import numpy as np

#tablice

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

b=np.array([[1,2,3,4,5], [6,7,8,9,10]])

#transpozycja b

np.transpose(b)

#utworzenie 100-elementowej tablicy

ar=np.arange(100)

print(ar)

#tablica 10 elemntów od 0 do 2

ar=np.linspace(0,2,num=10)

print(ar)

#tablica elementow od 0 do 100 z krokiem co 5

ar=np.arange(0,101,5)

print(ar)

#liczby losowe

#tablica 20 liczb rozkladu normalnego i zaokraglenie jej do 2 miejsc po przecinku

ar=np.random.randn(20)

ar=np.around(ar,2)

print(ar)

#tablica o rozmiarze 100 losowych liczb calkowitych z zakresu 1-1000

ar=np.random.randint(1,1001,size=100)

print(ar)

#tablica 0 3x2

ar=np.zeros((3,2))

print(ar)

#tablica jedynek 3x2

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

print(ar)

#macierz losowych liczb calkowitych o rozmiarze 5x5

ar=np.random.randint(1000,size=(5,5),dtype='int32')

print(ar)

a = np.linspace(0.0, 10.0, num=10)

b = a.astype(int)

a = np.around(a,0)

a = a.astype(int)

print('a = ',a)

print('b = ',b)

#np.around zaokragla nam liczbe w gore od 5 wzwyz, natomiast astype ucina nam koncowke po przecinku

#Selekcja danych

ar=np.array([[1,2,3,4,5], [6,7,8,9,10]],dtype=np.int32)

#ilosc wymiarow

print(np.ndim(ar))

#ilosc elementow

print(np.size(ar))

#wybór elementu o indeksie 2 i 4 począwszy od 0

print(ar[0,2],',',ar[0,4])

#lub wybór dokładnej wartości 2 i 4

print(ar[0,1],',',ar[0,3])

#wybór pierwszego wiersza tablicy

print(ar[0,:])

#wierszy z pierwszej kolumny

print(ar[:,0])

#losowanie macierzy liczb calkowitych z przedziału 0-100 za pomocą funkcji randint oraz utworzenie macierzy 20x7

ar=np.random.randint(101,size=(20,7))

print(ar)

print(ar[:,(0,1,2,3)])

#operacje matematyczne i logiczne

#stworzenie macierzy w przedziale od 0 do 10 o rozmiarze 3x3

a=np.random.randint(0,11,size=(3,3))

b=np.random.randint(0,11,size=(3,3))

#dodawanie

print(a+b)

#mnozenie

print(a\*b)

#dzielenie

print(a/b)

#potegowanie

print(a\*\*b)

#sprawdzenie czy a >= 4

print(a>=4)

#sprawdzenie warunku 1>= <= 4, czyli patrząc od końca a<=4, ale mamy warunek 1>=a, więc sprawdzamy czy a należy do przedziału <0;1>

print(a<=1)

#sumowanie głownej przekątnej macierzy b

print(np.trace(b))

#dane statystyczne

#wykorzystujemy macierz b z poprzedniego zadania

print(b.sum(), b.min(), b.max(), np.std(b))

#0 oznacza kolumny 1 oznacza wiersze

print(b.mean(axis=1))

print(b.mean(axis=0))

#rzutowanie wymiarów

#stworzenie macierzy z liczbami w zakresie 0-100 wielkosc 50

ar=np.random.randint(0,101,size=50)

#reshape oraz resize do macierzy 10x5

ar1=ar.reshape(10,5)

ar2=ar.resize(10,5)

#ravel zwraca jednowymiarowa tablicę

print(ar.ravel())

a=np.random.randint(0,11,size=5)

b=np.random.randint(0,11,size=4)

#nie możemy dodać dwóch macierzy o różnych rozmiarach

#print(a+b)

#newaxis podnosi wymiar macierzy o 1 do góry

b\_new=b[:,np.newaxis]

#dzieki temu mamy mozliwosc dodania macierzy

print(a+b\_new)

#sortowanie danych

a=np.random.randn(5,5)

#sortowanie kolumn macierzy

print(-np.sort(a,axis=0))

#sortowanie wierszy macierzy

print(np.sort(a,axis=1))

b=np.array([(1,'MZ','mazowieckie'),(2,'ZP','zachodniopomorskie'),(3,'ML','malopolskie')])

print(b)

#nie do konca wiem jak rozwaizac to zadanie

#PODSUMOWANIE

#zadanie 1

#tworzymy macierz 10x5 liczb calkowitych i używamy metod z polecenia

ar=np.random.randint(0,101,size=(10,5))

print('Suma glownej przekatnej: ', ar.trace())

print('Glowna przekątna: ', np.diag(ar))

#zadanie 2

#tworze macierze z rozkaldu noramlengo i je mnoze

a = np.random.randn(14)

b = np.random.randn(14)

print(a\*b)

#zadanie 3

a = np.random.randint(1, 101, size=15).reshape(3,5)

b = np.random.randint(1, 101, size=15).reshape(3,5)

print(a+b)

#zadanie 4

a = np.random.randint(0,11,(5,4))

b = np.random.randint(0,11,(4,5))

ans = a+b.swapaxes(0,1)

print(ans)

#zadanie 5

print(a[:,(2,3)]\*b.swapaxes(0, 1)[:,(2,3)])

#zadanie 6

a=np.random.normal(size=10)

print('a: mean =',a.mean(),'std =',a.std(),'var =',a.var())

b=np.random.uniform(size=10)

print('b: mean =',b.mean(),'std =',b.std(),'var =',b.var())

#zadanie 7

a=np.random.randint(0,51,size=(2,2))

b=np.random.randint(0,51,size=(2,2))

print('a\*b =',a\*b,'\na.dot(b) =',a.dot(b))

#dot to procedura mnożenia mecierzy, a samo a\*b oferuje nam wymnożenie danych elementów macierzy przez siebie

#zadanie 8

x = np.array([[0, 1, 2, 3, 4],

[5, 6, 7, 8, 9],

[10, 11, 12, 13, 14],

[15, 16, 17, 18, 19]], dtype=np.int32)

print(x.strides)

#x - nasz tablica z ktorej wybieramy dane

#(2,3) 2 - ilosc wierszy, ktore zostana wyswietlone w nowej tablicy i analogicznie 3 liczba kolumn

y=np.lib.stride\_tricks.as\_strided(x,(2,3))

print(y)

#zadanie 9

a = np.random.randint(0, 21, size=10)

b = np.random.randint(0, 21, size=10)

print(np.vstack((a,b)))

#vstack tworzy macierz, rozmiar musi się pokrywać

print(np.hstack((a,b)))

#hstack łączy tablice w jedną

#zadanie 10

#nie umiem go rozwiązać