Cryptography (Due 14 Sep 2020)

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 and lower case letters and digits from *a* to *z*. 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 x 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.

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.

Original Padded Message

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

Rotated Padded Message

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

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

Input: You will read from standard input. The first line is a string P (1 ≤ length (P) ≤ 100) that you will have to encrypt according to the following scheme. The second line is a string Q (1 ≤ length (Q) ≤ 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 run your code like so:

```
$ python3 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 may **NOT** change the names of the functions but you may add as many helper functions as needed. You will follow the standard coding conventions 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 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 and abide by the ground rules as stated in that paper. If you are working with a partner then only one of you will submit the code. Make sure that in the header in HackerRank that you have your name and UT EID and your partner's name and UT EID. If you are working alone then you will just have your name and your UT EID.

Use the *HackerRank* platform to submit your code. We should receive your work by 11 PM on Monday, 14 Sep 2020. There will be substantial penalties if you do not adhere to the guidelines. HackerRank will not assign late penalties (if any), we will make the adjustments.

- Your code must run before submission.
- You should be submitting your file through the web based HackerRank platform. We will not accept files e-mailed to us.

References

- Simon Singh On Cryptography
- History of Cryptography
- An Overview of Cryptography