238. Product of Array Except Self (Medium)

Given an integer array `nums`, return an array `output` where `output[i]` is the product of all the elements of `nums` except `nums[i]`.

Each product is guaranteed to fit in a 32-bit integer.

Follow-up: Could you solve it in O(n)$O(n)$ time without using the division operation?

Example 1:

```
Input: nums = [1,2,4,6]

Output: [48,24,12,8]
```

Example 2:

```
Input: nums = [-1,0,1,2,3]

Output: [0,-6,0,0,0]
```

Constraints:

- `2 <= nums.length <= 1000`
- `20 <= nums[i] <= 20`

▼ 思路:

作法一：紀錄全數相乘並獨立出現0的情況

T:$O(n)$, S:$O(1)$

作法二：prefix跟postfix相乘

T:$O(n)$, S:$O(n)$

作法一

```
class Solution {
public:
    vector<int> productExceptSelf(vector<int>& nums) {
        int sum=1,zeronum = 0;
        for(int num : nums){
            if(num!=0) sum *= num;
            else zeronum += 1;
        }
        for(int i = 0;i<nums.size();i++){
            if(zeronum>1) nums[i] = 0;
            else if(zeronum) nums[i]= (nums[i] == 0) ? sum:0;
            else nums[i] = sum/nums[i];
        }
        return nums;
    }
};
```

作法二

```
class Solution {
public:
    vector<int> productExceptSelf(vector<int>& nums) {
        vector<int> pre(nums.size()+1,1);
        vector<int> post(nums.size()+1,1);
        vector<int> ans(nums.size(),1);
        for(int i=1; i<nums.size();i++){
            pre[i] = nums[i-1]* pre[i-1]; // 0,1,..,n-1,0
        }
        for(int i=nums.size()-1; i>=0;i--){
            ans[i] = pre[i]* post[i+1]; // 0,n-1,...,1,0
            post[i] = nums[i]*post[i+1];
        }
        return ans;
    }
};
```