Strong Password ☆

142/563 challenges solved

Rank: **4351** | Points: **3344.07** ⊘

| Problem | Submissions | Leaderboard | Editorial |

RATE THIS CHALLENGE

☆ ☆ ☆ ☆ ☆

Louise joined a social networking site to stay in touch with her friends. The signup page required her to input a name and a password. However, the password must be strong. The website considers a password to be strong if it satisfies the following criteria:

- Its length is at least 6.
- It contains at least one digit.
- It contains at least one lowercase English character.
- It contains at least one uppercase English character.
- It contains at least one special character. The special characters are: !@#$%^&*()-+

She typed a random string of length n in the password field but wasn't sure if it was strong. Given the string she typed, can you find the minimum number of characters she must add to make her password strong?

Note: Here's the set of types of characters in a form you can paste in your solution:

```
numbers = "0123456789"
lower_case = "abcdefghijklmnopqrstuvwxyz"
upper_case = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
special_characters = "!@#$%^&*()-+"
```

Input Format

The first line contains an integer n denoting the length of the string.

The second line contains a string consisting of n characters, the password typed by Louise. Each character is either a lowercase/uppercase English alphabet, a digit, or a special character.

Constraints

- $1 \le n \le 100$

Output Format

Print a single line containing a single integer denoting the answer to the problem.

Sample Input 0

```
3
Ab1
```

Sample Output 0

3

Explanation 0

She can make the password strong by adding 3 characters, for example, $hk, turning the password into `Ab1$hk` which is strong.

2 characters aren't enough since the length must be at least 6.

Sample Input 1

```
11
#HackerRank
```

Sample Output 1

```
1
```

Explanation 1

The password isn't strong, but she can make it strong by adding a single digit.

Current Buffer (saved locally, editable)   Java 7 ⌄

```java
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Solution {

    static int minimumNumber(int lengthOfString, String password) {
        int minLengthOfPassword = 6;
        int missingKeyChars = 0;

        Pattern pattern_LowercaseEnglishCharacters = Pattern.compile("\\p{Lower}");
        Matcher matcher_LowercaseEnglishCharacters =
    pattern_LowercaseEnglishCharacters.matcher(password);
        missingKeyChars += matcher_LowercaseEnglishCharacters.find() ? 0 : 1;

        Pattern pattern_UppercaseEnglishCharacters = Pattern.compile("\\p{Upper}");
        Matcher matcher_UppercaseEnglishCharacters =
    pattern_UppercaseEnglishCharacters.matcher(password);
        missingKeyChars += matcher_UppercaseEnglishCharacters.find() ? 0 : 1;

        Pattern pattern_SpecialhCharacters = Pattern.compile("[!@#$%^&*()+//-]");
        Matcher matcher_SpecialhCharacters =
    pattern_SpecialhCharacters.matcher(password);
        missingKeyChars += matcher_SpecialhCharacters.find() ? 0 : 1;

```

```java
23          Pattern pattern_Digits = Pattern.compile("\\d");
24          Matcher matcher_Digits = pattern_Digits.matcher(password);
25          missingKeyChars += matcher_Digits.find() ? 0 : 1;
26
27          int numberOfCharsToBeAdded = Math.max(missingKeyChars, minLengthOfPassword -
    lengthOfString);
28
29          return numberOfCharsToBeAdded;
30      }
31
32 ▼    public static void main(String[] args) {
33          Scanner reader = new Scanner(System.in);
34          int lengthOfString = reader.nextInt();
35          reader.nextLine();
36
37          String password = reader.nextLine();
38          reader.close();
39
40          int answer = minimumNumber(lengthOfString, password);
41          System.out.println(answer);
42      }
43 }
```

Line: 1 Col: 1

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

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