

在一个由小写字母构成的字符串 `S` 中，包含由一些连续的相同字符所构成的分组。

例如，在字符串 `S = "abbxxxxzzy"` 中，就含有 `"a"`, `"bb"`, `"xxxx"`, `"z"` 和 `"yy"` 这样的一些分组。

我们称所有包含大于或等于三个连续字符的分组为较大分组。找到每一个较大分组的起始和终止位置。

最终结果按照字典顺序输出。

#### 示例 1:

输入: `"abbxxxxzzy"`

输出: `[[3,6]]`

解释: `"xxxx"` 是一个起始于 3 且终止于 6 的较大分组。

#### 示例 2:

输入: `"abc"`

输出: `[]`

解释: `"a"`, `"b"` 和 `"c"` 均不是符合要求的较大分组。

#### 示例 3:

输入: `"abcdddeeeeaabbbcd"`

输出: `[[3,5],[6,9],[12,14]]`

说明: `1 <= S.length <= 1000`

思路，直接找连续的数字，大于3后，当其不连续后进行判断，前面连续大于等于3，加入数组，不大于，放弃。数组遍历到最后再进行一次遍历即可

```
class Solution {
public:
    vector<vector<int>> largeGroupPositions(string S) {
        vector<vector<int>> result;
        int num=1;
        int n=S.size();
        for(int i=1;i<n;++i){
            if(S[i]==S[i-1]){
                if(i==n-1&&num>1){
                    vector<int> re;
                    re.push_back(i-num);
                    re.push_back(i);
                    result.push_back(re);
                }
                num++;
            }else if(num>2){

```

```
        vector<int> re;  
        re.push_back(i-num);  
        re.push_back(i-1);  
        result.push_back(re);  
        num=1;  
    }else{  
        num=1;  
    }  
}  
return result;  
}  
};
```