CS150 - Lab Prep 3
Due: Friday Sept. 29, at the beginning of class

For the lab on Friday, we're going to be playing with basic cryptography. In cryptography, the two key functions are encryption, where we take a message and encrypt it, and decryption, where we take an encrypted message and give back the original message. To get you ready for lab on Friday, we're going to be going through some key concepts here and play with some of the encryption schemes by hand.

1 Caesar's scheme

One of the simpler and more famous encryption schemes is credited to Julius Ceasar. The encryption scheme is called a "substitution cipher" in that each character is substituted for a different character in the message. The Caesar approach was to substitute a given letter with the letter that was some fixed number of letters up in the alphabet. For example, if you chose 2 as your fixed number up:

```
alphabet: a b c d e f g h i j k l m n o p q r s t u v w x  y  z ‘ ’
key:      c d e f g h i j k l m n o p q r s t u v w x y z ‘ ’ a  b
```

or if we chose 4

```
original: a b c d e f g h i j k l m n o p q r s t u v  w  x y z ‘ ’
key:      e f g h i j k l m n o p q r s t u v w x y z ‘ ’ a b c  d
```

Notice that we include the space as a character (written as ‘ ’) and that we wrap around when we're done.

When encrypting, you simply replace each character in your message with the encrypted character (i.e. the character the fixed number up in the alphabet) and to decrypt it is the reverse process.

Do the following based on Caesar's method and write the answer on a piece of paper to be handed in:

1. Encrypt `'this is a test'` with a spacing of 2 (to encrypt, substitute letters from "original" to letters in "encrypt")

2. Decrypt `'kbnqxgbeubencuu'` with a spacing of 2 (to decrypt, substitute the letters from "encrypt" to letters in "original")

3. Decrypt `'stenewjfqqceit'` with a spacing of 5

2 General substitution cyphers

As mentioned, Caesar's scheme is a specific example of a substitution cipher. In general, for a substitution cipher any letter can be substituted for any other letter. For example:

```
alphabet: a b c d e f g h i j k l m n o p q r s t u v w x  y  z ' '
key:      h v i e k s y r b d a j q w n c x m g u f l t p ' ' o  z
```

4. Decrypt the following message with this substitution and add it to your answers from above: `'urkzak zbgzvhwhwhg'`

The key idea when programming something like this is to note that if we store both the original alphabet and the encryption alphabet as strings, we can find the `index` in the original alphabet string and use that to lookup the correct character in the encryption alphabet string during encryption and vice versa during decryption. Specifically:

```
alphabet: a b c d e f g h i j k  l  m  n  o  p  q  r  s  t  u  v  w  x  y  z  ' '
          0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
key:      h v i e k s y r b d a  j  q  w  n  c  x  m  g  u  f  l  t  p ' ' o  z
```

Answer the following questions:

5. If we have a variable called ALPHABET initialized as follows:

   ```
   ALPHABET = "abcdefghijklmnopqrstuvwxyz "
   ```

   (again notice the space at the end) and we have a letter stored in a variable called `letter`, write a Python expression that will **find** (*hint*) the index in `ALPHABET` where that letter occurs

6. Finally, if we stored that index in a variable called `index` and the encrypted letters corresponding to `ALPHABET` are stored in a variable called `key`, e.g.

   ```
   key = "hvieksyrbdajqwncxmgufltp oz"
   ```

   write a Python expression that will find the corresponding encrypted letter to the `letter`. Hint: you'll use `index` and it should be something quite simple.

Something to think about: (You don't actually need to answer this) Why aren't substitution cyphers very good? Put another way, why are they relatively easy to break?