to solve the exercise.
2. Finish the assignment on your **OWN** (**Any student find copying/sharing from classmates or internet will get '0' points!!!**)
3. Once you finish the Assignment **convert your .ipynb file into PDF** (both .pynb and .pdf file will be required!)
4. Create **.zip** file and include your two files:
    1. Your Jupyter Notebook file (**001_Python_Assignment_01.ipynb**)

2. Your PDF converted file of 001_Python_Assignment_01.ipynb (**001_Python_Assignment_01.pdf**)

5. Name your .zip file as your student number and name. example: **0318021907632 Milan().zip**

2 Python Assignment 01

2.0.1 Question 1:

Write a python program that generates a list containing only common elements between the two lists (without duplicates). Make sure your program works on two lists of different sizes.

Expected Output:

```
List 1: [0, 2, 4, 6, 12, 13, 14, 18, 20, 24, 25, 26, 27]
List 2: [0, 4, 7, 9, 10, 11, 13, 14, 17, 18, 20, 33, 39]
List of common elements are:  [0, 4, 13, 14, 18, 20]
```

For extra points: 1. Generate the two list randomly to test this 2. Generate each list in one line of Python.

```
In [27]: # Solution 1:
         import random
         num=random.randint(10,30)
         a=[random.randint(0,40) for i in range(num)]
         b=[random.randint(0,40) for i in range(num)]

         def common_elements(ls1,ls2):
             ls1=list(set(ls1))
             ls=[]
             for l1 in ls1:
                 if l1 in ls2:
                     ls.append(l1)
             ls.sort()
             return ls

         common_elements(a,b)
```

```
Out[27]: [3, 5, 14, 15, 18, 19, 20, 21, 23, 30, 39]
```

2.0.2 Question 2:

Write a python program to find the gravitational force acting between two objects.

$$F = G\frac{m_1 m_2}{r^2}$$

Expected Output:

```
Enter the first mass (m1): 5000000
Enter the second mass (m2): 900000
Enter the distance between the centres of the masses (N): 30
Hence, the Gravitational Force is:  0.33 N
```

```
In [30]: # Solution 2:
         G = 6.66*pow(10, -11)

         def force(m1,m2,r):
             return G*m1*m2/r/r

         force(5000000,900000,30)
```

```
Out[30]: 0.333
```

2.0.3 Question 3:

Write a python program that generates a new list that contains only even elements from the randomly generated list.
Expected Output:

```
Randomly generated list: [64, 63, 90, 13, 38, 27, 19, 51, 97, 32, 18, 75]
List of even elements: [64, 90, 38, 32, 18]
```

```
In [34]: # Solution 3:
         num=random.randint(10,30)
         a=[random.randint(0,100) for i in range(num)]
         print(list(set(a)))
         ls=[]
         for one in set(a):
             if one%2==0:
                 ls.append(one)
         ls
```

```
[2, 5, 8, 80, 49, 84, 90, 59, 29, 94]


Out[34]: [2, 8, 80, 84, 90, 94]
```

2.0.4 Question 4:

Write a python program to check if a substring is present in a given string.
Expected Output:

```
Enter string: Hello world
Enter word: world
Substring in string!
```

```
In [35]: # Solution 4:

         def ifasubstr(string,word):
             if word in string:
                 print('Substring in string!')
             else:
```

```
                print('Substring not in string!')

        ifasubstr('Hello world','world')
```

```
Substring in string!
```

2.0.5 Question 5:

Write a python program that asks the user last 2 digit of (your) student number and generates Fibonacci series.

Expected Output:

```
How many numbers that generates?: 12
Fibonacci series:
 [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144]
```

```
In [37]: # Solution 5:
         num=int(input('How many numbers that generates?: '))
         ls=[]
         for i in range(num):
             if i==0 or i==1:
                 ls.append(1)
             else:
                 ls.append(ls[-1]+ls[-2])
         ls
```

```
How many numbers that generates?: 12
```

```
Out[37]: [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144]
```

2.0.6 Question 6:

Write a python program using function that generates a new list that contains all the elements of the first list and removing all the duplicates.

Expected Output:

```
List:  [1, 2, 3, 4, 3, 2, 1]
Result List using loop:  [1, 2, 3, 4]
Result List using sets:  [1, 2, 3, 4]
```

For extra points: 1. Generate the result using two different functions using: - one using a loop and constructing a list - sets

```
In [40]: # Solution 6:

         def useloop(a):
             ls=[]
             for one in a:
```

```
                if one not in ls:
                    ls.append(one)
            print(ls)
        def useset(a):
            ls=list(set(a))
            print(ls)
        useloop([1, 2, 3, 4, 3, 2, 1])
        useset([1, 2, 3, 4, 3, 2, 1])
```

```
[1, 2, 3, 4]
[1, 2, 3, 4]
```

2.0.7 Question 7:

Write a python program using functions that asks the user for a string containing multiple words and print back to the user the same string, except with the words in reverse order.

Expected Output:

```
Please enter a sentence: My name is Milaan
The reverse sentence is: Milaan is name My
```

```
In [41]: # Solution 7:
         def revthesen(s):
             ls=s.split()
             ls.reverse()
             s=' '.join(ls)
             return s
         revthesen('My name is Milaan')
```

```
Out[41]: 'Milaan is name My'
```

2.0.8 Question 8:

Write a python program using function that encrypts a given input with these steps:

Input: "apple"

- Step 1: Reverse the input: "elppa"
- Step 2: Replace all vowels using the following chart:

```
a => 0
e => 1
i => 2
o => 2
u => 3
# 1lpp0
```

- Step 3: Add "aca" to the end of the word: "1lpp0aca"

Expected Output:

```
Word:  apple
Encrypted word: 1lpp0aca
```

More Examples:

```
encrypt("banana")  "0n0n0baca"
encrypt("karaca")  "0c0r0kaca"
encrypt("burak")  "k0r3baca"
encrypt("alpaca")  "0c0pl0aca"
```

```
In [47]: # Solution 8:
         def encrypt(word):
             s=''
             for i in word:
                 if i=='a':
                     i='0'
                 elif i=='e':
                     i='1'
                 elif i=='i':
                     i='2'
                 elif i=='o':
                     i='2'
                 elif i=='u':
                     i='3'
                 s=i+s
             return s+'aca'
         encrypt('banana')
         encrypt("burak")
```

```
Out[47]: 'k0r3baca'
```

2.0.9 Question 9:

Write a python program using function that takes a number num and returns its length.

Expected Output:

```
Enter number: 963969
Total digits in given number:  6
```

```
In [49]: # Solution 9:
         def len_num(num):
             return len(str(int(num)))

         len_num('963969')
```

```
Out[49]: 6
```

2.0.10 Question 10:

Write a python program using function that takes a string and returns the number (count) of vowels contained within it.

Expected Output:

```
Enter string: Celebration
Total vowels in the string: 5
Identified vowels are:  ['e', 'e', 'a', 'i', 'o']
```

More examples:

```
count_vowels("Palm")  1
count_vowels("Prediction")  4
```

```
In [6]: # Solution 10:
        def vowels(word):
            a=word.count('a')
            e=word.count('e')
            i=word.count('i')
            o=word.count('o')
            u=word.count('u')
            return a+e+i+o+u
        vowels('Prediction')
```

```
Out[6]: 4
```

2.0.11 Question 11:

Write a python program to draw pattern as below:

Expected Output:

```
--- --- ---
|   |   |   |
 --- --- ---
|   |   |   |
 --- --- ---
|   |   |   |
 --- --- ---
```

For extra point: 1. Generate solution by asking the user what size game board they want to draw, and draw it for them to the screen using Python's print statement.

Expected Output:

```
Enter the size of board you want to draw: 4
 --- --- --- ---
|   |   |   |   |
 --- --- --- ---
|   |   |   |   |
 --- --- --- ---
```

```
|   |   |   |   |
 --- --- --- ---
|   |   |   |   |
 --- --- --- ---
```

```
In [13]: # Solution 11:
         num=int(input('Enter the size of board you want to draw:'))
         for i in range(num*2+1):
             for j in range(num):
                 if i%2==0:
                     print(' ---',end='')
                 else:
                     print('|   ',end='')
             if i%2==0:
                 print()
             else:
                 print('|')
```

```
Enter the size of board you want to draw: 4
```

```
 --- --- --- ---
|   |   |   |   |
 --- --- --- ---
|   |   |   |   |
 --- --- --- ---
|   |   |   |   |
 --- --- --- ---
|   |   |   |   |
 --- --- --- ---
```

2.0.12 Question 12:

Write a python program to ask user for a string and then perform following operations: 1. Calculate the num of digits 2. Calculate the num of characters 3. Calculate the num of vowels 4. Calculate the num of lowercase letters 5. replace ' ' with '_' in the string 6. Print and Store the ouput to 'output.txt' file.

Expected Output:

```
Enter string: Hello World 123
Output printed in'output.txt'
```

Expected Output in output.txt:

```
The entered string is: Hello World 123
The number of digits is: 3
The number of characters is: 15
The number of vowels is: 3
The number of lowercase letters is: 8
The modified string is: Hello_World_123
```

```
In [14]: # Solution 12:
         s=input('The entered string is:')
         ch=len(s.strip())
         d=0
         v=0
         l=0
         for a in s:
             if a.isdigit():
                 d=d+1
             if a.islower():
                 l=l+1
             if a in 'aeiou':
                 v=v+1

         with open('output.txt','w') as f:
             f.write('The entered string is: %s\n' % s.strip())
             f.write('The number of digits is: %d\n' % d)
             f.write('The number of characters is: %d\n' % ch)
             f.write('The number of vowels is: %d\n' % v)
             f.write('The number of lowercase letters is: %d\n' % l)
             f.write('The modified string is: %s\n' % s.strip().replace(' ','_'))
         print("Output printed in'output.txt'")
```

```
The entered string is: Hello World 123


Output printed in'output.txt'
```

2.0.13 Question 13:

Write a python program using function that takes as input three variables from user, and returns the largest of the three. Do this without using the Python `max()` function!

Expected Output:

```
Please enter three integers separated by comma: 12, 66, 31
The maximum value is: 66
```

```
In [18]: # Solution 13:

         def maxinthree(a,b,c):
             if a>b:
                 if a>c:
                     return a
                 else:
                     return c
             else:
                 if b>c:
```

```
            return b
        else:
            return c

s=input('Please enter three integers separated by comma:')
ls=s.split(',')
r=maxinthree(int(ls[0]),int(ls[1]),int(ls[2]))
print('The maximum value is: %d'%r)
```

```
Please enter three integers separated by comma: 12, 66, 31


The maximum value is: 66
```

2.0.14 Question 14:

Write a python program where user, will have a number in head between 0 and 100. The program will guess a number, and you, the user, will say whether it is too "high", too "low", or your number. Also, in the end program should print out how many guesses it took to get your number.

Expected Output:

```
Guess a number between 0 and 100 and tell whether high or low when prompted!
My guess is 50. Is that high, low or same? low
My guess is 75. Is that high, low or same? low
My guess is 88. Is that high, low or same? low
My guess is 94. Is that high, low or same? low
My guess is 97. Is that high, low or same? low
My guess is 99. Is that high, low or same? same
Congrats to me! I guessed it in 6 tries.
```

```
In [20]: # Solution 14:
import random
num=random.randint(0,100)
count=0
while True:
    x=int(input('My guess is '))
    count=count+1
    if x==num:
        print('Is that high, low or same? same')
        break
    elif x>num:
        print('Is that high, low or same? high')
    else:
        print('Is that high, low or same? low')
print('Congrats to me! I guessed it in %d tries.'%count)
```

```
My guess is  50
```

```
Is that high, low or same? high

My guess is  30

Is that high, low or same? high

My guess is  15

Is that high, low or same? low

My guess is  20

Is that high, low or same? low

My guess is  25

Is that high, low or same? high

My guess is  23

Is that high, low or same? high

My guess is  21

Is that high, low or same? low

My guess is  22

Is that high, low or same? same
Congrats to me! I guessed it in 8 tries.
```

2.0.15 Question 15:

Write a python program using function that takes an list(ordered) of numbers (from smallest to largest) and another number. The function decides whether or not the given number is inside the list and returns (then prints) an appropriate boolean.

Hint: Use binary search.

Expected Output:

```
List:  [2, 4, 6, 8, 10]
Find  '5': False
Find '10': True
Find '-1': False
Find  '2': True
```

For extra point: 1. Generate list randomly and select he number randomly to be search from the list.

```
In [2]: # Solution 15:
        import random
        num=random.randint(5,15)
        ls=[random.randint(-10,10) for i in range(num)]
        ls=list(set(ls))
        ls.sort()
        print('List: %s' % ls)
        an=[random.randint(-10,10) for i in range(4)]

        def search(ls,a):
            x1=0
            x2=len(ls)
            while True:
                b=ls[int((x1+x2)/2)]
                if a==b:
                    return True
                elif a>b:
                    x1=int((x1+x2)/2)
                else:
                    x2=int((x1+x2)/2)
                if (x1+1)>=x2:
                    return False

        for a in an:
            print("Find  '%d': %s"%(a,search(ls,a)))


List: [-9, -7, -6, -3, -2, 3, 7, 10]
Find  '6': False
Find  '0': False
```

```
Find  '1': False
Find  '3': True
```

2.0.16 Question 16:

Write a python program to generate password. Be creative with how you generate passwords - strong passwords have a mix of lowercase letters, uppercase letters, numbers, and symbols. The passwords should be random, generating a new password every time the user asks for a new password. Include your code in a main method.

Expected Output:

```
Please choose strong or weak:
strong
password: 6|Av.0T^9
do you want a new password? y/n
n
```

For extra points: 1. Ask the user if they want password to be strong(9 characters) or weak(6 characters)?

```
In [9]: # Solution 16:
        import random,string,re
        while True:
            s=input('Please choose strong or weak:')
            num=9 if s=='strong' else 6
            passwd=''
            while True:
                tmp = random.sample(string.ascii_letters + string.digits + '!@#$%^&*()><?', nu
                passwd = ''.join(tmp)
                if re.search('[0-9]', passwd) and re.search('[A-Z]', passwd) and re.search('[a
                    break
            print('password: %s'%passwd)
            s=input('do you want a new password? y/n')
            if s=='n':
                break
```

```
Please choose strong or weak: strong


password: 8ah*>JQFg


do you want a new password? y/n y
Please choose strong or weak: wesk


password: 9@Nkz(
```

```
do you want a new password? y/n n
```

2.0.17 Question 17:

Write a python program using function that picks a random word from a list of words from the **dictionary**. Each line in the file contains a single word.

Hint: use the Python random library for picking a random word.

Expected Output:

```
Random word: POTENTIATING
```

```
In [10]: # Solution 17:
         import random
         with open('sowpods.txt','r') as f:
             data=f.read().split('\n')
         i=random.randint(0,len(data)-1)
         print('Random word: %s'%data[i])
```

```
Random word: ASSUMINGLY
```

2.0.18 Question 18:

Write a python program where a text(.txt) file is given **nameslist.txt** that contains list of a bunch of names, count how many of each name there are in the file, and print out the results to the screen.

Expected Output:

```
{'Darth': 31, 'Luke': 15, 'Leia': 54}
```

For extra point: 1. Instead of using the **nameslist.txt** file from above (or instead of, if you want the challenge), take this **SUN_Database.txt** file, and count how many of each "category" of each image there are. This text file is actually a list of files corresponding to the SUN database scene recognition database, and lists the file directory hierarchy for the images. Once you take a look at the first line or two of the file, it will be clear which part represents the scene category.

Expected Output:

```
abbey: 50
airplane_cabin: 50
airport_terminal: 50
alley: 50
amphitheater: 50
...
...
...
wrestling_ring: 50
yard: 50
youth_hostel: 50
```

In [11]:
```
# Solution 18:

with open('nameslist.txt','r') as f:
    data=f.read().split('\n')
dic={}
for one in data:
    if one in dic:
        dic[one]=dic[one]+1
    else:
        dic[one]=1
print(dic)
```

```
{'Darth': 31, 'Luke': 15, 'Leia': 54}
```

2.0.19 Question 19:

Write a python program where two .txt files are given that have lists of numbers in them, find the numbers that are overlapping. One '**primenumbers1_1000.txt**' file has a list of all prime numbers under 1000, and the other '**happynumbers1_1000.txt**' file has a list of **happy numbers** up to 1000.

Expected Output:

```
The list of overlapping numbers:
 [7, 13, 19, 23, 31, 79, 97, 103, 109, 139, 167, 193, 239, 263, 293, 313, 331, 367, 379, 383, 3
```

For extra point: 1. Generate solution with functions using list comprehensions

In [13]:
```
# Solution 19:

with open('primenumbers1_1000.txt','r') as f:
    data1=f.read().split('\n')
with open('happynumbers1_1000.txt','r') as f:
    data2=f.read().split('\n')
ls=[]
for a in data1:
    if a in data2:
        ls.append(int(a))
print(ls)
```

```
[7, 13, 19, 23, 31, 79, 97, 103, 109, 139, 167, 193, 239, 263, 293, 313, 331, 367, 379, 383, 39
```

2.0.20 Question 20:

Create a function that takes a string as an argument and returns the Morse code equivalent.

For example:

```
encode_morse("HELP ME !")  ".... . .-.. .--.   -- .   -.-.--"
```

Expected Output:

```
Enter a sentence: I love
..   .-.. --- ...- .
Enter morse code: .--. -.-- - .... --- -.
PYTHON
```

This dictionary can be used for coding:

```
char_to_dots = {
    'A': '.-',   'B': '-...', 'C': '-.-.', 'D': '-..',  'E': '.',    'F': '..-.',
    'G': '--.',  'H': '....', 'I': '..',   'J': '.---', 'K': '-.-',  'L': '.-..',
    'M': '--',   'N': '-.',   'O': '---',  'P': '.--.', 'Q': '--.-', 'R': '.-.',
    'S': '...',  'T': '-',    'U': '..-',  'V': '...-', 'W': '.--',  'X': '-..-',
    'Y': '-.--', 'Z': '--..',

    '0': '-----', '1': '.----', '2': '..---', '3': '...--', '4': '....-',
    '5': '.....', '6': '-....', '7': '--...', '8': '---..', '9': '----.',

    ' ': ' ',      '&': '.-...',  "'": '.----.', '@': '.--.-.', ')': '-.--.-',
    '(': '-.--.',  ':': '---...', ',': '--..--', '=': '-...-',  '!': '-.-.--',
    '.': '.-.-.-', '-': '-....-', '+': '.-.-.', '"': '.-..-.',  '?': '..--..',
    '/': '-..-.'
}
```

```
In [20]: # Solution 20:
         char_to_dots = {
             'A': '.-',   'B': '-...', 'C': '-.-.', 'D': '-..',  'E': '.',    'F': '..-.',
             'G': '--.',  'H': '....', 'I': '..',   'J': '.---', 'K': '-.-',  'L': '.-..',
             'M': '--',   'N': '-.',   'O': '---',  'P': '.--.', 'Q': '--.-', 'R': '.-.',
             'S': '...',  'T': '-',    'U': '..-',  'V': '...-', 'W': '.--',  'X': '-..-',
             'Y': '-.--', 'Z': '--..',

             '0': '-----', '1': '.----', '2': '..---', '3': '...--', '4': '....-',
             '5': '.....', '6': '-....', '7': '--...', '8': '---..', '9': '----.',

             ' ': ' ',      '&': '.-...',  "'": '.----.', '@': '.--.-.', ')': '-.--.-',
             '(': '-.--.',  ':': '---...', ',': '--..--', '=': '-...-',  '!': '-.-.--',
             '.': '.-.-.-', '-': '-....-', '+': '.-.-.', '"': '.-..-.',  '?': '..--..',
             '/': '-..-.'
         }

         def encode_morse(s):
             r=''
             s=s.upper()
             for a in s:
                 r=r+char_to_dots[a]+' '
             return r

         def decode_morse(s):
```

```
            r=''
            s=s.split()
            for a in s:
                for key,value in char_to_dots.items():
                    if a==value:
                        r=r+key
            return r

        print(encode_morse('I love'))
        print(decode_morse('.--. -.-- - .... --- -.'))
```

```
..   .-.. --- ...- .
PYTHON
```

In [ ]: