PROBLEM

Xoring and Clearing

Easy Accuracy: 56.19% Submissions: 27K+ Points: 2

You are given an array **arr[]** of **size n**. You need to do the following:

1. You need to calculate the **bitwise XOR** of each element in the array with its **corresponding index** (indices start from 0).
2. After step1, **print the array.**
3. Now, **set all the elements of arr[] to zero**.
4. Now, **print arr[].**

Example 1:

Input:
n = 10
arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
Output:
1 3 1 7 1 3 1 15 1 3
0 0 0 0 0 0 0 0 0 0
Explanation:
First we take xor of every element with
their indices, like (1xor0), (2xor1), (3xor2), (4xor3) and so on.
Now print the resultant array
Now set all the elements of array to zero
Now print the resultant array

Example 2:

Input:
n = 4
arr[] = {10, 20, 30, 40}
Output:
10 21 28 43
0 0 0 0
Explanation:
First we take xor of every element with
their indices, like (1xor0), (2xor1), (3xor2).
Now print the resultant array
Now set all the elements of array to zero
Now print the resultant array

Your Task:
Since this is a function problem, you don't need to take any input. Just complete the provided functions. In a new line, print the **output**.

Expected Time Complexity: O(n)
Expected Auxiliary Space: O(1)

Constraints:
1 <= n <= 1000
1 <= arr[i] <= 1000

CODE

```
class Solution:
    def printArr(self, n, arr):
        # printing array elements with spaces
        print(*arr)

    def setToZero(self, n, arr):
        # setting all elements to zero
        for i in range(n):
            arr[i] = 0

    def xor1ToN(self, n, arr):
        # doing xor with indices optimized code
        for i in range(n):
            arr[i] ^= i
```