Find the Intersection of Two Arrays

Problem Statement:

You are given two arrays, `nums1` and `nums2`, both containing integers. Your goal is to find the intersection of these two arrays. Each element in the intersection should appear as many times as it shows in both arrays. Return an array containing the intersection elements, ensuring that the result array does not contain duplicates.

Examples:

Example 1:

Input:

```
nums1 = [1, 2, 2, 1], nums2 = [2, 2]
```

Output:

```
[2, 2]
```

Explanation: The intersection of the two arrays is `[2, 2]`, as 2 appears twice in both arrays.

Example 2:

Input:

```
nums1 = [4, 9, 5], nums2 = [9, 4, 9, 8, 4]
```

Output:

```
[4, 9]
```

Explanation: The intersection of the two arrays is `[4, 9]`, as these elements appear in both arrays.

Reverse a number using recursion

Problem Statement:

Given an integer N, reverse its digits using recursion.
Examples
• **Example 1:**
Input: N = 1234
Output: 4321

Merge Sort

Problem Statement:

You are given an array of integers. You need to sort the array using the **Merge Sort** algorithm. Return the sorted array.

Examples:

Example 1:

Input:

```
arr = [5, 2, 9, 1, 5, 6]
```

Output:

```
[1, 2, 5, 5, 6, 9]
```

Explanation: The array is sorted using Merge Sort, and the sorted result is `[1, 2, 5, 5, 6, 9]`.

Example 2:

Input:

```
arr = [3, 4, 1, 2]
```

Output:

```
[1, 2, 3, 4]
```

Explanation: After applying Merge Sort, the sorted array is `[1, 2, 3, 4]`.

Apply Operations to an Array

You are given a **0-indexed** array `nums` of size `n` consisting of **non-negative** integers.
You need to apply `n - 1` operations to this array where, in the `i`th operation (**0-indexed**), you will apply the following on the `i`th element of `nums`:

- If `nums[i] == nums[i + 1]`, then multiply `nums[i]` by 2and set `nums[i + 1]` to `0`. Otherwise, you skip this operation.

After performing **all** the operations, **shift** all the `0`'s to the **end** of the array.

- For example, the array `[1,0,2,0,0,1]` after shifting all its `0`'s to the end, is `[1,2,1,0,0,0]`.

Return *the resulting array*.
Note that the operations are applied **sequentially**, not all at once.

Example 1:

Input: nums = [1,2,2,1,1,0]

Output: [1,4,2,0,0,0]

Explanation: We do the following operations:
- i = 0: nums[0] and nums[1] are not equal, so we skip this operation.
- i = 1: nums[1] and nums[2] are equal, we multiply nums[1] by 2 and change nums[2] to 0. The array becomes [1,**4**,**0**,1,1,0].
- i = 2: nums[2] and nums[3] are not equal, so we skip this operation.
- i = 3: nums[3] and nums[4] are equal, we multiply nums[3] by 2 and change nums[4] to 0. The array becomes [1,4,0,**2**,**0**,0].
- i = 4: nums[4] and nums[5] are equal, we multiply nums[4] by 2 and change nums[5] to 0. The array becomes [1,4,0,2,**0**,**0**].

After that, we shift the 0's to the end, which gives the array [1,4,2,0,0,0].

Example 2:

Input: nums = [0,1]

Output: [1,0]

Explanation: No operation can be applied, we just shift the 0 to the end.