

One purpose of this homework is to make sure you can run Python in various ways: 1) interactively from the Python interpreter, 2) in a Jupyter Notebook and/or Google Colabs, and 3) as a script that you invoke from the command line. So, part of the assignment is to get these going first. Note that we will use Python 3 exclusively in this course (Python 2 is no longer supported). Send your results to me in a single email (each file as a separate attachment or all zipped together) by 11:59pm on Sunday, September 6.

1) Answer the questions given in Exercise 1.1 and in Exercise 1.2 questions 1 and 2 (skip 3) at the end of Chapter 1 in the text by simply trying each in the interpreter and then copying/pasting the results into a single text file called “hw01a.txt”. Include the whole interaction but you don’t need to add further comment to the file. Do include Python version number as below, given on startup.

Example:

```
Python 3.6.9 (default, Jul 17 2020, 12:50:27)
```

```
[GCC 8.4.0] on linux
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> x x
```

```
File "<stdin>", line 1
```

```
  x x
```

```
  ^
```

```
SyntaxError: invalid syntax
```

```
>>> 0110
```

```
File "<stdin>", line 1
```

```
  0110
```

```
  ^
```

```
SyntaxError: invalid token
```

```
>>>
```

```
Etc.
```

2) Repeat question 1 for Exercise 2.1 and in Exercise 2.2 questions 1 (skip 2 and 3) at the end of Chapter 2. These should be in a file called “hw01b.txt” Note that for the value of “pi” you can first “import math” and then you can use “math.pi” for its value.

3) In Jupyter Notebook or Google Colabs write a program that will compute the date of Easter after prompting for a year (algorithm below). Running the program should result in **exactly** the following (given the example year of 2020):

```
Enter year: 2020
```

```
Date of Easter in 2020 is 4/12
```

Use Python’s built-in “divmod()” to compute and return both the quotient and remainder in one operation. For example

```
x,y = divmod(13, 4)    # x is 3 (quotient) and y is 1 (remainder)
```

```
_,z = divmod(13, 4)    # underscore is a dummy value; z is set to 1
```

Recall that expressions can be used in place of values. Save your results in a “.ipynb” file called “hw01c.ipynb”.

Use the algorithm below

ALGORITHM TO DETERMINE THE DATE OF EASTER:

Divide	by	Quotient	Remainder
Year	19	-	A
Year	100	B	C
B	4	D	E
B + 8	25	F	-
B - F + 1	3	G	-
19*A + B - D - G + 15	30	-	H
C	4	I	K
32 + 2*E + 2*I - H - K	7	-	L
A + 11*H + 22*L	451	M	-
H + L - 7*M + 114	31	N	P

Month = N (March = 3, April = 4)

Day = P + 1

4) Write a python module called hw01d.py that you can run from the command line that implements the [“C” code given in the file here](#). (The link is also given on this course’s “assignments” page. You may recognize this code as part of an ECE 177 lab.) Implement in Python in a manner as close to the given C code as you can, including comments, control flow and output of each “print()” statement. Resist the urge to write it in a more “Pythonic” way.

Notes:

- 1) Use a “docstring” for the big comment block at the top.
- 2) Don’t put in unnecessary parenthesis (except for readability) or semicolons.
- 3) In Python “True” is used for a logically “true” value (as in the “while” statement).
- 4) Note:

```
for (x=0; x<limit; x++) {           // C version
for x in range(limit):             # Python equivalent
```
- 5) Note:

```
x = y < 7 ? 3 : 2                   // C version
x = 3 if y < 7 else 2               # Python equivalent
```
- 6) Note:

```
print("Hi")                          # Prints "Hi" and then a newline
print("Hi", end='')                  # Prints "Hi" and then '' (nothing)
```