

给定一个包含大写字母和小写字母的字符串，找到通过这些字母构造的最长的回文串。  
在构造过程中，请注意区分大小写。比如 `"Aa"` 不能当做一个回文字符串。

注意：

假设字符串的长度不会超过 1010。

示例 1:

输入：

`"abcccccdd"`

输出：

7

解释：

我们可以构造的最长的回文串是 `"dccaccd"`，它的长度是 7。

思路，map存储数据，然后遍历map，偶数个给拿出来统计，若有单独奇数个，奇数个数统计为1，最后返回偶数和奇数之和就得到答案

代码

```
class Solution {
public:
    int longestPalindrome(string s) {
        map<char, int> nums;
        map<char, int>::iterator iter;
        for(int i=0; i<s.size(); i++) {
            nums[s[i]]++;
        }
        int num=0;
        int flag=0;
        iter=nums.begin();
        while(iter!=nums.end()) {
            if(iter->second%2==1) {
                flag=1;
            }
            num+=iter->second/2*2;
            iter++;
        }
        return num+flag;
    }
};
```