

## 409. Longest Palindrome

Easy
1779
115
Add to List
Share

Given a string `s` which consists of lowercase or uppercase letters, return the length of the **longest palindrome** that can be built with those letters.

Letters are **case sensitive**, for example, `"Aa"` is not considered a palindrome here.

### Example 1:

**Input:** `s = "abcccd"`

**Output:** 7

**Explanation:**

One longest palindrome that can be built is `"dcca"`, whose length is 7.

### Example 2:

**Input:** `s = "a"`

**Output:** 1

### Example 3:

**Input:** `s = "bb"`

**Output:** 2

### Constraints:

- `1 <= s.length <= 2000`
- `s` consists of lowercase **and/or** uppercase English letters only.

```

1 class Solution:
2     def longestPalindrome(self, s: str) -> int:
3
4         di = {}
5         for i in s:
6
7             if i not in di:
8                 di.update({i: 1})
9             else :
10                di[i] += 1
11
12        print(di)
13        output,c = 0,0
14        for i in di.values() :
15
16            if i%2 ==0 :
17                output += i
18            else :
19                c+=1
20                output += i-1
21
22        return (output + c)

```

Testcase
Run Code Result
Debugger

Accepted

Runtime: 42 ms

Your input

"abcccd"

stdout

{'a': 1, 'b': 1, 'c': 4, 'd': 2}

Output

7

Diff

Expected

7