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 = "!@#$%^&*()-+"
```

Example

$password = '2bbbb'$

This password is 5 characters long and is missing an uppercase and a special character. The minimum number of characters to add is 2.

$password = '2bb\#A'$

This password is 5 characters long and has at least one of each character type. The minimum number of characters to add is 1.

Function Description

Complete the minimumNumber function in the editor below.

minimumNumber has the following parameters:

- int n: the length of the password
- string password: the password to test

Returns

- int: the minimum number of characters to add

Input Format

The first line contains an integer n, the length of the password.

The second line contains the password string. Each character is either a lowercase/uppercase English alphabet, a digit, or a special character.

Constraints

- $1 \leq n \leq 100$
- All characters in $password$ are in [a-z], [A-Z], [0-9], or [!@#$%^&*()-+].

```python
#
# Complete the 'minimumNumber' function below.
#
# The function is expected to return an INTEGER.
# The function accepts following parameters:
#  1. INTEGER n
#  2. STRING password
#

def minimumNumber(n, password):
    # Return the minimum number of characters to make the password strong
    numbers = "0123456789"
    lower_case = "abcdefghijklmnopqrstuvwxyz"
    upper_case = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    special_characters = "!@#$%^&*()-+"

    (lower, upper, special, number) = (False, False, False, False)
    for character in password:
        if numbers.find(character) != -1: number = True
        if lower_case.find(character) != -1: lower = True
        if upper_case.find(character) != -1: upper = True
        if special_characters.find(character) != -1: special = True

    count = 0
    if number is False: count += 1
    if lower is False: count += 1
    if upper is False: count += 1
    if special is False: count += 1

    if len(password) + count < 6:
        return count + (6 - (len(password) + count))
    else:
        return count
```

Line: 11 Col: 1

Upload Code as File Test against custom input

Run Code Submit Code

Congratulations

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

Next Challenge

Test case 83
Test case 84
Test case 85
Test case 86
Test case 87
Test case 88
Test case 89

Compiler Message

```
Success
```

Input (stdin) Download

```
3
Ab1
```

Expected Output Download

```
3
```