Beginselen van Programmeren
Exercise Session 6:
Sets and Dictionaries

Sets

Question 1.

Ask the user to input two integer numbers $m < n$. Create 2 sets so that the former contains all the even integers between m and n and the latter contains all the integers between m and n that are divisible by 3. Use the two sets to solve the following questions:

- How many integers between m and n that are divisible by 6 ?
- How many odd integers between m and n that are divisible by 3?
- How many integers between m and n that are divisible by either 2 or 3?

Question 2.

Create a function that receives a text as parameter and returns the vocabulary of words occurring in the text. The vocabulary should be implemented as a Python set.
Hints: You may need to use the built-in function *split* to split a string into words, and the following *cleanWord* function to get the cleaned word form.

```
import string
def cleanWord(w):
    return w.lower().strip(string.punctuation)
```

Question 3.

Use the function in Question 2 to write a program that builds word vocabulary from a sequence of texts. The program should have a loop allowing the user to type texts. The loop finishes when user types “QQQ". Whenever the program reads a text, it updates the vocabulary, then prints the new vocabulary size, the list of newly added words. After the user types “QQQ", the program prints the list of words occurring in more than 50% of the input texts.

```
Input a text:  Java is crazy.
New size of the vocabulary:  3
Words newly added:  'java', 'crazy', 'is'
```

```
Input a text:  Python is really crazy
New size of the vocabulary:  5
Words newly added:  'python', 'really'
Input a text:  QQQ
Words occurring in at more than 50% of the input texts:  crazy is
```

Question 4.

In a survey, fans are asked to choose their 3 favorite football teams (in any order) among “Bel", “Eng", “Ger", “Fra", “Ita", “Spa" and “Cam". Read file “survey.txt" that contains fan answers, then solve the following questions using Python set operators:

- Print the teams that are not selected by any fan.
- Print the total number of fans who like both “Bel" and “Ger".
- Print the total number of pair of fans have the same favorite teams (in any order).

Hints: You might need to use the following code to read a file:

```
f=open("survey.txt","r")
for line in f.readlines():  # read file line by line
    line=line.strip() # remove newline character
    #...  do something
f.close() # close file
```

Dictionaries

Question 5.

Create a dictionary whose keys and values are hexadecimal digits and their corresponding values in base-10 numeral system respectively.
Use the dictionary to write a function that converts hex numbers to decimal numbers.
Hints: `http://www.mathsisfun.com/binary-decimal-hexadecimal.html`

Question 6.

In cryptography, a Caesar cipher is a very simple encryption techniques in which each letter in the plain text is replaced by a letter some fixed number of positions down the alphabet. For example, with a shift of 3, A would be replaced by D, B would become E, and so on. The method is named after Julius Caesar, who used it to communicate with his generals. ROT-13 (“rotate by 13 places") is a widely used example of a Caesar cipher where the shift is 13. In Python, the key for ROT-13 may be represented by means of the following dictionary:

```
key = {"a":"n", "b":"o", "c":"p", "d":"q", "e":"r", "f":"s", "g":"t",
"h":"u", "i":"v", "j":"w", "k":"x", "l":"y", "m":"z", "n":"a", "o":"b",
"p":"c", "q":"d", "r":"e", "s":"f", "t":"g", "u":"h", "v":"i", "w":"j",
"x":"k", "y":"l", "z":"m", "A":"N", "B":"O", "C":"P", "D":"Q", "E":"R",
"F":"S", "G":"T", "H":"U", "I":"V", "J":"W", "K":"X", "L":"Y", "M":"Z",
"N":"A", "O":"B", "P":"C", "Q":"D", "R":"E", "S":"F", "T":"G", "U":"H",
"V":"I", "W":"J", "X":"K", "Y":"L", "Z":"M"}
```

Write an encoder/decoder of ROT-13. Once you are done, you will be able to read the following secret message:

```
Pnrfne pvcure?  V zhpu cersre Pnrfne fnynq!
```

Hints: Note that enumerating dictionary items should be avoided. You may need to use the built-in functions *ord* and *chr* as introduced in `https://docs.python.org/3/library/functions.html`.

Question 7.

Modify the program in Question 3 so that word vocabulary is implemented as a dictionary. The dictionary stores words and word frequencies as keys and values respectively.

Write a function that prints the frequencies of all the words occurring in the vocabulary as below:

```
word1        frequency1
word2        frequency2
...
```

Write a function that prints the *n* most common words.

Hints: To sort the items of a dictionary by values, you might need to use the built-in function *sorted*:

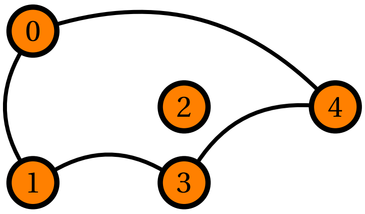
```
import operator
sorted(yourDictionary.items(), key=operator.itemgetter(1), reverse=True)
```

Question 8.

In Python, a graph can be easily built out of a dictionary mapping graph nodes to their set of successor nodes. A graph node is represented by its name which is a number in this exercise.

Undirected graph The undirected graph in the figure below is represented as:

```
undirectedGraph ={0:  {1,4}, 1:{0,3}, 2:{}, 3:  {1,4}, 4:  {0,3}}
```



Write a function that allows user to input an undirected graph by specifying its edges/nodes and return the dictionary representing the graph. For example, to obtain the graph as in the above figure, the input should be as follows:

```
Input an edge (Node1 Node2) or a node (Node1):  0 1
Input an edge (Node1 Node2) or a node (Node1):  1 3
Input an edge (Node1 Node2) or a node (Node1):  3 4
Input an edge (Node1 Node2) or a node (Node1):  4 0
Input an edge (Node1 Node2) or a node (Node1):  2
Input an edge (Node1 Node2) or a node (Node1):  QQQ
```

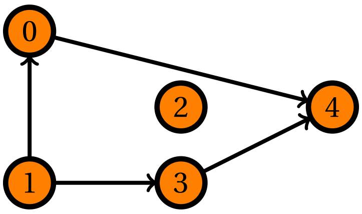
Write a function to check if a node is isolated.

Write a function to check if there is any loop in the graph (if there is an edge that connects a node to itself).

Write a function to check if two nodes are both directly connected to a third node.

Write a function to check if there is a path between two nodes.

Directed graph Modify the program to work with directed graphs.



Extra Excercises: R8.17, R8.19, P8.5, P8.10