FINDING REPEATED AND MISSING ELEMENT

☆ In this problem, we are supposed to return the element which is repeating within an array of size n.

Eg : [4 3 6 2 1 1]

This will give out 1 as the repeating element and 5 as the missing element.

Brute force solution is to check for appearance of all N elements, if count is 2 then its repeating but if count is 0, its missing. Time complexity is $O(N^2)$.

Better solution is to use a hashmap. We will create a hashmap in one full iteration and then just look for count of 0 and 2. A hasharray will also work in this case as we know the size will be N.

Optimal solution by math is to sum the entire array and to compare it with sum of first N natural numbers. Difference of both is also

equal to the difference between repeated and missing element. If we subtract sum of squares of both elements, we get difference of squares of missing element and repeated elements

So we have

$$X - Y = \_$$
$$X^2 - Y^2 = \_$$

With this we find value of $X+Y$ and then X, Y.

Another optimal solution is using XOR.

Pseudocode :

```
missingAndRepeated(arr, N) {

    sum1 = N*(N+1)/2
    sum2 = N*(N+1)*(2N-1)/6
    s1, s2 = 0, 0
    for(i = 0 → N) {
        s1 += arr[i]
        s2 += arr[i] * arr[i]
    }
    x_y = s1 - sum1
    x2_y2 = s2 - sum2
    x__y = x2_y2/x_y
    x = (x_y + x__y)/2
    y = (x__y - x_y)/2
    return x, y
```