## A10: Ciphers

* Assignment A10 should be complete individually or with one partner.
* If you work with a partner, be sure to follow good pair programming practices.
* Be sure to read the entire prompt and understand the problem before beginning coding.

## Learning Objectives

* Work with both input and output files
* Working with classes, particularly creating objects and calling methods
* Explore ciphers, encrypting, and decrypting

## Translation algorithms used to crack centuries-old code

Read the following article published on 25 October 25, 2011, by Mark Brown

From: <https://www.wired.com/2011/10/copiale-cipher-crack/>

Computer scientists from Sweden and the United States have applied modern-day, statistical translation techniques -- the sort of which that are used in Google Translate -- to decode a 250-year old secret message.

The original document, nicknamed the Copiale Cipher, was written in the late 18th century and found in the East Berlin Academy after the Cold War. It's since been kept in a private collection, and the 105-page, slightly yellowed tome has withheld its secrets ever since.

But in 2011, University of Southern California Viterbi School of Engineering computer scientist Kevin Knight -- an expert in translation, not so much in cryptography -- and colleagues Beáta Megyesi and Christiane Schaefer of Uppsala University in Sweden, tracked down the document, transcribed a machine-readable version and set to work cracking the centuries-old code.

The book's pages -- bound in gold and green brocade paper -- contained about 75,000 characters in very neat handwriting. Outside of two words -- an owner's mark ("Philipp 1866") and a note in the end of the last page ("Copiales 3") -- the rest was encoded.

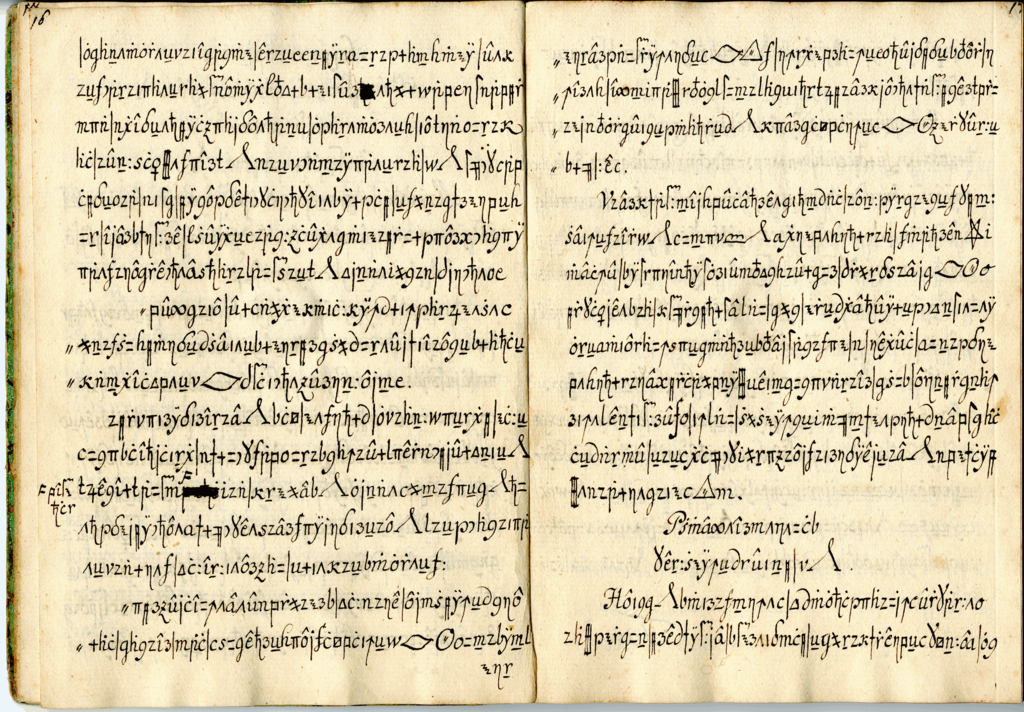
Some of the letters were obviously Roman and others were plainly Greek, while the rest were abstract symbols and doodles.

At first, Knight and his team isolated the Roman and Greek characters, figuring that they might be the real message, and attacked it with a home-made translation project. 80 different languages, and many hours later, and nothing happened. "It took quite a long time and resulted in complete failure," says Knight.

The team realized that the known characters were just there to mislead. So they booted them out and looked at the symbols. They theorized that abstract symbols with similar shapes might represent the same letter, or groups of letters. They tested this with different languages and when German was used, some meaningful words emerged -- "Ceremonies of Initiation", followed by "Secret Section".

A little computation later and a good chunk of the book had been decoded and transcribed. The document revealed the rituals and political leanings of a German secret society, and one that had a strange obsession with eyeballs, plucking eyebrows, eye surgery and ophthalmology. You can read the entire, weird, manifesto in English at <http://stp.lingfil.uu.se/~bea/copiale/copiale-translation.txt>

Buoyant from his success, Knight is now planning on using his techniques and programs to tackle other codes including ones from the Zodiac Killer, a Northern Californian serial murderer from the 60s; "Kryptos," an encrypted message carved into a granite sculpture on the grounds of CIA headquarters; and the Voynich Manuscript, a medieval document that has baffled professional cryptographers for decades.



You may optionally see the entire manuscript describing the work at <http://www.isi.edu/natural-language/people/copiale-11.pdf>.

## The instructions

After you have read the above article, respond to the following prompts. You only need write a paragraph of 3-5 sentences or more for each. The goal is that your response shows you have thought about the article and added your own experience or insights to its ideas.

|  |  |
| --- | --- |
| If you had an encrypted document that you somehow planned to decrypt, what do you think would be important features to consider as you tried to crack the code? Explain. | 1. |
| The article says, "A little computation later and a good chunk of the book had been decoded and transcribed." What do kind of computation do you think would have been used? Explain in a paragraph. | 2. |
| As of March 12, 2014, [Google is encrypting search globally](http://www.washingtonpost.com/blogs/the-switch/wp/2014/03/12/google-is-encrypting-search-worldwide-thats-bad-for-the-nsa-and-china/). Why do you think Google has chosen to do this? | 3. |
| Do you think there are ways of writing code where decryption is not simply the opposite process? Explain your answer. | 4. |
| Describe some types of information you consider private, and would want encrypted when being shared across a network. Explain why this type of information needs to be kept private. | 5. |

## Caesar Ciphers

Encryption has been in the news a great deal recently:

* July 11, 2013: [Microsoft handed the NSA access to encrypted messages](http://www.theguardian.com/world/2013/jul/11/microsoft-nsa-collaboration-user-data)
* December 21, 2013: [Report: NSA Paid RSA $10M to Create 'Back Door' in Encryption Software](http://www.pcmag.com/article2/0,2817,2428642,00.asp)
* March 12, 2014: [Google is encrypting search globally. That’s bad for the NSA and China’s censors](http://www.washingtonpost.com/blogs/the-switch/wp/2014/03/12/google-is-encrypting-search-worldwide-thats-bad-for-the-nsa-and-china/)
* February 17, 2016: [Apple, The FBI And iPhone Encryption: A Look At What's At Stake](http://www.npr.org/sections/thetwo-way/2016/02/17/467096705/apple-the-fbi-and-iphone-encryption-a-look-at-whats-at-stake)

The Google article proclaims the following:

*"Google has begun routinely encrypting Web searches conducted in China, posing a bold new challenge to that nation’s powerful system for censoring the Internet and tracking what individual users are viewing online.*

*The company says the move is part of a global expansion of privacy technology designed to thwart surveillance by government intelligence agencies, police and hackers who, with widely available tools, can view e-mails, search queries and video chats when that content is unprotected."*

Encryption and privacy are clearly big news.

Suppose you intercepted an email between two people that was using some kind of encryption, thereby creating a cryptogram:

pyhv ngyvr owf nrjrw urovn omy yhv posbrvn qvyhmbs pyvsb yw sbxn gywsxwrws, o wrt wosxyw, gywgrxjrf xw kxqrvsu, owf frfxgosrf sy sbr dvydynxsxyw sbos okk lrw ovr gvrosrf rzhok.

wyt tr ovr rwmomrf xw o mvros gxjxk tov, srnsxwm tbrsbrv sbos wosxyw, yv owu wosxyw ny gywgrxjrf owf ny frfxgosrf, gow kywm rwfhvr. tr ovr lrs yw o mvros qosskr-pxrkf yp sbos tov. tr bojr gylr sy frfxgosr o dyvsxyw yp sbos pxrkf, on o pxwok vrnsxwm dkogr pyv sbynr tby brvr mojr sbrxv kxjrn sbos sbos wosxyw lxmbs kxjr. xs xn oksymrsbrv pxssxwm owf dvydrv sbos tr nbyhkf fy sbxn.

qhs, xw o kovmrv nrwnr, tr gow wys frfxgosr -- tr gow wys gywnrgvosr -- tr gow wys bokkyt -- sbxn mvyhwf. sbr qvojr lrw, kxjxwm owf frof, tby nsvhmmkrf brvr, bojr gywnrgvosrf xs, pov oqyjr yhv dyyv dytrv sy off yv frsvogs. sbr tyvkf txkk kxsskr wysr, wyv kywm vrlrlqrv tbos tr nou brvr, qhs xs gow wrjrv pyvmrs tbos sbru fxf brvr. xs xn pyv hn sbr kxjxwm, vosbrv, sy qr frfxgosrf brvr sy sbr hwpxwxnbrf tyvc tbxgb sbru tby pyhmbs brvr bojr sbhn pov ny wyqku ofjowgrf. xs xn vosbrv pyv hn sy qr brvr frfxgosrf sy sbr mvros sonc vrloxwxwm qrpyvr hn -- sbos pvyl sbrnr bywyvrf frof tr socr xwgvronrf frjysxyw sy sbos gohnr pyv tbxgb sbru mojr sbr kons phkk lronhvr yp frjysxyw -- sbos tr brvr bxmbku vrnykjr sbos sbrnr frof nbokk wys bojr fxrf xw joxw -- sbos sbxn wosxyw, hwfrv myf, nbokk bojr o wrt qxvsb yp pvrrfyl -- owf sbos myjrvwlrws yp sbr drydkr, qu sbr drydkr, pyv sbr drydkr, nbokk wys drvxnb pvyl sbr rovsb.

Looks like gibberish, right? Actually, what was used was a simple cipher-substitution in which each letter ("a" through "z") in the original text was replaced by a different, random one. The cryptogram above, for example, has had every occurrence of the letter 'e' replaced with an 'r', every occurrence of 'w' by a 't', and so forth. The word "tr", which is the sixth word in the last paragraph, was therefore originally "we". The trick is to find which letter substitution was used and you can then reverse the process to get the original message.

Though not the first cipher, one of the simplest and most notable ciphers is the Caesar cipher, used by Julius Caesar to send messages to his generals during war. The cipher was simple; each letter is simply shifted in the alphabet by a number, which Caesar and the general would agree upon ahead of time. So a shift of +3 would equate to 'a' becoming 'd', 'b' would become 'e', and so on. The following Class demonstrates a Caesar cipher (be sure to download all files and place them in the same directory):

* [a10\_caesar\_cipher.py](https://drive.google.com/open?id=0B0J8Yj0B6KRSWG9EYktMbkYwNFE)
* [a10\_caesar\_cipher\_test\_suite.py](https://drive.google.com/file/d/0B0J8Yj0B6KRSTnVodzczZGE1eDA/view?usp=sharing)
* [message\_input.txt](https://drive.google.com/open?id=0B0J8Yj0B6KRSTmh5aXY0blV0ZFk)

## CRC Card for the CaesarCipher

|  |  |
| --- | --- |
| **Class name: CaesarCipher** | |
| **Class Attributes:** | **Class Collaborations (other classes):** |
| * alphabet:   + Class variable containing "ABCDEFGHIJKLMNOPQRSTUVWXYZ" which is the alphabet used to do our shifts * self.input\_file:   + is the file to be encrypted or decrypted * self.key:   + is the integer amount each message/cipher will be shifted * self.message:   + holds the message * self.cipher:   + holds the cipher * self.crypt\_type:   + holds either "encrypt" or "decrypt" | * None |
| **Class Methods:** | **Class Collaborations (other classes):** |
| * \_\_init\_\_():   + constructor for the CaesarCipher class * import\_file():   + imports a file stored in the variable self.input\_file and imports a string representing the contents of the file * export\_file():   + exports a file * encrypt():   + converts an original message into a ciphered message which is returned with each letter shifted to the right by the key * decrypt():   + converts a ciphertext and returns the original message which is found by shifting each letter to the left by the key | * None |

## The instructions

Modify the [a10\_caesar\_cipher.py](https://drive.google.com/open?id=0B0J8Yj0B6KRSWG9EYktMbkYwNFE) file to complete the following tasks:

Caesar has two letters to send (be sure to download these files into the same directory as your code!):

* [*letter\_to\_friend\_1.txt*](https://drive.google.com/open?id=0B0J8Yj0B6KRSbldSZTJYUlZvLW8)
* [*letter\_to\_friend\_2.txt*](https://drive.google.com/open?id=0B0J8Yj0B6KRSMzR5dGRTeDBaQk0)

Complete the code in main() to encrypt the above messages and generate two new encrypted files:

* *cipher\_to\_friend\_1.txt*
* *cipher\_to\_friend\_2.txt*

Caesar also has multiple letters from a friend which he needs to decrypt (again, be sure to download this file into the same directory as your code!). Here is one of these letters. Note that your code should be able to work with any letter which was encrypted using the Caesar cipher:

* [*letter\_from\_friend\_3.txt*](https://drive.google.com/open?id=0B0J8Yj0B6KRSc256Ti1VU2wwRDg)

The CaesarCipher Class is incomplete; while the program can encrypt a message, the program should also be able to decrypt the message. Complete the following tasks related to decryption:

* Complete the decrypt() function for decrypting the message inside the CaesarCipher Class.
* Once you've completed the decrypt function, make the correct calls in the main() function to construct a new CaesarCipher object and decrypt the message.
* Use the export\_file() function to write your decrypted message to a file named "*message\_from\_friend\_3.txt"*

## Submission Instructions

1. Review the requirements above to ensure you have completed everything that was required of you. **The test suite should help with that!**
2. Save your code as **A10\_ciphers\_*username*.py**. Replace *username* with your Berea usernames. For example, the TA Bianca Marrero’s file would be **A10\_ciphers\_marrerob.py.**
3. Save the following output text files:
   1. *cipher\_to\_friend\_1.txt*
   2. *cipher\_to\_friend\_2.txt*
   3. *message\_from\_friend\_3.txt*
4. Zip the code and the above three files together.
5. Upload the Zip file to Moodle by the due date listed on the course website: <https://trello.com/b/w7bIrLoV/>.
6. If you worked with a partner, your partner should upload a file named **A10\_ciphers\_*usernames*.txt** and include both partner’s name in the document.