credit

TODO

- prompt for user input
- multiply every other digit
- sum those digits
- add to remaining digits
- validate checksum
- validate company’s identifier
- validate number’s length

TODO

- [ ] prompt for user input
- [ ] multiply every other digit
- [ ] sum those digits
- [ ] add to remaining digits
- [ ] validate checksum
- [ ] validate company’s identifier
- [ ] validate number’s length

`get_long_long`

- CS50 Library function
- ensures that the user inputs an integer
  - positive integers
  - negative integers
  - `0`
  - otherwise, user is prompted to "`Retry: `".

checksum

- starting with second-to-last digit, multiply every other digit by 2
- add those products' digits together
- add the sum to the sum of the digits that weren't multiplied by 2
- if last digit is 0, number is valid

TODO

- ☑ prompt for user input
- ☐ multiply every other digit
- ☐ sum those digits
- ☐ add to remaining digits
- ☐ validate checksum
- ☐ validate company’s identifier
- ☐ validate number's length

378282246310005

378282246310005

- starting at second to last digit, multiply every other digit by 2

378282246310005

- starting at second to last digit, multiply every other digit by 2

- last digit: `cc_number % 10`
- how do you access second to last digit?

378282246310005

- starting at second to last digit, multiply every other digit by 2

```
0 * 2 = 0
0 * 2 = 0
3 * 2 = 6
4 * 2 = 8
```

```
2 * 2 = 4
2 * 2 = 4
7 * 2 = 14
```

378282246310005

- add those products' digits together

```
0 * 2 = 0
```

- 0 +

378282246310005

- add those products' digits together

```
0 * 2 = 0
0 * 2 = 0
```

- 0 + 0

378282246310005

- add those products' digits together

```
0 * 2 = 0
0 * 2 = 0
3 * 2 = 6
```

- `0 + 0 + 6`

3782822463100 05

- add those products' digits together

```
0 * 2 = 0
0 * 2 = 0
3 * 2 = 6
4 * 2 = 8
```

- 0 + 0 + 6 + 8

378282246310005

- add those products' digits together

```
0 * 2 = 0
0 * 2 = 0
3 * 2 = 6
4 * 2 = 8
```

```
2 * 2 = 4
```

- 0 + 0 + 6 + 8 + 4

378282246310005

- add those products' digits together

```
0 * 2 = 0
0 * 2 = 0
3 * 2 = 6
4 * 2 = 8
```

```
2 * 2 = 4
2 * 2 = 4
```

- 0 + 0 + 6 + 8 + 4 + 4

378282246310005

- add those products' digits together

```
0 * 2 = 0
0 * 2 = 0
3 * 2 = 6
4 * 2 = 8
```

```
2 * 2 = 4
2 * 2 = 4
7 * 2 = 14
```

- 0 + 0 + 6 + 8 + 4 + 4 + 1 + 4

378282246310005

- add those products' digits together

```
0 * 2 = 0
0 * 2 = 0
3 * 2 = 6
4 * 2 = 8
```

```
2 * 2 = 4
2 * 2 = 4
7 * 2 = 14
```

- 0 + 0 + 6 + 8 + 4 + 4 + 1 + 4 = 27

378282246310005

- 0 + 0 + 6 + 8 + 4 + 4 + 1 + 4 = 27

- add that product to the sum of the digits that were not multiplied by 2

378282246310005

- 0 + 0 + 6 + 8 + 4 + 4 + 1 + 4 = 27
- add that product to the sum of the digits that were not multiplied by 2
- 27 + 5 + 0 + 1 + 6 + 2 + 8 + 8 + 3 = 60

TODO

- ☑ prompt for user input
- ☑ multiply every other digit
- ☑ sum those digits
- ☑ add to remaining digits
- ☐ validate checksum
- ☐ validate company's identifier
- ☐ validate number's length

TODO

- ☑ prompt for user input
- ☑ multiply every other digit
- ☑ sum those digits
- ☑ add to remaining digits
- ☑ validate checksum
- ☐ validate company's identifier
- ☐ validate number's length

company identifiers

- Amex
  - 15 digits
  - start with 34, 37
- MasterCard
  - 16 digits
  - start with 51, 52, 53, 54, 55
- Visa
  - 13 and 16
  - start with 4

378282246310005

- keep track of:
  - first two digits of card
  - number’s length

TODO

- ☑ prompt for user input
- ☑ multiply every other digit
- ☑ sum those digits
- ☑ add to remaining digits
- ☑ validate checksum
- ☑ validate company's identifier
- ☑ validate number's length

this was credit