

### Lab 3: Shift Cipher

This is an INDIVIDUAL assignment. Due date is as indicated on BeachBoard. Follow ALL instructions otherwise you will lose points. In this lab, you will be implementing two functions. This will require the use

#### Background:

Shift Cipher: Encrypts messages by “shifting” the letter to the right a certain number of characters.

The most renowned shift cipher is Caesar Cipher which shifts everything by 13 letters

	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
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
Shifted:	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
	13	14	15	16	17	18	19	20	21	22	23	24	25	0	1	2	3	4	5	6	7	8	9	10	11	12

To encode text, add 13 to the character's value. Then, do mod 26 to accommodate overflow (overflow means that the value is not within our 0-25 character encoding..

For example, given the text “CECS”

1.  $C(2) + 13 = 15 \text{ mod } 26 = 15$  (P)
2.  $E(4) + 13 = 17 \text{ mod } 26 = 17$  (R)
3.  $S(18) + 13 = 31 \text{ mod } 26 = 5$  (F)

Thus CECS -> PRPF

To decrypt text, subtract 13 from the encrypted character. Then, do mod 26 to accommodate overflow.

For example, given the text “RHAVPR”

1.  $R(17) - 13 = 4 \text{ mod } 26 = 4$  (E)
2.  $H(7) - 13 = -6 \text{ mod } 26 = 20$  (U)
3.  $A(0) - 13 = 13 \text{ mod } 26 = 13$  (N)
4.  $V(21) - 13 = 8 \text{ mod } 26 = 8$  (I)
5.  $P(15) - 13 = 2 \text{ mod } 26 = 2$  (C)

Thus RHAVPR -> EUNICE

There are also general shift ciphers. These encrypt any letter by shifting the character right by  $n$  characters instead of strictly 13 like in the Caesar Cipher.

To encrypt text, add  $n$  to character's value. Then, do mod 26 to accommodate overflow

For example, given text “CECS” and  $n=15$

1.  $C(2) + 15 = 17 \text{ mod } 26 = 17$  (R)
2.  $E(4) + 15 = 19 \text{ mod } 26 = 19$  (T)
3.  $S(18) + 15 = 33 \text{ mod } 26 = 7$  (H)

Thus CECS -> RTRH

To decrypt text, subtract  $n$  from character's encoding. Then, do mod 26 to accommodate overflow

For example, given text: “RTRH” and  $n=15$  (decrypting what we just encrypted)

1.  $R(17) - 15 = 2 \text{ mod } 26 = 2$  (C)
2.  $T(19) - 15 = 4 \text{ mod } 26 = 4$  (E)
3.  $H(7) - 15 = -8 \text{ mod } 26 = 18$  (S)

Thus RTRH -> CECS

### Instructions:

1. Take a close look at the `shift.py` file. There are two empty functions:  
`shift_cipher_encode(string,n)` and `shift_cipher_decode(string,n)`.  
Read through both of their descriptions carefully. Remember, you will lose points if you do not follow the instructions. We are using a grading script
2. Your job is to implement both of these functions so that it passes any test case.  
There are some sample test cases provided for you, but these are not the only cases that we will test. There will be 30 test cases in addition to the ones that you see.
3. **Do NOT use a dictionary or list to encode the characters! Using a dictionary or list will result in a zero!**
4. **If there are characters other than letters, do NOT try to encrypt these. These should be left alone!**
5. After completing these functions, comment out the test cases (or delete them) or else the grading script will pick it up and mark your program as incorrect.
6. Convert your `shift.py` file to a `.txt`. Submit your `shift.py` file and your `.txt` file on BeachBoard. Do NOT submit it in compressed folder.
7. Do not email us your code asking us to verify it. We will answer general questions, but we will not debug your code over email.

Some helpful functions (click on function to go to reference link). Please note that you do not have to use any of these. They may be useful though:

<a href="#"><code>string name.isalpha()</code></a>	Returns True if all characters in string are letters of the alphabet. Otherwise, returns False
<a href="#"><code>ord(character)</code></a>	Gets the ascii value of <code>character</code>
<a href="#"><code>chr(num)</code></a>	Gets the character of ascii value <code>num</code>
<a href="#"><code>string name.isupper()</code></a>	Returns True if all characters in string are upper case. Otherwise, returns False
<a href="#"><code>string name.islower()</code></a>	Returns True if all characters in string are lower case. Otherwise, returns False
<a href="#"><code>string name.upper()</code></a>	Converts the string to all upper case and returns it
<a href="#"><code>string name.lower()</code></a>	Converts the string to all lower case and returns it

Ascii table for your reference

Ascii encoding	ascii Character		Ascii encoding	ascii Character
A	65		a	97
B	66		b	98
C	67		c	99
D	68		d	100
E	69		e	101
F	70		f	102
G	71		g	103
H	72		h	104
I	73		i	105
J	74		j	106
K	75		k	107
L	76		l	108
M	77		m	109
N	78		n	110
O	79		o	111
P	80		p	112
Q	81		q	113
R	82		r	114
S	83		s	115
T	84		t	116
U	85		u	117
V	86		v	118
W	87		w	119
X	88		x	120
Y	89		y	121
Z	90		z	122

Grading rubric

Points	Requirement
5	Correct submission (2 files, not in any folder), did not use a dictionary-all or nothing
5	Passes the test cases listed in <code>shift.py</code> and followed instructions by deleting/commenting the test cases in the file (all or nothing)
15	Passes the remaining 15test cases for <code>shift_cipher_encode</code>
15	Passes the remaining 15test cases for <code>shift_cipher_decode</code>