

## 只出现一次的数字（多文件测试）

由于正式机考时，多文件题由于WebIDE自测和调试比较局限和麻烦，本地IDE可能更加方便一点，你可以事先熟悉一下本地IDE多文件的编译和运行

- 对于Clion：新建项目后，右键项目文件夹添加cpp头文件与源文件，一般会自动配置，直接运行即可，如果提示找不到头文件请检查CMakeLists
- 对于VsCode：请确保你的所有文件在同一目录下，打开该目录，选择Main.cpp，点击右上角三角运行C/C++文件，调试配置选择C/C++:g++.exe 生成和调试活动文件，此时大概率报错undefined reference，然后打开刚刚生成的.vscode文件夹下的tasks.json，找到tasks-args下的"\${file}"修改为"\${fileDirname}\\\*.cpp"，然后运行即可

```
"args": [  
    "-fdiagnostics-color=always",  
    "-g",  
    "${fileDirname}\\*.cpp", // 修改此处  
    "-o",  
    "${fileDirname}\\${fileBasenameNoExtension}.exe"  
],
```

### 题目描述

给你一个**非空**整数数组`nums`，除了某个元素只出现一次以外，其余每个元素均出现两次。找出那个只出现了一次的元素。

示例 1：

输入：nums = [2,2,1] 输出：1

示例 2：

输入：nums = [1] 输出：1

示例 3：

输入：nums = [1,1,2,2,3,3,4,5,5,6,6] 输出：4

### 框架代码

#### Solution.h

```
#include <vector>  
  
int singleNumber(std::vector<int>& nums);
```

#### Solution.cpp

```
#include "Solution.h"

int singleNumber(std::vector<int>& nums) {
    // TODO
    return 0;
}
```

## Main.cpp

```
#include <cassert>
#include <functional>
#include <iomanip>
#include <iostream>
#include "Solution.h"

using namespace std;

void TEST_0();
void TEST_1();
void TEST_2();

#define REGISTER_TEST_CASE(name) {#name, name}

int main() {
    std::unordered_map<std::string, std::function<void()>>
        test_functions_by_name = {
            REGISTER_TEST_CASE(TEST_0),
            REGISTER_TEST_CASE(TEST_1),
            REGISTER_TEST_CASE(TEST_2)};
    std::string test_case_name;
    std::cin >> test_case_name;
    auto it = test_functions_by_name.find(test_case_name);
    assert(it != test_functions_by_name.end());
    auto fn = it->second;
    fn();
    return 0;
}

void TEST_0() {
    vector<int> nums = {2, 2, 1};
    cout << singleNumber(nums) << endl;
}

void TEST_1() {
    vector<int> nums = {1};
    cout << singleNumber(nums) << endl;
}

void TEST_2() {
    vector<int> nums = {1, 1, 2, 2, 3, 3, 4, 5, 5, 6, 6};
}
```

```
    cout << singleNumber(nums) << endl;  
}
```