

APT: Luhn Algorithm

Problem Statement

The *Luhn Checksum* helps validate credit card numbers (Mastercard, Visa, American Express, etc.), Canadian Social Insurance Numbers, and several other numbers. You must research how the Luhn algorithm (also called the mod 10 algorithm) calculates validity and write a method to determine if a sequence of digits stored in an array is valid according to this method.

The parameter `number` is a sequence of digits, the element at index zero represents the left-most digit, the right-most digit is the last element in the array.

The method `isValid` should return the String "YES" if the array `number` represents a valid sequence according to the Luhn algorithm and "NO" if the sequence is not valid.

Class

```
public class Luhn {
    public String isValid(int[] number) {
        // fill in code here
    }
}
```

Notes and Constraints

- The array `number` represents a sequence of digits to be checked for validity. The element at index zero of the array represents the left-most digit in the sequence.
- Parameter `number` will have at least one element and no more than 50 elements.
- Each value in `number` will be between 0 and 9 inclusive.

Examples

1. `number = [1,2,1,4]`
Returns: "YES"

The Luhn algorithm generates a sum from right-to-left of $4 + 1*2 + 2 + 1*2 = 10$ which is divisible by 10 and thus a valid number/sequence.

2. `number = [1,7,2,3,6]`
Returns: "YES"

The Luhn algorithm generates (from right-to-left) $6 + 6 + 2 + 5 + 1 = 20$ which is divisible by 10.

3. `number = [7,8,3]`
Returns: "NO"

The Luhn algorithm generates (from right-to-left) $3 + 7 + 7 = 17$ which is not divisible by 10.



This work is copyright © Owen Astrachan and is licensed under a [Creative Commons Attribution-Share Alike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/).