



Lab Assignment 2

In this assignment you are to write a program to solve the following problem. As with all assignments, remember the following submission steps:

- Make sure your code passes at least all the provided JUnit tests
- Save, commit, and push all code changes
- Confirm the latest code is visible via the “Repository” tab of your repository website
- Confirm that the instructor has Developer access
- Note: you do **NOT** need to document your code

Problem a (LA2a.java)

In this assignment you are to implement a challenge-response system¹ to protect a user’s numeric PIN, which they will enter via a phone. First you will be supplied a 10-digit numeric string (the *challenge*). Then, a PIN will be supplied as though typed into a phone:



So, “AGOT” would signify PIN 2468. To compute a *response*, substitute each PIN digit with the corresponding challenge digit. For example, assume this challenge (indexes shown for convenience):

SEQUENCE	3	2	3	1	1	3	2	2	1	3
INDEX	0	1	2	3	4	5	6	7	8	9

If supplied “AGOT” via phone (PIN: 2468), you would respond 3121 (the value at index 2, followed by index 4, ...). Note that the PIN can be any length (1 or more characters). Also note that if the challenge has repeated digits, the PIN itself is protected because many input PINs could have the same response (and future communications can use a different challenge).

You will need to validate the challenge (10 all-digit characters) and the PIN (all digits) and then output the response. For example:

¹ https://en.wikipedia.org/wiki/Challenge%E2%80%93response_authentication

Enter value sequence: 12345
Enter PIN: HELLO
Invalid sequence

Enter value sequence: 0.12345678
Enter PIN: HELLO
Invalid sequence

Enter value sequence: 0123456789
Enter PIN: Hello
Invalid PIN

Enter value sequence: 0123456789
Enter PIN: HELLO
Response: 43556

Enter value sequence: 3231132213
Enter PIN: AGOT
Response: 3121

To build up to this capability, you will have to implement several methods, each with their own tests. It is recommended that you first implement the following method:

- **letterToPhone** converts letters to numbers via the phone keypad

and then proceed to:

- **getResponse** computes the response given challenge & PIN

followed by these helpers:

- **stringIsKDigits** helps validate the challenge (is a string a specified length?)
- **allDigits** helps validate the challenge (is a string all digits?)
- **allUppercaseLetters** helps validate the PIN (is a string all uppercase letters?)
- **digitStringToIntArray** helps access challenge digits via index in an array

and then finally implement the **main** method to complete the program.

Bonus

While not required, there are two optional methods, each with their own tests, that can earn you extra points. The goal is to compute, for a given challenge, how many PINs would produce the same response. For example, given the challenge on page 1, 3121 could have been produced by 108 PINs (3 occurs four times in the challenge, 1 occurs three times, 2 occurs three times: $4 \times 3 \times 3 \times 3 = 108$).

- **countValues** finds how many times a digit is found within an array (i.e. the challenge)
- **numPossible** multiplies the counts from a full response