

Chapter 3 - Exercise 3: Thao tác trên mảng nhiều chiều, và đổi chiều với kết quả cho trước:

```
In [1]: import numpy as np
```

```
In [2]: # Câu 1: Tạo array arr có kích thước 3x3 với tất cả các giá trị đều là True.  
arr = np.ones((3,3), dtype=bool)  
# Hiển thị các phần tử của array arr  
print(arr)
```

```
[[ True  True  True]  
 [ True  True  True]  
 [ True  True  True]]
```

```
In [3]: # Câu 2: Cho array arr_1D = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8]).  
arr_1D = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8])  
print(arr_1D)  
# Hãy chuyển thành array 2 chiều có kích thước 3x3 và Lưu vào arr_2D.  
arr_2D = arr_1D.reshape((3,3))  
# Xem danh sách các phần tử của arr_2D.  
print(arr_2D)  
# Trong arr_2D. Chuyển cột 1 sang cột 3 và ngược lại. =>  
# Xem lại danh sách các phần tử của arr_2D  
arr_2D = arr_2D[:,[2,1,0]]  
print(arr_2D)
```

```
[0 1 2 3 4 5 6 7 8]  
[[0 1 2]  
 [3 4 5]  
 [6 7 8]]  
[[2 1 0]  
 [5 4 3]  
 [8 7 6]]
```

```
In [4]: print(arr_2D)
```

```
[[2 1 0]  
 [5 4 3]  
 [8 7 6]]
```

```
In [5]: # Câu 3: Sử dụng array arr_2D của câu 2 (sau khi đổi thứ tự cột),  
# chuyển dòng 1 sang dòng 2 và ngược lại  
# code here
```

Nhấn vào đây để xem kết quả!

```
[[5 4 3]  
 [2 1 0]  
 [8 7 6]]
```



```
In [6]: # Câu 4: Sử dụng array arr_2D của câu 3,  
# Đảo ngược các dòng của array arr_2D  
# code here
```

Nhấn vào đây để xem kết quả!

```
[[8 7 6]  
 [2 1 0]  
 [5 4 3]]
```

```
In [7]: # Câu 5: Sử dụng array arr_2D của câu 4, Đảo ngược các cột của array arr_2D =>  
# Xem lại danh sách các phần tử của arr_2D  
# code here
```

Nhấn vào đây để xem kết quả!

```
[[6 7 8]  
 [0 1 2]  
 [3 4 5]]
```

```
In [8]: # Câu 6: Cho arr_2D_null = np.array([[1, 2, 3], [np.NaN, 5, 6], [7, np.NaN, 9],  
#      [4, 5, 6]])  
arr_2D_null = np.array([[1, 2, 3], [np.NaN, 5, 6], [7, np.NaN, 9], [4, 5, 6]])  
print(arr_2D_null)  
print(np.where(np.isnan(arr_2D_null).any()==True,  
              'array có giá trị rỗng', 'array không có giá trị rỗng'))
```

```
[[ 1.  2.  3.]  
 [nan  5.  6.]  
 [ 7. nan  9.]  
 [ 4.  5.  6.]]  
array có giá trị rỗng
```

```
In [9]: # Câu 7: Sử dụng array arr_2D_null của câu 6, thay thế giá trị null bằng 0 =>  
# Xem lại danh sách các phần tử của arr_2D_null  
print('Before:')  
print(arr_2D_null)  
arr_2D_null[np.isnan(arr_2D_null)] = 0  
print('After:')  
print(arr_2D_null)
```

```
Before:  
[[ 1.  2.  3.]  
 [nan  5.  6.]  
 [ 7. nan  9.]  
 [ 4.  5.  6.]]  
After:  
[[1.  2.  3.]  
 [0.  5.  6.]  
 [7.  0.  9.]  
 [4.  5.  6.]]
```

```
In [ ]:
```


