GCS10004-07 Fall 2022

Handong Honor Code

- You are responsible for understanding and complying with Handong Honor Code.
- If you copy someone else's coding or homework assignments or use someone else's creative work without any indication, you will get an F grade. Anyone who shares the code with you will also get an F grade.

Jaeyoung Chun | School of Applied Artificial Intelligence | Handong Global University

Mid-Term Exam

We will test whether you understand some of the concepts we've learned for the past seven weeks. More importantly, we'd like to assess whether you can use those programming concepts to create a computer program.

- String manipulation
- Data types such as lists, tuples, dictionary
- Conditional statements (e.g. `if`)
- Iterative statements (e.g. `for`)
- Functions
- Compound data structure (e.g. storing/accessing tuples in a list or a dictionary)

1 ---

You are provided with the `data` variable (of string type) that contains the full text of the first epistle of Peter (*1 Peter*) and the second epistle of Peter (*2 Peter*) in the Bible. One problem with the `data` variable is all the verses are concatenated one after another so that it's a bit difficult to read, search, and etc.

For example, below are the first two verses extracted from the `data` variable.

```
Pe1|1|1| Peter, an apostle of Jesus Christ, to the strangers scattered thr
oughout Pontus, Galatia, Cappadocia, Asia, and Bithynia,~\nPe1|1|2| Elect
 according to the foreknowledge of God the Father, through sanctification
 of the Spirit, unto obedience and sprinkling of the blood of Jesus Chris
t: Grace unto you, and peace, be multiplied.~\n
```

Our goal is creating a program that allows users to quickly find and extract a verse of interest using Python.

Construct a new list in a way that each element in the list contains one verse. Your code will be part of the function named `load_data`. Once you've successfully constructed the list, make sure you return that list as an output of the function.

In [1]:

```
def load_data():
    with open("peter.txt", "rt") as fin:
        data = fin.read()

    ### YOUR CODE STARTS HERE

    ### YOUR CODE ENDS HERE


lines = load_data()
```

For example, if you did it correctly, `lines[0]` should have:

```
Pe1|1|1| Peter, an apostle of Jesus Christ, to the strangers scattered thr
oughout Pontus, Galatia, Cappadocia, Asia, and Bithynia,
```

The last element of the `lines` variable should have:

```
Pe2|3|18| But grow in grace, and in the knowledge of our Lord and Saviour
 Jesus Christ. To him be glory both now and for ever. Amen.
```

Take a very careful look at the `data` variable (probably you want to print it on the screen to explore the data). You will easily recognize that each verse ends with special characters, which can help you separate one verse from another. If there is any empty string in the list, make sure to remove it.

There are 166 verses in the first and second epistles of Peter, meaning the `lines` variable should have exactly 166 elements (verses).

2 ---

This problem is a continuation of the first problem. You cannot proceed without first completing `E01_01.py`. The `load_data` function and the `lines` variable from `E01_01.py` will be automatically available to you in `E01_02.py`. If you do not understand what this means, please contact me.

Use a `for` or `while` loop and iterate through each verse in the `lines` variable until you find `Pe1|4|7` (implying *1 Peter 4:7*). When you find this verse, store the text in the `text` variable and print it on the screen. Make sure to store and print *only* the text, nothng else. Remove spaces at the beginning and the end of the text if any.

Terminology

```
1 Peter 1:25 but the word of the Lord endures forever. (omitted)
^^^^^^^ ^^^^ ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  (1)    (2)                  (3)
```

- `(1)` : name of the book
- `(2)` : chater number and verse number separated by a colon
- `(3)` : text

In [6]:

```
from E01_01 import load_data
lines = load_data()
### YOUR CODE STARTS HERE

### YOUR CODE ENDS HERE

"""
### YOUR EXPLANATION STARTS HERE

### YOUR EXPLANATION ENDS HERE
"""
```

3 ---

This problem is a continuation of the first problem. You cannot proceed without first completing `E01_01.py`*. The* `load_data` *function and the* `lines` *variable from* `E01_01.py` *will be automatically available to you in* `E01_03.py`*. If you do not understand what this means, please contact me.*

Previously, we created a way to search a Bible verse, but it wasn't really user friendly. For example, finding a specific verse from `lines` was relatively easy, but we had to do some string manipulation in order to extract a text. We'd like to improve on this.

Create a new list variable called `bible`. Initially, this list will be empty.

Use a `for` or `while` loop and iterate through each verse in the `lines` variable. When you iterate, construct a tuple with four elements:

1. Name of the book
2. Chapter number
3. Verse number
4. Text

Then append this tuple to the `bible` list.

When the loop ends, 65th element in the `bible` list (i.e. `bible[64]`) should be:

```
('Pe1', 3, 15, 'But sanctify the Lord God in your hearts: and be ready alw
ays to give an answer to every man that asketh you a reason of the hope th
at is in you with meekness and fear:')
```

Again, there are 166 verses in the first and second epistles of Peter. So, `len(bible)` should return 166. Remove spaces at the beginning and the end of the text if any.

Use a `for` or `while` loop and iterate through each verse in the `bible` variable until you find *1 Peter 1:10*. When you find this verse, print the text on the screen and break out from the loop. Make sure to print *only* the text.

In [9]:

```
from E01_01 import load_data
lines = load_data()
### YOUR CODE STARTS HERE

### YOUR CODE ENDS HERE

"""
### YOUR EXPLANATION STARTS HERE

### YOUR EXPLANATION ENDS HERE
"""
```

4 ---

This problem is a continuation of the first problem. You cannot proceed without first completing `E01_01.py`*. The* `load_data` *function and the* `lines` *variable from* `E01_01.py` *will be automatically available to you in* `E01_04.py`*. If you do not understand what this means, please contact me.*

In the previous problem, we used a list and tuples together to make the search process simpler and easier. However, we know that iterating through every single element in the list to find something is not a good idea. It's like flipping hundreds of thousands of pages. Here, we're going to use Python's dictionary instead to make the search process super fast and simple.

Create a new dictionary variable called `bible`. Initially, this dictionary will be empty.

Use a `for` or `while` loop and iterate through each verse in the `lines` variable. When you iterate, construct a tuple with the following three elements:

1. Name of the book
2. Chapter number
3. Verse number

Then use this tuple as the key, the text as the value, and add this key-value pair to the `bible` dictionary.

Again, there are 166 verses in the first and second epistles of Peter. So, there should be 166 keys and 166 values in the `bible` dictionary.

Print the text of *1 Peter 4:13* on the screen using the `bible` dictionary you just created.

In [14]:

```
from E01_01 import load_data
lines = load_data()
### YOUR CODE STARTS HERE

### YOUR CODE ENDS HERE

"""
### YOUR EXPLANATION STARTS HERE

### YOUR EXPLANATION ENDS HERE
"""
```