

day36 cipher

Due: Thursday 11/3, 6th period

Friday 11/4, 7th period

We'll do today's program in repl.

Today's task is to write a program that encodes text using a Caesar Cipher, a simple method of encryption historically credited to Julius Caesar. See the following Wikipedia link for more information:

http://en.wikipedia.org/wiki/Caesar_cipher

To make a Caesar cipher, you first choose an offset (a number between 1 and 25 inclusive), and then map all letters of the alphabet to new letters of the alphabet shifted over.

For example, if you use an offset of 3, the following happens:

```
English:  abcdefghijklmnopqrstuvwxyz
Cipher:   defghijklmnopqrstuvwxyzabc
```

Look at the top line for the "c". Notice that below it in the Cipher line the letter right there is "f". This means in the encrypted text every letter that was a c will be replaced with an f. The English word "cat" becomes "fdw". The phrase "I wish I knew more palindromic cyclops primes" becomes "l zlvk l nqhz pruh sdolqgurplf fbforsv sulphv ". Not at all obvious, eh?!

This kind of cipher is susceptible to hacking using a frequency analysis, or just brute force. If you had enough text to work with you could run it through a frequency analysis program and then figure out which letter was "e", the usual most common letter in English text, figure out the offset to e, and you've most likely broken the code. Obviously this type of cipher isn't used where security matters anymore.

However, it's a great puzzle. So, your task is to create a program that does the following:

- uses a try/except structure to ask for an offset
- the program then verifies that the integer is one that will work with how you have set up the offset procedure (Make sure the user enters a number between 1 and 25 inclusive. 0 and 26 should not be allowed (they would mean that you weren't encrypting anything!))
- asks the user for text to encode, make the text lowercase
- encrypts the text and prints it

Then on the same run:

- asks the user for text to decode, make what they enter lowercase
- prints the decoded text (using the same offset as the encoding, don't ask for an offset here)
- if you enter nothing on any encode/decode line, the program quits

I give you the code you can use to encrypt and decrypt in a repl attached to this assignment on the Google classroom. You just have to do everything around it, making the program look good. Tips and sample output on next page.

Tips:

- You only use a try/except structure for making sure the user is entering an integer. Inside the try/except structure use an if statement to make sure that the offset is between 1 and 25 inclusive. Use a break if their number is valid, give an error if the number is not.
- Make the user-entered text all lowercase before you try to encrypt or decrypt.
- Then print the encrypted text, letter by letter, using the code I gave you in the repl attached to this assignment in the Google Classroom.
- To decrypt, use the decryption code from the repl mentioned above.

In both code snippets the text to be encrypted or decrypted is in a variable named **text** and the offset you are using is in a variable called **offset**.

For 20% of your grade, copy and paste a sample run of you using your program to encrypt and then decrypt and then enter nothing to quit in triple quotes after your code.

See sample output here:

```
Enter cipher offset: 5
Enter a message to encode: palindromic cyclops primes
Cipher text: ufqnsiwtrnh hdhqtux uwnrjx

Enter a message to decode: hdhqtux
English text: cyclops
```

Optional extra credit:

Ask the user to enter encrypted text and decrypt it using all 25 possible decryptions (print them all). One will clearly be the English phrase. Do this in a copy of your program named day36 extra.py and email me that link so I know you attempted the extra credit.