caesar

TODO

- get the key
- get the plaintext
- encipher
- print ciphertext

C

```
$ ./caesar 2
ABCDEFGHIJKL
CDEFGHIJKLMN

$ ./caesar 2
This is CS50!
Vjku ku EU50!
```

C

```
$ ./caesar 2
ABCDEFGHIJKL
CDEFGHIJKLMN

$ ./caesar 2
This is CS50!
Vjku ku EU50!
```

Python

```
$ python caesar.py 2
ABCDEFGHIJKL
CDEFGHIJKLMN

$ python caesar.py 2
This is CS50!
Vjku ku EU50!
```

TODO

- [ ] get the key
- [ ] get the plaintext
- [ ] encipher
- [ ] print ciphertext

get the key

C	Python
`argv[1]`	`sys.argv[1]`
`atoi(argv[1])`	`int(sys.argv[1])`

TODO

- ☑ get the key
- ☐ get the plaintext
- ☐ encipher
- ☐ print ciphertext

prompt user

C

- `get_string`
- `#include <cs50.h>`

Python

- `get_string`
- `import cs50`

print()

```
# print something with a newline
print("hello, world")

# print something without a newline
print("hello, world", end="")

# print something 50 times
print("#" * 50)

# print newline
print()
```

TODO

- ☑ get the key
- ☑ get the plaintext
- ☐ encipher
- ☐ print ciphertext

encipher

for each character in the plaintext string
    if alphabetic
        shift character by key, preserving case

for loop: Python

```
name = "Zamyla"

for c in name:
    print(c)
```

checking characters

C	Python
isalpha	isalpha
isupper	isupper
islower	islower

$$c_i = (p_i + k) \% 26$$

- c_i : i^{th} ciphertext letter
- p_i : i^{th} plaintext letter
- k : key
- % 26: remainder after dividing by 26

'Y' + 2 = 'A' ?

ASCII Values

('Y' + 2) % 26

= (89 + 2) % 26
= 91 % 26
= 13

A = 65

alphabetical index

Y : 24

(24 + 2) % 26
= 26 % 26
= 0

A = 0

ASCII ⇄ alphabetical?

C

- ...

Python

- `ord`
- `chr`

TODO

- ☑ get the key
- ☑ get the plaintext
- ☑ encipher
- ☑ print ciphertext

this was caesar