

Cryptography

10/21/2022

Late **85/100 Points**

Attempt 2



Review Feedback
11/17/2022

Attempt 2 Score:
85/100



View Feedback

Anonymous Grading: **no**

Unlimited Attempts Allowed
12/9/2022

Details

Preparation

Online students please watch to string manipulation

Objectives

Create a program that assists in a basic form of cryptography, a substitution cypher. Write a program that will accept a phrase and convert it into code by substituting letters according to a key.

This program will also be an example of functions. You will be given a main function body. This function refers to a number of other functions which you will have to create. It's part of your job to figure out how these functions should be created and what their parameters and output should be.

The program is based on a standard text-based menu. You'll need to create methods (function) to display the menu, get input from it, and handle the details. Your program should encrypt and decrypt messages.

Sample Run

Here's a sample run of the basic program in action.

```
SECRET DECODER MENU

0) Quit
1) Encode
2) Decode

What is your choice: 1
text to be encoded: Python Rocks!
AQULEWKEMNJ

SECRET DECODER MENU

0) Quit
1) Encode
2) Decode
```

```

☐
What is your choice: 2
code to be decyphered: AQULEWKEMNJ
PYTHONROCKS

SECRET DECODER MENU

0) Quit
1) Encode
2) Decode

☐
What is your choice: 0
Thanks for doing secret spy stuff with me.

```

Starter Code

You may copy and paste the following code into your editor to get started. You may also type the code by hand, but your main method must be identical to the one posted here:

```

""" crypto.py
    Implements a simple substitution cypher
"""

alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
key = "XPMGTDHLYONZBWEARKJUFSCIQV"

def main():
    keepGoing = True
    while keepGoing:
        response = menu()
        if response == "1":
            plain = input("text to be encoded: ")
            print(encode(plain))
        elif response == "2":
            coded = input("code to be decyphered: ")
            print (decode(coded))
        elif response == "0":
            print ("Thanks for doing secret spy stuff with me.")
            keepGoing = False
        else:
            print ("I don't know what you want to do...")

#my starter code.

if __name__ == "__main__":

    main()

```

Notes

Please keep the following ideas in mind:

This is a substitution cypher

Each letter is replaced by another letter with no repetition

Ignore spaces and punctuation

You *must* account for spaces (I ignored them, but there are other legitimate responses.)

The best program will also account for punctuation and other symbols. (Ignoring them is fine.) Your program can ignore these special characters when they occur, but it should not crash if it encounters a strange character.

Manage Capitalization

In most cryptographic applications, all letters are converted to uppercase. Please follow this convention or deal with potential case conversion problems in another way.

Use Functions

You may not make significant changes to the main function. Determine which functions are necessary and create them. (The blackbelt project does allow for one slight modification, noted below.) *Do not turn in a program with only a main() function!!* Write the code necessary for Python to run the main function as expected

Pass data between functions

Your functions may require some input and return some output. Part of the task is to determine how the methods are to be created.

Using a key

There's several workable answers, but I used a string with all the alphabet characters and a second string with a "scrambled" version of the alphabet.

Useful String methods

This program is all about string manipulation, so you may want to use `help("str")` in your python shell to look up some useful techniques. Here are a few things you might need to know how to do:

- determine the length of a string
- convert a string to upper case
- find the position of a character in a string
- Determine the character at a certain position in a string

Don't send spy messages with this

The encryption used here is laughably weak. This is an interesting exercise in cryptography, but it's nowhere near useable as an encryption tool. Don't encode your secret recipe with this thing and expect it to remain secret.

Test your program

The best way to test your program is to encrypt a phrase, and then decrypt the encoded phrase to see if you got the original (sans spaces and other punctuation.)

Use standard code only

Python is an exceptionally high-level language and includes a number of libraries for simplifying encryption including a cool function called "maketrans" and a number of specific cryptographic libraries. You are

welcome to experiment with any of these techniques for the black-belt extension, but your main program should stay within the core Python language. There should be no need to import any libraries for your basic version. Python does have a `String.translate()` method, but **please do not use it** for the basic assignment.

Remember, the goal is not to create a crypto program, but to experiment with building your own translator to learn string manipulation and functions.

Note that array manipulation is not necessary for this project. All work can be done directly with strings. You may need some exception-handling, though.

Submission

Please submit the following on Canvas:

- Your .py file (NOT a link to your pythonanywhere page and NOT a word document)
- A .txt file describing your algorithm (congruent with the requirements for algorithm files described in [the announcement \(%24CANVAS OBJECT REFERENCE%24/discussion_topics/g6b4bf997b8a8e34ab09edb503b4187b4\)](#) about algorithm files)
- If you are turning in a blackbelt version, submit your blackbelt as a separate .py file from your basic .py file

Black Belt Option

You can improve this program dramatically by adding the ability to generate a new random key. The new key should have the following characteristics:

- It should be 26 characters long
- It should contain all 26 letters
- The order of the letters should be randomly determined
- No letters repeat or are missing

You can add a new menu item to allow the user to generate a random key. You'll need a new function for the random key. You'll also need to modify the `main()` function and the menu to accept new input.

File Name		Size	
	Cryptography.txt	3.58 KB	✓
	cryptography.py	1.13 KB	✓