Numpy Assignment

Q1. For a given numpy array, how will you change the dimensions to 5 using the existing parameters in the numpy array.
my_array = numpy.array([[[[1,3,4]]]])

    a. numpy.array(my_array, ndim=5)
    b. numpy.array(my_array, ndim=1)
    c. numpy.array(my_array, ndmin=5)
    d. numpy.array(my_array, ndim= my_array.ndmin + 1)

Q2. For the given numpy arrays, Array1 and Array2, what will be the dot product for the same. Array1 = array([[1, 2, 3], [4, 5, 6]])
Array2 = array([[2, 3],[3, 2]])

    a. Keyerror
    b. Valueerror
    c. array([[-9, 6, -1],[-12, 18, -7]])
    d. None of the Above

Q3. In a given numpy array, using the slicing operations print the reverse of the array.
array = np.array([10,3,1,203,404,204,20,302,30,402,192])

    a. array[-1:0]
    b. array1[len(array1):0]
    c. array[::-1]
    d. array1[-1:-len(array1)]

Q4. What is the output of the below code:
        a = [[1, 0], [0, 1]]
        b = [[4, 1], [2, 2]]
        np.dot(a, b)
    a. array([[1, 1], [2, 2]])
    b. array([[4, 1],[2, 2]])
    c. array([[4, 4],[2, 4]])
    d. None of the above

Q5. What is the output of the following code:
a = np.arange(3*4*5*6).reshape((3,4,5,6))
b = np.arange(3*4*5*6)[::-1].reshape((5,4,6,3))
np.dot(a, b)[2,3,2,1,2,2]
    a. 499128
    b. 399118
    c. 909128
    d. Value Error
Q6. What will be output for the following code?
import numpy as np

a = np.array([3, 7, 32], dtype = complex)
print(a)
    a. [3.+0.j 7.+0.j 32.+0.j]
    b. [3.0+0.j 7.0+0.j 32.0+0.j]
    c. [3.+0.k 7.+0.j 32.+0.l]
    d. Error

Q7. How to replace all values greater than a given value with a given cutoff? For example: In array 'ar1', replace all values greater than 30 to 30 and less than 10 to 10.
Input:
np.random.seed(100)
ar1 = np.random.uniform(1,50, 20)
    a. np.clip(a, a_min=10, a_max=30)
    b. np.where(a < 10, 10, np.where(a > 30, 30, a))
    c. Both of the above
    d. None of the above

Q8. For a given numpy array, how are you going to insert a new value at the specified position? array = np.array([10,3,1,203,404,204,20,302,30,402,192])
Elem_to_be_inserted = [1,2,3,4]
The position to be inserted at = before 404

    a. numpy.insert(array2, 404, array2)

    b. numpy.insert(array2, np.index(404), array2)

    c. numpy.insert(array1, np.where(404), array2)

    d. numpy.insert(array1, 4, array2)

Q9. Create a 3x3 matrix using numpy, and all the values of the matrix must be a constant k.

    a. numpy.eye('k')

    b. numpy.random.random((3,3), 'k')

    c. numpy.full((3,3), 'k')

    d. numpy.ones(3,3)

Q10. For the given python code that implements bubble sort, what will be the output for the given numpy array.

```
def bub_sort(array):
    for i in range(0, len(array)):
```

```
        for j in range(0, len(array) - i - 1):
          if array[j] < array[j + 1]:
            temp = array[j]
            array[j] = array[j + 1]
            array[j+1] = temp
      return array

    my_array = numpy.array([20,14,25,16,45,60,12,9])
```

   a. array([9, 12, 14, 16, 20, 25, 45, 60])
   b. array([60, 45, 25, 20, 16, 14, 12, 9])
   c. array([60, 45, 20, 25, 16, 14, 12, 9])
   d. array([9, 12, 16, 14, 20, 25, 45, 60])

Q11. For a given numpy array of the shape (2,5) ,How will you reshape the array in the shape (5,2).

   a. numpy.shape(5,2)
   b. numpy.arange(5,2)
   c. numpy.reshape((2,5) == (5,2))
   d. numpy.reshape(5,2)

Q12. What will be the shape of the sample numpy array after flattening it?
Sample = numpy.array([[1,2],[3,4],[5,6],[7,8]])

   a. array([1, 2, 3, 4, 5, 6, 7, 8])
   b. array([1, 2], [3, 4], [5, 6], [7, 8])
   c. array([1], [2], [3], [4], [5], [6], [7], [8])
   d. None of the above

Q13. In the given array, how can we get the following output - array([2, 5, 8]). Sample = numpy.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12],[13,14,15]])

   a. array[0:3, 1]
   b. array[0:3][1]
   c. array[[0:3], [1]]
   d. None of the above

Q14. Given two numpy arrays, we will perform Horizontally stack the given arrays array1 and array2. What will be the output of the above operation? The sample arrays are as follows.

Array1 = numpy.arange(20,2)
Array2 = numpy.array([1,2,3,4,5,6,7,8,9,10])

   A. array([1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
   B. array([[1, 3, 5, 7, 9, 11, 13, 15, 17, 19], [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]])
   C. array([[1, 3, 5, 7, 9, 11, 13, 15, 17, 19], [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]])
   D. None of the above

Q15. For the given arrays, array1 and array2, if we stack the two arrays row wise, what will be the output?
array1 = np.array([[1,2],[3,4]])
array2 = np.array([[1,2],[3,4],[5,6],[7,8],[9,10]])

   a. array([[1, 2],[3, 4],[1, 2],[3, 4],[5, 6],[7, 8],[9, 10]])
   b. array([[1, 2, 3, 4],[1, 2, 3, 4],[5, 6, 7, 8],[9, 10]])
   c. array([[1, 2, 3, 4, 1], [2, 3, 4, 5, 6],[7, 8, 9, 10]])
   d. None of the above

Q16. For the given arrays, array1 and array2, if we stack the two arrays column wise, what will be the output?
array1 = np.array([[1,2],[3,4]])
array2 = np.array([[5,6],[7,8]])

   a. array([[1, 2, 5, 6],[3, 4, 7, 8]])
   b. array([[1, 2, 5, 6 ,3, 4, 7, 8]])
   c. array([[1, 2], [5, 6],[3, 4], [7, 8]])
   d. None of the above

Q17. Given two vectors A and B, find the cross product between the two
vectors. A = numpy.array([[4],[12],[29]])
B = numpy.array([[13],[21],[4]])

   A. array([[-561 361 -72]])
   B. array([[-560 360 -71]])
   C. array([[-555 361 -72]])
   D. array([[-561 360 -72]])

Q18. Given two vectors A and B, find the correlation coefficient of the following vectors.
A = numpy.array([1,3,5,7,9,11,13,15,17,19,21,23,25])
B = numpy.array([0,2,4,6,8,10,12,14,16,18,20, 22, 24])
   A. array([[1., 1.],[1., 1.]])
   B. array([[0., 1.],[1., 1.]])
   C. array([[1., 0.],[1., 1.]])
   D. array([[1., 1.],[0., 1.]])

Q19. Create a nested numpy array from a given dictionary data.
sample = {1: [1,2],
      2: [[1],[2]],
      3: [[1,2], [3,4], [4,5]],
      4: [1],
      5: [1,2,3,4,5]}
   A. array([[1, list([1, 2])], [2, list([[1], [2]])], [3, list([[1, 2], [3, 4], [4, 5]])], [4, list([1])], [5, list([1, 2, 3, 4, 5])]], dtype=object)
   B. array([[1, list([1, 1])], [2, list([[1], [2]])], [3, list([[1, 2], [3, 4], [4, 5]])], [4, list([1])], [5, list([1, 2, 3, 4, 5])]], dtype=object)
   C. array([[1, list([1, 2])], [2, list([[1], [2]])], [3, list([[1, 2], [1, 4], [4, 5]])], [4, list([1])], [5, list([1, 2, 3, 4, 5])]], dtype=object)
   D. array([[1, list([1, 2])], [2, list([[1], [2]])], [3, list([[1, 2], [3, 4], [4, 5]])], [4, list([1])], [5, list([1, 2, 3, 10, 5])]], dtype=object)

Q20. Create a 2-dimensional array with 3 rows and 3 columns containing random numbers from 1 to 9. Find the difference between the maximum element across the columns and the minimum element across the rows.

   A. [7 3 2]
   B. [7 1 2]
   C. [8 3 2]
   D. [7 3 1]