```
/*
在一个「平衡字符串」中，'L' 和 'R' 字符的数量是相同的。

给出一个平衡字符串 s，请你将它分割成尽可能多的平衡字符串。

返回可以通过分割得到的平衡字符串的最大数量。



示例 1：

输入：s = "RLRRLLRLRL"
输出：4
解释：s 可以分割为 "RL", "RRLL", "RL", "RL", 每个子字符串中都包含相同数量的 'L' 和 'R'。
示例 2：

输入：s = "RLLLLRRRLR"
输出：3
解释：s 可以分割为 "RL", "LLLRRR", "LR", 每个子字符串中都包含相同数量的 'L' 和 'R'。
示例 3：

输入：s = "LLLLRRRR"
输出：1
解释：s 只能保持原样 "LLLLRRRR".


提示：

1 <= s.length <= 1000
s[i] = 'L' 或 'R'

来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/split-a-string-in-balanced-strings
著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
*/
```

分析:
从最左端开始遍历字符串元素,遇到L则对应计数变量L_count自加,遇到R则对应计数变量R_count自加,若两个计数变量相等,则可以拆出一个分割,返回值加一,然后两个计数变量清零,继续遍历.

方法一:C++_遍历

```
class Solution
{
    public:
        int balancedStringSplit(string s)
        {
            int ret_val = 0;
            int i       = 0;
            int count   = 0'
            for(i=0;i<s.size();i++)
```

```
            {
                if(s[i] == 'L')
                {
                    count++;
                }
                else
                {
                    count--;
                }
                if(count==0)
                {
                    ++ret_val;
                }
            }
            return ret_val;
        }
};

/*
执行结果：
通过
显示详情
执行用时 :4 ms, 在所有 cpp 提交中击败了70.43%的用户
内存消耗 :8.4 MB, 在所有 cpp 提交中击败了100.00%的用户
*/
```

AlimyBreak
2019.10.24