print(np.all([10, 20, 0, -50]), np.all([10, 20, -50]))

Output của code trên là  Flase True

Hàm all(iterator) trả vể true nếu iterator rỗng hoặc tất cả giá trị là true, ngược lại trả về false

M = np.array([[6, 1, 1],

[4, -2, 5],

[2, 8, 7]])

print(np.linalg.matrix\_power(M,2))

Output của code trên là

[[42 12 18] [26 48 29] [58 42 91]]

numpy.linalg.matrix\_power(matrix,n) là raise matrix to the power of n

import numpy as np  
x = np.arange(1,28).reshape(3,3,3)  
print(x[Trả lời])

Điền dữ liệu thích hợp vào chỗ trống trong đoạn code trên để output là:

[[ 1 4 7]  
[10 13 16]  
[19 22 25]]

**Không được sử dụng ký tự trắng trong trả lời của bạn.**

:,:,0

import numpy as np

y = np.ones((2,3),dtype=np.int16)

print(y)

[[1 1 1] [1 1 1]]

ones(ind1, ind2) tạo ra ma tan ind1 hàng ind2 cột toàn bộ mang giá trị là 1

import numpy as np

x = np.array([7,2,3])

y = np.array([4,5,6])

z = x-y

print(z)

[3 -3 -3]

Lấy phần tử ở các vị trí đối xứng với nhau trừ nhau

#phep toan nhi phan

print(np.right\_shift(20,2))

20 🡪 0001 0100 🡪 shift\_right 2 🡪 0000 0101 🡪 4+1=5

5

x = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])

# Chi so co the âm

print(x[np.array([1, 3, -3])])

[2 4 7]

import numpy as np

y = np.ones((2,3))

print(y)

[[1. 1. 1.] [1. 1. 1.]]

import numpy as np  
m = np.array([48,36,12,20,28])  
print(m\*3)

[144 108 36 60 84]

import numpy as np  
m = np.array([48,36,22,20,28])  
print(m-1)

[47 35 21 19 27]

import numpy as np  
x = np.array([[2, 11, 0], [4, 3, 5]])  
y = x.sum(axis=1)  
print(y)

[13 12]

Trong một mảng 2 chiều, ví dụ, có hai trục: trục 0 (axis=0) là hàng và trục 1 (axis=1) là cột.

import numpy as np  
print(np.arange(9).reshape(3,3)[Trả lời])

Sử dụng slicing, điền dữ liệu thích hợp vào ô trống trong đoạn code trên để output của nó là:

[4 7]

**Không được sử dụng ký tự trắng trong trả lời của bạn.**

1:,1

Kiểm tra xem mảng có bao nhiêu chiều:

import numpy as np

x = np.array([[2,8],[3,7]])

print(x.Trả lời)

ndim

import numpy as np  
m = np.array([48,36,22,20,28])  
print(m+5)

[53 41 27 25 33]

x = [1, 2, 8]

x = np.append(x, [[4, 5, 9], [7, 3, 6]])

print(x)

[1 2 8 4 5 9 7 3 6]

.append(….) tác dụng:

🡪 dúng hết

import numpy as np

x = np.array([7,2,3])

y = np.array([4,5,6])

z = x\*y

print(z)

[28 10 18]

import numpy as np

x = np.array([1,2,3])

y = np.array([4,5,6])

z = x+y

print(z)

[5 7 9]

print(np.any([10, 20, 0, -50]), np.any([10, 20, -50]))

True True

any(iterator) : bất kì phần tử nào đúng trả về true

Tạo mảng numpy:

import numpy as np

x = np.Trả lời([7,2,3])

array

import numpy as np

x = np.array([7,2,3])

y = np.array([4,5,6])

z = x/y

print(z)

[1.75 0.4 0.5]

import numpy as np

x = np.arange(1,6)

print(x)

[1 2 3 4 5]

hàm arange(start,end,step) : tạo dãy số

import numpy as np  
x = np.arange(1,28).reshape(3,3,3)  
print(x[Trả lời])

Điền dữ liệu thích hợp vào chỗ trống trong đoạn code trên để output là:

[[ 1 4 7]  
[10 13 16]  
[19 22 25]]

**Không được sử dụng ký tự trắng trong trả lời của bạn.**

:,:,0

Tạo mảng 0D chứa giá tri 18:

import numpy as np

x = np.Trả lời

array(18)

x = [1, 2, 8]

x = np.append(x, [[4, 5, 9], [7, 3, 6]])

print(x)

[1 2 8 4 5 9 7 3 6]

x = np.arange(10, 1, -2)

#Chỉ số được đặc tả bên trong np.array

print(x[np.array([3,1,2)])

[4 8 6]

Or ERROR

x = np.array([10, 10, 20, 30, 30], float)   
y = np.array([0, 40, 60], float)   
print(x.put([0, 4], y))

None

Hàm put(…) có tác dụng:

import numpy as np  
print(np.arange(1,28).reshape(3,3,3)[Trả lời])

Sử dụng slicing, điền dữ liệu thích hợp vào ô trống trong đoạn code trên để output của nó là:

[[[ 1 2]  
[ 4 5]]  
[[10 11]  
[13 14]]]

**Không được sử dụng ký tự trắng trong trả lời của bạn.**

:2,:2,:2

import numpy as np  
x = np.array([[2, 1, 0], [4, 3, 5]])  
y = x.sum()  
print(y)

15

x = np.array([0,1,2,4,6,8])

y = [1,3,4,5,7]

print(np.setdiff1d(x,y))

[0 2 6 8]

Tính tổng phần tử thứ 3 và phần tử thứ 4 :

import numpy as np

x = np.array([1,2,3,4])

print(x[2]+x[3])

7

🡪 sai 2

x = np.arange(10, 1, -2)  
print(x)

[10 8 6 4 2]

import numpy as np  
a = np.array([[1,2,3,4,5], [6,7,8,9,10]])  
print('Phần tử thứ hai trên dòng thứ nhất: ', a[Trả lời])

0,1

🡪 đúng 1

Tạo mảng 0D chứa giá tri 18:

import numpy as np

x = np.array(18)

import numpy as np

x = np.arange(1,6)

print(x)

[1 2 3 4 5]

import numpy as np

y = np.ones((2,3))

print(y)

[[1. 1. 1.] [1. 1. 1.]]

import numpy as np  
def f(A):  
    return A[:(A.shape[1]Trả lời]  
print(f(np.arange(16).reshape(4,4)))  
print(f(np.arange(9).reshape(3,3)))

Điền phần lệnh đang bị thiếu vào chỗ trống trong đoạn code trên để hàm f trả về phần ma trận con thuộc góc trên bên trái của ma trận vuông A

Output của code trên là:

[[0 1]  
[4 5]]  
[[0 1]  
[3 4]]

-A.shape[1]//2),:2 🡪 sai

return A[:, : A.shape[1] // 2] # Select all rows, columns up to half

🡪 đúng 3

Tính tổng phần tử thứ 3 và phần tử thứ 4 :

import numpy as np

x = np.array([1,2,3,4])

print(x[2]+x[3])

7

import numpy as np  
print(np.arange(9).reshape(3,3)[Trả lời])

Điền dữ liệu thích hợp vào ô trống trong đoạn code trên để output của nó là:

[[1]  
[4]  
[7]]

**Không được sử dụng ký tự trắng trong trả lời của bạn.**

:,1:2

x = np.arange(10, 1, -2)  
print(x)

[10 8 6 4 2]

import numpy as np

y = np.ones((2,3),dtype=np.int16)

print(y)

[[1 1 1] [1 1 1]]

x = [1,5]

print(np.tile(Trả lời))

Viết phần lệnh bị thiếu để output là  [1 5 1 5 1 5]

x,3

hàm tile(….) có tác dụng:

x = np.array([0, 10, 20, 40, 60])

y = [0, 40]

print(np.in1d(x, y))

[True False False True False]

Hàm in1d(interator1, iterator2) có tác dụng gì:

m = np.array([10, 40, 80, 50, 100])

# Chỉ số bằng biểu thức kiểu boolean

print(m[m>50])

[80 100]

import numpy as np  
def f(A):  
    return A[Trả lời]  
x= np.array([1,4,7,10,13,16,19,22,25]).reshape(3,3)  
print(f(x))

Điền dữ liệu thích hợp vào chỗ trống trong đoạn code trên để hàm f trả về các phần tử trên hình chữ thập băng qua tâm của ma trận A(3x3).

Output của code trên là mảng:  
[10 13 16 4 22]

**Không được sử dụng ký tự trắng trong trả lời của bạn.**

[1,1,1,0,2],[0,1,2,1,1]

import numpy as np  
m = np.array([48,36,22,20,28])  
print(m+5)

[53 41 27 25 33]

#phep toan nhi phan

print(np.right\_shift(20,2))

5

import numpy as np

x = np.array([7,2,3])

y = np.array([4,5,6])

z = x\*y

print(z)

[28 10 18]

import numpy as np  
x = np.array([[2, 1, 0], [4, 3, 5]])  
y = x.sum()  
print(y)

15

import numpy as np  
m = np.array([48,36,22,20,28])  
print(m-1)

[47 35 21 19 27]

import numpy as np  
x = np.array([[2, 11, 7], [4, 3, 5]])  
y = x.min()  
print(y)

2

import numpy as np  
arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])  
print('Phần tử cuối cùng từ chiều thứ hai: ', arr[1, -1])

Output là:  Phần tử cuối cùng từ chiều thứ hai:

10

🡪 12/15

x = np.array([[1, 1], [2, 3]])

print(np.unique(x))

[1 2 3]

import numpy as np  
x = np.array([[2, 1, 0], [4, 3, 5]])  
y = x.sum()  
print(y)

15

M = np.array([[6, 1, 1],

[4, -2, 5],

[2, 8, 7]])

print(np.trace(M))

Output của code trên là

11

x = [1,5]

print(np.tile(Trả lời))

Viết phần lệnh bị thiếu để output là  [1 5 1 5 1 5]

x,3

import numpy as np  
m = np.array([48,36,22,20,28])  
print(m/2)

[24. 18. 11. 10. 14.]

import numpy as np  
x = np.array([[5],[2],[8]])  
y = np.array([[7],[6],[1]])  
z = Trả lời

Sử dụng hàm stack thích hợp điền vào chỗ trống để z là:

array([[[5, 7]], [[2, 6]], [[8, 1]]])

**Không được sử dụng ký tự trắng trong trả lời của bạn.**

np.stack((x,y),axis=2)

import numpy as np

x = np.array([7,2,3])

y = np.array([4,5,6])

z = x-y

print(z)

[3 -3 -3]

import numpy as np  
x = np.array([[2, 11, 7], [4, 3, 5]])  
y = x.min()  
print(y)

2

Tạo mảng 0D chứa giá tri 18:

import numpy as np

x = np.Trả lời

array(18)

import numpy as np  
x = np.array([[2, 11, 0], [4, 3, 5]])  
y = x.sum(axis=1)  
print(y)

[13 12]

import numpy as np  
x = np.array([9,2,3])  
y = np.array([6,7,4])  
z = Trả lời

Sử dụng hàm stack thích hợp điền vào chỗ trống để z là:

array([[[6, 9], [7, 2], [4, 3]]])

**Không được sử dụng ký tự trắng trong trả lời của bạn.**

np.stack((y,x),axis=1)

import numpy as np

x = np.array([1,2,3])

y = np.array([4,5,6])

z = x+y

print(z)

[5 7 9]

x = np.array([0, 10, 20, 40, 60])

y = [0, 40]

print(np.in1d(x, y))

Output của code trên là

[True False False True False]

import numpy as np  
m = np.array([48,36,12,20,28])  
print(m\*3)

[144 108 36 60 84]

x = np.array([7, 10, 20, 40, 60])

y = [7, 30, 10]

print(np.intersect1d(x, y))

[7 10]

🡪 11/15

import numpy as np  
x = np.array([[2, 11, 7], [4, 3, 18]])  
y = x.max()  
print(y)

18

x = np.array([1,2,3], dtype=np.float64)

print(x.size, x.itemsize, x.nbytes)

3 8 24

#phep toan nhi phan

print(np.bitwise\_and(10,11))

10

🡪 đúng 3 câu

Hàm f có công dụng biến đổi mảng 1 chiều x thành mảng 2 chiều y(n,m) với các dòng của y được lấy từ x khi duyệt x với bước duyệt s

import numpy as np  
def f(x,m,s):  
    # n = ...

    return np.array([x[k:k+m] Trả lời])  
print('Vidu 1:\n',f(np.arange(10),4,3))  
print('Vidu 2:\n',f(np.arange(10),5,2))

Giả sử n đã được xác định, hãy điền phần lệnh bị thiếu vào ô trống trong đoạn code trên để output của code trên là:

Vidu 1:  
[[0 1 2 3]  
[3 4 5 6]  
[6 7 8 9]]  
Vidu 2:  
[[0 1 2 3 4]  
[2 3 4 5 6]  
[4 5 6 7 8]]

**Không được sử dụng ký tự trắng trong trả lời của bạn trừ khi bắt buộc phải phân cách giữa các từ khóa, định danh**

for i in range(n)

Điền 4 ký tự vào chỗ trống để in ra phần tử thứ 2 của x

x = ["xoai", "mit", "chuoi"]  
print(Trả lời)

x[1]

import numpy as np  
m = np.array([48,36,22,20,28])  
print(m-1)

[47 35 21 19 27]

#phep toan nhi phan

print(np.bitwise\_or([2, 8],[3, 3]))

Output của code trên là

[3 11]

Tạo mảng numpy:

import numpy as np

x = np.Trả lời([7,2,3])

array

Kiểm tra xem mảng có bao nhiêu chiều:

import numpy as np

x = np.array([[2,8],[3,7]])

print(x.Trả lời)

ndim

import numpy as np  
x = np.array([[[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]])  
print(x[0, 1, 2])

6

#phep toan nhi phan

print(np.right\_shift(20,2))

Output của code trên là

5

import numpy as np  
print(np.arange(1,28).reshape(3,3,3)[Trả lời])

Sử dụng slicing, điền dữ liệu thích hợp vào ô trống trong đoạn code trên để output của nó là:

[[[ 1 2]  
[ 4 5]]  
[[10 11]  
[13 14]]]

**Không được sử dụng ký tự trắng trong trả lời của bạn.**

:,:,0

import numpy as np  
arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])  
print('Phần tử cuối cùng từ chiều thứ hai: ', arr[1, -1])

10

import numpy as np  
a = np.array([[1,2,3,4,5], [6,7,8,9,10]])  
print('Phần tử thứ hai trên dòng thứ nhất: ', a[Trả lời])

0,1

M = np.array([[6, 1, 1],

[4, -2, 5],

[2, 8, 7]])

print(np.linalg.det(M))

-306.0

import numpy as np

y = np.ones((2,3))

print(y)

[[1. 1. 1.] [1. 1. 1.]]

x = np.array([0,1,2,4,6,8])

y = [1,3,4,5,7]

print(np.setdiff1d(x,y))

[0 2 6 8]

import numpy as np

x = np.array([7,2,3])

y = np.array([4,5,6])

z = x\*y

print(z)

[28 10 18]

13/14