NumPy Test

Name - Utkrist Jaiswal

Roll - 21051526

Sec - CSE-36

Question 1

How to create an empty and a full NumPy array?

In [2]:

```
import numpy as np
emptyArray = np.empty((2,2))
print("Empty Array: ", emptyArray)
```

```
Empty Array:  [[4.45061032e-308 9.34604358e-307]
 [1.11259601e-306 2.04721870e-306]]
```

In [3]:

```
fullArray = np.full((2,2), 3)
print("Full Array: ", fullArray)
```

```
Full Array:  [[3 3]
 [3 3]]
```

Question 2

Create a NumPy arraay filled with all zeroes.

In [4]:

```
zeroArray = np.full((1,3), 0)
print("Array with zeroes: ", zeroArray)
```

```
Array with zeroes:  [[0 0 0]]
```

Question 3

Create a NumPy array filled with all ones.

In [5]:

```
onesArray = np.full((1,4), 1)
print("Array with ones: ", onesArray)
```

```
Array with ones:  [[1 1 1 1]]
```

Question 4

Check whether a NumPy array contains a specified row.

In [6]:

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

specifiedRow = np.array([1,3,4])
row = np.any(np.all(arr == specifiedRow, axis=1))

print(row)
```

```
True
```

Question 5

How to remove rows in NumPy array that contains non-numerical values?

In [7]:

```
arr = np.array([[1,3,4],
                [2,np.nan,6],
                [7,5,9]])

nonNumericRows = np.isnan(arr).any(axis=1)

print("Original Array: ", arr)
print("Non Numeric Array: ", arr[~nonNumericRows])
```

```
Original Array:  [[ 1.   3.   4.]
 [ 2. nan  6.]
 [ 7.   5.   9.]]
Non Numeric Array:  [[1. 3. 4.]
 [7. 5. 9.]]
```

Question 6

Remove single-dimensional entries from the shape of an array.

In [8]:

```
arr = np.array([[1,3,1,4,7]])
removed = np.squeeze(arr)
print("Squeezed Array: ", removed.shape)
```

```
Squeezed Array:  (5,)
```

Question 7

Find the number of occurences of a sequence in a NumPy array.

In [9]:

```
arr = np.array([[1,2,3],
               [1,2,4]])
sequence = np.array([1,2,3])

arrStr = np.array2string(arr)
sequenceStr = np.array2string(sequence)

occurence = arrStr.count(sequenceStr)
print("No. of Occurences: ", occurence)
```

```
No. of Occurences:  1
```

Question 8

Find the most frequent value in a Numpy array.

In [10]:

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

arrayFlatten = arr.flatten()
count = np.bincount(arrayFlatten)
FrequentValue = np.argmax(count)

print("Most Frequent Value: ", FrequentValue)
```

```
Most Frequent Value:  2
```

Question 9

Combining a one and a two-dimensional Numpy Array.

In [11]:

```
arr1 = np.array([1,2,3])
arr2 = np.array([[2,3,4],
                [6,8,4]])

combinedArr = np.vstack((arr1,arr2))
print(combinedArr)
```

```
[[1 2 3]
 [2 3 4]
 [6 8 4]]
```

Question 10

How to build an array of all combinations of two Numpy arrays?

In [12]:

```
arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])

gridArr1, gridArr2 = np.meshgrid(arr1, arr2)

combinedArray = np.column_stack((gridArr1.ravel(), gridArr2.ravel()))

print("Combined Array:", combinedArray)
```

```
Combined Array: [[1 4]
 [2 4]
 [3 4]
 [1 5]
 [2 5]
 [3 5]
 [1 6]
 [2 6]
 [3 6]]
```

Question 11

How to add a border around Numpy array?

In [13]:

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

BorderSize = 2

BorderArr = np.pad(arr, pad_width=BorderSize, mode='constant')
print(BorderArr)
```

```
[[0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]
 [0 0 1 2 3 0 0]
 [0 0 2 5 6 0 0]
 [0 0 7 8 9 0 0]
 [0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]]
```

Question 12

How to compare two Numpy arrays?

In [14]:

```
arr1 = np.array([1,2,3])
arr2 = np.array([2,4,3])

result = arr1==arr2
print(result)
```

```
[False False  True]
```

Question 13

How to check whether specified values are present in Numpy array?

In [15]:

```
arr = np.array([1,2,3,4,5])
specified = [2,5,6]
result = np.isin(arr, specified)
print(result)
```

```
[False  True False False  True]
```

Question 14

How to get all 2D diagonals of a 3D Numpy array?

In [16]:

```
arr3d = np.array([[[1,2,3],[4,5,6],[7,8,9],[10,11,12]]])
diagonals2D = [np.diag(arr3d[i]) for i in range(arr3d.shape[0])]

for diagonals in diagonals2D:
    print(diagonals)
```

```
[1 5 9]
```

Question 15

Flatten a matrix in Python

In [17]:

```
matrix = np.array([[1,2,3],[2,4,5]])
flatten = matrix.flatten()
print(flatten)
```

```
[1 2 3 2 4 5]
```

Question 16

Flatten a 2D numpy into 1D

In [18]:

```
arr2d = np.array([[1,2,3],[1,5,7]])
flatten = np.ravel(arr2d)
print(flatten)
```

```
[1 2 3 1 5 7]
```

Question 17

Move axes of an array to new positions

In [19]:

```
arr = np.zeros((2,3,4))
movedArr = np.moveaxis(arr,(2,0,1),(1,2,0))
print(arr.shape)
print(movedArr.shape)
```

```
(2, 3, 4)
(3, 4, 2)
```

Question 18

Interchange two axes of an array

In [20]:

```
arr = np.zeros((2,3,4))
InterchangedArr = np.swapaxes(arr, 0 , 1)
print(arr.shape)
print(InterchangedArr.shape)
```

```
(2, 3, 4)
(3, 2, 4)
```

Question 19

Numpy - Fibonacci Series using Binet Formula

In [21]:

```
def fibonacciBinet(n):
    sqrt5 = np.sqrt(5)
    phi = (1+sqrt5)/2
    fibN = np.rint((phi**n-(-1/phi)**n)/sqrt5)
    return fibN.astype(int)

n = 10
fibSeries = [fibonacciBinet(i) for i in range(n)]
print(fibSeries)
```

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

Question 20

Count the number of non-zero values in the array

In [22]:

```
arr = np.array([[1,0,3],[2,0,0],[9,1,2]])
NonZeroCount = np.count_nonzero(arr)
print(NonZeroCount)
```

```
6
```

Question 21

Count the number of elements along the given axis

In [23]:

```
arr = np.array([[1,2,3], [2,3,4],[7,4,3]])
print(arr.shape[0])
print(arr.shape[1])
```

```
3
3
```

Question 22

Trim the leading and/or trailing zeroes from a 1-D Array

In [24]:

```
arr = np.array([0,0,0,0,1,1,2,3,4,5,0,0,0])
trimArrLeading = arr[np.argmax(arr!=0):]
trimArrTrailing = arr[:len(arr)-np.argmax(arr[::-1]!=0)]
print(arr)
print(trimArrLeading)
print(trimArrTrailing)
```

```
[0 0 0 0 1 1 2 3 4 5 0 0 0]
[1 1 2 3 4 5 0 0 0]
[0 0 0 0 1 1 2 3 4 5]
```

Question 23

Change data type of a given numpy array

In [25]:

```
arr = np.array([1,2,3,4,5])
arrFloat = arr.astype(float)
print(arrFloat)
```

```
[1. 2. 3. 4. 5.]
```

Question 24

Reverse a numpy array

In [26]:

```
arr = np.array([1,2,3,4,5])
ReversedArr = arr[::-1]
print(arr)
print(ReversedArr)
```

```
[1 2 3 4 5]
[5 4 3 2 1]
```

Question 25

How to make a Numpy array read-only?

In [27]:

```
arr = np.array([1,2,3,4,5])
arr.setflags(write=False)
try:
    arr[0] = 10
except ValueError as e:
    print("Error", e)

print(not arr.flags.writeable)
```

```
Error assignment destination is read-only
True
```

Questions on Numpy Matrix

Question 26

Get the maximum value of a given matrix

In [28]:

```
matrix = np.array([[1,2,3], [1,4,5],[8,9,3]])
maxVal = np.amax(matrix)
print(maxVal)
```

```
9
```

Question 27

Get the minimum value of a matrix

In [29]:

```
matrix = np.array([[1,2,3], [1,4,5],[8,9,3]])
minVal = np.amin(matrix)
print(minVal)
```

```
1
```

Question 28

Find the number of rows and columns of a given matrix using Numpy.

In [30]:

```
matrix = np.array([[1,2,3],[2,3,4],[7,8,9]])
print(np.shape(matrix))
```

```
(3, 3)
```

Question 29

Select the elements from a given matrix

In [31]:

```
matrix = np.array([[1,2,3],[2,3,4],[7,8,9]])
SelectedEle = matrix[[0,1,2],[0,1,2]]
print(SelectedEle)
```

```
[1 3 9]
```

Question 30

Find the sum of values

In [32]:

```
matrix = np.array([[1,2,3],[2,3,4],[7,8,9]])
print(np.sum(matrix))
```

```
39
```

Question 31

Calculate the sum of the diagonal elements of a Numpy array

In [33]:

```
matrix = np.array([[1,2,3],[2,3,4],[7,8,9]])
print(np.trace(matrix))
```

```
13
```

Question 32

Adding and Subtracting Matrices in Python.

In [34]:

```
matrix1 = np.array([[1,2,3],[2,3,4],[7,8,9]])
matrix2 = np.array([[10,12,13],[12,23,43],[17,48,79]])

msum = matrix1 + matrix2
print(msum)
mdiff = matrix1 - matrix2
print(mdiff)
```

```
[[11 14 16]
 [14 26 47]
 [24 56 88]]
[[ -9 -10 -10]
 [-10 -20 -39]
 [-10 -40 -70]]
```

Question 33

Ways to add rows/columns in numpy array.

In [35]:

```
arr = np.array([[1,2,3],[2,3,4]])
newRow = np.array([7,8,9])
ArrWithNewRow = np.vstack([arr,newRow])

newCol = np.array([[10],[11]])
ArrWithNewCol = np.hstack([arr,newCol])
print(ArrWithNewRow)
print(ArrWithNewCol)
```

```
[[1 2 3]
 [2 3 4]
 [7 8 9]]
[[ 1  2  3 10]
 [ 2  3  4 11]]
```

Question 34

Matrix Multiplication in Numpy.

In [36]:

```
matrix = np.array([[1,2,3],[2,3,4],[7,8,9]])
matrix2 = np.array([[13,2,23],[12,3,14],[7,28,39]])
result = np.dot(matrix,matrix2)
print(result)
```

```
[[ 58  92 168]
 [ 90 125 244]
 [250 290 624]]
```

Question 35

Get the eigen values of a matrix

In [37]:

```
matrix = np.array([[1,2],[3,4]])
eigenvalue = np.linalg.eigvals(matrix)
print(eigenvalue)
```

```
[-0.37228132  5.37228132]
```

Question 36

How to Calculate the determinant of a matrix using Numpy

In [38]:

```
matrix = np.array([[1,2,3],[2,3,4],[7,8,9]])
determinant = np.linalg.det(matrix)
print(determinant)
```

```
0.0
```

Question 37

How to inverse a matrix using Numpy?

In [39]:

```
matrix = np.array([[1,2],[3,4]])
Inverse = np.linalg.inv(matrix)
print(Inverse)
```

```
[[-2.   1. ]
 [ 1.5 -0.5]]
```

Question 38

How to count the frequency of unique values in Numpy array?

In [40]:

```
matrix = np.array([[1,2],[3,4]])
unique, counts = np.unique(arr,return_counts=True)
for value, count in zip(unique, counts):
    print(f"Value:{value},Count:{count}")
```

```
Value:1,Count:1
Value:2,Count:2
Value:3,Count:2
Value:4,Count:1
```

Question 39

Multiply matrices of a complex number using Numpy in Python

In [41]:

```
n1 = np.array([[1+2j,2+3j],[3+4j,4+5j]])
n2 = np.array([[5+6j,6+7j],[8+9j,9+10j]])

result = np.dot(n1,n2)
print(result)
```

```
[[-18. +58.j -20. +66.j]
 [-22.+114.j -24.+130.j]]
```

Question 40

Compute the outer product of two given vectors using Numpy

In [42]:

```
vector1 = np.array([1,2,3])
vector2 = np.array([2,4,5])
outer = np.outer(vector1,vector2)
print(outer)
```

```
[[ 2  4  5]
 [ 4  8 10]
 [ 6 12 15]]
```

Quetsion 41

Calculate the inner outer and cross poduct of matrices and vectors using numpy arrays

In [44]:

```
matrix = np.array([[1,2],[3,4]])
matrix2 = np.array([[1,21],[32,41]])
vector1 = np.array([1,2,3])
vector2 = np.array([2,4,5])
innerpro = np.inner(vector1, vector2)
outerpro = np.outer(vector1, vector2)
crosspro = np.cross(vector1, vector2)
matrixpro = np.dot(matrix, matrix2)

print(innerpro)
print(outerpro)
print(crosspro)
print(matrixpro)
```

```
25
[[ 2  4  5]
 [ 4  8 10]
 [ 6 12 15]]
[-2  1  0]
[[ 65 103]
 [131 227]]
```

Question 42

Compute the covariance matrix of two given numpy arrays

In [45]:

```
arr1 = np.array([1,2,3])
arr2 = np.array([4,5,6])
cov = np.cov(arr1,arr2)
print(cov)
```

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

Question 43

Convert the covariance matrix to correlation matrix using python.

In [46]:

```
cov = np.array([[2.0,1.0],[1.0,3.0]])
std_dev = np.sqrt(np.diag(cov))
corr = cov/np.outer(std_dev,std_dev)
print(corr)
```

```
[[1.         0.40824829]
 [0.40824829 1.        ]]
```

Question 44

Compute the Kronecker product of two multidimensionNumpy Arrays

In [47]:

```
arr1 = np.array([[1,2],[3,4]])
arr2 = np.array([[5,6],[7,8]])

kron = np.kron(arr1,arr2)
print(kron)
```

```
[[ 5  6 10 12]
 [ 7  8 14 16]
 [15 18 20 24]
 [21 24 28 32]]
```

Question 45

Convert the matrix into list

In [49]:

```
matrix = np.array([[1,2,3],[4,5,6],[7,8,9]])
matrixList = matrix.tolist()
print(matrixList)
```

```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```