

DATA 690 Homework 1 (10 points - Due on Sunday, September 18, 2022 by 11:00 pm ET)

The output of this assignment for submission should be in PDF format **AND** .py or .ipynb. The name of the file should be as follows: Lastname_Firstname_Homework1.pdf (example: Thomas_Sunela_Homework1.pdf) **AND** Lastname_Firstname_Homework1.ipynb (example: Thomas_Sunela_Assignment1.ipynb. In short, you are submitting the python notebook as well as the pdf of that notebook. Do **NOT** submit .html file, the system will give you an error.

Incorrect file name will cost you points!

Instructions for converting a Jupyter Python notebook to PDF: Go to the menu and choose, File --> Download As --> html. Open that html file and print it to PDF. Submit the PDF file **NOT** the html file.

If you are using Google Colab, remember to review the PDF before submitting to ensure that all cells and answers are displayed in the PDF.

Things to note:

- Each cell should display an output
- Use only the basic Python concepts and methods
- Use both Markdown and code comments in the Jupyter Notebook as needed

Problem 1

Calculate the quantity: 7^{19} and assign this value to $prob_1$.

```
In [1]: # Answer
prob_1 = 7 ** 19
prob_1
```

Out[1]: 11398895185373143

Problem 2

Find the remainder of 745463478 divided by 234 using the modulo operator and assign this value to $prob_2$

```
In [2]: # Answer
prob_2 = 745463478 % 234
prob_2
```

Out[2]: 84

Problem 3

Import the `*math*` library and calculate the square root of 143, round this number to 2 decimal places. Assign this value to $prob_3$

```
In [3]: # Answer
import math
prob_3 = round(math.sqrt(143),2)
prob_3
```

Out[3]: 11.96

Problem 4

Determine if 56^{90} is bigger than 57^{89} using a boolean expression. Assign your answer to $prob_4$

```
In [4]: # Answer
prob_4 = (56 ** 90) > (57 ** 89)
prob_4
```

Out[4]: True

Problem 5

Suppose I have 5 boolean values: $TrueFalseTrueTrueFalse$. What proportion of the 5 values are $True$? Use your knowledge of arithmetic with Booleans to calculate this proportion using Python. Assign your answer to $prob_5$. (Hint: add the boolean values and divide by the number of values added.)

```
In [5]: # Answer

Prob_5 =(True + False + True + True + False)/5
Prob_5
```

Out[5]: 0.6

Hence, 0.6 proportion of the 5 values are True (3 Trues out of 5 booleans i.e., $3/5=0.6 \Rightarrow (1+0+1+1+0)/5$)

Problem 6

Create a string called 'my_first_string' and set it equal to the following string: 'Once upon a time, the quick brown fox jumps over the lazy dog and they all lived happily ever after.'

```
In [6]: # Answer
my_first_string = 'Once upon a time, the quick brown fox jumps over the lazy dog and they all lived happily ever after.'
my_first_string
```

Out[6]: 'Once upon a time, the quick brown fox jumps over the lazy dog and they all lived happily ever after.'

```
In [7]: print(my_first_string)
```

Once upon a time, the quick brown fox jumps over the lazy dog and they all lived happily ever after.

Problem 7

Determine if the letter pattern 'ne' is contained in $myfirststr \in g$

```
In [13]: print(my_first_string)
```

Once upon a time, the quick brown fox jumps over the lazy dog and they all lived happily ever after.

```
In [8]: # Answer
'ne' in my_first_string
```

Out[8]: False

'ne' pattern is not present in my_first_string

Problem 8

Determine which position the 'z' in $myfirststr \in g$

```
In [14]: print(my_first_string)
```

Once upon a time, the quick brown fox jumps over the lazy dog and they all lived happily ever after.

```
In [9]: # Answer
my_first_string.find('z')
```

Out[9]: 55

z is present in the 55 position

Problem 9

Replace all instances of 'o' in $myfirststr \in g$ with '[:(&)]' then replace all instances of 'e' with 'zzz', replace all periods with exclamation points, and make the entire string uppercase. Assign the resulting string to $prob_9$

```
In [15]: print(my_first_string)
```

Once upon a time, the quick brown fox jumps over the lazy dog and they all lived happily ever after.

```
In [10]: # Answer
prob_9 = (my_first_string.replace('o','[:(&)]').replace('e','zzz').replace('.', '!')).upper()
prob_9
```

Out[10]: 'ONCZZZ UP[:(&)]N A TIMZZZ, THZZZ QUICK BR[:(&)]WN F[:(&)]X JUMPS [:(&)]VZZZR THZZZ LAZY D[:(&)]G AND THZZZY ALL LIVZZZD HAPPILY ZZZVZZZR AFTZZZR!'

```
In [ ]: #If capital 'O' should also be replaced
```

```
In [17]: prob_9 = (my_first_string.replace('O','[:(&)]').replace('o','[:(&)]').replace('e','zzz').replace('.', '!')).upper()
prob_9
```

Out[17]: '[:(&)]NCZZZ UP[:(&)]N A TIMZZZ, THZZZ QUICK BR[:(&)]WN F[:(&)]X JUMPS [:(&)]VZZZR THZZZ LAZY D[:(&)]G AND THZZZY ALL LIVZZZD HAPPILY ZZZVZZZR AFTZZZR!'

Problem 10

Create an extra long string by adding $prob_9$ to the string 'I love UMBC!' with no space between the two strings. Assign the answer to $prob_{10}$.

```
In [19]: # Answer
prob_10 = 'I love UMBC!' + prob_9
prob_10
```

Out[19]: 'I love UMBC![:(&)]NCZZZ UP[:(&)]N A TIMZZZ, THZZZ QUICK BR[:(&)]WN F[:(&)]X JUMPS [:(&)]VZZZR THZZZ LAZY D[:(&)]G AND THZZZY ALL LIVZZZD HAPPILY ZZZVZZZR AFTZZZR!'

Problem 11

Find the length of $prob_{10}$ without counting and assign this value to $prob_{11}$.

```
In [12]: # Answer
prob_11 = len(prob_10)
prob_11
```

Out[12]: 157

Length of prob_10 is 157

```
In [ ]:
```