Assignment 2

CS329e - Elements of Software Design
Simplified Cryptography
(100 points)
Due Date on Canvas and Gradescope

1 Description

Cryptography is an ancient study of secret writing. There is a wealth of literature in this field. An extremely readable book on this subject is *The Code Book* by Simon Singh. This is a field of study that is of particular relevance in Computer Science. Given the widespread use of computers, one of the things people are interested in is making transactions over the internet more secure.

Here is a simple and clever way to encrypt plain text. Assume that the message contains only upper case letters, lower case letters and digits. Let L be the length of the original message, and M the smallest square number greater than or equal to L. Add (M-L) asterisks to the message, giving a padded message with length M. Use the padded message to fill a table of size $K \times K$, where $K^2 = M$. Fill the table in row-major order (left to right in each column, top to bottom for each row).

Now to encrypt, rotate the table 90° clockwise. The encrypted message comes from reading the message in row-major order from the rotated table, omitting any asterisks and maintaining the case of each character from the original message.

Let us say the original message is *gonewiththewind*. The message length L = 15 and so M = 16. The padded message is *gonewiththewind**. Here are two tables showing the padded message and the padded message after rotation.

g	o	n	e
w	i	t	h
t	h	e	w
i	n	d	*

Table 1: Original Padded Message

i	t	w	g
n	h	i	o
d	e	t	n
*	w	h	e

Table 2: Rotated Padded Message

So the encrypted message (ignoring the asterisks) is *itwgnhiodetnwhe*.

Decrypting a message would just be the reverse process of encrypting. Let us consider the encrypted message *osotvtnheitersec*.

o	s	o	t
v	t	n	h
e	i	t	e
r	s	e	c

Table 3: Original Padded Message

t	h	e	c
o	n	t	e
s	t	i	s
o	v	e	r

Table 4: Rotated Padded Message

So the decrypted message is `thecontestisover`.

Input:

You will read from standard input.

- The first line is a string P $(1 \leq length(P) \leq 100)$ that you will have to encrypt according to the following scheme.
- The second line is a string Q $(1 \leq length(Q) \leq 100)$ that you will have to decrypt.

Assume that both strings have only upper case letters, lower case letters, and digits.
Here is the format of your input **cipher.in**:

```
gonewiththewind
osotvtnheitersec
```

You will read your input from *stdin* like so:

Mac: `python3 Cipher.py < cipher.in`
Windows: `python Cipher.py < cipher.in`

Output:

You will print your output to standard out.

- The first line will be the encryption of string *P* and
- the second line will be the decryption of the string *Q*.

This is the format of your output cipher.out.

```
itwgnhiodetnwhe
thecontestisover
```

The file `Cipher.py` that you will be submitting will have the following structure. You will follow the standard coding conventions[1] in Python.

```
# Input: strng is a string of 100 or less of upper case, lower case,
#         and digits
# Output: function returns an encrypted string
def encrypt ( strng ):

# Input: strng is a string of 100 or less of upper case, lower case,
#         and digits
# Output: function returns an encrypted string
def decrypt ( strng ):

def main():
  # read the two strings P and Q from standard imput

  # encrypt the string P

  # decrypt the string Q

  # print the encrypted string of P and the
  # decrypted string of Q to standard out

if __name__ == "__main__":
  main()
```

You may not change the names of the functions listed. They must have the functionality as given in the specifications. You can always add more functions than those listed.

For this assignment you may work with a partner. Both of you must read the paper on Pair Programming[2] and abide by the ground rules as stated in that paper. If you are working with a partner then only one of you will be submitting the code. But make sure that your partner's name and UT EID is in the header. If you are working alone then remove the partner's name and eid from the header.

1.1 Turnin

Turn in your assignment on time on Gradescope system on Canvas. For the due date of the assignments, please see the Gradescope and Canvas systems.

1.2 Academic Misconduct Regarding Programming

In a programming class like our class, there is sometimes a very fine line between ”cheating” and acceptable and beneficial interaction between students (In different assignment groups). Thus, it is very important that you fully understand what is and what is not allowed in terms of collaboration with your classmates. We want to be 100% precise, so that there can be no confusion.

The rule on collaboration and communication with your classmates is very simple: you cannot transmit or receive code from or to anyone in the class in any way – visually (by showing someone your code), electronically (by emailing, posting, or otherwise sending someone your code), verbally (by reading code to someone) or in any other way we have not yet imagined. Any other collaboration is acceptable.

The rule on collaboration and communication with people who are not your classmates (or your TAs or instructor) is also very simple: it is not allowed in any way, period. This disallows (for example) posting any questions of any nature to programming forums such as **StackOverflow**. As far as going to the web and

[1] PEP 8 – Style Guide for Python Code `https://www.python.org/dev/peps/pep-0008/`

[2] Read this paper about Pair Programming `https://collaboration.csc.ncsu.edu/laurie/Papers/Kindergarten.PDF`

using Google, we will apply the **”two line rule”**. Go to any web page you like and do any search that you like. But you cannot take more than two lines of code from an external resource and actually include it in your assignment in any form. Note that changing variable names or otherwise transforming or obfuscating code you found on the web does not render the ”two line rule” inapplicable. It is still a violation to obtain more than two lines of code from an external resource and turn it in, whatever you do to those two lines after you first obtain them.

Furthermore, you should cite your sources. Add a comment to your code that includes the URL(s) that you consulted when constructing your solution. This turns out to be very helpful when you're looking at something you wrote a while ago and you need to remind yourself what you were thinking.

We will use the following Code plagiarism Detection Software to automatically detect plagiarism.

- Staford MOSS

  `https://theory.stanford.edu/~aiken/moss/`

- Jplag - Detecting Software Plagiarism

  `https://github.com/jplag/jplag` and `https://jplag.ipd.kit.edu/`