

Name: .....

# Lab TEST Week 6 - Strings

Skills Needed to complete this Lab

- Index, slice strings
- Iterate over strings
- use string methods
- Loops
- Variables

## Caesar Cipher

The Caesar Cipher is an encryption/decryption method that shifts the alphabet. For instance, if you have a cipher that shifts by 1, A would become B, B would become C, Z would wrap and become A. To decrypt that cipher you simply shift by -1. It is a very simple cipher that is easily broken. [http://en.wikipedia.org/wiki/Caesar\\_cipher](http://en.wikipedia.org/wiki/Caesar_cipher)

We need you to write a utility that Encodes and Decodes a Cipher.

### The Rules.

- The main program loop should ask the user if they want to.
  - 1. Encode a string
  - 2. Decode a string
  - Q. Quit
- If the user chooses anything other than 1, 2, or Q, they get an error and it tries again.
- If they want to encrypt a word or sentence, ask for the phrase and then ask for the # to shift by, once you encrypt the string, output the result to the user.
- If they want to decrypt a word or sentence, ask for the string and the # of to shift by. Then display the decrypted string to the user.
- Go back to the main menu until they select quit.

### Program Requirements :

#### Development Notes :

- The string module has some useful constants. Strings of only upper case characters, lower case characters, digits.
 

```
>>> import string
>>> string.ascii_uppercase
'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
>>>
```
- Strings have two methods that could be useful. `.lower()` and `.upper()`. These methods return a string that is lowercased or upper cased respectively.
  - `abcLower = "ABC".lower()` # abcLower is "abc"
  - `abcUpper = "abc".upper()` # abcUpper is "ABC"
- The membership operator `in` can be useful as well, but is not required.
- Strings have a method `.find()` that will return a -1 if the substring or character provided is not in the string.
 

```
>>> "ABC".find("Z")
-1
```

```
>>> "ABC".find("A")
0
>>>
```

- String indexing will probably be useful here as well.
- You can use the Python chr() and ord() function get the numeric value of each character and change it back to a character.

```
>>> ch = "A"
>>> ch_value = ord(ch)
>>> ch
'A'
>>> ch_value
65
>>> ch_new = chr(ch_value + 2)
>>> ch_new
'C'
>>>
```

- **You must use functional decomposition to get full credit for this program.**

### Example

MAIN MENU:

1) Encode a string

2) Decode a string

Q) Quit

Enter your selection ==> 1

Enter (brief) text to encrypt: hello world

Enter the number to shift letters by: 3

Encrypted: KHOOR ZRUOG

MAIN MENU:

1) Encode a string

2) Decode a string

Q) Quit

Enter your selection ==> 2

Enter (brief) text to decrypt: koor zruog

Enter the number to shift letters by: 3

Decrypted: HELLO WORLD

MAIN MENU:

1) Encode a string

2) Decode a string

Q) Quit

Enter your selection ==> 1

Enter (brief) text to encrypt: eleven leaping lavender antelope

Enter the number to shift letters by: 3

Encrypted: HOHYHQ OHDSLQJ ODYHQGHU DQWHORSH

MAIN MENU:

1) Encode a string

2) Decode a string

Q) Quit

Enter your selection ==> Q

>>>

### Grading and Turning In

Turn in your program before the end of the lab. Only upload the solution.py file, as other files will be ignored. Include the screenshots of your output. Upload the file on GitHub and submit the link on Canvas.