William Kelley
ITE315-Assignment 05

```
$ python3 assn05.py
hello world how are you today hello today
{'hello': 2, 'world': 1, 'how': 1, 'are': 1, 'you': 1, 'today': 2}
Enter a month number:
5
      May 2019
Mo Tu We Th Fr Sa Su
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

37
```

```
#!/usr/bin/python

# William Kelley
# ITE315-Assignment 05
# July 5, 2019

# Problem Number 2 to me is more difficult than problem 4.
# Working on a better implementation
# of it, I just wanted to make sure I got something decent to submit.

# Sources:
# https://teamtreehouse.com/community/counting-words-in-a-string-
using-a-dictionary-python-collection-challenge
#

import calendar

#Problem 1
def wordCounter(countMe):
  print(countMe)
  countMe = countMe.lower().split()
  countDict = {}
  for i in countMe:
    if i in countDict:
      countDict[i] += 1
    else:
      countDict[i] = 1
  return countDict
```

```
#Problem 2 - Only other way I could think would be a lot of if else
statements or switch may resubmit with the other way
def calendarPrint(month):
  return (calendar.month(2019, int(month, 10)))

#Problem 3
def countString(string):
  return len(string)


def main():
  print(wordCounter("hello world how are you today hello today"))
  month=input("Enter a month number:\n")
  print(calendarPrint(month))
  print(countString("hello this is something about strings"))



if __name__ == "__main__":
  main()
```

```
#Problem 4
$ python3 lab16.py
Random 1000 bases
{'A': [272, 262, 241], 'C': [241, 233, 250], 'G': [263, 267, 257],
'T': [224, 238, 252]}
Mutated 1000 bases
{'A': [265, 277, 239], 'C': [241, 220, 249], 'G': [269, 271, 252],
'T': [225, 232, 260]}
```

```
#!/usr/bin/python

import random

def frequencyTable(dnaList):
  n = max([len(dna) for dna in dnaList])
  frequency_matrix = {
    'A': [0]*n,
    'C': [0]*n,
    'G': [0]*n,
    'T': [0]*n
  }
  for dna in dnaList:
    for index, base in enumerate(dna):
      frequency_matrix[base][index] += 1

  return frequency_matrix
```

```
def mutateDna(dnaList):
  for _ in range(100):
    #get a random sequence
    index = random.randint(0, 999)
    #assign it as a list to be able to edit one gene
    mutate = list(dnaList[index])
    #insert the mutated gene
    mutate = generateString(3)
    #convert back to a string
    dnaList[index] = ''.join(mutate)
  return dnaList

def generateString(N, alphabet=list('ATGC')):
  dnaList = [random.choice(alphabet) for i in range(N)]
  dnaList = ''.join(dnaList)
  return dnaList

def main():
  listDna=list()
  for _ in range(1000):
    listDna.append((generateString(3)))
  print("Random 1000 bases")
  print(frequencyTable(listDna))
  print("Mutated 1000 bases")
  print(frequencyTable(mutateDna(listDna)))

if __name__ == '__main__':
  main()
```