Introduction to Python

Luhn's Algorithm

Checking the Validity of Credit Card Numbers

Luhn's Algorithm

When you mistype your credit card number in the browser, the browser will alert you immediately.

This response does not come from the merchant(e.g., Amazon). It happens locally on your computer: Javascript code downloaded to your computer runs a quick check using a simple algorithm called Luhn's algorithm.

This algorithm is meant to correct simple mistakes not malicious attacks.

It's possible, of course, that the credit card number passes Luhn's algorithm check but is still an invalid credit card account.

Luhn's Algorithm

7	9	9	2	7	3	9	8	7	1	3

Luhn's Algorithm

7	9	9	2	7	3	9	8	7	1	3
	18		4		6		16		2	

Starting from the right column, every second digit is doubled.

Luhn's Algorithm

7	9	9	2	7	3	9	8	7	1	3
	18		4		6		16		2	
	9		4		6		7		2	

If the doubled value is >= 10, add the digits.

Luhn's Algorithm

7	9	9	2	7	3	9	8	7	1	3	
	18		4		6		16		2		
7	9	9	4	7	6	9	7	7	2	3	

Carry down the other columns.

Luhn's Algorithm

7	9	9	2	7	3	9	8	7	1	3	
	18		4		6		16		2		
7	9	9	4	7	6	9	7	7	2	3	= 70

Then add the all the entries in the last row.
If the result is a multiple of 10, then the number passes Luhn's algorithm and is a POSSIBLE credit card number.

Different Types of Credit Cards

In addition to Luhn's check, different credit cards have additional constraints.

A Visa card for example must have either 16 or 13 digits AND must start with the digit 4.

A MasterCard must have exactly 16 digits AND the first digit must be a 5 AND the second digit must be between 1 and 5 inclusive.

Luhn's Algorithm

Write a program on repl.it that checks if a given number is a valid Visa OR MasterCard credit card number. Your program should implement 4 functions:

1) is_credit which accepts a string of digits and returns whether the digits passes Luhn's algorithm. USE ONLY ONE FOR LOOP to traverse through the digits.

2) is_visa which accepts a string of digits and returns whether it passes the additional Visa requirements.

3) is_mastercard which accepts a string of digits and returns whether it passes the additional MasterCard requirements.

4) main() function. See the next slide for additional requirements.

is_credit

To test if is_credit is correct, try the following inputs:

is_credit("6011000000000004") should return True
is_credit("79927398713") should return True
is_credit("30000000000004") should return True
is_credit("6911600873502604") should return False
is_credit("1234567890") should return False

Luhn's Algorithm

The main function should accept two inputs: type of credit card and credit card number separated by a space. The type of credit card number should be case insensitive, for example "visA", "Visa", "viSa" should all work.

Before calling is_credit, is_visa or is_mastercard, the main function should check that the credit number entered is even a string of numbers. Use the string isnumeric() method. For example s.isnumeric() returns True if all of the character is s are numbers.

See the next slide for some sample runs that you can use to check your program

Luhn's Algorithm

Enter type and credit card number(Visa, MasterCard Only):Visa 4111111111111111

Valid

Enter type and card number(Visa, MasterCard): visA 4111111111111111

Valid

Enter type and card number(Visa, MasterCard): mastercard 5500000000000004

Valid

Enter type and credit card number(Visa, MasterCard): mastercard 123ACBCDEG4

Invalid

Enter type and credit card number(Visa, MasterCard): AmericanExpress 340000000000009

Invalid