**<u>Name: Shatadru Banerjee</u>**
**<u>Roll No.: 2105580</u>**

# NumPy Test

1.A) Numerical Python
2.B) np.array([1, 2, 3, 4, 5])
3.A) [[1, 2, 3], [4, 5, 6]]
4.B) arr.ndim
5.B) print(myArr[0])
6.B) print(arr[1, 2])
7.B) print(arr[2:5])
8.A) print(arr[3:])
9.B) print(arr[::2])
10.A) arr.dtype
11.C)arr = np.array([1, 2, 3, 4], dtype=np.float)
12.B) The view SHOULD BE Affected by the changes made to the original array.
13.C) The copy SHOULD NOT be affected by the changes made to the original array.
14.C) The shape is the number of elements in each dimensions.
15.A) arr.shape
16.A) Concatenate()
17.A) array_split()
18.A) where()
19.A) np.where(arr==4)
19.C) sort()
20.A) np.random.randint(100)
21.B) random.normal(size=1000, loc=50, scale=0.2)
22.B) np.add(arr1, arr2)
23.D) np.subtract(arr1, arr2)
24.A) All the other 3 are rounding methods in NumPy
25.B) [1 3 6]
26.D) All the above
27.B) array([2, 3, 4, 5, 6, 7])
28.C) 3
29.C) It returns the byte size of each element of the array

30.A) 6
31.B) array([1, 2, 3, 4, 5])
32.B) a = np.array([(1, 2, 3), (4, 5, 6)]); a.reshape(2, 4)
33.D) float64
34.B) It contains 1s in all the diagonals
35.A) array([1, 2, 3, 4, 5, 6])
36.B) arr = np.array([[1, 2, 3], [4, 5, 6]]); np.hstack((arr, arr))
37.C) full()
38.B) a1 = np.array([1, 2, 3, 3]); a2 = np.array([0, 4, 9]); np.add(a1, a2)
39.C) A.T
40.B) 108
41.A) number of items
42.A) 8
43.D) reshape()
44.C) To create a matrix with all elements as 0
45.A) [[[1]], [[2]], [[3]], [[4]]]
46.D) All of the mentioned above
47.A) array([[0, 2], [1, 3]])
48.A)
```
[[[10]]
 [[20]]
 [[30]]
 [[40]]]
```
49.A) ndarray
50.C) Negative one