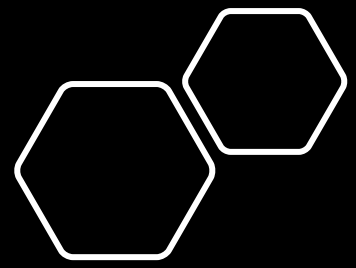
```
238 | Medium | Product of Array Except Self | Array

Given an integer array nums, return an array answer such that answer[i] is equal
to the product of all the elements of nums except nums[i].
The product of any prefix or suffix of nums is guaranteed to fit in a 32-bit integer.
You must write an algorithm that runs in O(n) time and without using the division operation.

Constraints:
2 <= nums.length <= 105
-30 <= nums[i] <= 30
The product of any prefix or suffix of nums is guaranteed to fit in a 32-bit integer.
```



Example 1:

```
Input: nums = [1,2,3,4]
Output: [24,12,8,6]
```

Example 2:

```
Input: nums = [-1,1,0,-3,3]
Output: [0,0,9,0,0]
```

```cpp
vector<int> productExceptSelf(vector<int>& nums) {
    int n = nums.size();
    vector<int> prefix(n, 1), suffix(n, 1), result(n, 1);

    prefix[0] = nums[0];
    suffix[n-1] = nums[n-1];
    for(int i=1; i<n; i++) {
        prefix[i] = prefix[i-1]*nums[i];
        suffix[n-i-1] = suffix[n-i]*nums[n-i-1];
    }

    result[0] = suffix[1];
    result[n-1] = prefix[n-2];
    for(int i=1; i<n-1; i++) {
        result[i] = prefix[i-1] * suffix[i+1];
    }
    return result;
}
```

#100daysofDSA





/rvislive

Rakesh Vishwakarma