

LAB 6_ Đáp án : Numpy

1. Tạo một ma trận 4x4 toàn các giá trị False

```
import numpy as np
a = np.full((4, 4), False, dtype = bool)
print(a)
```

2. Cho một dãy số nguyên 100 phần tử, hãy tách lấy tất cả những phần tử lẻ cho vào một mảng

```
import numpy as np
a = np.random.randint(1000, size=100)
b = a[a % 2 == 1]
print(b)
```

3. Cho một dãy số tự nhiên 20 phần tử, hãy thay thế tất cả những phần tử lẻ bằng số -1

```
import numpy as np
a = np.random.randint(100, size=20)
a[a % 2 == 1] = -1
print(a)
```

4. Hai mảng a và b có cùng số dòng, hãy ghép chúng theo các dòng thành mảng c, các cột của a rồi đến các cột của b

```
import numpy as np
a = np.array([[1,2,3],[4,5,6]])
b = np.array([[1,2,3],[4,5,6]])
#dùng hàm concatenate
c = np.concatenate([a, b], axis=1)
print(c)
```

5. Mảng a và b có cùng số cột, hãy ghép chúng theo các cột thành mảng c, các dòng của a rồi đến của b

```
import numpy as np
a = np.array([[1,2,3],[4,5,6]])
b = np.array([[1,2,3],[4,5,6]])
#dùng hàm concatenate
c = np.concatenate([a, b], axis=0)
print(c)
```

6. Cho một mảng a, hãy in ra tất cả những phần tử trong khoảng từ 5 đến 10

```
import numpy as np
a = np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15])
print(a[5:11])
```

7. Tạo ra một mảng số thực có 1000 phần tử, các phần tử nằm trong khoảng từ -0.5 đến <0.5

```
import numpy as np
a = np.random.uniform(low=-0.5, high=0.5, size=(1000))
print(a)
```

8. Tạo ra một ma trận 3x5 gồm các số ngẫu nhiên từ 0 đến nhỏ hơn 10, tính và in ra số lớn nhất trên mỗi dòng của ma trận

```
import numpy as np
a = np.random.uniform(low=0, high=10, size=(15))
b = a.reshape(3,5)
#In Mảng 1 chiều có 3x5=15 phần tử
print(a, "\n")
# In ra ma trận(3,5)
print(b)
print("Phần tử lớn nhất là :", np.max(b))
```

9. Nhập mảng a và b có 10 phần tử, tính khoảng cách euclid giữa a và b
Hướng dẫn:

$\text{dist} = \sqrt{(a_1-b_1)^2 + (a_2-b_2)^2 + (a_3-b_3)^2 \dots}$

```
import numpy as np
import math
a = np.random.randint(100,size=10)
print(a)
b = np.random.randint(100,size=10)
print(b)
def kc(a,b):
    return math.sqrt(np.sum(a-b)**2)
print (kc(a,b))
```