Lab 8 – Sets, Dictionaries

Q1 – Write a function named **count** that given a list of words, it returns a dictionary of the words that shows how many time(s) each word has been repeated in the list. Note that your function is case-insensitive.

For example, if the list is like [“is”, “the”, “season”, “The”, “computer”, “the”, “Is”, ”COMPUTER”], then your function must return a dictionary like {“season”:1, “computer”:2, “is”:2, “the”:3}

```
def count(ls):
   d = {}
   for word in ls:
      word = word.lower()
      d[word] =  d.get(word,0) + 1
   return d
```

Q2 - Write a function named **getStat** that given a file name, it returns a dictionary of the words and number of their occurrences in the file. Note that your function is case-insensitive.

```
#precondition: the fileName is valid
def getStat(fileName):
   f = open(fileName, 'r')
   content = f.read()
   f.close()
   d = {}
   words = content.split()
   for word in words:
      word = clean(word)
      if word != None :
         d[word] =  d.get(word,0) + 1
   return d
```

Q3 - Now write a program that calls **getStat** function from previous question to read the contents of two text files and then compares them in the following ways by developing the following functions:

1. A function named **uniqueWords** that given two file names, it finds and returns a list of all the unique words (one occurrence) that exists in both files.
2. A function named **moreThanOnce** that given two file names, it finds and returns a list of words that appear more than once in both files.
3. A function named **onlyInFirst** that given two file names, it finds and returns a list of words that appear in the first file but not the second.
4. A function named **moreThanTwice** that given two file names, it finds and returns a list of words that appear more than twice in either the first or second file, but not both.
5. Write main function that calls all the above functions to test them. You can write your own functions to finish the program if it is needed.

The text files are also uploaded on BB for testing your program.

```
#This is a sample main function to test all the other functions
def main():
   #geeting two valide filename from user
   fname1 = getValidName()
   fname2 = getValidName()
   #calling the functions
   print(getStat(fname1))
   print(uniqueWords(fname1, fname2))
   print(moreThanOnce(fname1, fname2))
   print(onlyInFirst(fname1, fname2))
   print(moreThanTwice(fname1, fname2))

def getValidName():
   name = input("Enter a correct name: ")
   while True :
      try:
         open(name, 'r')
         return name
      except IOError as err:
         print("Error - ", err)
         name = input("Enter a correct name: ")

#precondition: the fileName is valid
def getStat(fileName):
   f = open(fileName, 'r')
   content = f.read()
   f.close()
   d = {}
   words = content.split()
   for word in words:
      word = clean(word)
      if word != None :
         d[word] =  d.get(word,0) + 1
   return d

def clean(w):
    w = w.lower()
    w = w.strip('\'<>{}()[]-=+@#$%^&*|\",.!?#&0123456789')
    if w == '' :
        return None
    return w

def uniqueTokens(filename):
    s = set()
    d = getStat(filename)
    for key in d.keys():
        if d[key] == 1 :
            s.add(key)
    return s
```

```
def uniqueWords(f1, f2):
    set1 = uniqueTokens(f1)
    set2 = uniqueTokens(f2)
    return set1 & set2

def moreThanOnce(f1, f2):
    set1 = moreThanNum(f1, 1)
    set2 = moreThanNum(f2, 1)
    return set1 & set2

def moreThanNum(filename, num):
    s = set()
    d = getStat(filename)
    for key in d.keys():
        if d[key] > num :
            s.add(key)
    return s

def onlyInFirst(f1 , f2):
    set1 = set(getStat(f1))
    set2 = set(getStat(f2))
    return set1 - set2

def moreThanTwice(f1 , f2):
    set1 = moreThanNum(f1, 2)
    set2 = moreThanNum(f2, 2)
    return (set1 | set2 ) - (set1 & set2) # set1 ^ set2

main()
```