

## Introduction to programming – ASSIGNMENTS 8

### General instructions:

- Type in and test your programs using Python Idle.
- Bring in your answers to the next assignment (i.e. demonstration) session in memory stick, or save them to a web folder accessible in class. Alternatively, you can bring written answers with you, though this is not recommended.
- Remember to comment your code – this does NOT mean, that every single line should be commented. Comment the important parts in your program.
- Prepare to present your solution to the class.

1.

Write a program that queries the user for persons' names and ages and saves them into a dictionary with name as a key and age as a value. When the user enters an empty string as a name, the program outputs the dictionary and terminates.

Example run:

```
Enter a name or an empty string to stop: James
Enter age: 25
Enter a name or an empty string to stop: Jane
Enter age: 31
Enter a name or an empty string to stop:
{'James' : 25, 'Jane' : 31 }
```

2.

Write a function `uniqueItems(list1, list2)`, which gets two lists as an argument, and returns a new list that contains all unique items from both lists. Utilize **sets** to create the list.

Example run (in Python Shell):

```
>>> l1 = [1,2,1,3]
>>> l2 = [2,3,4,5]
>>> l3 = uniqueItems(l1,l2)
>>> print l3
[1, 2, 3, 4, 5]
```

3.

Write a function **dictToList(dict)** , which gets a dictionary as an argument. The function creates a new list containing all keys and values from the dictionary as tuples. Each tuple consists of a key and the corresponding value (in that order). Finally, the generated list is returned.

Example run (in Python Shell):

```
>>> dict = {0 : "abc", 2 : "bcd", 3: "cde", 4 : "def"}
>>> myList = dictToList(dict)
>>> print myList
[(0, "abc"), (2, "bcd"), (3, "cde"), (4, "def")]
```

4.

Write a function **minMaxAvg(dict)** , which gets a dictionary as an argument. You can assume that all *values* in dictionary are numbers. The function returns a tuple, which contains the smallest value, the largest value and the average of all items stored in the dictionary.

Example run (in Python Shell):

```
>>> d = {"a":0, "b":-1, "c":3, "d":6, "e":11, "f":8}
>>> print minMaxAvg(d)
(-1, 11, 4.5)
```

5.

Write a function **vowelProportions(str)** , which gets a string as an argument. The function generates and returns a dictionary, containing proportions of each vowel (a, e, i, o, u, y) as a percentage of all characters in the string. The vowel should be stored as a key, and the proportion as a value.

To calculate the proportion of a letter 'x' in a string, use formula  $\frac{\text{number of letters } x}{\text{total number of letters}} \cdot 100$

Example run (in Python Shell):

```
>>> print vowelProportions("aaccfedubbbyyy")
{'a': 13, 'e': 6, 'i': 0, 'o': 0, 'u': 6, 'y': 26}
```

6. \*\* Expert assignment (double points)

Note: you should use the function **LCM** written in Assignments 5, Question 4 to solve this.

Write a program that queries the user for numbers A and B, and outputs the lowest common multiple of the numbers. *However*, instead of just using the function LCM, the program should have a dictionary with already solved multiples. In dictionary, values A and B are used as a key, and the LCM for A and B as a value.

When the user enters the values, the program:

1. Terminates, if either of the values A or B is a zero
2. Tries to find the LCM for A and B by looking at the dictionary. Note, that the order of A and B should not matter; if there is an LCM for numbers 2 and 5 in the dictionary, the program should be able to use that to solve LCM for numbers 5 and 2 as well!
3. If the LCM is not found in the dictionary, the program calculates it using the function, and then saves this new LCM to dictionary.
4. After the LCM is output, the program queries for new numbers.

Example output:

```
Give A: 3
Give B: 5
Not found in dictionary!
LCM for A and B is 15.
Give A:4
Give B:2
Not found in dictionary!
LCM for A and B is 4.
Give A: 5
Give B: 3
Found in dictionary!
LCM for A and B is 15.
Give A: 0
Give B: 1
Bye!
```