题目1

给你类 ZeroEvenOdd 的一个实例,该类中有三个函数: zero、even 和 odd。 ZeroEvenOdd 的相同实例将会传递给三个不同线程: + **线程 1**: 调用 zero(), 只输出0+**线程 2**: 调用 even(), 只输出偶数 + **线程 3**: 调用 odd(), 只输出奇数 修改给出的类,以输出序列 "010203040506…",其中序列的长度必须为 2n。

实现 ZeroEvenOdd 类:

- ZeroEvenOdd(int n) 用数字 n 初始化对象,表示需要输出的数。
- void zero(printNumber) 调用 print 以输出一个 0。
- void even(printNumber) 调用 print 以输出偶数。
- void odd(printNumber) 调用 print 以输出奇数。

模板代码

```
public class ZeroEvenOdd {
    private int n;

public ZeroEvenOdd(int n) {
        this.n = n;
    }

public void Zero(Action<int> printNumber) {
    }

public void Even(Action<int> printNumber) {
    }

public void Odd(Action<int> printNumber) {
    }
```

题目2

给你一个包含'('和')'的字符串,找出有效 (以上个子串结尾后英文字符至'('开头格式正确且连续) 括号子串。 格式正确指括号外仅包含英文大小写字符,且长度不为**0**

示例 1:

```
输入: s = "print((\"hello world\")"
输出: []
```

示例 2:

```
输入: s = "add(1, 2)del(2, 1))"
输出: ["add(1, 2)", "del(2, 1)"]
```

示例 3:

```
输入: s = "print(\"hello\");print(\" \")print(\"world\")print((\"hello world\")"
world\")"
输出: ["print(\"hello\")", "print(\" \")", "print(\"world\")"]
```

模板代码

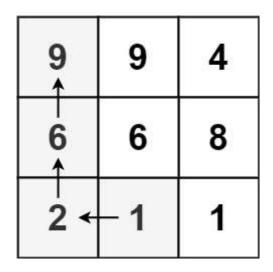
```
public class Solution {
   public string[] ValidParentheses(string s) {
   }
}
```

题目3

给定一个m x n 整数矩阵matrix , 找出其中 最长递增路径 的长度。

对于每个单元格,你可以往上,下,左,右四个方向移动。 你 不能 在 对角线 方向上移动或移动到 边界外 (即不允许环绕)。

示例 1:

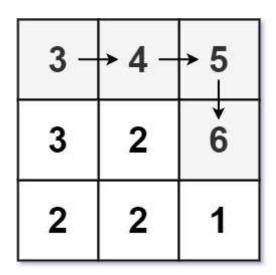


输入: matrix = [[9,9,4],[6,6,8],[2,1,1]]

输出: 4

解释: 最长递增路径为 [1, 2, 6, 9]。

示例 2:



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

输出: 4

解释:最长递增路径是[3,4,5,6]。注意不允许在对角线方向上移动。

示例 3:

输入: matrix = [[1]]

输出: 1

模板代码

```
public class Solution {
    public int LongestIncreasingPath(int[][] matrix) {
    }
}
```