day36 cipher
Due: Wednesday 11/15/23

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 (an integer from 1 to 25 inclusive).
- 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)
- ask the user if they want to run again (yes/no)

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.

Open this repl and fork it, then get started. See steps on the next page.

Steps:

1. Above the block I gave you called "encryption" do a try/except to get the offset. Make sure the entry is from 1 to 25 inclusive.
2. Do an input to get the text, do text = text.lower() on the next line
3. Run the program now, and it should encrypt whatever you enter. Make sure this is working before you move on.
4. After my encryption block, but before the decryption block, add another input to ask the user for text to decrypt. Make what they enter lowercase again.
5. Uncomment the second block of code (remove the three """ that are before and after it).
6. Run your code, you should now encrypt and decrypt just fine.
7. Select your whole program, indent it, put a while True above it.
8. Add a "run again" prompt after the decryption block of code to ask the user if they want to run again. If they do not, break out of the loop.

Both code blocks of code that I give you use a variable named **text** for the text to process and a variable named **offset** for which offset to use. Please don't change these parts.

Please run your program, encoding and decoding at least five words of text, then copy and put this into triple quotes after your program.

See sample output here:

```
Welcome to the Encryptor 2000!

Enter offset:
Invalid, try again.

Enter offset: 30
Invalid offset, must be between 1 and 25 inclusive.

Enter offset: 4

Enter something to encrypt: hello there
lipps xlivi

Enter text to decrypt: lipps xlivi
hello there

Run again? (y/n) n
Goodbye
```

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 and email me that link so I know you completed the extra credit.