Problem 2 - vector distance

In [139]:

```
import numpy as np

u = np.random.randint(10)
v = np.random.randint(10)

vector1 = np.array([u,v]) # vector => u, v

u = np.random.randint(10)
v = np.random.randint(10)

vector2 = np.array([u,v]) # vector => u, v

print(vector1, vector2)
```

```
[9 7] [9 1]
```

Define functions

In [140]:

```
def dist_euclid(v1, v2):
    '''get Euclidean distance'''
    return np.sqrt(np.sum((v1-v2)**2))
```

In [141]:

```
def dist_cityblock(v1, v2):
    '''get Manhattan distance'''
    return np.sum(np.abs(v1-v2))
```

In [142]:

```
def dist_hamming(v1, v2):
    '''get Hamming distance'''
    dist = 0
    for i in range(len(v1)):
        if(v1[i] != v2[i]):
            dist += 1
    return dist
```

In [143]:

```
from numpy import dot
from numpy.linalg import norm

def dist_cosin(v1, v2):
    '''get Cosin distance'''
    return dot(v1, v2) / (norm(v1) * norm(v2))
```

In [144]:

```
def dist_jaccard(v1, v2):
    '''get Tanimoto distance'''
    union = set(v1).union(set(v2))
    intersection = set(v1).intersection(set(v2))
    return len(intersection) / len(union)
```

Test

In [145]:

```
print(dist_euclid(vector1, vector2))
print(dist_cityblock(vector1, vector2))
print(dist_hamming(vector1, vector2))
print(dist_cosin(vector1, vector2))
print(dist_jaccard(vector1, vector2))
```

```
6.0
6
1
0.8523227286486657
0.3333333333333333
```