Homework #2 Data Structures — Due: Feb. 10 (Saturday at 11:59 PM)

Objects: Practice designing a robust program and "reviewing" Python file usage. (NOTE: be sure to review the **example programs found** at: www.cs.uni.edu/~fienup/cs1520s18/homework/example_programs_hw2.zip)

File Encryption/Decryption Program: You are to **design** and write a program that allows a user to interactively select between encrypting or decrypting files:

- encrypting a user-specified text-file (`.txt` extension) using a Vigenere-cipher encryption scheme to generate a new encoded file with the same name, except with a `.zzz` extension. The Vigenere-cipher is a form of substitution cipher based on a user-specified keyword. (see below example)
- decrypting a user-specified Vigenere-cipher file (`.zzz` extension) using a user-specified keyword to generate a new text-file with the same name, except with a `.txt` extension.

Vigenere-cipher encryption example: For the keyword "bobwhite" and the subset of 32 characters shown in the top row below (i.e., letters, ' '/space, '\n'/new-line, , /comma, ./period, ?/question-mark, and !/exclamation-point), the Vigenere-cipher table would be:

Letter / **row#**	**a** **0**	**b** **1**	**c** **2**	**d** **3**	**e** **4**	**f** **5**	**g** **6**	**h** **7**	**i** **8**	**j** **9**	**k** **10**	**l** **11**	**m** **12**	**n** **13**	**o** **14**	**p** **15**	**q** **16**	**r** **17**	**s** **18**	**t** **19**	**u** **20**	**v** **21**	**w** **22**	**x** **23**	**y** **24**	**z** **25**	**' '** **26**	**'\n'** **27**	**,** **28**	**.** **29**	**?** **30**	**!** **31**	**Additional Positions**		
0	**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	' '	'\n'	,	.	?	!	a	8	16	24
1	**o**	p	q	r	s	t	u	v	w	x	y	z	' '	'\n'	,	.	?	!	a	b	c	d	e	f	g	h	i	j	k	l	m	n	9	17	25
2	**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	' '	'\n'	,	.	?	!	a	10	18	26
3	**w**	x	y	z	' '	'\n'	,	.	?	!	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	11	19	27
4	**h**	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	' '	'\n'	,	.	?	!	a	b	c	d	e	f	g	12	20	28
5	**i**	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	' '	'\n'	,	.	?	!	a	b	c	d	e	f	g	h	13	21	29
6	**t**	u	v	w	x	y	z	' '	'\n'	,	.	?	!	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	14	22	30
7	**e**	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	' '	'\n'	,	.	?	!	a	b	c	d	15	23	31

To encrypt a message with the keyword "bobwhite", the first character of message (at position 0) uses row 0 ('a' maps to a 'b', 'b' maps to a 'c', etc.). The second character of the message (at position 1) uses row 1 ('a' maps to a 'o', 'b' maps to a 'p', etc.). If the message is longer than the keyword, then every 8th character (i.e., the length of "bobwhite") uses the same row (e.g., characters of the message at positions 1, 9, 17, 25, 33, etc. all use row 1). All upper-case letters can be converted to lower-case before encryption. Any characters not in the top row can be completely ignored (i.e., discarded) without incrementing the position in the message.

The text-file `message.txt` (with new-line characters shown as'\n') would be encrypted to `message.zzz` and then decrypted back to `decryptedMessage.txt` as shown below:

`message.txt`

```
Meet by the Union\n
at noon today!\n
-Sam\n
```

`message.zzz`

```
nsfjbjl?uvfq\n
v\n
sojbjbvbsoiuekild,abcc
```

`decryptedMessage.txt`

```
meet by the union\n
at noon today!\n
sam\n
```

Hints:

- You don't actually need to create the Vigenere-cipher table shown above. Instead just use the position of each character from the overall sequence, and the position of each keyword character in the overall sequence

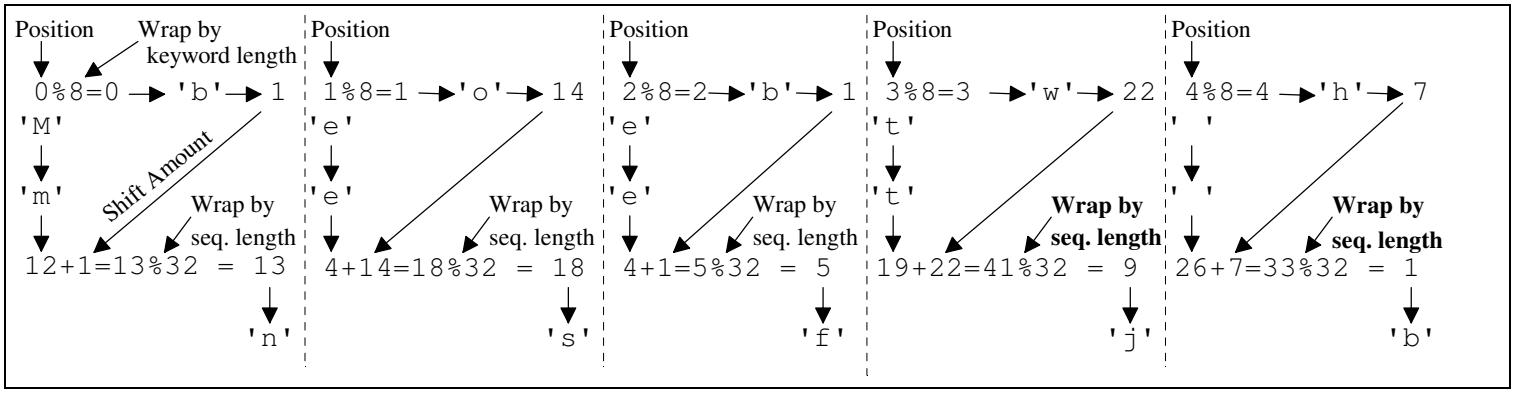
Seq. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Letter	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**	**n**	**o**	**p**	**q**	**r**	**s**	**t**	**u**	**v**	**w**	**x**	**y**	**z**	**' '**	**'\n'**	**,**	**.**	**?**	**!**

Position	0	1	2	3	4	5	6	7
Letter	**b**	**o**	**b**	**w**	**h**	**i**	**t**	**e**
Letter seq. #	1	14	1	22	7	8	19	4

- You might also want to create a dictionary to enable quick lookup of characters position in the sequence

  (e.g. 'm' → 12, '?' → 30, etc.)

For example, encryption of first part of the message ('Meet '):



Your program also needs to: (See www.cs.uni.edu/~fienup/cs1520s18/homeworks/example_programs_hw2.zip)

- validate a correct selection whether to encrypt, decrypt, or exit.
- validate that the user specified file to encrypt/decrypt exists and force the user to reenter until they specify an existing file. Feel free to make the program more user-friendly by listing all files of the appropriate type `.txt` or `.zzz` extensions.
- check if the user-specified file name for the encrypting (or decrypting) exists before opening it for write. If it already exists, ask the user if they are okay with it being wiped out. If they are not, ask them to pick a different file name to receive the encrypted (or decrypted) text.

Your program's interaction should looks something like: (Student input shown in **bold**.)

```
Welcome to the Vigenere-cipher Encryption/Decryption Program

Would you like to (e)ncrypt a file, (d)ecrypt a file, or e(x)it (enter e, d, or x)? encrypt
Sorry, that's an invalid choice.  Please enter only e, d, or x: e

What keyword would you like to use for encryption? bobwhite

Enter the text-file name to encrypt: mesage.txt
Sorry the file 'mesage.txt' does NOT exist -- please try again!
Enter the text-file name to encrypt: messaeg.txt
Sorry the file 'messaeg.txt' does NOT exist -- please try again!
Enter the text-file name to encrypt: message.txt

The file 'message.txt' was successfully encrypted using keyword 'bobwhite' to the file 'message.zzz'

Would you like to (e)ncrypt a file, (d)ecrypt a file, or e(x)it (enter e, d, or x)? d

What keyword would you like to use for decryption? bobwhite

Enter the text-file name to decrypt: message.zzz

WARNING:  The file 'message.txt' already exists!
Is it okay to wipe it out (y/n)?  n
Enter the file name that should be used (.txt extension will automatically be added): decryptedMessage

The file 'message.zzz' was successfully decrypted using keyword 'bobwhite' to the file 'decryptedMessage.txt'

Would you like (e)ncrypt a file, (d)ecrypt a file, or e(x)it (enter e, d, or x)? x
```

When you write your program, be sure you:

- save your program in a file called `VigenereCipher.py`
- make your program robust by validating the correctness of user input: verify that user-entered file names exists (`import os.path`) before opening them, and check for correct menu-option, etc.
- think about the functional-decomposition (top-down) design **before you start to write code**! You will need to turn in a design document (see lab 1 for an example: diagram and sentence about each function).
- don't use global variables, except for global constants with good style (ALL_CAPS_AND_UNDERSCORES). Variables should be passed as parameters into functions and returned from functions
- use meaningful variable names with good style (i.e., useCamelCase)
- use comments (""" Multi-line Comment """) at the start of the program **and** immediately after each function `def`inition describing what they do (see lab1 `diceOutcomes.py` program)
- use a main function (see lab1 `diceOutcomes.py` program) located at the top of program **with a call to it at the bottom to start execution**

Submit a single zipped file called hw2.zip containing the following:

- **`VigenereCipher.py`** (your Python program)
- **design.doc** (or design.pdf, or design.txt, or design.rtf or design.jpg) a document describing the design of your program including a functional-decomposition diagram (**can be hand-drawn**) with text describing each function (see lab1 description)

The steps for the homework submission system are:

1. Design, write, debug, and test your program in the hw2 folder. When you are ready to submit your homework, zip the whole folder by right-clicking on it and selecting `Send to | Compressed (zipped) folder`. This will create a new file called `hw2.zip` which you will submit electronically. (This assumes Windows OS....)
2. Log on to the submission system at: **https://www.cs.uni.edu/~schafer/submit/which_course.cgi**
3. Select the course and section number of "CS 1520, Data Structures, Fienup". Click the "Continue".
4. Select the homework that you wish to submit: "HW 2: VigenereCipher". Click the "Continue" button.
5. Specify how many extra files you want to submit. Just leave it at 0. Click the "Continue" button.
6. Upload your program by Browsing and selecting your `hw2.zip` file. Click the "Continue" button.
7. The next page reports on the status of the upload(s). You can always continue to upload a better version of the program until the deadline. The newer file will replace an older file of the same name.

(If you miss the deadline, you'll need to submit it as above, but select "Late Homeworks" in step 4 above.)