# *Advanced Topics in Computer Science I (420-G40-HR)*

# *Assignment 4 – Crypto-Quotes*

Date assigned: Wednesday, November 24, 2021

Date due: Wednesday, December 15, 2021

**Learning Objectives**

Upon successful completion of this assignment, the student will be able to:

* Scrape data from a website.

To do:

**A cryptogram puzzle…what’s that?**

1. A cryptogram is a word puzzle featuring encrypted text that the user decrypts to reveal a message of some sort. Once used for message security, cryptograms are now typically only used for entertainment purposes in newspapers and magazines.
2. The original message is encoded by randomly changing each different letter in the original quote to a new letter…such as all occurrences of the letter S change to Q, and all occurrences of the letter U change to Y, etc. All letters in the quote are changed and the encoded quote is displayed to the user to try and solve. So, the quote “Corona Virus Sucks” might be changed to “ZTBTCF ADBYQ QYKPQ”
3. The user selects a letter to convert and the letter is converted in the quote. Once the user converts all of the letters, the quote is revealed and the user wins. For an example of how the game works try: <https://api.razzlepuzzles.com/cryptogram>. You are going to build a simple CLI version of this puzzle.
4. If the user wants to, they can change their mind as to which letter is converted. This can be done by changing the letter to a different letter, or changing it to a space to leave it blank.
5. In this assignment you can get the quotes you will use by reading the following site. The site contains a list of famous quotes.

|  |  |
| --- | --- |
| <https://litemind.com/best-famous-quotes/> | Quotes are the data in the paragraphs that have the class=wp\_quotepage\_quote. (NOTE: The numbers are not part of the quote and neither is the author) |

1. The process is as follows:
   1. Get the web page and parse out all the quotes.
   2. **Randomly** select a quote from the page to use.
   3. Encode the quote **randomly** (the encoding should change with every game play).
   4. Show the encoded quote to the user and allow them to solve the puzzle.
   5. Time how long it takes the player to complete the puzzle (use the time standard library).
2. The user interaction is as follows:
   1. Allow the user to enter a letter to convert, and the letter to convert to. (For example, R converted to X). The letter to convert can be a letter that is already converted – this is how the user can change their mind. However, you must prevent the same letter from being used twice. The user can also convert a letter back to blank (space).
   2. If the user enters an invalid character, then display an error message.
   3. Allow the user to get one (and only one) hint. The hint should convert a random letter.
   4. Allow the user to find all mistakes. This should remove all incorrect conversions, but leave the correct conversions. The user can then continue trying to solve the puzzle with the mistakes removed.
   5. Once the quote is complete, which means all letters are converted, (and correct), display a congratulation message that includes how long it took to solve the puzzle, and prompt the user to play again.
   6. If the quote is complete but incorrect, display a message to the user and ask them if they want to find all mistakes (functionality in step d) or play a new game.

DO:

1. Use urllib in the standard library to read the page you are getting the quotes from (just like you did in the lab).
2. Use HTMLParser in the html.parser standard library to parse out the quotes (just like you did in the lab).   
   NOTE: You can randomly select which quote you will use and only parse that one or parse them all and then select the one you are going to use. Either way is fine.
3. Determine your encoding string (I suggest using random.sample or random.shuffle from the standard library) for the entire alphabet.
4. Convert your selected quote and encode it. All quotes should be displayed in ALL UPPERCASE and no numbers or special characters (hyphen, apostrophe, period, etc) should be converted. Quote authors should not be converted.
5. In CLI, display the encoded quote for the user as well as an easy to read view of the 26 letters of the alphabet along with their currently converted value. If a letter has not yet been converted, then its value should be shown as blank.
6. In CLI, allow the user to enter the current value and the new value, as well as the option for a hint (use?) or the option to find all mistakes (!)
7. Read the values entered by the user and convert the letters as required.
8. Use colours to differentiate the guessed letters from the encoded letters. When finding mistakes use colours to show mistakes. When providing a letter use colours to show the letter provided. Colorama is a good library for this, but there are others like “prompt-toolkit” and “chalk” or “yachalk”.
9. Check to see if the quote is complete and correct. If not, redisplay the updated quote and prompt the user for more letters. If it is complete and correct display the quote, a congratulations, how long they took in minutes and seconds (mm:ss) and ask the user if they want to play again.
10. Code Pythonically; marks are allocated for Pythonic code. Use exceptions for error handling. Use comprehensions where possible. Use dunder methods in classes. Follow the PEP-8 standard.

DO NOT:

1. Convert numbers or punctuation, but display them as is in the quote.
2. Differentiate between case. All values are converted to uppercase regardless of what is entered.

**Steps to complete:**

1. Write your breakdown of tasks and how you will approach the project.
2. Write your black box test case scenarios for this program. This is the same thing you have always done – use these requirements and generate the list of scenarios needed to fully test the program.
3. Code using proper Python techniques and classes as applicable. If you want to use some functional programming that is fine, but it is not required.

**To submit**

1. A ZIP format (*YourInitials*B43A04.zip) containing all submitted files on Moodle.