

## ★ 551 Student Attendance Record I

2018年4月14日 16:42

You are given a string representing an attendance record for a student. The record only contains the following three characters:

1. 'A': Absent.
2. 'L': Late.
3. 'P': Present.

A student could be rewarded if his attendance record doesn't contain **more than one 'A' (absent)** or **more than two continuous 'L' (late)**.

You need to return whether the student could be rewarded according to his attendance record.

**Example 1:**

**Input:** "PPALLP"

**Output:** True

**Example 2:**

**Input:** "PPALLL"

**Output:** False

来自 <<https://leetcode.com/problems/student-attendance-record-i/description/>>

给定一个字符串来代表一个学生的出勤纪录，这个纪录仅包含以下三个字符：

1. 'A': Absent, 缺勤
2. 'L': Late, 迟到
3. 'P': Present, 到场

如果一个学生的出勤纪录中不超过一个'A'(缺勤)并且不超过两个连续的'L'(迟到),那么这个学生会被奖赏。

你需要根据这个学生的出勤纪录判断他是否会被奖赏。

### Solution for Python3:

```
1 class Solution1:
2     def checkRecord(self, s):
3         """
4         :type s: str
5         :rtype: bool
6         """
7         Acnt, Lcnt = 0, 0
8         for i in range(len(s)):
9             if s[i] == 'L':
10                 Lcnt += 1
11             else:
12                 if s[i] == 'A':
13                     Acnt += 1
14                     Lcnt = 0
15                 if Acnt > 1 or Lcnt > 2:
16                     return False
17             return True
18
19 class Solution2:
20     def checkRecord(self, s):
21         """
22         :type s: str
23         :rtype: bool
24         """
25         import re
26         # return re.match(r'^\d{3}\-\d{3,8}$', '010-12345')
27         return not re.match(r'[ALP]*[L]{3}[ALP]*| [ALP]*A[ALP]*A[ALP]*', s)
28
29 class Solution3:
30     def checkRecord(self, s):
```

```

31         """
32         :type s: str
33         :rtype: bool
34         """
35         import re
36         # return not re.match(r'*.?LLL.*|.?.*A.*', s)
37         return not re.match(r'*.?(LLL|A.*A).*', s)
38
39 class Solution4:
40     def checkRecord(self, s):
41         """
42         :type s: str
43         :rtype: bool
44         """
45         import re
46         return not re.search('A.*A|LLL', s)
47
48
49 class Solution5:
50     def checkRecord(self, s):
51         """
52         :type s: str
53         :rtype: bool
54         """
55         return s.find('A') == s.rfind('A') and s.find('LLL') == -1
56         #find不存在时返回-1, index不存在报错
57
58 class Solution6:
59     def checkRecord(self, s):
60         """
61         :type s: str
62         :rtype: bool
63         """
63         return not (s.count('A') > 1) or 'LLL' in s)

```

## Solution for C++:

```

1  class Solution1 {
2  public:
3      bool checkRecord(string s) {
4          int Acnt = 0, Lcnt = 0;
5          for (int i = 0; i < s.length(); i++) {
6              if (s[i] == 'L') {
7                  if (++Lcnt == 3)
8                      return false;
9              } else {
10                 if (s[i] == 'A') {
11                     if (++Acnt == 2)
12                         return false;
13                 }
14                 Lcnt = 0;
15             }
16         }
17         return true;
18     }

```

```

19 };
20
21
22 class Solution2 {
23 public:
24     bool checkRecord(string s) {
25         return !regex_match(s, regex(".*(LLL|A.*A).*"));
26         // return !regex_match(s, regex(".*LLL.*|.*A.*A.*"));
27     }
28 };
29
30 class Solution3 {
31 public:
32     bool checkRecord(string s) {
33         return !regex_search(s, regex("A.*A|LLL"));
34     }
35 };
36
37 class Solution4 {
38 public:
39     bool checkRecord(string s) {
40         return s.find_first_of('A') == s.find_last_of('A') && s.find("LLL")
41 == -1;
42     }
43 };

```