

▼ Chapter 3 - Exercise 1c:

▼ Các kiến thức sử dụng trong bài tập:

Tạo mảng (numpy array):

1. Tạo mảng là 1 dãy số bắt đầu từ a đến b - 1: hàm **arange**
2. Tạo mảng từ 1 dãy số cho trước
3. Tạo mảng với các phần tử là số 0: hàm **zeros**
4. Tạo mảng theo điều kiện từ mảng đã có

Các thao tác trên mảng:

1. Thêm phần tử vào cuối mảng đang có: hàm **append**
2. Chèn thêm phần tử vào vị trí bất kỳ trong mảng đang có: hàm **insert**
3. Xóa phần tử trong mảng: hàm **delete**

▼ Thực hiện các yêu cầu sau và đối chiếu với kết quả cho trước:

```
import numpy as np
```

```
# Câu 1: Tạo array arr_zeros có 10 phần tử 0, cập nhật phần tử ở vị trí thứ 5 là 1
arr_zeros = np.zeros(10)
print(arr_zeros)
```

```
arr_zeros[4] = 1
print(arr_zeros)
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
```

► Nhấn vào đây để xem kết quả !

```
# Câu 2: Tạo và in array arr_h có giá trị từ 10 đến 24
arr_h = np.arange(10, 25)
print(arr_h)
```

```
# In danh sách các phần tử theo tứ tự đảo ngược của arr_h vừa tạo
print(arr_h[::-1])
```

```
[10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]
[24 23 22 21 20 19 18 17 16 15 14 13 12 11 10]
```

► Nhấn vào đây để xem kết quả !

```
# Câu 3: Cho array arr_k = [1, 2, 0, 8, 2, 0, 1, 3, 0, 5, 0]
# Tạo array arr_l từ arr_k với các phần tử khác 0

arr_k = np.array([1, 2, 0, 8, 2, 0, 1, 3, 0, 5, 0] )
print(arr_k)

arr_l = arr_k[arr_k != 0]
print(arr_l)
```

```
[1 2 0 8 2 0 1 3 0 5 0]
[1 2 8 2 1 3 5]
```

► Nhấn vào đây để xem kết quả !

```
# Câu 4: Từ array arr_l của câu 3, thêm 2 phần tử có giá trị là 10 và 20 vào cuối array
arr_l = np.append(arr_l, [10, 20])
print(arr_l)
```

```
[ 1  2  8  2  1  3  5 10 20]
```

► Nhấn vào đây để xem kết quả !

```
# Câu 5: Từ array của câu 4, thêm phần tử có giá trị 100 vào vị trí có index = 5

arr_l = np.insert(arr_l, 5, 100)
print(arr_l)
```

```
[ 1  2  8  2  1 100  3  5 10 20]
```

► Nhấn vào đây để xem kết quả !

```
# Câu 6: Từ array của câu 5, xóa các phần tử tại vị trí có index = 0, 1, 2

arr_l = np.delete(arr_l, [0, 1, 2])
print(arr_l)
```

```
[ 2  1 100  3  5 10 20]
```

► Nhấn vào đây để xem kết quả !

