Problem 1

1. **Program for calculating cosine similarity** – This program is implemented in python. It calculates cosine similarity between each of the five queries and the given documents. Input to this program are documents and each query. The output of this query is a python dictionary containing first 10 documents with the largest score. The key of dictionary represents filename and value represents score.

```
import math

from collections import Counter

from string import punctuation

import os

import subprocess as sp

import zipfile

import operator

import itertools


def cos_sim(a,b):

  # Input text

  t1 = open(a, 'r')

  textwords1=t1.read().split()

  t2 = open(b, 'r')

  textwords2=t2.read().split()


  # count word occurrences

  a_vals = Counter(textwords1)

  b_vals = Counter(textwords2)


  # convert to word-vectors

  words  = list(a_vals.keys() | b_vals.keys())

  a_vect = [a_vals.get(word, 0) for word in words]
  b_vect = [b_vals.get(word, 0) for word in words]


  # find cosine
  len_a  = sum(av*av for av in a_vect) ** 0.5
  len_b  = sum(bv*bv for bv in b_vect) ** 0.5
  dot    = sum(av*bv for av,bv in zip(a_vect, b_vect))
  cosinesim = dot / (len_a * len_b)
  return cosinesim



listOfFiles = os.listdir('docs')
sim_list = {}
sorted_list = {}
for entry in listOfFiles:
      str1 = "docs\\"
      str2 = str1 + entry
      cosinesim = cos_sim(str2,'1')
      if cosinesim > 0:
         sim_list.update({entry:cosinesim})


sorted_list = sorted(sim_list.items(), key=operator.itemgetter(1), reverse=True)
d = dict(itertools.islice(sorted_list, 10))
print("Cosine similarity ",d)
```

Output (First query and documents):

```
Cosine similarity {'51060': 0.5610469391504762, '51144': 0.27498597046143514, '51120': 0.23981604843963608, '51164': 0.230541983810352, '51135': 0.21650635094610968, '51158': 0.2100420126042015, '51184': 0.20100756305184242, '51171': 0.19364916731037085, '51130': 0.17770466332772772, '51161': 0.17251638983558856}
```

Program for calculating dot similarity - This program is implemented in python. It calculates dot similarity between each of the five queries and the given documents. Input to this program are documents and each query. The output of this query is a python dictionary containing first 10 documents with the largest score. The key of dictionary represents filename and value represents score.

```
import math
from collections import Counter
from string import punctuation
import os
import subprocess as sp
import zipfile
import operator
import itertools

def dot_sim(a,b):
   # Input text
   t1 = open(a, 'r')
   textwords1=t1.read().split()
   t2 = open(b, 'r')
   textwords2=t2.read().split()

   # count word occurrences
   a_vals = Counter(textwords1)
   b_vals = Counter(textwords2)

   # convert to word-vectors
   words  = list(a_vals.keys() | b_vals.keys())
   a_vect = [a_vals.get(word, 0) for word in words]
   b_vect = [b_vals.get(word, 0) for word in words]

   # find dot
   dot    = sum(av*bv for av,bv in zip(a_vect, b_vect))
   return dot


listOfFiles = os.listdir('docs')
sim_list = {}
sorted_list = {}
for entry in listOfFiles:
        str1 = "docs\\"
        str2 = str1 + entry
        dot = dot_sim(str2,'5')
        if dot > 0:
            sim_list.update({entry:dot})
sorted_list = sorted(sim_list.items(), key=operator.itemgetter(1), reverse=True)
d = dict(itertools.islice(sorted_list, 10))
      print("Dot similarity ",d)
```

Output (First query and documents):

```
Dot similarity{'51060': 170, '49960': 21, '51153': 11, '51165': 10, '51164': 9, '51120': 7,
'51144': 7, '51130': 6, '51135': 6, '51161': 5}
```

2.

- Cosine Similarity for first query and list of first 10 documents with largest similarity score as shown below.

  Here key represents document name and value represents score

  {'51060': 0.5610469391504762, '51144': 0.27498597046143514, '51120': 0.23981604843963608, '51164': 0.230541983810352, '51135': 0.21650635094610968, '51158': 0.2100420126042015, '51184': 0.20100756305184242, '51171': 0.19364916731037085, '51130': 0.17770466332772772, '51161': 0.17251638983558856}

- Cosine Similarity for second query and list of first 10 documents with largest similarity score as shown below.

  Here key represents document name and value represents score

  {'59905': 0.26579594988979643, '60103': 0.25147488875239266, '60170': 0.22541740866685805, '60210': 0.2198260025574647, '59850': 0.20988774008055677, '60195': 0.15075567228888181, '59909': 0.1495371511386448, '60198': 0.1307440900921227, '59870': 0.12947450111684894, '60200': 0.12751534261266767}

- Cosine Similarity for third query and list of first 10 documents with largest similarity score as shown below.

  Here key represents document name and value represents score

  {'10064': 0.22360679774997896, '10083': 0.17334381132038412, '10063': 0.17149858514250882, '10089': 0.13545709229571928, '10052': 0.13245323570650439, '10013': 0.12909944487358055, '10066': 0.11338934190276817, '59913': 0.09097816785925637, '10067': 0.08111071056538127, '101639': 0.07624928516630233}

- Cosine Similarity for fourth query and list of first 10 documents with largest similarity score as shown below.

  Here key represents document name and value represents score

  {'102656': 0.11180339887498948, '102660': 0.10540925533894598, '51207': 0.09759000729485333, '10027': 0.09325048082403138, '51151': 0.09263716756926595, '59849': 0.08679260732054925, '60213': 0.08272911497096004, '102675': 0.0778498944161523, '102626': 0.0659380473395787, '51206': 0.0659380473395787}

- Cosine Similarity for fifth query and list of first 10 documents with largest similarity score as shown below.

  Here key represents document name and value represents score

```
{'102598': 0.3922322702763681, '102610': 0.37369978746448757, '102647':
0.2773500981126145, '101666': 0.2305733966518285, '102609': 0.22011272658140596,
'100521': 0.20519567041703082, '102617': 0.19245008972987526, '102608':
0.173421993904824, '102634': 0.1643989873053573, '102622': 0.1608168802256692}
```

- Dot similarity for first query and list of first 10 documents with largest similarity score as shown below.

  Here key represents document name and value represents score

```
{'51060': 170, '49960': 21, '51153': 11, '51165': 10, '51164': 9, '51120': 7, '51144': 7,
'51130': 6, '51135': 6, '51161': 5}
```

- Dot similarity for second query and list of first 10 documents with largest similarity score as shown below.

  Here key represents document name and value represents score

```
{'59905': 64, '60103': 43, '59850': 30, '59874': 19, '59873': 17, '59909': 15, '59871': 12,
'59870': 11, '60210': 7, '59846': 6}
```

- Dot similarity for third query and list of first 10 documents with largest similarity score as shown below.

  Here key represents document name and value represents score

```
{'10011': 9, '59913': 7, '10083': 5, '59850': 5, '10089': 4, '10066': 3, '10074': 3, '59873': 3,
'59874': 3, '10013': 2}
```

- Dot similarity for fourth query and list of first 10 documents with largest similarity score as shown below.

  Here key represents document name and value represents score

```
{'59849': 10, '51151': 6, '59850': 6, '51211': 4, '59874': 3, '59905': 3, '60152': 3, '60213':
3, '101603': 2, '59908': 2}
```

- Dot similarity for fifth query and list of first 10 documents with largest similarity score as shown below.

  Here key represents document name and value represents score

```
{'102610': 23, '101666': 17, '102591': 7, '102604': 6, '102647': 6, '102609': 5, '102627': 5,
'100521': 4, '102598': 4, '102608': 4}
```

3. Cosine similarity takes into account the angle difference and ignores magnitude. However, dot product would be better similarity measure if magnitude plays a role.

- Here consider the cosine similarity and dot similarity of the documents with **first query**.

  For cosine similarity, the document with the second highest score is **51144**. For dot similarity, the document with second highest score is **49960**. If we compare these two documents, we could say that 49960 has more number of similar words to 1st query than 51144 document. So dot similarity has a better performance than cosine similarity in this case.

  49960 document contains words – christian, belief, religion

  51144 document contain words – religious

- Here consider the cosine similarity and dot similarity of the documents **with second query**.

  For cosine similarity, the document with the third highest score is **60170**. For dot similarity, the document with second highest score is **59850**. If we compare these two documents, we could say that 59850 has more number of similar words to 2nd query than 60170 document. So dot similarity has a better performance than cosine similarity in this case.

  59850 document contains words – astronaut, astronomy

  60170 document contain words - moon