**best\_shuffle.rs**：这个错误是由于数组索引越界导致的。

**calenda.rs**：lazy\_static 的问题：在 Rust 2018 版本之后，不再需要使用 extern crate。直接在 Cargo.toml 文件中添加依赖，然后在代码中直接使用。MONTHS 变量没有正确定义或导入到当前作用域。

**convert\_decimal\_number\_to\_rational.rs**：死循环，程序运行超时。

**cuban\_primes.rs**：1、不必要的可变性警告：在 push\_back 和 pop\_back 函数中，current 变量被声明为可变的，但实际上不需要。2、借用冲突错误：在 while 循环中，借用了 temp.next，但在循环结束后又试图修改它，这违反了 Rust 的借用规则。

**farey\_sequence.rs:**

1、类型不匹配错误：在 farey\_len 函数中，std::u64::MAX 和 std::usize::MAX 的类型不匹配。2、未使用的变量警告：在 farey 函数中，k 变量被赋值但从未被使用。

**flatten\_a\_list.rs**：试图获取一个不可变变量的可变引用。在 Rust 中，默认情况下所有变量都是不可变的，除非我们明确地使用 mut 关键字将其声明为可变。

**last\_letter\_first\_letter.rs**：移除了 Ref 结构体的 Clone trait 派生，但是代码中仍然需要使用 Clone 特性。Copy trait 要求类型也实现 Clone trait。当我们移除 Clone 时，Copy 无法正常工作。

**nimber\_arithmetic.rs**：对一个整数进行过大的右移操作时，程序运行时发生了右移溢出。

**one\_dimensional\_cellular\_automata.rs**：在同一作用域内尝试同时使用可变借用和不可变借用。在创建 c\_slice 和 b\_slice 时，对 c 和 b 进行了可变借用。然后在 while 循环中，尝试通过 c.iter() 对 c 进行不可变借用，这与之前的可变借用冲突。

**parse\_an\_ip\_address.rs**：代码中使用了 ptr 模块，但没有导入它

**priority\_queue.rs**：数组越界

**run\_length\_encoding.rs，sexy\_primes.rs**：减法溢出

CA (Computational Accuracy)：语义正确性，即测试用例通过率；

Compiled:测试用例编译通过率；

AS-P: 将通过测试的文件中safe API百分比相加求平均;

AS-A: 将所有文件中safe API百分比相加求平均（未通过测试的视为0）;

CS-P: 将通过测试的文件中safe code百分比相加求平均;

CS-A: 将所有文件中safe code百分比相加求平均（未通过测试的视为0）