In [16]:
```
n = int(input('Enter the numbers in the vector :'))

#creating vectors
vect1 = []
vect2 = []

print('Enter the elements of the first vector')
for i in range(n):
    inp = int(input())
    vect1.append(inp)

print('Enter the elements of the first vector')
for i in range(n):
    inp1 = int(input())
    vect2.append(inp1)
```

```
Enter the numbers in the vector :4
Enter the elements of the first vector
3
2
7
5
Enter the elements of the first vector
9
3
2
5
```

In [17]:
```
from math import*

def euclidean_dist(x,y):
    return sqrt(sum(pow(a-b, 2) for a,b in zip(x,y)))

def manhatten_dist(x,y):
    return sum(abs(a-b) for a, b in zip(x,y))

from decimal import Decimal

def nth_value(value , n_root):
    root_value = 1/float(n_root)
    return round(Decimal(value)**Decimal(root_value) , 3)
```

```
def minkowski_dist(x,y,p_value):
    return nth_value(sum(pow(abs(a-b),p_value) for a,b in zip(x,y)), p_value)
```

In [18]:
```
print('Euclidean Distance', euclidean_dist(vect1,vect2))
```

```
Euclidean Distance 7.874007874011811
```

In [19]:
```
print('manhattan Distance', manhatten_dist(vect1,vect2))
```

```
manhattan Distance 12
```

In [20]:
```
print('minkowski Distance', minkowski_dist(vect1,vect2,3))
```

```
minkowski Distance 6.993
```

In [35]:
```
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity

n = int(input('Enter the numbers in the vector :'))

#creating vectors
vect1 = []
vect2 = []

print('Enter the elements of the first vector')
for i in range(n):
    inp = int(input())
    vect1.append(inp)

print('Enter the elements of the first vector')
for i in range(n):
    inp1 = int(input())
    vect2.append(inp1)

print("A:", vect1)
print("B:", vect2)
```

```
Enter the numbers in the vector :4
Enter the elements of the first vector
2
3
2
4
Enter the elements of the first vector
5
3
4
1
A: [2, 3, 2, 4]
B: [5, 3, 4, 1]
```

In [36]:
```
from numpy.linalg import norm
```

In [37]:
```
cosine = np.dot(vect1,vect2)/ (norm(vect1)*norm(vect2))
print("Cosine Similarity :", cosine)
```

```
Cosine Similarity : 0.755648171598035
```

In [ ]: