```
#************************** CS421: Assignment 7 **************
#
#  Most of the implementation for the cryptoquote algorithm is done
#
#  You only need to implement section 5.
#
# Lines from 28 and 35 reflect different test cases.
# Start with simple test cases and progress to the bigger test cases
#
#  Use pythontutor.com to implement so that you can visualize how your
lists and variables are changing.
#  Save the complete implementation to a file called
"a_cryptoquote.py" and submit the file to Google Classroom
#
# How to save the code from PythonTutor to a file?
#           Select the entire code (Ctrl + A)
#           Copy the entire code (Ctrl + C)
#           Open notepad or any other editor you use to write text
files.
#           Paste the entire code (Ctrl + V)
#           Save the code to a file (Ctrl +S)
# ********************************************************


# Get the quote. This is the input.

#We will have different test cases
# start from simple test cases
# only when simple test cases are passing, go the next test case
quote = "aaaa"
quote = "abab"
quote = "ab ab"
quote = "abc abc abc"
quote = "abc! abc? abc@gmail.com"
quote = "Siva Jasthi"
quote  = "An eye for an eye makes the whole world blind!"
quote  = "An eye for an eye makes the whole world blind! - Mahatma
Gandhi"

# Once all the above test cases are passing
# we can ask the user for inputting the quote
# At that time, we can ask the user for the input
#quote = input("Enter a quotation: ")


# [1] I need a list of all letters from the input quote
#https://stackoverflow.com/questions/4978787/how-to-split-a-string-
into-an-array-of-characters-in-python

# since cryptoquotes deal with only upper case letters
# let us make a temporary string to hold the upper-cased quote
quote_upper = quote.upper()

#convert quote_upper into a list
quote_list = list(quote_upper)
```

```
# create a temporary quote_list_temp for holding the swaps
quote_list_temp = list(quote_list)
print(quote_list_temp)

#print what we got so far
print("This is the input quote")
print(quote)
print("This is the upper cased quote")
print(quote_upper)
print("This is the list of characters in the quote")
print(quote_list)



# [2]  We need an alphabet list which is ordered
# a,b,c,...z

# How do I create a ordered list of alphabest?
#https://www.geeksforgeeks.org/python-ways-to-initialize-list-with-
alphabets/

#import the string module
import string

# initializing empty list
ordered_list = []

# using string module for filling alphabets
# Since all cryptograms are not case sensitive,
# let us go with upper case
ordered_list = list(string.ascii_uppercase)

# printing the ordered list
print ("Ordered Alphabet List : ")
print(ordered_list)



# [3]. We need a randomized list
# How do I randomize this ordered list
# so that I can maintain the mapping
# between original (ordered) and the randomized list

import random

# make a copy of the original list
random_list = list(ordered_list);

#using the raondom module to shuffle (randomize) the input list
random.shuffle(random_list);
print ("Shuffled Alphabet List : ")
print(random_list)



# All you code goes between BEGIN and END lines.
```

```
# BEGIN ============= Student Code ============

#[4]  We will now create the crypto quote
#   per the psedo-code given below

# we now have these four lists
#  quote_list
#  quote_lit_temp
#  ordered_list
#  random_list
#
# Our algorithm depends on all these lists
#

#Once the for loop is done,
#we will have quote_list_temp reflecting the cryptoquote

# we need to keep track of the position
# at which we are doing the substitution
q_pos = -1

for x in quote_list:

    # bump up the position for the character
    # this is the character we are currently processing
    q_pos = q_pos + 1

    # we need to swap only the characters
    # symbols and characters can be ignored
    if (x in ordered_list):
        # if we are here, it means that we are processing a character

        # find out where this character is in the ordered alphabet
list
        # the x_pos should be between 0 and 25
        x_pos = ordered_list.index(x)

        # Now find out our mapping letter for the substitution
        y_char = random_list[x_pos]

        # Now do the swap in the quote_list_temp
        quote_list_temp[q_pos] = y_char

# print the quote_list_temp
print("Here is quote list temp after the substitutions: ")
print(quote_list_temp)

# END ============== Student Code ============


#5. We need to convert the quote_list_temp (the cryptoquote) to a
string
# How do I convert a list into a string?
# see  https://www.geeksforgeeks.org/jin-function-python/
# Variables for holding the final crypto quote
# We are joining the list into a single string
```

```
crypto= crypto.join(quote_list_temp)

#6. We need to show one hint to the users.
# Since quote_list_temp contains the cryptic character
# and the quote_list contains the original character
# We can simply show the first character mapping from both the lists
hint = "Hint => " + quote_list_temp[0] + " = " + quote_list[0]



# All previous print statements can be commended
# once the program is working.

#6 Now print all the variables in the end
# to verify that our algorithm is working
#print the quote
print("========== Your quote: ============ ")
print(quote)


#print the crypto
print("========== Crypto quote: ============ ")
print(crypto)
print("Hint: ", hint)
```