Table of Contents

Credit

tl;dr

Implement a program that determines whether a provided credit card number is valid according to Luhn's algorithm.

```
$ python credit.py
Number: 378282246310005
AMEX
```

Specification

- In `credit.py` in `~/workspace/pset6/credit/`, write a program that prompts the user for a credit card number and then reports (via `print`) whether it is a valid American Express, MasterCard, or Visa card number, exactly as you did in Problem Set 1

(https://lab.cs50.io/cs50/labs/2019/x/credit/), except that your program this time should be written (a) in Python and (b) in CS50 IDE.

- So that we can automate some tests of your code, we ask that your program's last line of output be `AMEX\n` or `MASTERCARD\n` or `VISA\n` or `INVALID\n`, nothing more, nothing less.

- For simplicity, you may assume that the user's input will be entirely numeric (i.e., devoid of hyphens, as might be printed on an actual card).

- Best to use `get_int` or `get_string` from CS50's library to get users' input, depending on how you to decide to implement this one.

Walkthrough

credit (Python)

Usage

Your program should behave per the example below. Assume that the underlined text is what some user has typed.

```
$ python credit.py
Number: 378282246310005
AMEX
```

```
$ python credit.py
Number: 3782-822-463-10005
Number: foo
Number: 378282246310005
AMEX
```

```
$ python credit.py
Number: 6176292929
INVALID
```

Testing

Correctness

```
check50 cs50/problems/2019/x/sentimental/credit
```

Style

```
style50 credit.py
```

Staff Solution

```
~cs50/2019/x/pset6/credit
```

How to Submit

Execute the below, logging in with your GitHub username and password when prompted. For security, you'll see asterisks (`*`) instead of the actual characters in your password.

```
submit50 cs50/problems/2019/x/sentimental/credit
```

You can then go to [https://cs50.me/cs50x (https://cs50.me/cs50x)](https://cs50.me/cs50x) to view your current scores!

Hints

Test out your program with a whole bunch of inputs, both valid and invalid. (We certainly will!) Here are a few card numbers that PayPal recommends for testing:

[https://developer.paypal.com/docs/classic/payflow/payflow-pro/payflow-pro-testing/#credit-card-numbers-for-testing (https://developer.paypal.com/docs/classic/payflow/payflow-pro/payflow-pro-testing/#credit-card-numbers-for-testing)](https://developer.paypal.com/docs/classic/payflow/payflow-pro/payflow-pro-testing/#credit-card-numbers-for-testing)

Google (or perhaps a roommate's wallet) should turn up more. (If your roommate asks what you're doing, don't mention us.) If your program behaves incorrectly on some inputs, time to debug!