

题目：LCR 012. 寻找数组的中心下标

思想：利用前缀和思想，用一个变量记录前缀和，判断条件： $2 * \text{前缀和} + \text{当前} = \text{数组总和}$

```
1 class Solution {
2 public:
3     int pivotIndex(vector<int>& nums) {
4         int sum = 0;
5         for (auto e : nums) {
6             sum += e;
7         }
8         int total = 0;
9         for (int i = 0; i < nums.size(); i++) {
10
11             if (2 * total + nums[i] == sum)
12                 return i;
13             total += nums[i];
14         }
15         return -1;
16     }
17 }; LCR 012. 寻找数组的中心下标
```

题目：DP35 【模板】二维前缀和

思想：类似于动态规划，我的问题 忘记求和的时候冗余数字

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main() {
6      int n, m, q;
7      cin >> n >> m >> q;
8
9      // 初始化二维数组
10     vector<vector<long long >> vv(n + 1, vector<long long>(m + 1, 0));
11
12     // 输入矩阵数据, 从 (1,1) 开始
13     for (int i = 1; i <= n; i++) {
14         for (int j = 1; j <= m; j++) {
15             cin >> vv[i][j];
16         }
17     }
18
19     // 构建前缀和
20     for (int i = 1; i <= n; i++) {
21         for (int j = 1; j <= m; j++) {
22             vv[i][j] = vv[i][j] + vv[i - 1][j] + vv[i][j - 1] - vv[i - 1][j - 1];
23         }
24     }
25
26     // 处理每个查询
27     while (q--) {
28         int x1, y1, x2, y2;
29         cin >> x1 >> y1 >> x2 >> y2;
30
31         // 计算子矩阵的和
32
33
34         cout << vv[x2][y2]
35              - (x1 > 1 ? vv[x1 - 1][y2] : 0)
36              - (y1 > 1 ? vv[x2][y1 - 1] : 0)
37              + (x1 > 1 && y1 > 1 ? vv[x1 - 1][y1 - 1] : 0)
38              << endl;
39     }
```

```
40
41     return 0;
42 }
```

题目：238. 除自身以外数组的乘积

```
1  class Solution {
2  public:
3      vector<int> productExceptSelf(vector<int>& nums) {
4          int n = nums.size();
5          vector<int> pre(n, 1);
6          for (int i = 1; i < n; i++) {
7              pre[i] = pre[i - 1] * nums[i - 1];
8          }
9
10         vector<int> suf(n, 1);
11         for (int i = n - 2; i >= 0; i--) {
12             suf[i] = suf[i + 1] * nums[i + 1];
13         }
14
15         vector<int> ans(n);
16         for (int i = 0; i < n; i++) {
17             ans[i] = pre[i] * suf[i];
18         }
19         return ans;
20     }
21 };
```