实验五 Python数据结构与数据模型

班级： 21计科1

学号： 202302200000

姓名： 张三

Github地址：https://github.com/yourusername/python_course

CodeWars地址：https://www.codewars.com/users/yourusername

实验目的

1. 学习Python数据结构的高级用法
2. 学习Python的数据模型

实验环境

1. Git
2. Python 3.10
3. VSCode
4. VSCode插件

实验内容和步骤

第一部分

在Codewars网站注册账号，完成下列Kata挑战：

第一题：停止逆转我的单词

难度： 6kyu

编写一个函数，接收一个或多个单词的字符串，并返回相同的字符串，但所有5个或更多的字母单词都是相反的（就像这个Kata的名字一样）。传入的字符串将只由字母和空格组成。只有当出现一个以上

的单词时，才会包括空格。
例如：

```
spinWords( "Hey fellow warriors" ) => returns "Hey wollef sroirraw"
spinWords( "This is a test") => returns "This is a test"
spinWords( "This is another test" )=> returns "This is rehtona test"
```

代码提交地址：
https://www.codewars.com/kata/5264d2b162488dc400000001

提示：

- 利用str的split方法可以将字符串分为单词列表
  例如：

```
words = "hey fellow warrior".split()
# words should be ['hey', 'fellow', 'warrior']
```

- 利用列表推导将长度大于等于5的单词反转(利用切片word[::-1])
- 最后使用str的join方法连结列表中的单词。

第二题： 发现离群的数(Find The Parity Outlier)

难度：6kyu

给你一个包含整数的数组（其长度至少为3，但可能非常大）。该数组要么完全由奇数组成，要么完全由偶数组成，除了一个整数N。请写一个方法，以该数组为参数，返回这个 "离群 "的N。

例如：

```
[2, 4, 0, 100, 4, 11, 2602, 36]
# Should return: 11 (the only odd number)

[160, 3, 1719, 19, 11, 13, -21]
# Should return: 160 (the only even number)
```

代码提交地址：
https://www.codewars.com/kata/5526fc09a1bbd946250002dc

第三题： 检测Pangram

难度：6kyu

pangram是一个至少包含每个字母一次的句子。例如，"The quick brown fox jumps over the lazy dog "这个句子就是一个pangram，因为它至少使用了一次字母A-Z（大小写不相关）。

给定一个字符串，检测它是否是一个pangram。如果是则返回 `True` ，如果不是则返回 `False` 。忽略数字和标点符号。
代码提交地址：
https://www.codewars.com/kata/545cedaa9943f7fe7b000048

第四题： 数独解决方案验证

难度：6kyu

数独背景

数独是一种在 9x9 网格上进行的游戏。游戏的目标是用 1 到 9 的数字填充网格的所有单元格，以便每一列、每一行和九个 3x3 子网格（也称为块）中的都包含数字 1 到 9。更多信息请访问：
http://en.wikipedia.org/wiki/Sudoku

编写一个函数接受一个代表数独板的二维数组，如果它是一个有效的解决方案则返回 true，否则返回 false。数独板的单元格也可能包含 0，这将代表空单元格。包含一个或多个零的棋盘被认为是无效的解决方案。棋盘总是 9 x 9 格，每个格只包含 0 到 9 之间的整数。

代码提交地址：
https://www.codewars.com/kata/63d1bac72de941033dbf87ae

第五题： 疯狂的彩色三角形

难度： 2kyu

一个彩色的三角形是由一排颜色组成的，每一排都是红色、绿色或蓝色。连续的几行，每一行都比上一行少一种颜色，是通过考虑前一行中的两个相接触的颜色而产生的。如果这些颜色是相同的，那么新的一行就使用相同的颜色。如果它们不同，则在新的一行中使用缺失的颜色。这个过程一直持续到最后一行，只有一种颜色被生成。

例如：

```
Colour here:            G G        B G        R G        B R
Becomes colour here:     G          R          B          G
```

一个更大的三角形例子：

```
R R G B R G B B
 R B R G B R B
  G G B R G G
   G R G B G
    B B R R
     B G R
      R B
       G
```

你将得到三角形的第一行字符串，你的工作是返回最后的颜色，这将出现在最下面一行的字符串。在上面的例子中，你将得到 "RRGBRGBB"，你应该返回 "G"。
限制条件： 1 <= length(row) <= 10 ** 5
输入的字符串将只包含大写字母'B'、'G'或'R'。

例如：

```
triangle('B') == 'B'
triangle('GB') == 'R'
triangle('RRR') == 'R'
triangle('RGBG') == 'B'
triangle('RBRGBRB') == 'G'
triangle('RBRGBRBGGRRRBGBBBGG') == 'G'
```

代码提交地址：
https://www.codewars.com/kata/5a331ea7ee1aae8f24000175

提示：请参考下面的链接，利用三进制的特点来进行计算。
https://stackoverflow.com/questions/53585022/three-colors-triangles

第二部分

使用Mermaid绘制程序流程图

安装VSCode插件：

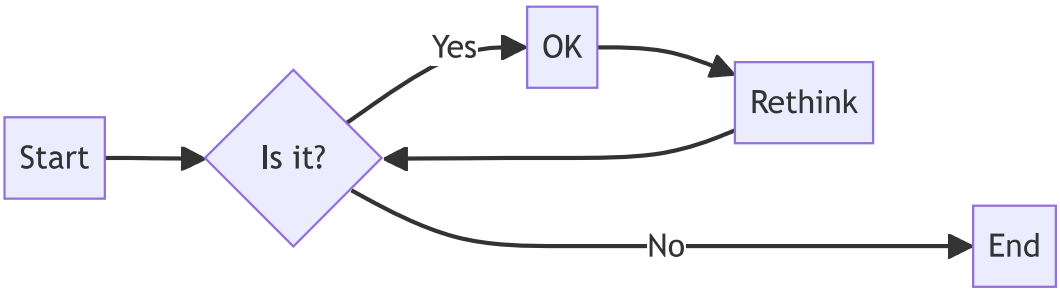
- Markdown Preview Mermaid Support

- Mermaid Markdown Syntax Highlighting

使用Markdown语法绘制你的程序绘制程序流程图（至少一个），Markdown代码如下：

```
flowchart TD
    A[Start] --> B{Is it?}
    B -->|Yes| C[OK]
    C --> D[Rethink]
    D --> B
    B ----->|No| E[End]
```

显示效果如下：



查看Mermaid流程图语法-->点击这里

使用Markdown编辑器（例如VScode）编写本次实验的实验报告，包括实验过程与结果、实验考查和实验总结，并将其导出为 **PDF格式** 来提交。

实验过程与结果

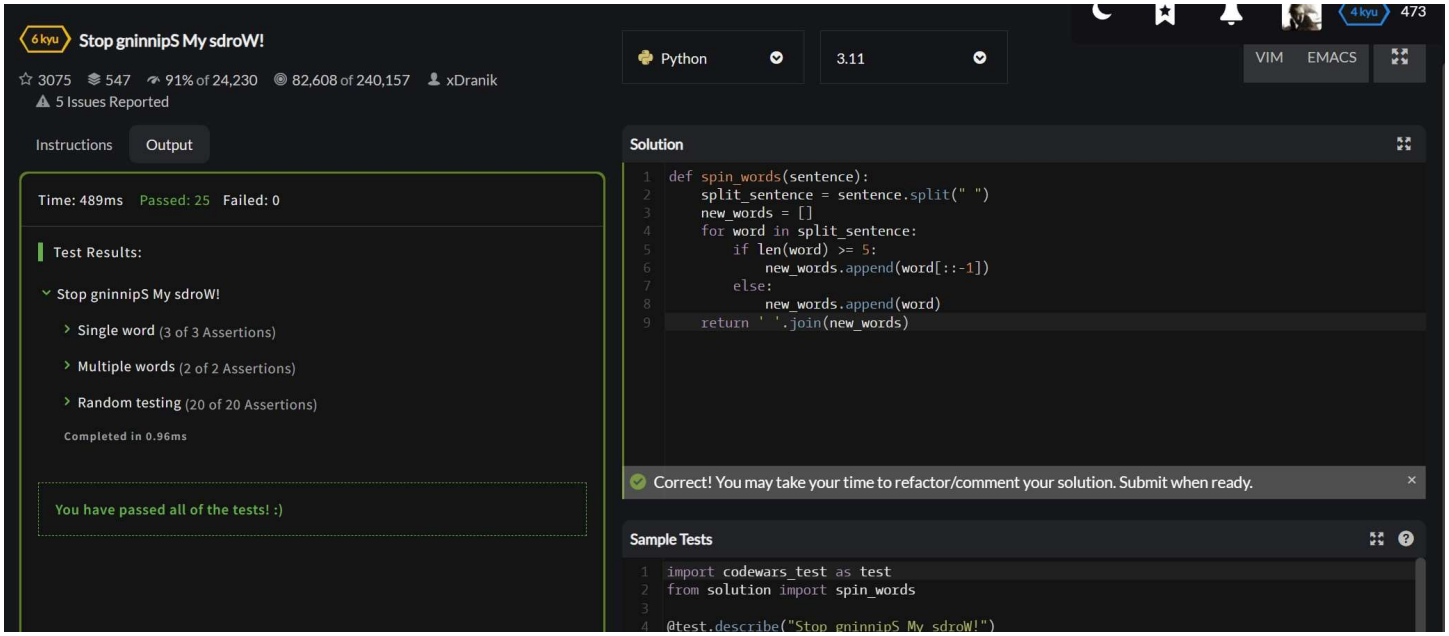
第一部分 第一部分 Codewars Kata挑战

第一题:停止逆转我的单词

代码:

```python
def spin_words(sentence):
    split_sentence = sentence.split(" ")
    new_words = []
    for word in split_sentence:
        if len(word) >= 5:
            new_words.append(word[::-1])
        else:
            new_words.append(word)
    return ' '.join(new_words)
```

运行成功截图:

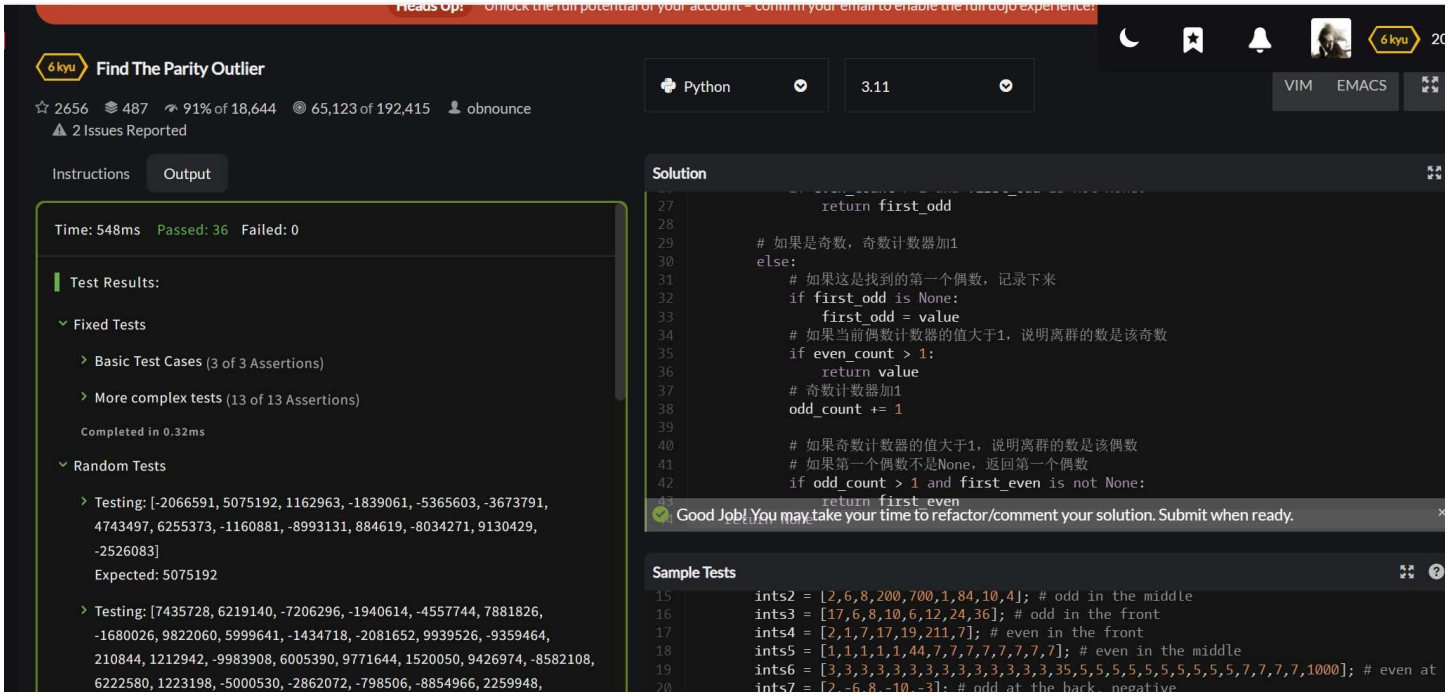


第二题：发现离群的数(Find The Parity Outlier)

代码:

```
def find_outlier(integers):
    # 设置偶数和奇数计数器，初始值都为0
    even_count = 0
    odd_count = 0

    # 设置找到的第一个奇数和偶数，初始值都为None
    first_odd = None
    first_even = None

    # 遍历整数列表
    for value in integers:
        # 如果是偶数
        if value % 2 == 0:
            # 如果这是找到的第一个奇数，记录下来
            if first_even is None:
                first_even = value
            # 如果当前奇数计数器的值大于1，说明离群的数是该偶数
            if odd_count > 1:
                return value

            # 偶数计数器加1
            even_count += 1

            # 如果偶数计数器的值大于1，说明离群的数是该奇数
            # 如果第一个奇数不是None，返回第一个奇数
            if even_count > 1 and first_odd is not None:
                return first_odd

        # 如果是奇数，奇数计数器加1
        else:
            # 如果这是找到的第一个偶数，记录下来
            if first_odd is None:
                first_odd = value
            # 如果当前偶数计数器的值大于1，说明离群的数是该奇数
            if even_count > 1:
                return value
            # 奇数计数器加1
            odd_count += 1

            # 如果奇数计数器的值大于1，说明离群的数是该偶数
            # 如果第一个偶数不是None，返回第一个偶数
            if odd_count > 1 and first_even is not None:
                return first_even
    return None
```

运行成功截图:

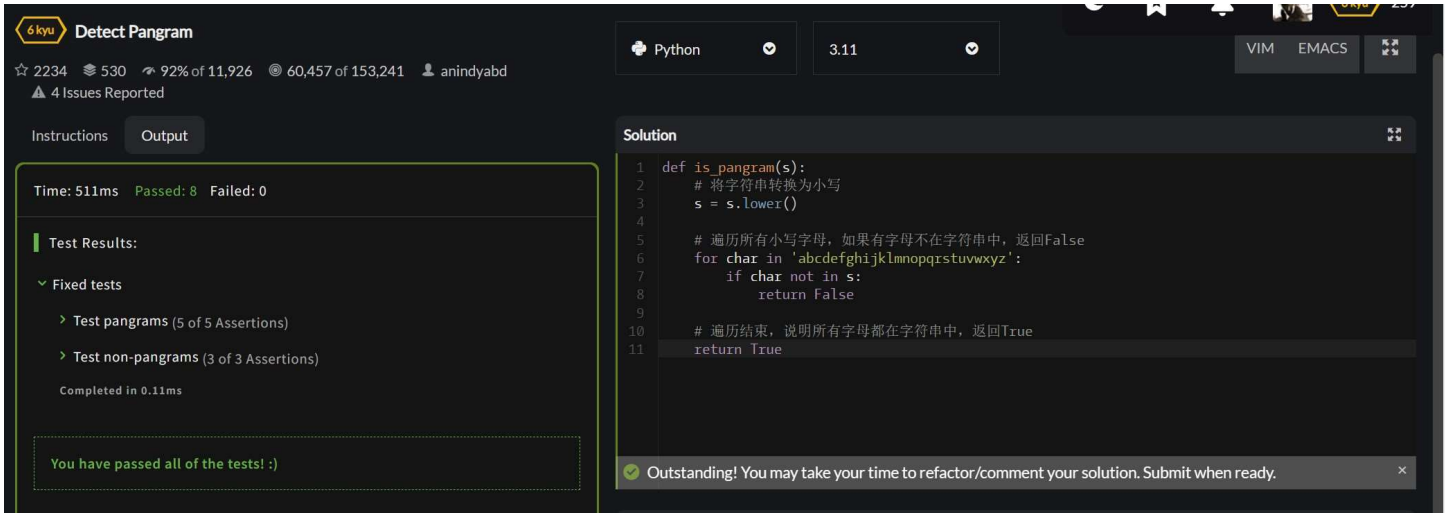


第三题：检测Pangram

代码:

```
def is_pangram(s):
    # 将字符串转换为小写
    s = s.lower()

    # 遍历所有小写字母，如果有字母不在字符串中，返回False
    for char in 'abcdefghijklmnopqrstuvwxyz':
        if char not in s:
            return False

    # 遍历结束，说明所有字母都在字符串中，返回True
    return True
```

运行成功截图:

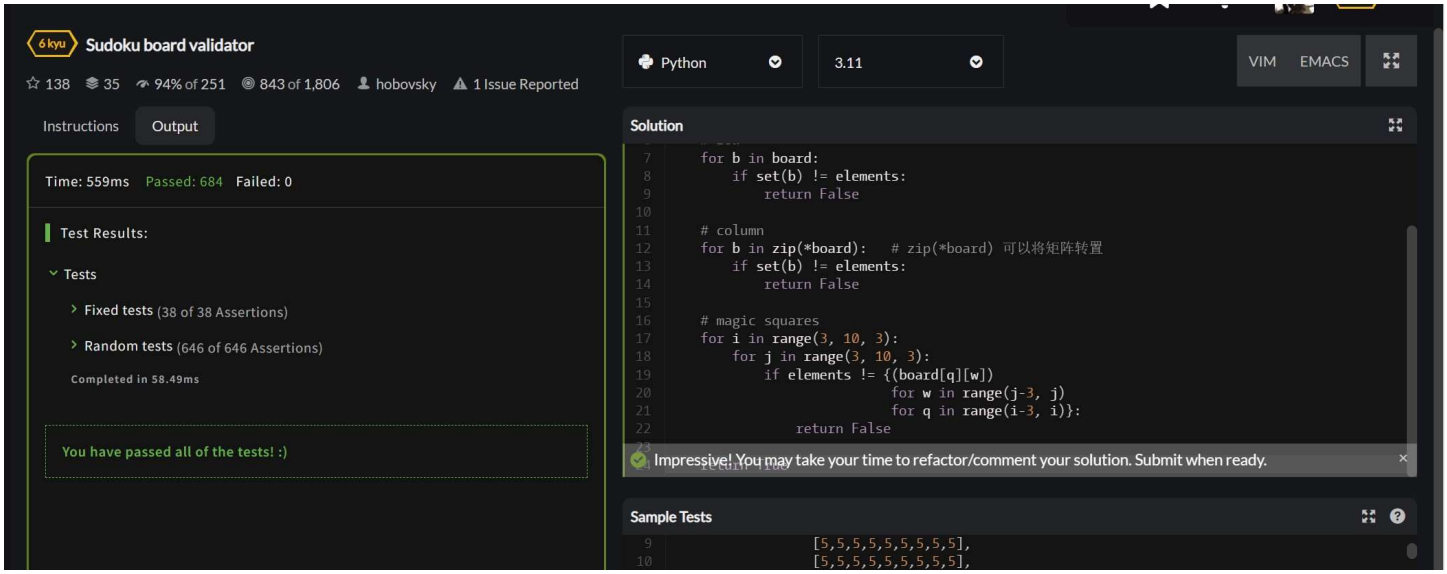


第四题：数独解决方案验证

代码:

```
def validate_sudoku(board):

    # 利用集合进行比较 {1,2,3,4,5,6,7,8,9}
    elements = set(range(1, 10))

    # row
    for b in board:
        if set(b) != elements:
            return False

    # column
    for b in zip(*board):   # zip(*board) 可以将矩阵转置
        if set(b) != elements:
            return False

    # magic squares
    for i in range(3, 10, 3):
        for j in range(3, 10, 3):
            if elements != {(board[q][w])
                            for w in range(j-3, j)
                            for q in range(i-3, i)}:
                return False

    return True
```

运行成功截图:

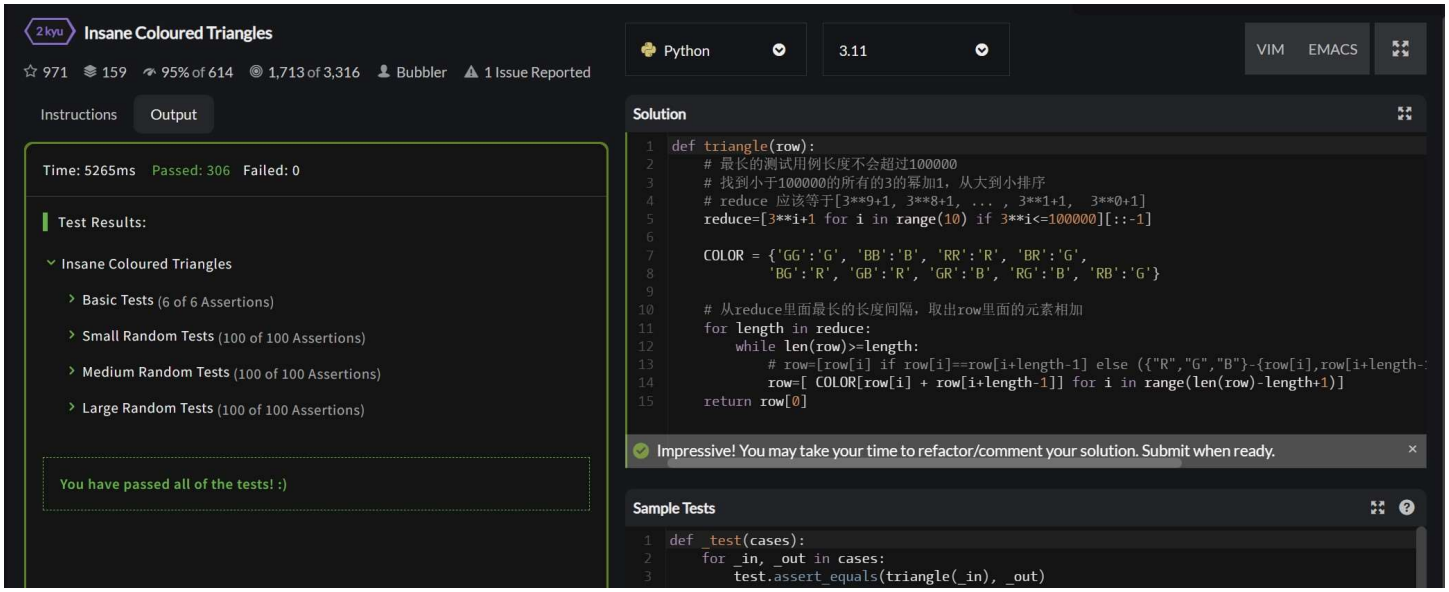


第五题：疯狂的彩色三角形

代码:

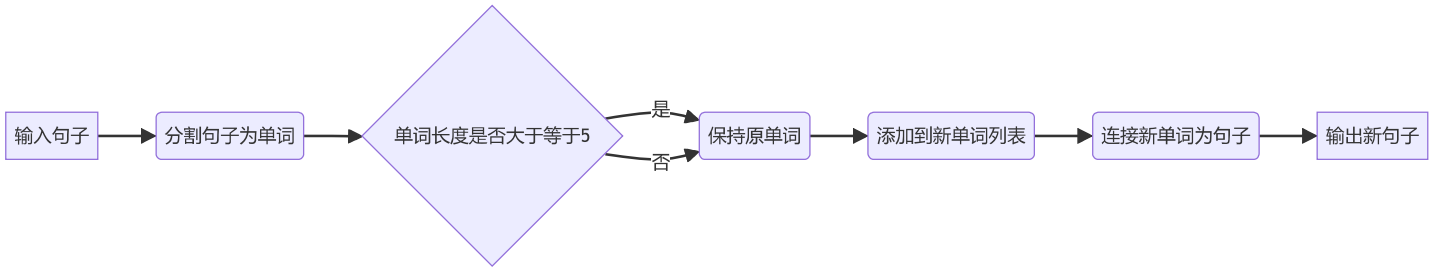
```
def triangle(row):
    # 最长的测试用例长度不会超过100000
    # 找到小于100000的所有的3的幂加1，从大到小排序
    # reduce 应该等于[3**9+1, 3**8+1, ... , 3**1+1,  3**0+1]
    reduce=[3**i+1 for i in range(10) if 3**i<=100000][::-1]

    COLOR = {'GG':'G', 'BB':'B', 'RR':'R', 'BR':'G',
             'BG':'R', 'GB':'R', 'GR':'B', 'RG':'B', 'RB':'G'}

    # 从reduce里面最长的长度间隔，取出row里面的元素相加
    for length in reduce:
        while len(row)>=length:
            row=[ COLOR[row[i] + row[i+length-1]] for i in range(len(row)-length+1)]
    return row[0]
```

运行成功截图:

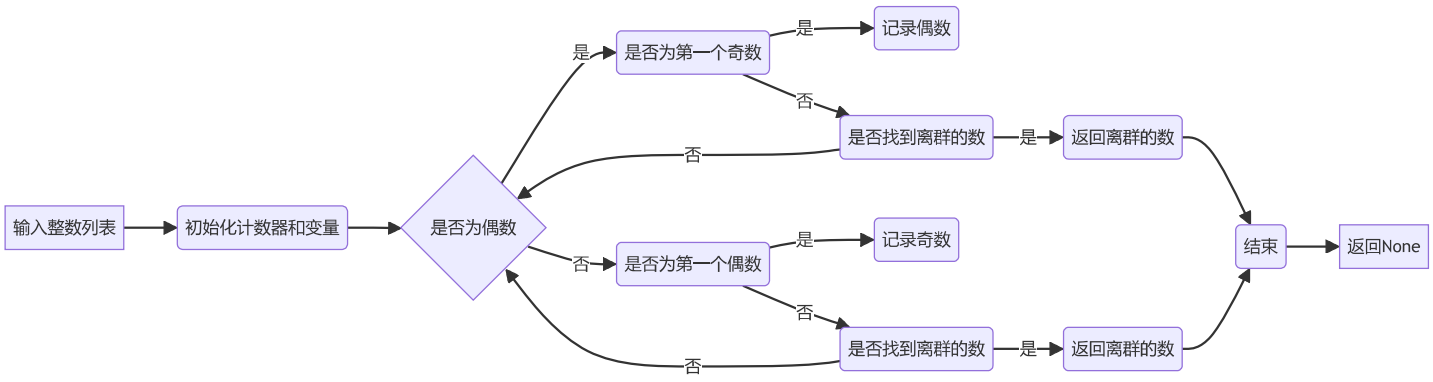


第二部分 使用Mermaid绘制程序流程图

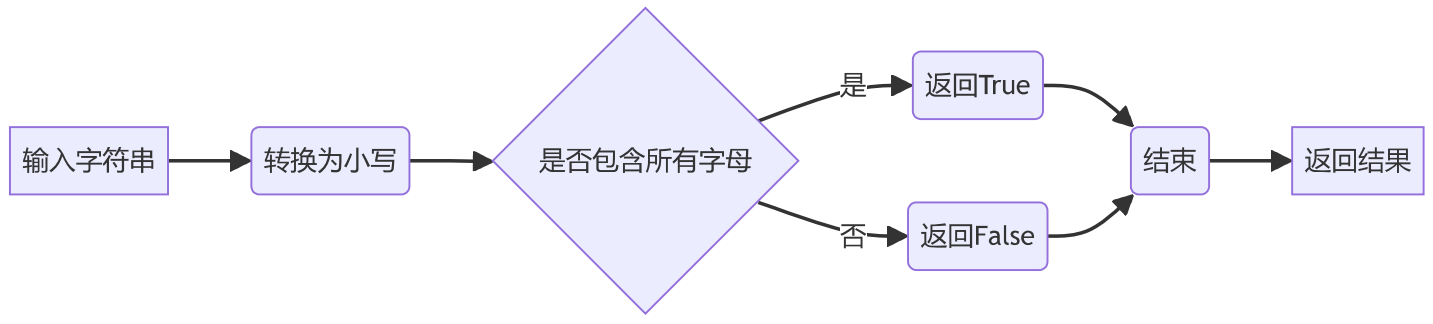
第一题: 停止逆转我的单词



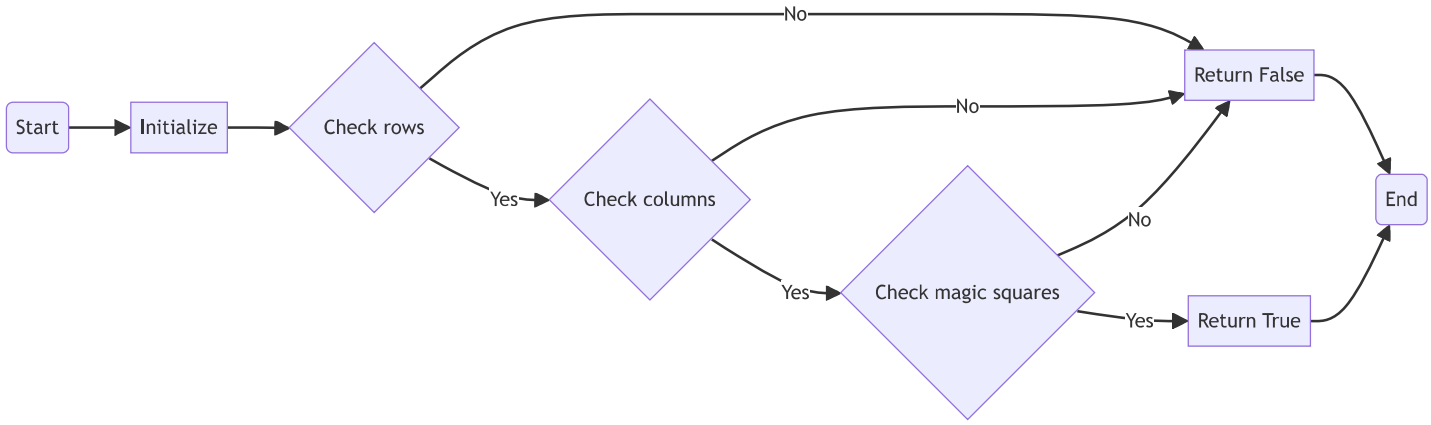
第二题:发现离群的数(Find The Parity Outlier)



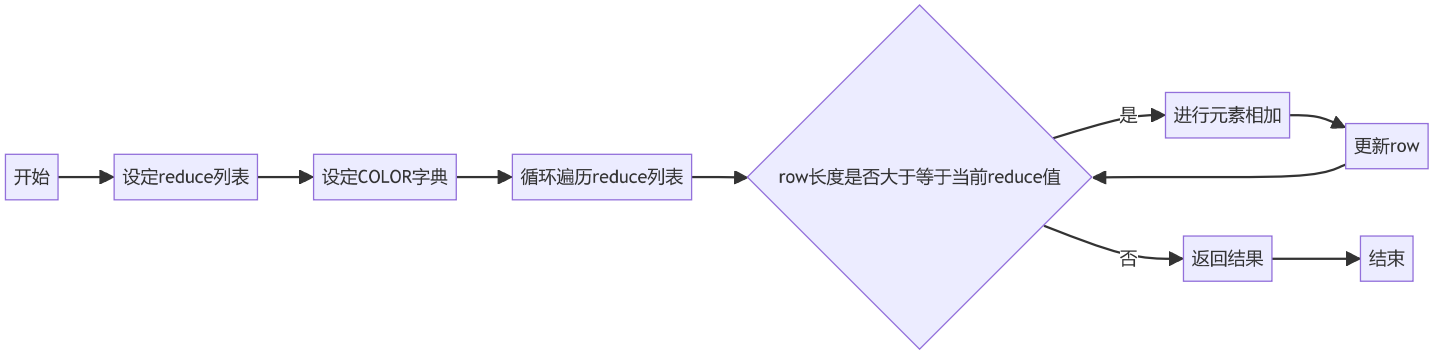
第三题:检测Pangram



第四题:数独解决方案验证



第五题:疯狂的彩色三角形



实验考查

请使用自己的语言并使用尽量简短代码示例回答下面的问题，这些问题将在实验检查时用于提问和答辩以及实际的操作。

1. 集合（set）类型有什么特点？它和列表（list）类型有什么区别？

集合（set）是一种数学中的数据类型，它具有以下几个特点：

- 集合中的元素是唯一的，不会有重复的元素。
- 集合是无序的，即集合中的元素没有固定的顺序。
- 集合不支持索引，即你不能通过索引来访问集合中的元素。
- 集合可以进行交集、并集、差集等数学运算。

与列表（list）类型相比，集合（set）类型主要有以下区别：

- 列表是有序的，而集合是无序的。列表中的元素可以根据索引进行访问，而集合则不能。
- 列表可以包含重复的元素，而集合中的元素是唯一的，不会有重复的元素。
- 列表可以进行各种复杂的操作，如添加、删除、修改元素等，而集合的操- 作相对简单，主要是添加、删除元素，以及进行数学运算。
- 列表适用于存储需要保持顺序的数据，而集合适用于需要对数据进行快速查找和筛选的情况。

-

2. 集合（set）类型主要有那些操作?

- 交集并集差集的运算：可以执行多个集合之间的交集、并集和差集运算，例如，将多个集合合并成一个新集合，或者从某个集合中排除其他集合中的元素。
- 集合运算的运算符：集合可以使用数学运算符（如+、-、*、/等）进行运算，例如，两个集合相加、相减、相乘或相除可以得到一个新的集合。
- 集合运算的函数：Python提供了许多用于集合运算的函数，例如，len()函数可以返回集合中元素的数量，not in运算符可以判断一个元素是否不在集合中，等等。
- 迭代集合：可以使用for循环遍历集合中的所有元素，例如，可以使用for循环打印出集合中的所有元素。
- 集合推导式：可以使用Python中的列表推导式或字典推导式来创建新的集合，例如，可以使用列表推导式将一个列表中的元素转换为另一个集合。
- 集合操作符的优先级：Python中的集合操作符优先级与其他运算符的优先级相同，例如，乘法运算符的优先级高于加法运算符。

3. 使用 * 操作符作用到列表上会产生什么效果？为什么不能使用 * 操作符作用到嵌套的列表上？使用简单的代码示例说明。

在Python中，*操作符可以用于对列表进行复制或者对元组进行解包。当我们对一个列表使用*操作符时，会返回该列表的复制。例如：

```
original_list = [1, 2, 3, 4, 5]
copied_list = *original_list
print(copied_list)  # 输出：[1, 2, 3, 4, 5]
```

这段代码创建了一个名为copied_list的新列表，它是original_list的复制。

然而，对于嵌套的列表，*操作符不能直接使用。这是因为*操作符在遇到嵌套列表时，会尝试对其中的每个元素进行解包，而不是对整个嵌套列表进行复制。例如：

```
nested_list = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
attempted_copy = *nested_list
print(attempted_copy)  # 输出：[1, 2, 3, [4, 5, 6], [7, 8, 9]]
```

这段代码的结果并非我们所期望的嵌套列表的复制，而是返回了原始列表中的元素以及嵌套列表本身。这是因为*操作符在此情况下尝试解包嵌套列表的每个元素，而非整个嵌套列表。

如果你想复制嵌套列表，你需要使用其他方式，例如使用递归函数或者循环来逐层复制内部的列表。

4. 总结列表,集合，字典的解析（comprehension）的使用方法。使用简单的代码示例说明。

在Python中，列表解析（List Comprehension）、集合解析（Set Comprehension）和字典解析（Dictionary Comprehension）是非常方便且强大的语法特性，它们可以简洁明了地生成列表、集合和字典。

列表解析：
列表解析是Python中的一种构造列表的简洁方式。它使用一个表达式 followed by for clause(s) and then a list comprehension.

例子：

```
# 创建一个列表，其中的元素是1到10的平方
squares = [x**2 for x in range(1, 11)]
print(squares)  # 输出：[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

集合解析：
集合解析与列表解析非常相似，但是生成的集合是无序的，且元素是唯一的。

例子：

```
# 创建一个集合，其中的元素是1到10的平方
squares = {x**2 for x in range(1, 11)}
print(squares)  # 输出：{1, 4, 9, 16, 25, 36, 49, 64, 81, 100}
```

字典解析：
字典解析可以用于创建字典，表达式中的键和值都可以由一个表达式来计算。

例子：

```
# 创建一个字典，其中的键是1到10的平方，值是键的平方根
squares = {x**2: x**0.5 for x in range(1, 11)}
print(squares)  # 输出：{1: 1.0, 4: 2.0, 9: 3.0, 16: 4.0, 25: 5.0, 36: 6.0, 49: 7.0, 64: 8.0, 8
```

这些解析（comprehension）都是Python中非常强大的工具，可以帮助你以更简洁高效的方式创建和操作列表、集合和字典。

实验总结

总结一下这次实验你学习和使用到的知识，例如：编程工具的使用、数据结构、程序语言的语法、算法、编程技巧、编程思想。

- 在这次实验中，我学习和使用到的知识非常丰富。首先，我注册了Codewars网站的账号，并完成了一系列的Kata挑战。通过这些挑战，我深入了解了Python编程语言的各种特性和用法。

- 在解决每个Kata挑战的过程中，我学习了如何使用不同的数据结构来组织和处理数据。比如，在停止逆转我的单词这个题目中，我使用了列表来保存字符串中的每个单词，并利用切片操作进行逆转。而在检测Pangram这个题目中，我使用了集合来存储字母，并进行对比和删除操作。这些数据结构的灵活运用使得代码更加简洁、高效。
- 此外，我还学习了一些重要的算法和编程技巧。例如，在发现离群的数这个题目中，我使用了遍历整数列表并分别统计奇偶数的方法，找到了唯一一个离群的数字。在数独解决方案验证这个题目中，我利用了双重循环遍历二维数组，并使用集合来检查每行、每列和每个小方块是否满足数独规则。这些算法和技巧的应用极大地提高了我的问题解决能力。
- 在编程思想方面，我运用了迭代、条件判断、遍历等常见的编程思维方式。这些思想帮助我更好地理解和分析问题，并找到合适的解决方案。同时，我也注重代码的可读性和可维护性，通过合理的命名、注释和代码结构，使得代码更加清晰易懂，并方便后续的维护和修改。
- 最后，在使用Mermaid绘制程序流程图的过程中，我更加深入地思考和设计了程序的执行流程。通过绘制流程图，我可以清晰地将整个程序拆分成不同的模块，并明确它们之间的关系和交互。这种系统化的思考和设计有助于提升我的程序设计能力，使得代码更加可靠和高效。
- 总而言之，这次实验让我学习到了丰富的知识和技能。通过注册Codewars账号并完成Kata挑战，我巩固了Python语言的基本语法和数据结构的应用。同时，通过解决各个题目的算法和编程技巧，我提高了自己的问题解决能力。最重要的是，我意识到编程不仅仅是语法和工具的应用，更是一种思维方式和解决问题的能力。这次实验为我打开了深入学习编程的大门，让我更加热爱和熟悉这个领域。