Sayan Adhikari 2105577

Q1 :A) Numerical Python
Q2 :B)np.array([1,2,3,4,5])
Q3 :A)[[1,2,3,],[4,5,6]]
Q4 :B) arr.ndim
Q5 :B)print(myArr[0])
Q6 :B)print(arr[1,2])
Q7 :A)print(arr[2:6])
Q8 :A)print(arr[3:])
Q9 :B)print(arr[::2])
Q10 :A)arr.dtype
Q11 :C) arr = np.array([1, 2, 3, 4], dtype=np.float)
Q12 :B) The view SHOULD BE Affected by the changes made to the original array
Q13 :C) The copy SHOULD NOT be affected by the changes made to the original array.
Q14 :C) The shape is the number of elements in each dimensions.
Q15 :A) arr.shape
Q16 :A)concatenate()
Q17 :A) array_split()
Q18 :A) where()
Q19 :A) np.where(arr==4)
Q19 :C) sort()
Q20 :A) np.random.randint(100)
Q21 :B) random.normal(size=1000, loc=50, scale=0.2)
Q22 :B) np.add(arr1, arr2)
Q23 :D) np.subtract(arr1, arr2)
Q24 :A) All the other 3 are rounding methods in NumPy.
Q25 :B) [1 3 6]
Q26 :B) array()
Q27 :B) array([2, 3, 4, 5, 6, 7])
Q28 :C) 3
Q29 :C) It returns the byte size of each element of the array
Q30 :A) 6
Q31 :B) array([1, 2, 3, 4, 5])
Q32 :B)a = np.array([(1, 2, 3), (4, 5, 6)]); a.reshape(2, 4)
Q33 :D) float64
Q34 :C) We can create an identity matrix using the identity() function
Q35 :A) array([1, 2, 3, 4, 5, 6])
Q36 :B) arr = np.array([[1, 2, 3], [4, 5, 6]]); np.hstack((arr, arr))
Q37 :C) full()

Q38 :B) a1 = np.array([1, 2, 3, 3]); a2 = np.array([0, 4, 9]); np.add(a1, a2)
Q39 :C) A.T
Q40 :B) 108
Q41 :A) number of items
Q42 :A) 8
Q43 :D) reshape()
Q44 :C) To create a matrix with all elements as 0
Q45 :A) [[[1]], [[2]], [[3]], [[4]]]
Q46 :D) All of the mentioned above
Q47 :A) array([[0, 2], [1, 3]])
Q48 :A) [[[10]]
 [[20]]
 [[30]]
 [[40]]]
Q49 :A) ndarray
Q50 :C) Negative one