Final Assignment Report

JILIN UNIVERSITY OF FINANCE AND ECONOMICS

Department of College of Managment Science and Information Engineering

BSc in Data Science and Big Data Technology

(2021)

Final Assignment: Part 01

21/06/2021

MODULE: Data Mining
Major: Information management and information system
Class number:1854
Submitted by: Hedongxu(何东旭) 0314021805409 (1854)
QQ: 1455536901
Github ID: 520Degree

Instructions:

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. **Accept this assignment** from the **Github Clasroom link** This will create private repository of the assignment in your Github account.
4. In your repository `Clone` -> `Download ZIP` in your computer.
5. Change your: **Major**, **Name**, **Student number**, **Class number**, **QQ number** and **GitHub ID**
6. Once you finish the Assignment **convert your .ipynb file into PDF** (both .pynb and .pdf file will be required!)
7. Create Folder name "**Solution**" and copy your 3 files:
   A. Your Jupyter Notebook file (**.ipynb**).
   B. Your PDF converted file (**.pdf**).
   C. **.zip** file containing both .ipynb and .pdf files and name your .zip file as your student number and name. For example: **0318021907632 Milan(米兰).zip**
8. Finally, in your repository `Add files` -> `upload files` upload the "**Solution**" folder and `Commit changes`.

Python Assignment 01

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 [1]:
```python
# Solution 1:
List1=[0, 2, 4, 6, 12, 13, 14, 18, 20, 24, 25, 26, 27]
List2= [0, 4, 7, 9, 10, 11, 13, 14, 17, 18, 20, 33, 39]
a=[x for x in List1 if x in List2]
a
```

Out[1]:
```
[0, 4, 13, 14, 18, 20]
```

Question 2:

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

\begin{equation*} F=G\frac{{m_1}{m_2}}{{r^2}} \end{equation*}
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 [1]:
```python
# Solution 2:
G = 6.66*pow(10, -11)
m1 =float (input())
print("Enter the first mass (m1):",m1)
m2 = float(input())
print("Enter the first mass (m2):",m2)
r = float(input ("enter the distance (r):"))
f=(G*m1*m2)/(r**2)
print("Hence, the Gravitational Force is:",f)
```

```
5000000
Enter the first mass (m1): 5000000.0
900000
Enter the first mass (m2): 900000.0
enter the distance (r):30
Hence, the Gravitational Force is: 0.33299999999999996
```

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 [2]:
```python
# Solution 3:
list = [1,2,3,4,5,6,7,8]
print([li for li in list if list.index(li) % 2 == 1])
```

```
[2, 4, 6, 8]
```

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 [4]:
```python
# Solution 4:
def check(string, sub_str):
    if (string.find(sub_str) == -1):
        print("NO")
    else:
        print("Substring in string!")
string = "Hellow world"
sub_str ="world"
check(string, sub_str)
```

```
Substring in string!
```

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 [2]:
```python
# Solution 5:
student_id = input('please enter your student id of last 2 digit:')
print( 'How many numbers that generates?: {}.'.format(
    student_id))
def fibo(n):
    if n<2:
        return 1
    return fibo(n-1) + fibo(n-2)
for i in range(12):
    print('Fibonacci series:',fibo(i))
```

```
please enter your student id of last 2 digit:12
How many numbers that generates?: 12.
Fibonacci series: 1
Fibonacci series: 1
Fibonacci series: 2
Fibonacci series: 3
Fibonacci series: 5
Fibonacci series: 8
Fibonacci series: 13
Fibonacci series: 21
Fibonacci series: 34
Fibonacci series: 55
Fibonacci series: 89
Fibonacci series: 144
```

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 [3]:
```python
# Solution 6:
import itertools
num = [1,2,3,4,3,2,1]
num.sort()
print(list(num for num,_ in itertools.groupby(num)))
```

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

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 [5]:
```python
# Solution 7:
seq1 = "My name is Milaan"
seq2 = "Milaan is name My"
res = []
for x in seq1:
  if x in seq2:
    res.append(x)
print (res[::-1])
```

```
['n', 'a', 'a', 'l', 'i', 'M', ' ', 's', 'i', ' ', 'e', 'm', 'a', 'n', ' ', 'y', 'M']
```

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 [8]:
```python
# Solution 8:
n=input("word: ")
def reverse():
    return n[::-1]
a=reverse()
y=""
for z in a:
    if(z=='a'):
        y+='0'
    elif( z=='e'):
        y+='1'
    elif(z=='i'):
        y+='2'
    elif( z=='o'):
        y+='2'
    elif( z=='u'):
        y+='3'
    else:
        y+=z
print(y)
y2="aca"
print("Encrypted word: ",y+y2)
```

```
word: apple
1lpp0
Encrypted word:  1lpp0aca
```

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 [9]:
```python
# Solution 9:
str = input("Enter number:");
print("Total digits in given number:",len(str))
```

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

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 [11]:
```python
# Solution 10:
n=input("Enter string: ")
m=""
count=0
list=[]
for b in n:
    if(b=='a' or b=='A' or b=='E' or b=='I' or b=='O' or b=='U' or b=='a' or b=='e' or b=='i' or b=='o' or b=='
        count+=1
        list.append(b)
print("Total vowels in the string:",count)
print("Identified vowels are:",list)
```

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

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 [12]:
```python
# Solution 11:
def x(size):
    y=1
    while y<=size:
        draw(size)
        y+=1
    return
def draw(size):
    z=1
    a=1
    while z<=size:
        print ("--- ",end='')
        z+=1
    print()
    while a<=size+1:
        print ("|   ",end='')
        a+=1
    print()
    return
size=int(input("Enter the size of board you want to draw:"))
x(size)
for z in range(size):
    print ("--- ",end='')
```

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

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 [1]:
```python
# Solution 12:
import sys
str=input("The entered string is: ")
num=0
count=0
lower=0
for x in str:
    if x.isdigit():
        num = num + 1
for x in str:
    if(x=='A' or x=='E' or x=='I' or x=='O' or x=='U'):
        count+=1
for x in str:
    if x.islower():
        lower+= 1
sys.stdout = open("output.txt",'w')
print("The number of digits is:",num)
print("The number of characters is:",len(str))
print("The number of vowels is: ",count)
print("The number of lowercase letters is: ",lower)
print("The modified string is:",str.replace(" ", "_"))
```

```
The entered string is: Hello World 123
```

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 [5]:
```python
# Solution 13:
def Maxnum():
    data=input("Please enter three integers separated by comma:")
    num=data.split(",")
    Max=1
    for X in num:
        if int(X)>=Max:
            Max=int(X)
    return Max
print("The maximum value is:",Maxnum())
```

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

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 [4]:
```python
# Solution 14:
import random
def guess_number():
    true_num = random.randint(0, 100)
    user_num = int(input("请输入一个整数:"))
    count = 1
    while true_num != user_num:
        if true_num > user_num:
            print("low! ")
        elif true_num < user_num:
            print("high! ")
        elif true_num == user_num:
            print("high! ")
        count += 1
        user_num = int(input("请输入一个整数: "))
        print("Congrats to me! I guessed it in,", count,"tries.")
guess_number()
```

```
请输入一个整数:50
high!
请输入一个整数: 40
Congrats to me! I guessed it in, 2 tries.
high!
请输入一个整数: 20
Congrats to me! I guessed it in, 3 tries.
high!
请输入一个整数: 10
Congrats to me! I guessed it in, 4 tries.
low!
请输入一个整数: 15
Congrats to me! I guessed it in, 5 tries.
low!
请输入一个整数: 18
Congrats to me! I guessed it in, 6 tries.
```

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 [3]:
```python
# Solution 15:
test_list = [ 2,4,6,8,10 ]
k=input()
def f(k):
    for i in test_list:
        if int(i)== int(k):
            print ("true")
            break
    else:
        print("false")
f(k)
```

```
2
true
```

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 [ ]:
```python
# Solution 16:
import random

flag = 0
while flag == 0:
    choose = input('Please choose strong or weak:')
    if choose == 'strong':
        print('password:',
              ''.join(random.sample("1234567890abcdefghijklmnopqrstuvwxyz!@#$%^&*()QWERTYUIOPASDFGHJKLZXCVBNM",
    if choose == 'weak':
        print('password:', ''.join(random.sample("1234567890abcdefghijklmnopqrstuvwxyz!@#$%^&*()", 5)))
    print('do you want a new password? y/n')
    n = input()
    if n == 'n':
        flag = 1
```

```
Please choose strong or weak:strong
6|Av.0T^9
Please choose strong or weak:
```

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 [2]:
```python
# Solution 17:
import random
def choice():
    word = random.choice(words.read().split())
    print(word)
choice()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-a4d728f50846> in <module>
      4     word = random.choice(words.read().split())
      5     print(word)
----> 6 choice()

<ipython-input-2-a4d728f50846> in choice()
      2 import random
      3 def choice():
----> 4     word = random.choice(words.read().split())
      5     print(word)
      6 choice()

NameError: name 'words' is not defined
```

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 [1]:
```python
# Solution 18:
def all_list(list):
    result = {}
    for i in set(list):
        result[i] = list.count(i)
    return result
list=(data.read()).split()
all_list(list)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-4f8bb4b389cf> in <module>
      5         result[i] = list.count(i)
      6     return result
----> 7 list=(data.read()).split()
      8 all_list(list)

NameError: name 'data' is not defined
```

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, 397, 409,
487, 563, 617, 653, 673, 683, 709, 739, 761, 863, 881, 907, 937]
```

For extra point:

1. Generate solution with functions using list comprehensions

In [2]:
```python
# Solution 19:
def all_list(list):
    result = {}
    for i in set(list):
        result[i] = list.count(i)
    return result
list=(data.read()).split()
all_list(list)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-862d42519fce> in <module>
      5         result[i] = list.count(i)
      6     return result
----> 7 list=(data.read()).split()
      8 all_list(list)

NameError: name 'data' is not defined
```

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 [7]:
```python
# 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 morse(word, char_to_dots):
    mode = ''
    index = 0
    for i in word:
        index += 1
        str_mode = char_to_dots[i]
        if index == len(word):
            mode += str_mode
        else:
            mode += str_mode + ' '
    return mode


word = input('Enter a sentence: ')
word = word.upper()
print(morse(word, char_to_dots))
b = input('Enter morse code: ')
b_list = b.split(' ')
word = ''
for j in b_list:
    word += list(char_to_dots.keys())[list(char_to_dots.values()).index(j)][0]
print(word)
```

```
Enter a sentence: HELP ME !
Enter morse code: ..   .-.. --- ...- .
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-7-3ea45d21be11> in <module>
     37 word = ''
     38 for j in b_list:
---> 39     word += list(char_to_dots.keys())[list(char_to_dots.values()).index(j)][0]
     40 print(word)
     41

ValueError: '' is not in list
```

In [ ]:

In [ ]: