

# AI VIET NAM – COURSE 2024

## Numpy - Exercise

Ngày 21 tháng 7 năm 2024

Ngày thực hiện:	22/07/2024
Người thực hiện:	Đinh Thị Tâm
Nguồn:	AIO2024 - Module2 - Week2
Nguồn dữ liệu (nếu có):	<a href="#">Link of Data Sources</a>
Từ khóa:	Module 02 - Vector Exercise
Người tóm tắt:	Đinh Thị Tâm

### 1 Bài tập.

- Độ dài của vector:

– Code

```
1 def compute_vector_length(vector):  
2     len_of_vector = np.linalg.norm(vector)  
3     return len_of_vector
```

- Phép tích vô hướng

– Code

```
1 def compute_dot_product(vector1, vector2):  
2     result = np.dot(vector1, vector2)  
3     return result
```

- Nhân vector với ma trận

– Code

```
1 def matrix_multi_vector(matrix, vector):  
2     result = np.dot(matrix, vector)  
3     return result
```

- Nhân ma trận với ma trận:

– Code

```
1 def matrix_multi_matrix(matrix1, matrix2):  
2     result = np.dot(matrix1, matrix2)  
3     return result
```

- Ma trận nghịch đảo:

– Code

```
1 def inverse_matrix(matrix):
2     det_matrix = np.linalg.det(matrix)
3     if det_matrix != 0:
4         result = np.linalg.inv(matrix)
5     else:
6         result = None
7     return result
```

## 2 Trắc nghiệm:

– Câu 1:

```
1 import numpy as np
2 vector = np.array ([-2, 4, 9, 21])
3 result = compute_vector_length ([ vector ])
4 print ( round (result ,2))
```

– Câu 2:

```
1 v1 = np. array ([0 , 1, -1, 2])
2 v2 = np. array ([2 , 5, 1, 0])
3 result = compute_dot_product (v1 , v2)
4 print ( round (result ,2))
5
```

– Câu 3:

```
1 x = np. array ([[1 , 2],
2 [3, 4]])
3 k = np. array ([1 , 2])
4 print ('result \n', x.dot (k))
5
```

– Câu 4:

```
1 x = np.array ([[ -1 , 2],
2 [3, -4]])
3 k = np.array ([1 , 2])
4 print ('result \n', x@k)
5
```

– Câu 5:

```
1 m = np.array ([[ -1 , 1, 1], [0, -4, 9]])
2 v = np.array ([0 , 2, 1])
3 result = matrix_multi_vector (m, v)
4 print ( result )
5
```

– Câu 6:

```
1 m1 = np.array ([[0 , 1, 2], [2, -3, 1]])
2 m2 = np.array ([[1 , -3] ,[6 , 1], [0, -1]])
3 result = matrix_multi_matrix (m1 , m2)
4 print ( result )
5
```

– Câu 7:

```
1 m1 = np.eye (3)
2 m2 = np.array ([[1 , 1, 1] , [2 , 2, 2], [3, 3, 3]])
3 result = m1@m2
4 print ( result )
5
```

– Câu 8:

```
1 m1 = np.eye (2)
2 m1 = np.reshape (m1 ,( -1 ,4))[0]
3 m2 = np.array ([[1 , 1, 1, 1] , [2 , 2, 2, 2], [3, 3, 3, 3], [4, 4, 4, 4]])
4 result = m1@m2
5 print ( result )
6
```

– Câu 9:

```
1 m1 = np. array ([[1 , 2], [3, 4]])
2 m1 = np. reshape (m1 ,( -1 ,4) , "F")[0]
3 m2 = np. array ([[1 , 1, 1, 1] , [2 , 2, 2, 2], [3, 3, 3, 3], [4, 4, 4, 4]])
4 result = m1@m2
5 print ( result )
6
```

– Câu 10:

```
1 m1 = np.array ([[ -2 , 6], [8, -4]])
2 result = inverse_matrix (m1)
3 print ( result )
4
```

– Câu 11:

```
1 matrix = np.array ([[0.9 , 0.2] , [0.1 , 0.8]])
2 eigenvalues , eigenvectors = compute_eigenvalues_eigenvectors ( matrix )
3 print ( eigenvectors )
4
```

– Câu 12:

```
1 x = np.array ([1 , 2, 3, 4])
2 y = np.array ([1 , 0, 3, 0])
3 result = compute_cosine (x,y)
4 print ( round (result , 3))
5
```