Problem 3524: Find X Value of Array I

Problem Information

Difficulty: Medium
Acceptance Rate: 36.67%
Paid Only: No
Tags: Array, Math, Dynamic Programming

Problem Description

You are given an array of **positive** integers `nums`, and a **positive** integer `k`.

You are allowed to perform an operation **once** on `nums`, where in each operation you can remove any **non-overlapping** prefix and suffix from `nums` such that `nums` remains **non-empty**.

You need to find the **x-value** of `nums`, which is the number of ways to perform this operation so that the **product** of the remaining elements leaves a _remainder_ of `x` when divided by `k`.

Return an array `result` of size `k` where `result[x]` is the **x-value** of `nums` for `0 <= x <= k - 1`.

A **prefix** of an array is a subarray that starts from the beginning of the array and extends to any point within it.

A **suffix** of an array is a subarray that starts at any point within the array and extends to the end of the array.

Note that the prefix and suffix to be chosen for the operation can be **empty**.

Example 1:

Input: nums = [1,2,3,4,5], k = 3

Output: [9,2,4]

Explanation:

* For `x = 0`, the possible operations include all possible ways to remove non-overlapping prefix/suffix that do not remove `nums[2] == 3`. * For `x = 1`, the possible operations are: * Remove the empty prefix and the suffix `[2, 3, 4, 5]`. `nums` becomes `[1]`. * Remove the prefix `[1, 2, 3]` and the suffix `[5]`. `nums` becomes `[4]`. * For `x = 2`, the possible operations are: * Remove the empty prefix and the suffix `[3, 4, 5]`. `nums` becomes `[1, 2]`. * Remove the prefix `[1]` and the suffix `[3, 4, 5]`. `nums` becomes `[2]`. * Remove the prefix `[1, 2, 3]` and the empty suffix. `nums` becomes `[4, 5]`. * Remove the prefix `[1, 2, 3, 4]` and the empty suffix. `nums` becomes `[5]`.

Example 2:

Input: nums = [1,2,4,8,16,32], k = 4

Output: [18,1,2,0]

Explanation:

* For `x = 0`, the only operations that **do not** result in `x = 0` are: * Remove the empty prefix and the suffix `[4, 8, 16, 32]`. `nums` becomes `[1, 2]`. * Remove the empty prefix and the suffix `[2, 4, 8, 16, 32]`. `nums` becomes `[1]`. * Remove the prefix `[1]` and the suffix `[4, 8, 16, 32]`. `nums` becomes `[2]`. * For `x = 1`, the only possible operation is: * Remove the empty prefix and the suffix `[2, 4, 8, 16, 32]`. `nums` becomes `[1]`. * For `x = 2`, the possible operations are: * Remove the empty prefix and the suffix `[4, 8, 16, 32]`. `nums` becomes `[1, 2]`. * Remove the prefix `[1]` and the suffix `[4, 8, 16, 32]`. `nums` becomes `[2]`. * For `x = 3`, there is no possible way to perform the operation.

Example 3:

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

Output: [9,6]

Constraints:

* `1 <= nums[i] <= 109` * `1 <= nums.length <= 105` * `1 <= k <= 5`

Code Snippets

C++:

```
class Solution {
public:
vector<long long> resultArray(vector<int>& nums, int k) {

}
};
```

Java:

```
class Solution {
public long[] resultArray(int[] nums, int k) {

}
}
```

Python3:

```
class Solution:
def resultArray(self, nums: List[int], k: int) -> List[int]:
```