Table of Contents

Vigenère

tl;dr

Implement a program that encrypts messages using Vigenère’s cipher, per the below.

```
$ python vigenere.py ABC
plaintext:  HELLO
ciphertext: HFNLP
```

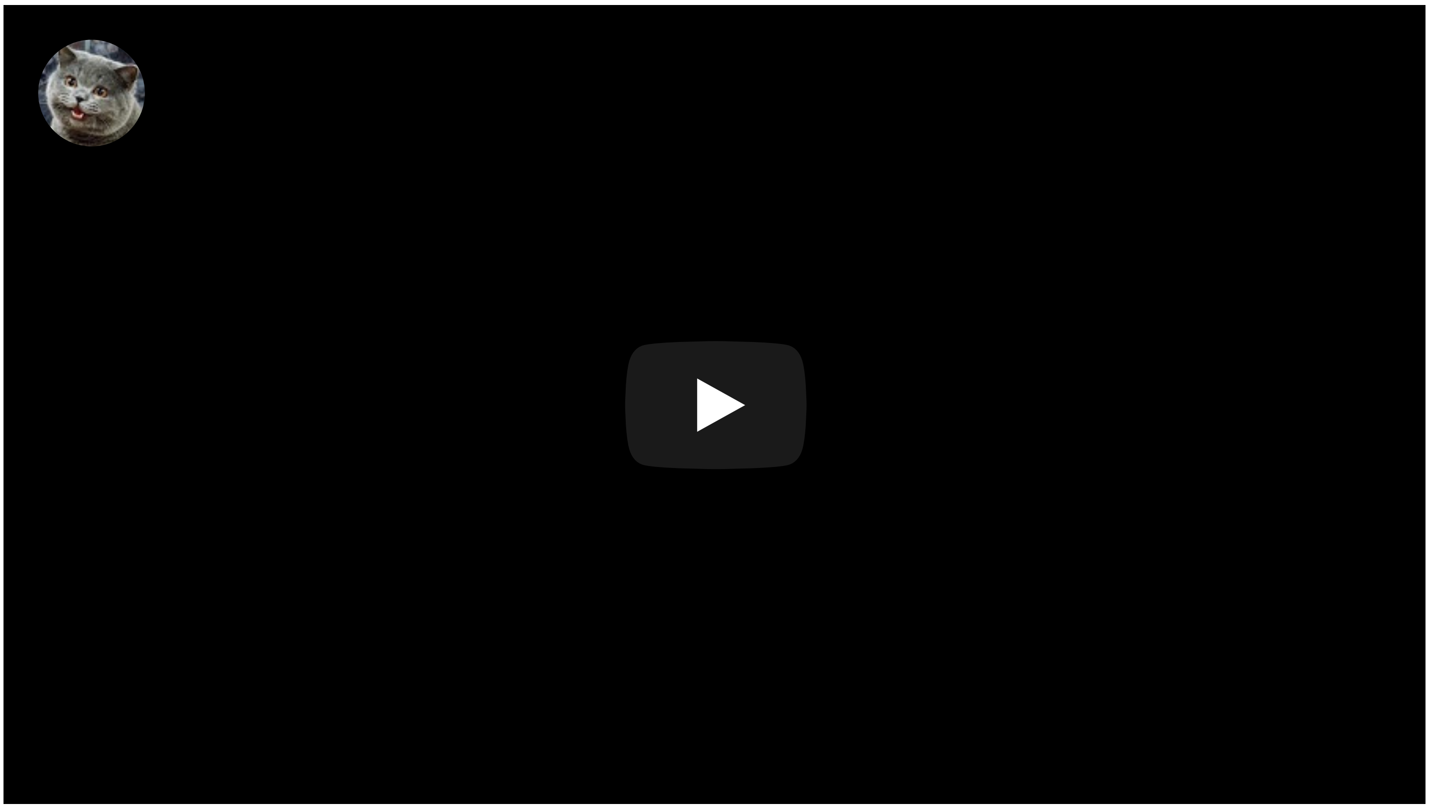
Specification

Design and implement a program that encrypts messages using Vigenère's cipher, exactly as you did in Problem Set 2 (https://lab.cs50.io/cs50/labs/2019/x/vigenere/), except that your program this time should be written (a) in Python and (b) in CS50 IDE.

- Implement your program in a file called `vigenere.py` in your `~/workspace/pset6/vigenere` directory (if it doesn't already exist, create it now!).
- Your program must accept a single command-line argument: a keyword, *k*, composed entirely of alphabetical characters.
- If your program is executed without any command-line arguments, with more than one command-line argument, or with one command-line argument that contains any non-alphabetical character, your program should print an error (of your choice) and exit (https://docs.python.org/3/library/sys.html#sys.exit) immediately with a status code of 1.
- Otherwise, your program must proceed to prompt the user for a string of plaintext, *p*, (as by a prompt for `plaintext:`) which it must then encrypt according to Vigenère's cipher with *k*, ultimately printing the result (prepended with `ciphertext:` and ending with a newline) and exiting.
- With respect to the characters in *k*, you must treat `A` and `a` as 0, `B` and `b` as 1, … , and `Z` and `z` as 25.
- Your program must only apply Vigenère's cipher to a character in *p* if that character is a letter. All other characters (numbers, symbols, spaces, punctuation marks, etc.) must be outputted unchanged. Moreover, if your code is about to apply the j^{th} character of *k* to the i^{th} character of *p*, but the latter proves to be a non-alphabetical character, you must wait to apply that j^{th} character of *k* to the next alphabetical character in *p*; you must not yet advance to the next character in *k*.

- Your program must preserve the case of each letter in *p*.

Walkthrough



Usage

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

```
$ python vigenere.py 13
Usage: python vigenere.py k
```

```
$ python vigenere.py
Usage: python vigenere.py k
```

```
$ python vigenere.py bacon and eggs
Usage: python vigenere.py k
```

```
$ python vigenere.py bacon
plaintext: Meet me at the park at eleven am
ciphertext: Negh zf av huf pcfx bt gzrwep oz
```

Testing

To help you test `vigenere`, we’ve written a program called `devigenere` for you that also takes one and only one command-line argument (a keyword) but whose job is to take ciphertext as input and produce plaintext as output. To use our program, execute

```
~cs50/pset2/devigenere k
```

at your prompt, where `k` is some keyword. Presumably you’ll want to paste your program’s output as input to our program; be sure, of course, to use the same key. Note that you do not need to implement `devigenere` yourself, only `vigenere`.

Correctness

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

Style

```
style50 vigenere.py
```

Staff's Solution

```
~cs50/2019/x/pset6/vigenere
```

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/vigenere
```

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

Hints

Not sure where to begin? As luck would have it, this program's pretty similar to `caesar` (../caesar/caesar.html)! Only this time, you need to decide which character in *k* to use as you iterate from character to character in *p*.