M2 - W1 Assignment: Advanced Data Structures and Typings

In this assignment you will work on your basic and intermediate Python skills. You will use your knowledge on basic and advanced built-in Python data structures and write simple functions.

1.

I.    Write a function (with the appropriate dockstrings and typings) that will take as input a list of strings and will return a dictionary of iterables, where each iterable contains the location of the key in the original list, for example, if the function takes as input the list [1,2,3,2,3,1] it should return {1:[0,5], 2:[1,3], 3[2,4]}.  **There will be bonus points if you can do it without the use of any for loops.**

Solution (I):

# Create list

my\_list = [3, 4, 5, 5, 4, 3]

# Create dictionary

my\_dict = {}

# Use iterable to find index of each value

for index, val in enumerate(my\_list) :

# Use logical function

if val not in my\_dict :

my\_dict[val] = [index, index]

else :

my\_dict[val][1] = index

# Display dictionary

print(my\_dict)

II.    Test your function by reading the list found here and saving the file under the name frequency\_of\_numbers.txt. You can either write your own code to save it or use any built-in Python library.

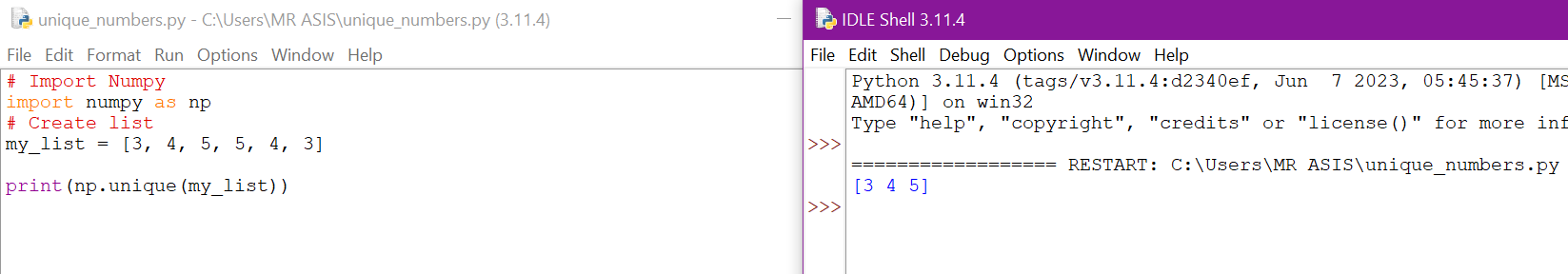
Solution (II):

Submitted as a separate .txt file.

2.

I.    Consider the same list found here. Write a function (with the appropriate typings and dockstrings) that returns a list of the unique elements in the original list. **Extra points if you can do this without the use of the built-in set function**.

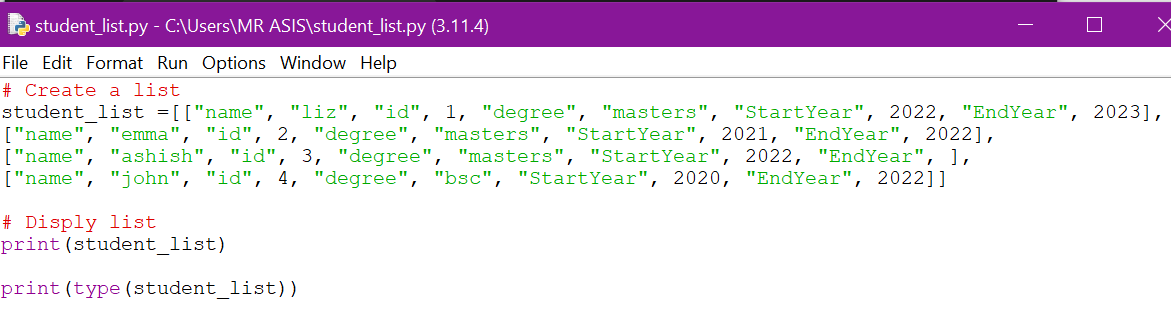
Solution (I):



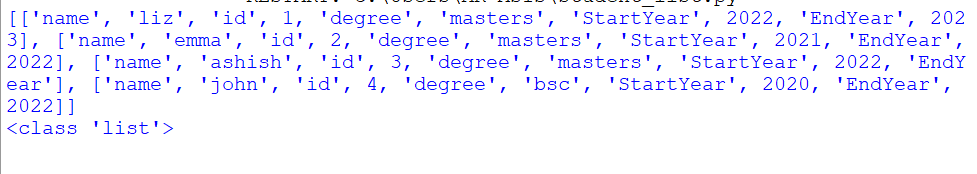
3.

I.    Create a structure to store multiple students’ data using Python’s basic data structures. Make sure each student will have the following features: Name, ID, Degree, StartYear, EndYear. Each student will have multiple courses taken from a specific lecturer (e.g., each student can take N different courses from M different lecturer, each of the information must be defined). Also, assume that some students might haven’t graduated yet so these students shouldn’t have an EndYear.

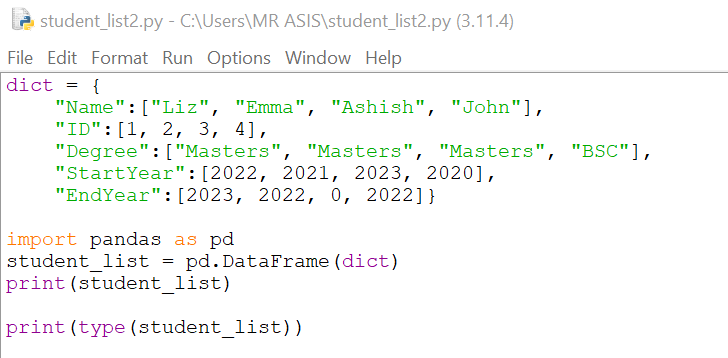
Process 1:



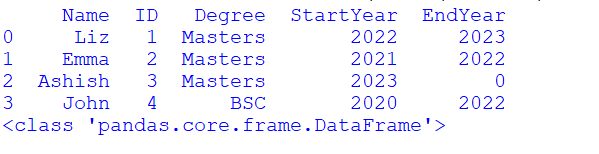
Output



Process 2:



Output



II.   Describe the typing structure that could fit the structure in Question 3.1. You na be as creative as you like. (no need for actual code, a full explanation and justification of the process is required).

Solution (II):

I have created the above student\_list where some are strings and some are integers. For the given problem, each student have name, id, degree, StartYear, and EndYear. So that in Process 1, I have created sublist for each student. Each sublist contains name as string, id as integer, degree as string, StartYear as integer and EndYear as integer. The data type is ‘list’. In Process 2, I have created a dictionary that contains name as string, id as integer, degree as string, StartYear as integer and EndYear as integer. The data type is DataFrame, which looks better than list.

**You should also submit your assignment via GitHub Classroom! Please accept the invitation to the task via the following link:**[**https://classroom.github.com/a/ibP0wjE1Links to an external site.**](https://classroom.github.com/a/ibP0wjE1)

Please submit a Jupyter Notebook (.ipynb file) or a normal Python script (.py)