CSCI 2040: Introduction to Python **2024-2025 Term 1**

Lab Assignment 3

Instructor: Yim Pan, CHUI Due: 23:59 on Wed. Nov. 6st, 2024

Notes

1. You are allowed to form a group of two to do this lab assignment.
2. You are strongly recommended to bring your own laptop to the lab with Anaconda[1] and Pycharm[2] installed. You don't even have to attend the lab session if you know what you are required to do by reading this assignment.
3. Only **Python 3.x** is acceptable.
4. Passing the test scripts we have provided does not guarantee full marks for your question as our grade scripts will test for more cases.
5. You may assume that all the corner cases we have not mentioned in this document will not appear in the hidden test cases, so you do not have to worry too much about wrong inputs unless you are required to do so.
6. Your code should only contain specified functions. Please delete all the debug statements (e.g. print) before submission.

Exercise 1 (20 marks)

Using lists, write a function `non_unique(list)` that takes a list as argument. The function should return a new list that contains all duplicated elements of the original list. The order of the elements in the returned list should be the same as their first appearance in the original list.
For example, if the input list is ['apple', 'banana', 'apple', 'orange', 'banana', 'grape'], the function should return ['apple', 'banana'].
Your program should contain the function in the format shown below:

```python
def non_unique(list):
    # Your codes here.
    return result # 'result' is a list.
```

Save your script for this exercise in `p1.py`

Exercise 2 (20 marks)

The quicksort algorithm is a recursive method that operates by choosing an element, known as the pivot, and splitting the initial list into three segments: those larger than the pivot, those

[1]An open data science platform powered by Python. `https://www.continuum.io/downloads`
[2]A powerful Python IDE. `https://www.jetbrains.com/pycharm/download/`

equal to the pivot, and those smaller than the pivot. It then sorts the first and third segments recursively, combining the results to achieve the final outcome. Develop a quicksort function that takes a `list` of numbers and employs ***list comprehension*** to build the function. It's important to note that the numbers must be sorted in descending order.
You **MUST** use **List Comprehension** to do the sorting. Otherwise, you will get zero marks for Exercise 2.
Your program should contain the function in the format shown below:

```
def quicksort(a):
    # Your codes here. Use List Comprehension and return the result.
```

Save your script for this exercise in `p2.py`

Exercise 3 (60 marks)

The given file `dept-prof.pydata` contains data about school departments and their corresponding professors.
Note that you need to place the data file to the same directory hierarchy as your script for this exercise. When reading a file, you should avoid the crash of your program in case that the file does not exist. (Hint: you may use `try...except..` statement). And you should include all the data files `*.pydata` for Exercise 3 when you submit lab 3 (see `Submission rules`).

(a) We can use `pickle` to do data storage and retrieval. Implement a function that takes the name of the data file (e.g., `dept-prof.pydata`) as argument, through which we can load the given data file `*.pydata` to a `dictionary`, where the keys are names of the departments and the value to each key is a list of professors who work for this department. And the function should return the `dictionary` load from `*.pydata`.**(15 marks)**

Your program should contain the function in the format shown below:

```
def load_data(file):
    # Your codes here.
    return result # result is the dictionary you loaded from the 'file'.
```

(b) Implement a function that takes a professor's name as argument (case sensitive), through which we can query which department(s) he/she works for. The function should return a `list` of department(s). **(15 marks)**

Your program should contain the function in the format shown below:

```
def query(prof_name):
    # Your codes here.
    return result # 'result' is a list.
```

(c) Imagine that it is the year 2028, and the school wants to establish some new departments and remove the outdated ones. Implement a function including the following operations to the dictionary we get from `dept-prof.pydata`.

- remove the old department `Artificial Intelligence`, and move all professors in this department to the department `Computer Science`
- add a new department `Space Engineering` with professors `Mark`, `Neo` and `Jane`
- use `pickle` to save the updated dictionary to the same directory hierarchy as your script for this exercise as file `dept-prof-updated.pydata`

There is no need to return anything. This function will be graded by testing the generated file `dept-prof-updated.pydata`.**(15 marks)**

Your program should contain the function in the format shown below:

```
def update():
    # Your codes here.
    # No need to return.
```

(d) Implement a function to analyze the updated data in `dept-prof-updated.pydata` which we get from (c), and return a new `dictionary` where the keys are names of the departments and the value to each key is the number of its professors. **(15 marks)**

Your program should contain the function in the format shown below:

```
def get_depts_size():
    # Your codes here.
    return result # 'result' is a dictionary.
```

Save your script for this exercise in `p3.py`

Submission rules

1. Please name the `functions`, script files and data files with the **exact** names specified in this assignment and test all your scripts. Any script that has any wrong name or syntax error will not be marked.
2. For each group, please pack all your script files as well as the data files `dept-prof.pydata` and `dept-prof-updated.pydata` in Exercise 3 as a single archive named as
   `<student-id1>_<student-id2>_lab3.zip`
   For example, `1155012345_1155054321_lab3.zip`, i.e., just replace `<student-id1>` and `<student-id2>` with your own student IDs. If you are doing the assignment alone, just leave `<student-id2>` empty, e.g, `1155012345_lab3.zip`.
3. Upload the zip file to your blackboard (`https://blackboard.cuhk.edu.hk`),

- Only one member of each group needs to upload the archive file.
- Subject of your file should be `<student-id1>_<student-id2>_lab3` if you are in a two-person group or `<student-id1>_lab3` if not.
- No later than 23:59 on Wed. Nov. 6st, 2024

4. Students in the same group would get the same marks. Marks will be `deducted` if you do not follow the submission rules. Anyone/Anygroup who is caught plagiarizing will get a 0 score!