

JLUFE

Fall

2021(Sep-Jan)

Final Assignment Report

JILIN UNIVERSITY OF FINANCE AND ECONOMICS

College of Managment Science and Information Engineering

BSc in Data Science and Big Data Technology

(2021)

MODULE: Intelligent Technology

Final Assignment

20/12/2021

Submitted by:

Milan(米兰) 0318021907632 (2005)

QQ: 3086215265 | Github ID: milaan9

Instructions:

1. I have added tips and required learning resources for each question, which helps you to solve the problems.
2. Finish the assignment on your **OWN**. **Any student find copying/sharing from classmates or internet will get '0' points!!!**
3. from → [GitHub Classroom link \(https://classroom.github.com/a/KX_0SzHL\)](https://classroom.github.com/a/KX_0SzHL).
This will create private repository of the assignment in your GitHub Clasroom account.
4. In your repository → 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](https://github.com/milaan9/91_Python_Mini_Projects/tree/main/001_Convert_IPython_to_PDF)
(https://github.com/milaan9/91_Python_Mini_Projects/tree/main/001_Convert_IPython_to_PDF)
(both **.ipynb** and **.pdf** file will be required!)
7. To submit your assignment, go to GitHub Classrom repository and → →
 - A. Replace the question (**.ipynb**) file with your solution (**.ipynb**) file.
 - B. Also, upload (**.pdf**) converted file of your solution (**.ipynb**) file.

Python Final Assignment

Question 1:

Write a python program to create a 3x3 matrix with values ranging from 12 to 20

Expected Output:

```
[[12 13 14]
 [15 16 17]
 [18 19 20]]
```

In [77]:

```
# Solution 1
import numpy
matrix = numpy.array([[12, 13, 14], [15, 16, 17], [18, 19, 20]])
matrix
```

Out[77]:

```
array([[12, 13, 14],
       [15, 16, 17],
       [18, 19, 20]])
```

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): 7000000
Enter the second mass (m2): 600000
Enter the distance between the centres of the masses (N): 16
Hence, the Gravitational Force is: 1.09 N
```

In [78]:

```
# Solution 2
m1 = input("Enter the first mass (m1):")
m2 = input("Enter the second mass (m2):")
N = input("Enter the distance between the centres of the masses (N): ")
Force = eval(f"(667*{m1}*{m2})/({N}*{N}))/10000000000000")
Force = "%0.2f N"%Force
Force
```

Traceback (most recent call last):

```
File "C:\Users\Dream\AppData\Local\Programs\Python\Python37\lib\site-packages\IPython\core\interactiveshell.py", line 3418, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)
```

```
File "<ipython-input-78-f94e37e52d7c>", line 5, in <module>
    Force = eval(f"(667*{m1}*{m2})/({N}*{N}))/10000000000000")
```

```
File "<string>", line 1
    (667**/(*) )/100000000000000
    ^
```

SyntaxError: invalid syntax

Question 3:

Write a python program to create a 9x9 array with random values and find the minimum and maximum values

💡 hint: min, max

In [79]:

```
# Solution 3
import numpy
matrix=numpy.random.randint(0,10,size=[9,9])
matrix
```

Out[79]:

```
array([[3, 8, 5, 0, 2, 2, 5, 3, 2],
       [7, 8, 2, 0, 5, 7, 8, 3, 5],
       [5, 0, 8, 7, 7, 3, 3, 0, 4],
       [2, 3, 2, 5, 7, 3, 3, 3, 9],
       [3, 9, 0, 7, 0, 9, 3, 3, 0],
       [1, 3, 6, 2, 8, 5, 7, 0, 4],
       [3, 4, 3, 3, 1, 2, 7, 3, 3],
       [3, 1, 5, 6, 7, 1, 8, 3, 8],
       [2, 2, 9, 9, 8, 4, 7, 3, 9]])
```

Question 4:

Write a python program that generates a list containing only common elements between the two randomly generated 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,]

List of common elements are: [0, 4, 13, 14, 18, 20]

In [80]:

```
# Solution 4

import random
list_1 = []
for i in range(13):
    list_1.append(random.randint(1,50) )
list_2 = []
for i in range(11):
    list_2.append(random.randint(1,50))
print(list_1)
print(list_2)

list_3=[]
for item in list_1:
    if item in list_2:
        list_3.append(item)
print(list_3)
```

Question 5:

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

Expected Output:

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

In [81]:

```
# Solution 5
student = int(input("Last two numbers of my student number:"))
numbers = int(input("How many numbers that generates?:"))
n = 0
a = 0
b = 1
Fibonacci_series=[]
while n < numbers :
    Fibonacci_series.append(b)
    a, b = b, a + b
    n = n + 1
Fibonacci_series
```

Question 6:

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 [82]:

```

# Solution 6
str1=''' -----
|   |   |   |   |
-----
|   |   |   |   |
-----
|   |   |   |   |
-----
|   |   |   |   |
-----'''
str1.split('\n')
[print(i) for i in str1]

```

Question 7:

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 [83]:

```
# Solution 7
vowels = ['e', 'e', 'a', 'i', 'o']
cun=0
str1 = input()
for s in str1:
    if s in vowels:
        cun=cun+1
cun
```

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 [84]:

```
# Solution 8
str1=input("input_str:")
one=str1[::-1]
zifudict={"a":0, "e": 1, "i": 2, "o":3, "u": 4,}
str2=''
for i in one:
    if i in zifudict:
        str2+=str(zifudict[i])
    else:
        str2+=i
str2+='aca'
str2
```

Question 9:

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: 3SPk|0}y
do you want a new password? y/n
n
```

In [85]:

```
# Solution 9
import random
import string
def defmima(strong):
    miam=''
    str_source = string.ascii_letters + string.digits + string.punctuation
    str_list = random.sample(str_source, strong)
    if str_list not in ['/', '\\', '@', 'l', '1', 'o', '0']:
        str_final = ''.join(str_list)
        miam+=str_final
    return miam
while True:
    mima=input("Please choose strong or weak:")
    if mima=='strong':
        m=defmima(10)
        print(f"password: {m}")
        input("do you want a new password? y/n:")
        break
    elif mima=='weak':
        m=defmima(6)
        print(f"password: {m}")
        input("do you want a new password? y/n:")
        break
    else:
        print("Please re-enter strong or weak")
```

Question 10:

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: Good Morning 369
Output printed in 'output.txt'

Expected Output in file output.txt:

The entered string is: Good Morning 369
The number of digits is: 3
The number of characters is: 16
The number of vowels is: 4
The number of lowercase letters is: 9
The modified string is: Good_Morning_369

In [86]:

```
# Solution 10
# Solution 10
str1 = input("input_str:")
digits = 0
vowels = 0
lowercase = 0
for i in str1:
    if i.isdigit():
        digits += 1
    elif i >= "a" and i <= "z":
        vowels += 1
    elif i >= "A" and i <= "Z":
        lowercase += 1
s1=f"The entered string is: {str1}"
s2=f"The number of digits is: {digits}"
s3=f"The number of characters is: {len(str1)}"
s4=f"The number of vowels is: {lowercase}"
s5=f"The number of lowercase letters is: {vowels}"
s6=f"The modified string is: {str1.replace(' ', '_')}"
shu=s1+'\n'+s2+'\n'+s3+'\n'+s4+'\n'+s5+'\n'+s6
print(shu)
with open('./output.txt', 'w', encoding='utf-8') as f:
    f.write(shu)
```

Question 11:

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. Refer the Expected output for example.

Expected Output:


```
WELCOME TO GUESS ME!
Let me select a number between 1 and 100
If your guess is more than 10 away from my number, I'll tell you you're COLD
If your guess is within 10 of my number, I'll tell you you're WARM
If your guess is farther than your most recent guess, I'll say you're getting COLDER
If your guess is closer than your most recent guess, I'll say you're getting WARMER
LET'S PLAY!
Make your guess
5
WARMER!
Make your guess
9
WARM!
Make your guess
3
WARM!
Make your guess
4
WARM!
Make your guess
6
CONGRATULATIONS!, you guessed the correct answer in 5 attempts.
```

In [87]:

```
# Solution 11

# Solution 11
import random
print('''WELCOME TO GUESS ME!
Let me select a number between 1 and 100
If your guess is more than 10 away from my number, I'll tell you you're COLD
If your guess is within 10 of my number, I'll tell you you're WARM
If your guess is farther than your most recent guess, I'll say you're getting COLDER
If your guess is closer than your most recent guess, I'll say you're getting WARMER
LET'S PLAY!''')

randomshu = random.randint(1,100)
cishu=0
while True:
    cishu+=1
    inputshu = int(input("Make your guess:"))
    if inputshu==randomshu:
        break
    elif inputshu > randomshu+10 or inputshu < randomshu-10:
        print("COLD!")
        print("you're getting COLDER")
    elif randomshu-10 < inputshu < randomshu+10:
        print("WARM!")
        print("you're getting WARM!")

if inputshu == randomshu:
    print(f"CONGRATULATIONS!, you guessed the correct answer in {cishu} attempts.")
```

Question 12:

Write a python program using function that picks a random word from a list of words from the [dictionary \(https://github.com/milaan9/JLUFE_Intelligent_Tech_2005-2006/blob/main/sowpods.txt\)](https://github.com/milaan9/JLUFE_Intelligent_Tech_2005-2006/blob/main/sowpods.txt). Each line in the file contains a single word.

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

Expected Output:

Random word: GRAYNESSES

In [88]:

```
# Solution 12
import requests
import random
response = requests.get('https://raw.githubusercontent.com/milaan9/JLUFE_Intelligent_Tech_2005-2006/
list1=response.text.split('\n')
print(random.choice(list1))
```

Question 13:

Write a python program where a text(.txt) file is given [nameslist.txt \(https://github.com/milaan9/JLUFE_Intelligent_Tech_2005-2006/blob/main/nameslist.txt\)](https://github.com/milaan9/JLUFE_Intelligent_Tech_2005-2006/blob/main/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 }

In [89]:

```
# Solution 13
import requests
response = requests.get('https://raw.githubusercontent.com/milaan9/JLUFE_Intelligent_Tech_2005-2006/
list1=response.text.split('\n')
jishudict={}
for i in list1:
    try:
        jishudict[i]+=1
    except:
        jishudict[i]=1
print(jishudict)
```

Question 14:

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 \(https://github.com/milaan9/JLUFE_Intelligent_Tech_2005-2006/blob/main/primenumbers1_1000.txt\)](https://github.com/milaan9/JLUFE_Intelligent_Tech_2005-2006/blob/main/primenumbers1_1000.txt)' file

has a list of all prime numbers under 1000, and the other '[happynumbers1_1000.txt](https://github.com/milaan9/JLUFETech2005-2006/blob/main/happynumbers1_1000.txt)' file (https://github.com/milaan9/JLUFETech2005-2006/blob/main/happynumbers1_1000.txt) file has a list of [happy numbers](https://en.wikipedia.org/wiki/Happy_number) (https://en.wikipedia.org/wiki/Happy_number) 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]
```

In [90]:

```
# Solution 14
import requests
primenumbers1_1000 = requests.get('https://raw.githubusercontent.com/milaan9/JLUFETech2005-2006/blob/main/happynumbers1_1000.txt')
happynumbers1_1000 = requests.get('https://raw.githubusercontent.com/milaan9/JLUFETech2005-2006/blob/main/happynumbers1_1000.txt')
primenumbers1_1000_list=primenumbers1_1000.text.split('\n')
happynumbers1_1000_list=happynumbers1_1000.text.split('\n')
shuzilist=[i for i in primenumbers1_1000_list if i in happynumbers1_1000_list]
print(shuzilist)
```

Question 15:

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 [91]:

```

# Solution 15
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': '----.',

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

str1=input('Enter a sentence:')
str2=''
for i in str1:
    str2+=char_to_dots[i.upper()]
print(str2)

```

In []: