

▼ CIND830 - Python Programming for Data Science

Assignment 1 (10% of the final grade)

Due on February 07, 2022 11:59 PM

This is a Jupyter Notebook document that extends a simple formatting syntax for authoring HTML and PDF. Review [this](#) website for more details on using Jupyter Notebooks.

Consider using a Jupyter Notebook platform to complete this assignment. Ensure using **Python 3.7** release or higher then complete the assignment by inserting your Python code wherever seeing the string `#INSERT YOUR ANSWER HERE`.

When you click the `File` button, from the top navigation bar, then select `Export Notebook to HTML`, an HTML document will be generated that includes both the assignment content and the output of any embedded Python code chunks.

Use [these](#) guidelines to submit **both** the IPYNB and the exported file (HTML). Failing to submit both files will be subject to mark deduction.

Please be advised that you cannot get more than 100% in this assignment, and the **BONUS** question (if there is any) will only be graded if all other questions have been submitted.

▼ Question 1 [25 pts]:

Write a python program to display all prime numbers within a given range

a) [5 pts] Take inputs from the user

- Take 2 inputs, a lower bound and an upper bound from the user

```
lowBound = int(input("Enter the lower bound:\n"))
upBound = int(input("Enter the upper bound:\n"))
```

```
Enter the lower bound:
2
Enter the upper bound:
10
```

b) [10 pts] Implement a solution using count-controlled loops (for loops)

- For the count-controlled loops, you can use the `range()` function to generate a sequence of numbers

```
for counter in range(lowBound, upBound + 1):
    prime = True
    for factor in range(2, counter):
        if counter % factor == 0:
            prime = False
    if prime:
        print(counter)

2
3
5
7
```

c) [10 pts] Implement a solution using condition-controlled loops (while loops)

```
counter = lowBound

while counter <= upBound:
    factor = 2
    prime = True
    while factor <= counter - 1:
        if counter % factor == 0:
            prime = False
        factor += 1
    if prime:
        print(counter)
    counter += 1

2
3
5
7
```

▼ **Question 2 [25 pts]:**

Write a python program that takes two numeric inputs x and y .

a) [5 pts] The program should take numeric inputs continuously.

b) [10 pts] The program should ensure the range of x and y is between 0 and 1. If either one of the input values is out of range, smaller than 0 or greater than 1, then the program should quit.

c) [10 pts] Implement XOR gate, where the program returns 1 if both x and y values are different; otherwise, it returns 0.

The program should work for floating-point values.

For example,

If x = .3 and y = .3, then xor output should be 0

If x = .6 and y = .3, then xor output should be 1

If x = .3 and y = .6, then xor output should be 1

If x = .6 and y = .6, then xor output should be 0

```
xcheck = True
ycheck = True

x = float(input("Enter first num between 0 and 1: "))
if (x < 0 or x > 1):
    print("Error, input value is not between 0 and 1.")
    xcheck = False

y = float(input("Enter second num between 0 and 1: "))
if (y < 0 or y > 1):
    print("Error, input value is not between 0 and 1.")
    ycheck = False

if xcheck == False or ycheck == False:
    print("No XOR output value.")
else:
    if(x != y):
        xor = 1
    else:
        xor = 0
    print("XOR output: ", xor)

Enter first num between 0 and 1: .2
Enter second num between 0 and 1: .3
XOR output: 1
```

▼ Question 3 [25 pts]:

a) [5 pts] Write a Python program to accept an integer and display its binary and octal equivalent. For example, if the user enters 5 then the output should be:

The binary equivalent of 5 is 0b101 The octal equivalent of 5 is 0o5

```
num = int(input("Enter an integer:\n"))

numBin = bin(num)
numOct = oct(num)
print("The binary equivalent of", num, "is", numBin)
print("The octal equivalent of", num, "is", numOct)
```

```
Enter an integer:
-5
The binary equivalent of -5 is -0b101
The octal equivalent of -5 is -0o5
```

b) [10 pts] Write a Python program to calculate the distance between two given points whose coordinates are (x_1, y_1) and (x_2, y_2)

according to the following formula $Distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

For example, if the user enters 2, 4 and 4, 8 then the output should be:

The distance between [2.0, 4.0] and [4.0, 8.0] is 4.47213595499958

```
x1 = float(input("Enter x-coordinate of the first point: "))
y1 = float(input("Enter y-coordinate of the first point: "))
x2 = float(input("Enter x-coordinate of the second point: "))
y2 = float(input("Enter y-coordinate of the second point: "))

dist = ((x2-x1)**2+(y2-y1)**2)**(1/2)

print("The distance between", "[", x1, ",", y1, "]" and "[", x2, ",", y2, "]" is", dist)
```

```
Enter x-coordinate of the first point: 10
Enter y-coordinate of the first point: 15
Enter x-coordinate of the second point: 4
Enter y-coordinate of the second point: 2
The distance between [ 10.0 , 15.0 ] and [ 4.0 , 2.0 ] is 14.317821063276353
```

c) [10 pts] Write a Python program to accept an integer number and display the digits at the ones, tens and hundreds place of that number.

For example, if the user enters 86421 then the output should be:

```
The digit at the ones place of 86421 is 1
The digit at the tens place of 86421 is 2
The digit at the hundreds place of 86421 is 4
```

```
num = int(input("Enter an integer:\n"))
```

```

ones = abs(num) % 10
tens = abs(num) // 10 % 10
hundreds = abs(num) // 100 % 10
print("The digit at the ones place of", num, "is", ones)
print("The digit at the tens place of", num, "is", tens)
print("The digit at the hundreds place of", num, "is", hundreds)

```

Enter an integer:

-4456

The digit at the ones place of -4456 is 6

The digit at the tens place of -4456 is 5

The digit at the hundreds place of -4456 is 4

▼ Question 4 [25 pts]:

a) [10 pts] Write a code that creates a set of different passwords. Each password is a combination of a random adjective, noun, two digits and two punctuation symbols.

For example, if the user asks for generating 4 passwords, then the code might generate the following list:

Password
supershine17*-
cheerfulmeet78~.
elitemango32&,
glossytrain14\$}

Hints:

1. You can use the [wonderwords](#) library to generate random words of the english language.
2. You can use the "functions for sequences" methods from the [random](#) module.

```
num = int(input("How many passwords to generate? "))
```

```
import wonderwords
```

```
import random
```

```
import string
```

```
from wonderwords import RandomWord
```

```
r = RandomWord()
```

```
for num in range(num):
```

```
    adj = r.word(include_parts_of_speech=["adjectives"])
```

```
    noun = r.word(include_parts_of_speech=["nouns"])
```

```
    int2 = random.choice(range(10,100))
```

```
    sym2 = ''.join(random.choice(string.punctuation) for i in range(2))
```

```
    print(adj, noun, int2, sym2, sep='')
```

How many passwords to generate? 7

```

crabbybeaver33+^
allegedcascade86+`
garruloustremor46,)
divergentcure52({
perfectclue52@<
statuesquechub91{{
evanescentpumpkin89!%

```

b) [10 pts] Count the characters of each generated password in Q4.a, then print the password along with its number of characters.

For example, if the user enters 4, then the output might be as follows:

Password	Length
supershine17*-	14
cheerfulmeet78~.	16
elitemango32&,	14
glossytrain14\$}	15

```

num = int(input("How many passwords to generate? "))

import wonderwords
import random
import string
from wonderwords import RandomWord
r = RandomWord()

for num in range(num):
    adj = r.word(include_parts_of_speech=["adjectives"])
    noun = r.word(include_parts_of_speech=["nouns"])
    int2 = random.choice(range(10,100))
    sym2 = ''.join(random.choice(string.punctuation) for i in range(2))
    password = adj + noun + str(int2) + sym2
    print(password, "\t", len(password))

How many passwords to generate? 5
hugechairlift80$~      17
raresympathy18!{      16
fertileflour54[?      16
verseddearest25=`      17
talentedread30<,      16

```

c) [5 pts] Write a code to print only the passwords with more than 14 characters and include either the letter 'o', the digit '3', or the punctuation symbol '~'.

For example, the code would filter the 4 passwords listed in 4.a, and print the following output:

Password	Length
cheerfulmeet78~.	16

Password	Length
----------	--------

```

num = int(input("How many passwords to generate? "))

import wonderwords
import random
import string
from wonderwords import RandomWord
r = RandomWord()

for num in range(num):
    adj = r.word(include_parts_of_speech=["adjectives"])
    noun = r.word(include_parts_of_speech=["nouns"])
    int2 = random.choice(range(10,100))
    sym2 = ''.join(random.choice(string.punctuation) for i in range(2))
    password = adj + noun + str(int2) + sym2

    if(len(password) > 14):
        if("o" in password or "3" in password):
            print(password, "\t", len(password))
        elif("~" in password):
            print(password, "\t", len(password))

How many passwords to generate? 12
apatheticsunbonnet48'" 22
brashhydrofoil64%_ 18
lovelyconcentration25"@ 23
understoodsparrow56>{ 21
massivetattler43\[ 18
pumpedrations27\< 17
fragilegossip35`@ 17
amusedrocket-ship10<= 21

```

BONUS [5 pts] Design an encoder that encrypts the passwords generated in Q4.a. by changing their letters into the octal form.

For example, if the user enters 4, then the output might be as follows

Password	Encryption
supershine17*-	0o1630o1650o1600o1450o1620o1630o1500o1510o1560o1450o610o670o520o55
cheerfulmeet78~.	0o1430o1500o1450o1450o1620o1460o1650o1540o1550o1450o1450o1640o670o700o1760o56
elitemango32&.	0o1450o1540o1510o1640o1450o1550o1410o1560o1470o1570o630o620o460o54
glossytrain14\$}	0o1470o1540o1570o1630o1630o1710o1640o1620o1410o1510o1560o610o640o440o175

```

num = int(input("How many passwords to generate? "))

import wonderwords
import random
import string
from wonderwords import RandomWord

```

```
r = RandomWord()

for num in range(num):
    adj = r.word(include_parts_of_speech=["adjectives"])
    noun = r.word(include_parts_of_speech=["nouns"])
    int2 = random.choice(range(10,100))
    sym2 = ''.join(random.choice(string.punctuation) for i in range(2))
    password = adj + noun + str(int2) + sym2

    octpw = []
    for char in password:
        octpw.append(oct(ord(char)))

    octpw = ''.join([str(octchar) for octchar in octpw])

    print(password, "\t", octpw)

How many passwords to generate? 3
ruralseal45}-      0o1620o1650o1620o1410o1540o1630o1450o1410o1540o640o650o1750o55
etherealmenorah34?.      0o1450o1640o1500o1450o1620o1450o1410o1540o1550o1450o1560o1570o
naughtywaste58-$      0o1560o1410o1650o1470o1500o1640o1710o1670o1410o1630o1640o1450o
```

This is the end of assignment 1

