**Assignments with Solutions**

**1. Convert a 1-D array into a 2-D array with 3 rows.**

Start with:

Assign-1 = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8])

Desired output:

[[ 0, 1, 2]

[3, 4, 5]

[6, 7, 8]]

*Solution:*

*import numpy as np*

*Assign-1 = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8])*

*print("The original array: \n", exercise\_2, "\n")*

*print("Method 1: \n", exercise\_2.reshape(3,-1), "\n")*

*print("Method 2: \n", exercise\_2.reshape(3,3))*

2. Replace all odd numbers in the given array with -1

Start with:

Assign-2 = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

Desired output:

*[ 0, -1, 2, -1, 4, -1, 6, -1, 8, -1]*

*Solution:*

*import numpy as np*

*Assign-2 = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])*

*Assign-2 [Assign-2 % 2 == 1] = -1*

*print(Assign-2)*

3. Find the positions of:

elements in x where its value is more than its corresponding element in y, and elements in x where its value is equals to its corresponding element in y.

Start with these:

x = np.array([21, 64, 86, 22, 74, 55, 81, 79, 90, 89])

y = np.array([21, 7, 3, 45, 10, 29, 55, 4, 37, 18])

Desired output:

(array([1, 2, 4, 5, 6, 7, 8, 9]),) and (array([0]),)

*Solution:*

*import numpy as np*

*x = np.array([21, 64, 86, 22, 74, 55, 81, 79, 90, 89])*

*y = np.array([21, 7, 3, 45, 10, 29, 55, 4, 37, 18])*

*print(np.where(x > y))*

*print(np.where(x == y))*

4. Extract the first four columns of this 2-D array.

Start with this:

Assign-4= np.arange(100).reshape(5,-1)

Desired output:

[[ 0 1 2 3]

[20 21 22 23]

[40 41 42 43]

[60 61 62 63]

[80 81 82 83]]

*Solution:*

*import numpy as np*

*Assign-4= np.arange(100).reshape(5,-1)*

*print("The original array: \n", Assign-4, "\n")*

*print("The extracted array:")*

*print(Assign-4 [:,:4])*

5. Generate a 1-D array of 10 random integers. Each integer should be a number between 30 and 40 (inclusive).

Sample of desired output:

[36, 30, 36, 38, 31, 35, 36, 30, 32, 34]

*Solution:*

*import numpy as np*

*Assign-5 = np.random.randint(30, 41, size = (10))*

*print(Assign-5)*

6.Consider the following matrices :

A= ((1, 2, 3), (4, 5, 6), (7, 8, 10)) and B = ((7, 8, 10) ,(4, 5, 6), (1, 2, 3))

Write a python program to perform the following using Numeric Python (numpy).

i) Add and Subtract of the Matrix A and B, print the resultant matrix C for add and E for subtract.

ii) Compute the sum of all elements of Matrix A, sum of each column of Matrix B and sum of each row of Matrix C

iii) Product of two matrices A and B, and print the resultant matrix D

iv) Sort the elements of resultant matrix C and print the resultant Matrix E.

v) Transpose the Matrix E and print the result

Solution:

import numpy as np

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

B = np.array(((7, 8, 10),(4,5,6), (1, 2, 3)))

print('Matrix A')

print(A)

print('Matrix B')

print(B)

# (i)

print('Addition of Matrix A and B')

C=np.add(A,B)

print(C)

print('Subtraction of Matrix A and B')

D=np.subtract(A,B)

print(D)

# (ii)

print('Sum of elements of Matrix A')

print(np.sum(A)) # Compute sum of all elements; prints "10"

print('Sum of each column of Matrix B')

print(np.sum(B, axis=0)) # Compute sum of each column; prints "[4 6]"

print('Sum of each row of Matrix C')

print(np.sum(C, axis=1)) # Compute sum of each row; prints "[3 7]"

# (iii)

print('The product of two matrix A and B')

E=np.dot(A,B) #C= np.mat(A)\*np.mat(B)

print(E)

# (iv)

print('Sorting of Matrix E ')

print np.sort(E)

# (v)

print('Transpose of Matrix E ')

print np.transpose(E)