

Q1. Comparison Returns!

What will be the output of the following code?

```
a = np.array([100, 200, 300, 400])
b = np.array([300, 200, 100, 400])
print(a == b)
```

- A. True
- B. False
- C. [True, False, True, False]
- D. [False, True, False, True]

Q2. Comparing in numpy

a. What is the last element of the output?

```
import numpy as np
print(np.sort(np.array(['Ram', 'Astha', 'Raghavendra'])))
```

b. What is the output of the code snippet given below?

```
arr1 = np.array(['Ram', 'Astha', 'Brahmat'])
arr2 = np.array(['Shyam', 'Kalyan', 'Naveen'])
arr1 > arr2
```

Choose the options which are answers to questions a and b.

- A. For block 'a', answer is 'Ram'.
- B. For block 'a', answer is 'Raghavendra'.
- C. For block 'b', answer is array([False, False, False]).
- D. For block 'b', answer is False.

There may be more than one correct answer. If so, please submit all those are correct.

Q3. What about output?

Which option is correct about the output of the following code snippet?

```
import numpy as np
x = np.array([-5, 9, 20, 25, -3, 5, 16, 10, -8])
x[(x >= -5) & (x <= 15)] *= -1
print(x)
```

- A. Given code will change the sign of all the numbers in x.
- B. Given code will change the sign of all the numbers in range [-5, 15] in x.
- C. Given code will change the sign of all the numbers greater than -5 in x.
- D. Given code will change the sign of all the numbers smaller than 15 in x.

Q4. Vectorized code

Given 3 arrays,

```
arr1 = np.array([1,2,3,6,3,2])
arr2 = np.array([4,2,1,3,3,2])
arr3 = np.zeros(len(arr1))
```

Which of the following are vectorized code(s) for array operations?

A.

```
for i in range(len(arr1)):
    arr3[i] = arr1[i] * arr2[i]
```

B.

```
arr3 = arr1*arr2
```

C.

```
for i in range(len(arr1)):
    if(arr1[i] < 0 ):
        arr1[i] = -1
    else:
        arr1[i] = 1
```

D.

```
np.where(arr1 > 0, 1, -1)
```

There may be more than one correct answer. If so, please submit all those are correct.

Q5. What about code?

```
import numpy as np
arr = 2 * np.arange(0,2,0.5)
if arr <= 0.6:
    print("condition satisfies")
else:
    print("condition doesn't satisfy")
```

In the above code, 'condition' implies the situation that **arr** has **at least one value smaller than or equal to 0.6**. Which option is true with respect to the code and condition?

- A. The Code will give the required output.
- B. The Code will throw ValueError, and np.any() should be used to get the required output.
- C. The Code will throw ValueError and np.all() should be used to get the required output.
- D. None of the given option is correct

Q6. Reshape me

What is the below code printing?

```
import numpy as np
a = np.arange(10,22).reshape((3, 4))
print(a)
```

- A. A 1D numpy array filled with values from 10 to 21
- B. A 3X4 matrix filled with values from 10 to 22
- C. A 3X4 matrix filled with values from 10 to 21
- D. A 1D numpy array filled with values from 10 to 22

Q7. Transpose Reshaped

What would the following code print?

```
import numpy as np
a = np.array([[6, 28], [8, 56], [7, 19]])
x = np.transpose(a).reshape(1,6)
print(x)
```

- A. [6 8 7 28 56 19]
- B. [[6 8 7 28 56 19]]
- C. [6 28 8 56 7 19]
- D. [[6 28 8 56 7 19]]

Q8. Batchmate's code

Your batchmate writes the following code :

```
import numpy as np
a = np.array([[16, 5], [81, 6], [33, 1]])
x=np.transpose(a).reshape(2,3)
print(x.flatten())
```

Which of the following is the correct output for the above code?

Note:

- flatten() function is used to transform a multi-dimensional array into a one-dimensional array.
- In other words, it "flattens" a multi-dimensional array structure into a simple linear sequence.

- A. [16 81 33 5 6 1]
- B. [16 33 81 5 1 6]
- C. [16 5 81 6 33 1]
- D. [[16 5 81 6 33 1]]

Q9. Indexed array

Given the following code, what will be the output?

```
import numpy as np
a = np.array([[34, 28, 55], [8, 56, 3], [77, 87, 19]])
print(a.transpose() [-2, -2])
```

- A. 55
- B. 28
- C. 56
- D. 3