# Validating Postal Codes ★

135 more points to get your gold badge!

Rank: **210951** | Points: **265/400**

**Your Validating Postal Codes submission got 80.00 points.**    শেয়ার ২    Tweet

You are now 135 points away from the gold level for your python badge.

**Try the next challenge** | **Try a Random Challenge**

Problem      Submissions      Leaderboard      Editorial 🔒

A valid postal code $P$ have to fullfil both below requirements:

1. $P$ must be a number in the range from **100000** to **999999** inclusive.
2. $P$ must not contain more than one alternating repetitive digit pair.

Alternating repetitive digits are digits which repeat immediately after the next digit. In other words, an alternating repetitive digit pair is formed by two equal digits that have just a single digit between them.

For example:

```
121426 # Here, 1 is an alternating repetitive digit.
523563 # Here, NO digit is an alternating repetitive digit.
552523 # Here, both 2 and 5 are alternating repetitive digits.
```

Your task is to provide two regular expressions `regex_integer_in_range` and `regex_alternating_repetitive_digit_pair`. Where:

`regex_integer_in_range` should match only integers range from **100000** to **999999** inclusive

`regex_alternating_repetitive_digit_pair` should find alternating repetitive digits pairs in a given string.

Both these regular expressions will be used by the provided code template to check if the input string $P$ is a valid postal code using the following expression:

```
(bool(re.match(regex_integer_in_range, P))
and len(re.findall(regex_alternating_repetitive_digit_pair, P)) < 2)
```

**Input Format**

Locked stub code in the editor reads a single string denoting $P$ from stdin and uses provided expression and your regular expressions to validate if $P$ is a valid postal code.

**Output Format**

You are not responsible for printing anything to stdout. Locked stub code in the editor does that.

**Sample Input 0**

```
110000
```

**Sample Output 0**

```
False
```

**Explanation 0**

1 1 0**0**0**0** : (0, 0) and (**0**, **0**) are two alternating digit pairs. Hence, it is an invalid postal code.

**Note:**

A score of **0** will be awarded for using `'if'` conditions in your code.
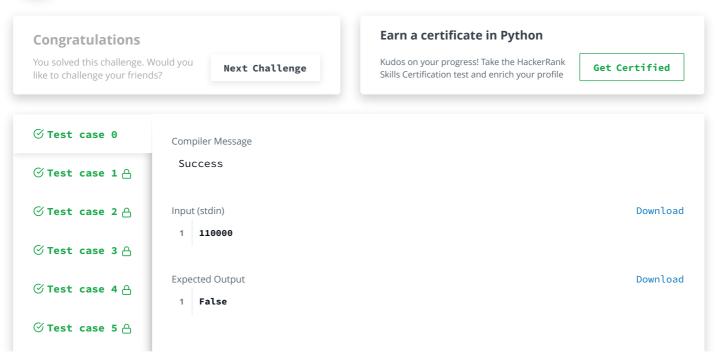
You have to pass all the testcases to get a positive score.

Change Theme     Language  Python 3          ▾   ⟳  ⤢  ⋮

```
 1   import re
 2   s=input()
 3   print (bool(re.match(r'^[1-9][\d]{5}$',s) and len(re.findall(r'(\d)(?=\d\1)',s))<2 ))
 4
 5
 6
 7
 8   import re
 9   P = input()
10
11   print (bool(re.match(regex_integer_in_range, P))
12   and len(re.findall(regex_alternating_repetitive_digit_pair, P)) < 2)
```

Line: 3 Col: 86

⬆ Upload Code as File     ☐ Test against custom input                    **Run Code**     **Submit Code**

You have earned 80.00 points!
You are now 135 points away from the gold level for your python badge.

**25%**                                               265/400

**Python**
★★★★

### Congratulations
You solved this challenge. Would you like to challenge your friends?

**Next Challenge**

### Earn a certificate in Python
Kudos on your progress! Take the HackerRank Skills Certification test and enrich your profile

**Get Certified**

⊘ **Test case 0**

⊘ **Test case 1** 🔒

⊘ **Test case 2** 🔒

⊘ **Test case 3** 🔒

⊘ **Test case 4** 🔒

⊘ **Test case 5** 🔒

Compiler Message

  Success

Input (stdin)                                        Download

  1   **110000**

Expected Output                                      Download

  1   **False**

⊘ **Test case 6** 🔒

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature

https://www.hackerrank.com/challenges/validating-postalcode/problem?h_r=next-challenge&h_v=zen&h_r=next-challenge&h_v=zen&h_r=next-ch…     3/3