LAB 05

1. **Tutorial: Pickle**
   Study the following tutorial on python pickling. Make sure you test all the code.
   https://www.pythoncentral.io/how-to-pickle-unpickle-tutorial/

2. **Logging**
   Use python logging features to add logging information to your code of the Tic-Tac-Toe game. Run your code for a while and then show the TA the log output generated.

3. **Target sum**
   Write a function targetSum(nums, target) that find four numbers from a list of integers (nums) such that the sum of these four integers is equal to a given number (target).

   For example:

   `targetSum([3, 0, -1, 0, -2, 5], 2)` will return

   ```
   >>> [[-2, -1, 0, 5], [-1, 0, 0, 3]]
   ```

4. **Sorting algorithms**
   a. Implement the insertion sort algorithm (https://youtu.be/uMqVuEEWJv4).
   b. Implement the merge sort algorithm (https://youtu.be/4VqmGXwpLqc).
   c. Compare their execution times (for the same sample list of integers) with the execution time of the built-in python *sorted(list)* function.
      Hint: Consider using the python *time* module to compute the execution time.
   d. Optional: Python language uses the Timsort algorithm to implement both *sorted()* and *sort()*. Read the documentation describing the Timsort algorithm. https://en.wikipedia.org/wiki/Timsort

5. **Custom exceptions**
   In python, it is possible to create user-defined exceptions. The following short tutorial shows how to do it:
   https://www.programiz.com/python-programming/user-defined-exception

a) Write a python code[1] that loops over a list of user data (tuples containing a username, email and age) and adds each user to a directory if the user is at least 16 years old. You do not need to store the age. (See example below).

Write a simple exception hierarchy which defines a different exception for each of these error conditions:

1. the username is not unique
2. the age is not a positive integer
3. the user is under 16
4. the email address is not valid (a simple check for a username, the @ symbol and a domain name is sufficient)

b) Raise these exceptions in your program where appropriate. Whenever an exception occurs, your program should move onto the next set of data in the list. Print a different error message for each different kind of exception.

Hints: You can consider an email address to be valid if it contains one @ symbol and has a non-empty username and domain name – you don't need to check for valid characters. You can assume that the age is already an integer value.

Running your application with the following as input:

```
usersList =

[("jane", "jane@example.com", 21),("bob", "bob@example",
19),("jane", "jane2@example.com", 25),("steve",
"steve@somewhere", 15),("joe", "joe", 23),("anna",
"anna@example.com", -3),]
```

should return:

```
Username 'jane' is in use.
User steve is underage.
'joe' is not a valid email address.
Invalid age:-3
```

6. **NumPy**

Installation: https://numpy.org/install/

Using a ***NumPy array***, write a python code that asks the user to enter the elements of the array and tests whether none of the elements of the array is zero.

---

[1] Exercise 2, Chapter 10, page 154, Object-oriented programming in python documentation, Release 1

Here is a sample output:

```
Enter the size of the array:5
Enter element:1
Enter element:2
Enter element:3
Enter element:4
Enter element:5
Test array: [1 2 3 4 5]
Test if none of the elements of the said array is zero: True

Enter the size of the array:5
Enter element:0
Enter element:1
Enter element:2
Enter element:3
Enter element:4
Test array: [0 1 2 3 4]
Test if none of the elements of the said array is zero: False
```

7. **Pandas**

Installation: https://pandas.pydata.org/docs/getting_started/install.html

Write a program to convert a python dictionary to a Pandas series.
Here is a sample output:

```
Enter the size of the dictionary: 3
Enter the key: a
Enter the value: 12
Enter the key: b
Enter the value: 24
Enter the key: c
Enter the value: 48

Original dictionary:
{'a': '12', 'b': '24', 'c': '48'}
Converted series:
a     12
b     24
c     48
dtype: object
```

End.