Basic Credit Card Processing

----------------------------

Imagine that you're writing software for a credit card provider. Implement a program that will add new credit card accounts, process charges and credits against them, and display summary information.

Requirements:

- three input commands must be handled, passed with space delimited arguments, via stdin or a file passed on the command line

- "Add" will create a new credit card for a given name, card number, and limit

- Card numbers should be validated using Luhn 10

- New cards start with a $0 balance

- "Charge" will increase the balance of the card associated with the provided name by the amount specified

- Charges that would raise the balance over the limit are ignored as if they were declined

- Charges against Luhn 10 invalid cards are ignored

- "Credit" will decrease the balance of the card associated with the provided name by the amount specified

- Credits that would drop the balance below $0 will create a negative balance

- Credits against Luhn 10 invalid cards are ignored

- when all input has been read and processed, a summary should be generated and written to stdout

- the summary should include the name of each person followed by a colon and balance

- the names should be displayed alphabetically

- display "error" instead of the balance if the credit card number does not pass Luhn 10

Input Assumptions:

- all input will be space delimited

- credit card numbers may vary in length, up to 19 characters

- credit card numbers will always be numeric

- amounts will always be prefixed with "$" and will be in whole dollars (no decimals)

Example Input:

```

Add Tom 4111111111111111 $1000

Add Lisa 5454545454545454 $3000

Add Quincy 1234567890123456 $2000

Charge Tom $500

Charge Tom $800

Charge Lisa $7

Credit Lisa $100

Credit Quincy $200

```

Example Output:

```

Lisa: $-93

Quincy: error

Tom: $500

```

Implement your solution in any programming language you wish, but keep in mind we may ask you to explain or extend your code. Please write tests and include them with your submission, along with a README that includes usage instructions, an overview of your design decisions, and why you picked the programming language you used for the solution.

\*\*\* Note: this information is confidential. It is prohibited to share, post online or otherwise publicize without Braintree's prior written consent. \*\*\*