Q1. Compute the BoW model, TF model, and IDF model for each of the terms in the following three sentences.
Then calculate the TF.IDF values

S1 “sunshine state enjoy sunshine”
S2 “brown fox jump high, brown fox run”
S3 “sunshine state fox run fast”

Q2. Compute the cosine similarity between S1 and S3.

ANSWEERS

```
import numpy as np
from numpy.linalg import norm


doc1 = "sunshine state enjoy sunshine"
doc2 = "brown fox jump high, brown fox run"
doc3="sunshine state fox run fast"
bow1 = doc1.split(" ")
bow2 = doc2.split(" ")
bow3 = doc3.split(" ")

terms = set(bow1).union(set(bow2)).union(set(bow3))

wordDict1 = dict.fromkeys(terms, 0)
wordDict2 = dict.fromkeys(terms, 0)
wordDict3 = dict.fromkeys(terms, 0)

for term in bow1:
    wordDict1[term]+=1
for term in bow2:
    wordDict2[term]+=1
for term in bow3:
    wordDict3[term]+=1
def computeTF(wordDict, bow):
    tfDict = {}
    bowCount = len(bow)
    for term, count in wordDict.items():
        tfDict[term] = count/float(bowCount)
    return tfDict
```

```
tfBow_1 = computeTF(wordDict1, bow1)
tfBow_2 = computeTF(wordDict2, bow2)
tfBow_3 = computeTF(wordDict3, bow3)

def computeIDF(docList):
    import math
    idfDict = {}
    N = len(docList)

    idfDict = dict.fromkeys(docList[0].keys(), 0)
    for doc in docList:
        for word, val in doc.items():
            if val > 0:
                idfDict[word] += 1

    for word, val in idfDict.items():
        idfDict[word] = math.log10(N / float(val))

    return idfDict

idf = computeIDF([wordDict1, wordDict2 , wordDict3])
def computeTFIDF(tfBow, idf):
    tf_idf = {}
    for term, val in tfBow.items():
        tf_idf[term] = val*idf[term]
    return tf_idf

tf_idf_Bow_1 = computeTFIDF(tfBow_1, idf)
tf_idf_Bow_2 = computeTFIDF(tfBow_2, idf)
tf_idf_Bow_3 = computeTFIDF(tfBow_3, idf)

tf_idf_Bow_1 = computeTFIDF(tfBow_1, idf)
tf_idf_Bow_2 = computeTFIDF(tfBow_2, idf)
tf_idf_Bow_3 = computeTFIDF(tfBow_3, idf)

pd.DataFrame([tf_idf_Bow_1, tf_idf_Bow_2 , tf_idf_Bow_3])
```

	sunshine	jump	high,	fox	fast	enjoy	brown	state	run
0	0.088046	0.00000	0.00000	0.000000	0.000000	0.11928	0.00000	0.044023	0.000000
1	0.000000	0.06816	0.06816	0.050312	0.000000	0.00000	0.13632	0.000000	0.025156
2	0.035218	0.00000	0.00000	0.035218	0.095424	0.00000	0.00000	0.035218	0.035218

```
import numpy as np
from numpy.linalg import norm


S1 = np.array([2,1,1,0,0,0,0,0,0])
S3 = np.array([1,1,0,0,1,0,0,1,1])
cose = np.dot(S1,S3)/(norm(S1)*norm(S3))
print("Cosine Similarity = ", cose)
```

```
Cosine Similarity =  0.5477225575051661
```