SINGLE NUMBER III

☆ In this problem, we are given an array of integers where each int occurs twice but two numbers appearing only once. Our job is to return these two numbers in any order.

Bruteforce solution is to obviously hash the array, and then iterate through it again to find out which elements occur only once.

If we iterate through the entire array and just XOR each element, we will be left with XOR of two distinct elements. Then we can use concept of buckets to find original elements. As XOR result is non zero, there must be atleast one bit that is different. Hence, we take the first set bit in the XOR result (that is the first bit that is different). This seggregates both of our required numbers so we just XOR both our buckets.

Pseudocode :

```
singleNumberIII (arr, N) {
```

```
initialXOR = 0
for (i = 0 → N-1) {
    initialXOR = initialXOR ^ arr[i]
}
firstSetBit = (initialXOR & initialXOR-1)
              & initialXOR

notSet = 0 , set = 0
for(i = 0 → N-1) {
    if (arr[i] & firstSetBit != 0) {
        set = set ^ arr[i]
    } else {
        notSet = notSet ^ arr[i]
    }
}
```