

1. Explain why in Python the expression  $10^2$  does not evaluate to 100. Find what are bitwise operators in Python
2. Truth table for the xor operator is given as

P	Q	P xor Q
True	True	False
False	True	True
True	False	True
False	False	False

Since there is no exclusive-or (xor) operator provided out-of-the-box by Python, we can construct such operator using existing operators. Find two ways of implementing xor in Python

3. Different languages provide a *sign(a)* function that returns -1 if *a* is negative and 1 otherwise. Python core does not have one by default. *Math* module includes a function *math.copysign(x, y)* which returns the absolute value of *x* with the sign of *y*. Modify this function to create a *sign(a)* function.
4. What does the code below print?

```
pset_time = 15
sleep_time = 8
print(sleep_time > pset_time)
derive = True
drink = False
both = drink and derive
print(both)
```

5. What is the value of L3 after you execute all the operations in the code below?

```
L1 = ['re']
L2 = ['ga']
L3 = ['sa']
L4 = L1 + L2
L3.extend(L4)
L3.sort()
del(L3[0])
L3.append(['ma', 'dha'])
```

6. Find bug in the loop *without coding*. Then write the correct code.

```
n = 10
i = 10
while i > 0:
    print i
    if i % 2 == 0:
        i = i / 2
    else:
        i = i + 1
```

In the correct code, print a table with two columns – first column tabulates the value of variable *i* while the second column tabulates the value of variable *n* during execution.

7. Create a 4 x 5 NumPy array of random values using `numpy.random.rand()` function. Write your own function to sort elements in your array in descending as well as ascending order. Denote your array as  $A$ . Find  $AA^T - A^T A$ .
8. Write programs to test the following NumPy broadcasting rule:  
When operating on two arrays, NumPy compares their shapes element-wise. It starts with the trailing or rightmost dimension and works its way left. Two dimensions are compatible when they are equal or one of them is 1.
9. Make an array with `dtype = uint8` and elements of your choosing. Keep adding to it until (one of) the items go over 255. What do you observe? Hint: make an array, and just add a constant to it. The constant will be added to all the items of the array element-wise.
10. Write a program to multiply all values below 100 in the following list by 2 until all values in the list are above 100. Use looping.  
`a = np.array([230, 10, 284, 39, 76, 53, 15, 12, 42, 50, 87, 100])`  
Create a new array from the previous array selecting elements that lie between 150 and 200.