Lab 6

Tuples and dictionaries

Laboratory Objectives

Exercises with:

- Tuples, dictionaries and other data structures.
- Algorithms that use data structures.

Tuples

- Similar to lists, but can not be modifiable

```
>>> t1 = (1,2,3,4,5)
>>> len(t1)
5
>>> t1[0]
1
>>> t1[0] =7
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    t1[0] =7
TypeError: 'tuple' object does not support
item assignment
```

Dictionaries

- ***Dictionnaires*** are not sequential.
- We have values (with different types) that require a specific index to access them, called a ***key*** (alphabetic, numerical, or other).

```
>>> d1 = {'computer': 'pc1',
        'keyboard': 'keyb1', 'mouse': 'mo1'}
>>> d1['printer'] = 'print1'
>>> d1
{'keyboard': 'keyb1', 'printer': 'print1',
'computer': 'pc1', 'mouse': 'mo1'}
>>> len(d1)
4
```

Dictionaries (cont.)

- Dictionaries are not modifiable. We can add or remove elements. We can modify the value of a key.

```
>>> d1['computer'] = ‘pc2'
>>> d1.keys()
dict_keys(['keyboard', 'printer', 'computer',
'mouse'])
>>> d1.values()
dict_values([‘keyb1', ‘print1', ‘pc2',
‘mo1'])
>>> 'printer' in d1
True
```

Construction of a histogram with a dictionary

```
>>> text ="the drinks are in the refrigerator"
>>> letters ={}
>>> for c in text:
            letters[c] =letters.get(c, 0) + 1

>>> print(letters)
{'t': 3, 'h': 2, 'e': 5, ' ': 5, 'd': 1, 'r': 6,
'i': 3, 'n': 2, 'k': 1, 's': 1, 'a': 2, 'f': 1,
'g': 1, 'o': 1}
>>> letters_sorted = list(letters.items())
>>> letters_sorted.sort()
>>> print(letters_sorted)
[(' ', 5), ('a', 2), ('d', 1), ('e', 5), ('f', 1),
('g', 1), ('h', 2), ('i', 3), ('k', 1), ('n', 2),
('o', 1), ('r', 6), ('s', 1), ('t', 3)]
```

Exercise 1

- Create a function that takes a character chain and returns a histogram in a dictionary format.
- The main program must take a dictionary and display it in alphabetical order.

Note d.get(val1, val2) will return the value of the key val1 from dictionary d, otherwise val2. If we try to access directly d[val1], we get an error if key val1 does not exist.

Exercise 2

- Create a function that takes a tuple and returns a dictionary that is a histogram that counts how many times each number gets into the tuple:
- The main program must take the dictionary, transform the items in a list, use *sort()* to order the list(in ascending order) and then display it.

```
def histo_n(x):
    '''(tuple)->dict
    Returns a dictionary
    Precondition: thee tuple contains integers
    >>> t = (1,2,-3,3,4,-3,3,3)
    >>> histo_n(t)
    {1: 1, 2: 1, 3: 3, 4: 1, -3: 2}
    '''
```

Exercise 3

- Create a function described by:

```
def sum_of_three(x):
    '''(tuple)->bool
    Returns True if the sum of 3 consecutive
     elements is zero
    Precondition: the tuple has at least 3
    elements
    >>> t = (1,2,-3,4,-1,3)
    >>> sum_of_three(t)
    True
    '''
```

Exercise 4

Create a function move_zeros whose parameter is a list of integers and moves all the zeros to the end of the list. For instance, if the list is [1, 0, 3, 0, 0, 5, 7] the result should be [1, 3, 5, 7, 0, 0, 0]

Derive THREE solutions

- move_zeros_v1 uses another list tmp to compute the new list and returns it as a result (easy one) . The initial list is not modified.
- move_zeros_v2 modifies the initial list inside the function and does not return anything.
- move_zeros_v3 moves the elements in the initial list without using any temporary list (harder one) . The function does not return anything. We can use a temporary variable to switch 2 elements, but we can not use the Python exchange `a,b=b,a`

Exercise 4 (cont.)

```
>>> x = [1, 0, 3, 0, 0, 5, 7]
>>> y=move_zeros_v1(x)
>>> print(x, y)
[1, 0, 3, 0, 0, 5, 7] [1, 3, 5, 7, 0, 0, 0]

>>> x = [1, 0, 3, 0, 0, 5, 7]
>>> z=move_zeros_v2(x)
>>> print(x, z)
[1, 3, 5, 7, 0, 0, 0] None

>>> x = [1, 0, 3, 0, 0, 5, 7]
>>> t=move_zeros_v3(x)
>>> print(x, t)
[1, 3, 5, 7, 0, 0, 0]  None
```