CPSC 1301, Computer Science I Programming Exercise

Final Programming Exercise, Encryption.py

1 Introduction

This activity consists of six programming exercises. The following exercises are open book and open note. You are free to use any written documentation you wish. However, these are individual exercises, and you cannot consult with each other in writing your programs. Name your program `Encryption_lastname.py`.

This programming exercise is a substantial assignment. The assignment is to write a program with will encrypt and decrypt messages. It has three parts, and each part has two sections, one to encrypt a message, and the second to decrypt the same message. Even though this is more difficult than your previous assignments, you have all you need to complete this successfully. Just be prepared to think very hard. The code is more difficult than in the previous exercises, but the thinking part is much harder.

Each letter of the alphabet has a numbered place, from 1 to 26. A = 1, B = 2, C = 3, and so on to Z = 26. Messages are encrypted by substituting each letter in the plain text message for another letter, which is determined by a key. Encrypted messages are decrypted by substituting each letter in the encoded message with the plain text key, again depending on the key. This is explained more fully below.

In the English speaking world, alphabetical letters are encoded by ACSII. You can look up the ASCII table to get the numerical code for each letter. For example, an uppercase "A" has the ASCII code 65, while the lower case "a" has the ASCII code 97. When letters are read by the computer, they are internally converted to the ASCII code. Python allows you to cast an alphabetical character to an integer, and cast an integer to a character provided that the integer is convertable to a character. In practice, this means that you are restricted to integers between 65 and 90 for upper case characters, and 97 to 122 for lower case characters.

Finally, users will enter invalid characters in their plain text message, such as blank spaces, numerical characters (we will not use numbers in this program), punctuation marks, etc. You will have to delete all these non-alphabetical characters. Part of the challenge is to "clean" the text the user enters by deleting all non-alphabetical characters (including digits, spaces, and punctuation) and convert all lower case characters to upper case characters.

2 Simple cipher

A very simple cipher entails the movement of each letter the number of places specified by a single letter key. For example, if the key were "C", each letter would be substituted for the third letter down from the place: A becomes D, B becomes E, C becomes F, W becomes Z, X becomes A, W becomes B, and Z becomes C. Notice that the letters at the end of the alphabet wrap around to the beginning. If the message was "We attack at dawn.", and the key was "C", the encoding would be done like this:

Plain	W	E	A	T	T	A	C	K	A	T	D	A	W	N
Key	C	C	C	C	C	C	C	C	C	C	C	C	C	C
Encoded	Z	H	D	W	W	D	F	N	D	W	G	D	Z	Q

Using numbers, the result would be:

ASCII	87	69	65	84	84	65	67	75	65	84	68	65	87	78
Key	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Code	90	72	68	87	87	68	70	78	68	87	71	68	90	81

The first line represents the ASCII values of the upper case alphabetical characters. The second line is the key, “C” using the nominal numerical value of “3”. The third line is the coded message. Obviously, you will have to convert back and forth between the ASCII values and the nominal numerical value, that is, converting 65 to 1, 66 to 2, 67 to 3, and so forth, wrapping around to the beginning as necessary.

Decrypting is the opposite, taking the third line, adding the key, and displaying the plain text. If you can write the first part, you can write the second part, if you think hard enough about it.

3 Less simple cipher

This is the same as the first part, but instead of using just one charcter as a key, you will use a word. In this example, the key is CAT. If the key is shorter than the message, you repeat the key as often as necessary. If you wrote the first part, you can write the second part, but it’s more difficult.

Plain	W	E	A	T	T	A	C	K	A	T	D	A	W	N
Key	C	A	T	C	A	T	C	A	T	C	A	T	C	A
Encoded	Z	F	U	W	U	U	F	L	U	W	E	U	Z	O

Here it is with numerical values:

ASCII	87	69	65	84	84	65	67	75	65	84	68	65	87	78
Key	3	1	20	3	1	20	3	1	20	3	1	20	3	1
Code	90	70	85	87	85	85	70	76	85	87	69	85	90	79

4 Still less simple cipher

This part has a key that’s a word, but the word isn’t repeated. Instead, the plain text becomes part of the key. Look at the example below. Notice that the second line, the encryptiion key, is “CATWEATTACKATD.” This technique makes the encrypted message much harder to crack.

Plain	W	E	A	T	T	A	C	K	A	T	D	A	W	N
Key	C	A	T	W	E	A	T	T	A	C	K	A	T	D
Encoded	Z	F	U	Q	Y	B	W	E	B	W	O	B	Q	R

Here it is with numerical values:

ASCII	87	69	65	84	84	65	67	75	65	84	68	65	87	78
Key	3	1	20	23	5	1	20	20	1	3	11	1	20	4
Code	90	70	85	81	89	66	87	69	66	87	79	66	81	82

Notice that in the fourth column, the plain text letter is “T” and the key is “W”. 84 (representing T in ASCII) and 23 (representing W in ASCII) resulting in 107 (84 + 23), which does not give the correct value. In order to wrap this character around, it’s necessary to subtract 26 from 107, which gives 81, the ASCII value of “Q”.

5 Exercise requirements

Part 1, Simple single character substitution cipher Write a function that will accept a plain text string to be encrypted. You will need to “clean” the input from the user to convert it to upper case and to delete everything that is not an alphabetical character. Write a function that accepts a single alphabetical character to use as a key. The encryption section will product a simple substitution cypher. The decryption will take the encoded text, the key, and decode the message.

Part 2, Simple multiplo character substitution cipher Write a function that will accept a plain text string to be encrypted. You will need to "clean" the input from the user to convert it to upper case and to delete everything that is not an alphabetical character. Write a function that accepts a multiple character string to use as a key, which will will also have to clean. The encryption section will product a simple substitution cypher using the string as a key. The decryption will take the encoded text, the key, and decode the message.

Part 3, Complex multiple character substitution cipher Write a function that will accept a plain text string to be encrypted. You will need to "clean" the input from the user to convert it to upper case and to delete everything that is not an alphabetical character. Write a function that accepts a multiple character string to use as a key, which will will also have to clean. The encryption section will product a simple substitution cypher using the string as a key as the start of the encryption, then will use the plain text message to encrypt the remainder of the plain text. The decryption will take the encoded text, the key, and decode the message.

Important instructions Do not, repeat, **DO NOT**, use any built-in functions, such as the `upper()` method. There are multiple ways to do this, including using regular expressions (which we have not covered), or converting each character to its `int` equivalent integer and manipulating the integer. Do as much as possible by hand, the hard way. You can use the normal Python iterators and list comprehensions.

6 Starter template

No starter code — you are on your own. Do it like you have been taught. It's just like all the other exercises.

7 Sample output

```
Encryption and decryption using a single key
Enter the text to encrypt:  We attack at dawn!
Enter a single character to use as a key:  c
The encrypted string using a single key is ZHDWWDFNDWGDZQ
------------------------------------
Enter the text to decrypt:  ZHDWWDFNDWGDZQ
Enter a single character to use as a key:  c
The decrypted string using a single key is WEATTACKATDAWN
------------------------------------
Encryption and decryption using a multi-character key
The text to encrypt is: 'We attack at dawn!'
Enter a character string to use as a multi-key:  cat
The encrypted string using a multi-character key is ZFUWUUFLUWEUZO
------------------------------------
Enter the text to decrypt:  ZFUWUUFLUWEUZO
Enter a character string to use as a multi-key:  cat
The decrypted string using a multi key is WEATTACKATDAWN
------------------------------------
Encryption and decryption using a continuous key
The text to encrypt is: 'We attack at dawn!'
Enter a character string to use to start a continuous key:  cat
The encrypted string using a continuous key is ZFUQYBWEBWOBQR
------------------------------------
```

```
Enter the text to decrypt:  ZFUQYBWEBWOBQR
Enter a character string to begin a continuous key:  cat
The decrypted string using a continuous key is WEATTACKATDAWN
```