Assignment 2

1) What is the difference between a scalar, vector and matrix in NumPy?

⇒ In NumPy, scalars can be represented as zero-dimensional arrays or simply as native Python numeric type that NumPy functions can operate on. For eg 5 or 3.14 are scalars

⇒ Vectors are typically created using 1D array. For example [1,2,3] or np.array([1,2,3]).

⇒ Matrices are created using 2D arrays as,

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

2) How can you create an array with evenly spaced values within a given range?

⇒ We can create an array with evenly spaced values within a given range as,

```
import numpy as np
evenly_spaced = ([range(1,21)])
evenly_spaced = np.array(2,20,2)
evenly_spaced
```

3) Explain the concept of array broadcasting in NumPy.

⇒ In NumPy broadcasting feature allows arithmetic operations between arrays with arrays with different shapes for example,

```
a = np.array = ([[1,2,3],[4,5,6]])
b = np.array ([10,20,30])
c = a+b
```

output c: ([11,22,33],[14,25,36])

4) How can you perform element-wise operations on NumPy arrays?

⇒ We can use element wise operation with the help of mathematical operators (*, /, +, -)

For example

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

add = a+b
sub = a-b
mul = a*b
div = a/b
```

5) What is the purpose of np.newaxis in NumPy?

⇒ It is used to increase the dimension of the existing array by one more dimension when used once. It also facilitates broadcasting

For example.

```
arr = np.arange(4)
row_vec = arr[np.newaxis, :] # convert to row
col_vec = arr[:, np.newaxis] # to column
```

Also,

```
n1 = np.array([1,2,3,4,5])
n2 = np.array([5,4,3])
n1_new = n1[:, np.newaxis]
result = n1_new + n2
```

5. How can you sort a Numpy array along a specific axis?

⇒ We can use numpy function sort() to sort array

As;

```
a: np.array( [[1,3,12], [9,17,10]] )
sorted_a: np.sort ( a, axis=0 )
sorted_a
```

7) Explain the difference between np.array and np.asarray.

⇒ np.array always create a new Numpy array where as np.array may or may not create a new array.

⇒ ~~It~~ np.array copies input data to new array where as in np.asarray if input is not in a way it creates a new array i.e do not copy the input data unless necessary.

⇒ np.array is useful when you want to ensure that a new array is created whereas np.asarray is useful when we want to work with input array directly

8) What are the advantages of using Numpy over Python's built-in list for numerical operations?

⇒ Numpy provides better memory efficiency, faster performance and convient functionality compared to Python list. Also Numpy ~~array~~ supports built in functions for various scientific operation, such as linear algebra & basic stat.

9) How can you save and load NumPy arrays to/from disk

=> We use load and save functions as:

```
a = np.array([1,2,3,4])
np.save("data.npy", a)
b = np.load("data.npy")
```

10) What is the purpose of np.where function in NumPy?

=> It returns the indices of elements in an input array where the given condition is satisfied.

Syntax: numpy.where(condition)

Example

```
a = np.array([[1,2,3],[4,5,6]])
b = np.where(a<4)
```