CSE222 / BİL505
Data Structures and Algorithms
Homework #5

In this homework, you are asked to write a complete Java program that can encrypt and decrypt a string using **Vigenère Cipher**.

Vigenère Cipher

Vigenère Cipher is a poly-alphabetic substitution algorithm. It generates a table, where each row is the alphabet, (circularly) shifted left by *row_index* times. For a better understanding, analyze the figure below:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

As seen, the first row is the alphabet itself, since it is shifted 0 times (because the row index is 0). On the other hand, the second row starts with “B, C, D, ...” and ends with “A”, because in this row (row index is 1) the alphabet has been shifted left once.

The table generation is not dynamic, i.e., the same table will be used for any input. The letters outside of the table are the indicators (they will be used to find the related row and column in the table).

In order to perform encryption/decryption, we need a text (plaintext or ciphertext) and a key. Both the text and key are strings with at least 1 letter. Additionally, there is a *keystream* which is generated from key, based on the length of the plaintext. There are three possible cases to generate the keystream:

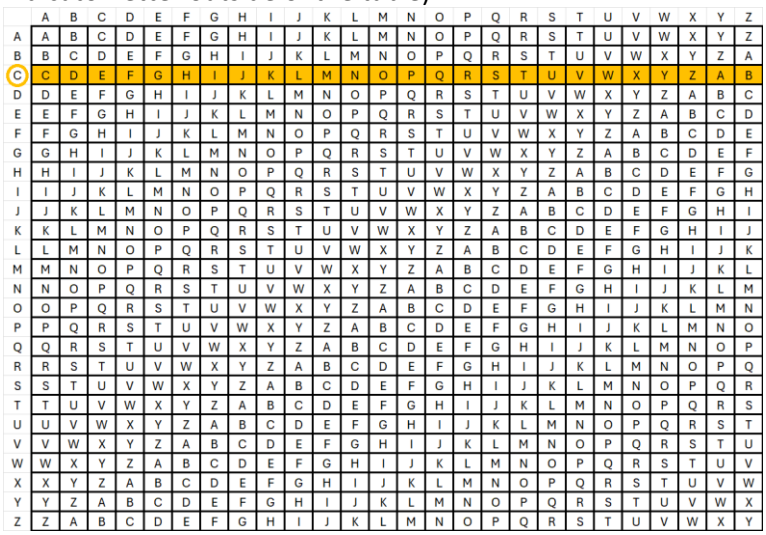
1) If the text is shorter than the key, the keystream is the first *length_of_text* letters of the key.
   Example:
   Text: DATA
   Key: ALGORITHM
   Keystream: ALGO
2) If the text is longer than the key, we repeatedly add the key to the end of itself to generate the keystream.
   Example:
   Text: ALGORITHM
   Key: DATA
   Keystream: DATADATAD
3) If the length of the text and the key is equal, then the keystream is the same as key.
   Example:
   Text: ALGO
   Key: DATA
   Keystream: DATA

Encryption

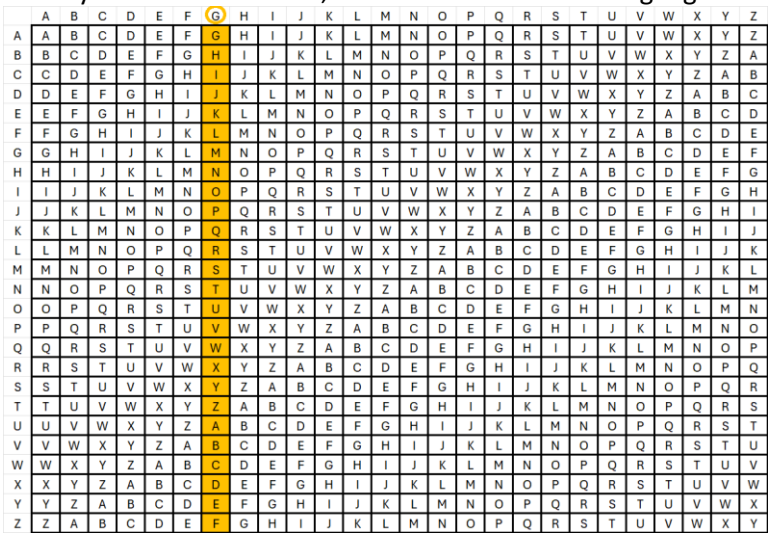
Once the keystream is generated, the ciphertext is being formed by using the plaintext, the keystream, and the table. The encryption should be handled letter by letter. For $0 \leq i < PlaintextLength$, $Ciphertext[i]$ is generated by using$Plaintext[i]$ and $Keystream[i]$. Let’s assume that the plain text is “CSE” and the key is “GTU”. In this case, keysteam is the same as key, “GTU”. In order to generate the first letter of the ciphertext, we should use the first letter of plain text, which is “C”, and the first letter of the keystream, which is “G”. Once we define the plaintext letter and keystream letter, we do the following:

1) **Find the relevant row in the table:** This is decided by using the letter of plaintext. Since the plaintext letter is “C”, we should focus on the highlighted row (this is decided by using the

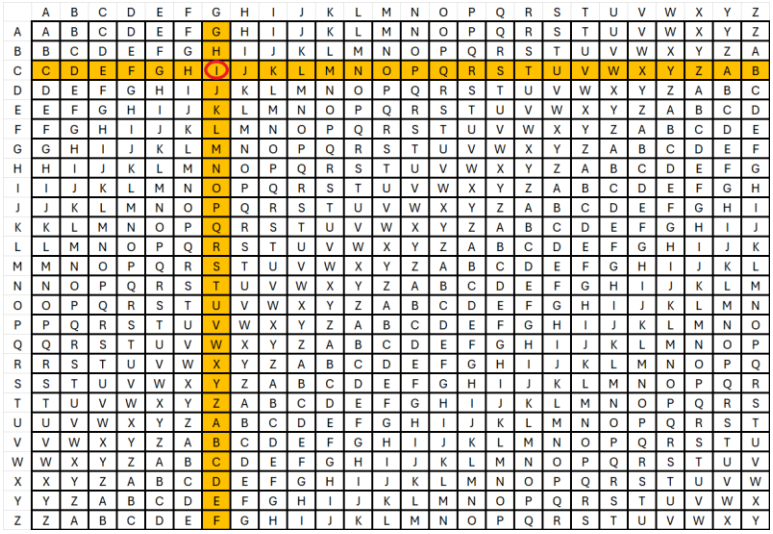
indicator letter outside of the table).



2) **Find the relevant column in the table:** This is decided by using the letter of keystream. Since the keystream letter is “G”, we should focus on the highlighted column.



3) **Find the ciphertext letter:** This is the letter on the intersection of the row and column. In this example, the ciphertext letter is “I”.

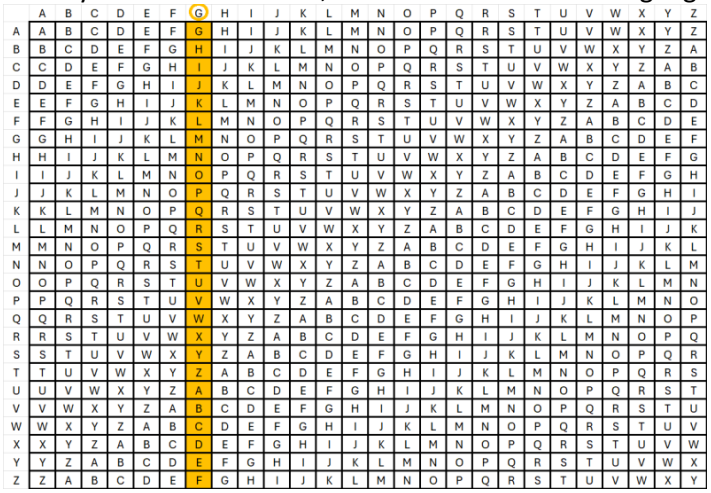


If we repeat these steps for the remaining two letters, we will obtain the ciphertext which is “ILY”.

Decryption

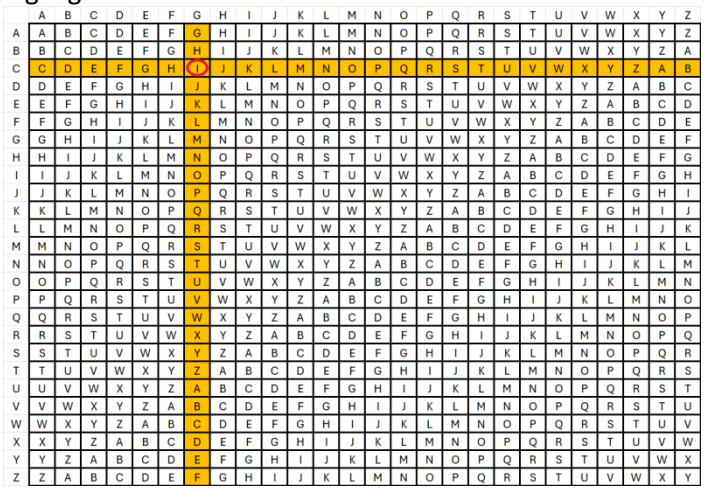
For the decryption of a ciphertext, we use the table to reverse the encryption. Assuming the ciphertext is “ILY” and the keystream is “GTU” (same example as in encryption), we should be able to obtain the plain text “CSE”. The decryption should also be done letter by letter. For the decryption of the first letter, we should do the following:

1) **Find the relevant column in the table:** This is decided by using the letter of keystream. Since the keystream letter is “G”, we should focus on the highlighted column.



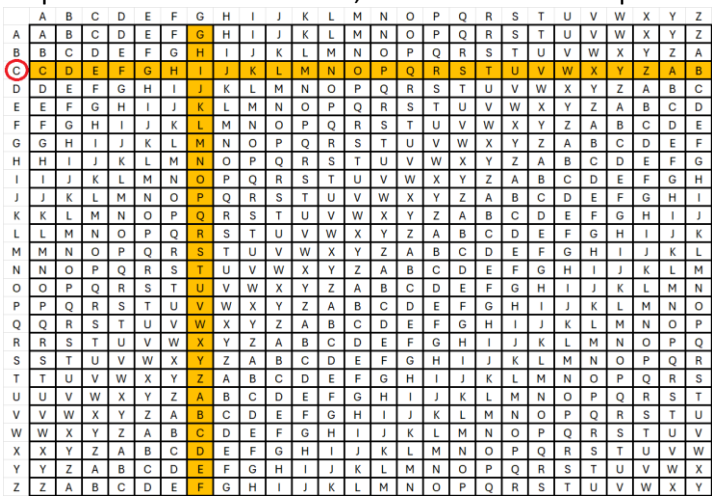
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

2) **Find the relevant row in the table:** This is decided by the row that contains the ciphertext letter on the highlighted column. Since the ciphertext letter is “I” we look for “I” in the highlighted column.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

3) **Find the plaintext letter:** This is the indicator letter which represents the row we found on Step 2. Since that letter is “C”, the first letter of the plain text is “C”.



If we repeat these steps for the remaining two letters, we will obtain the plaintext which is “CSE.

How to implement it in Java?

In this homework, you are asked to write a complete program that will

- generate the table,
- preprocess the inputs,
- generate the keystream, and
- encrypt / decrypt.

The table will be a Map: `Map<Character, Map<Character, Character>>` As seen the key of the map is a character, this is the indicator letter of each row. The value of the key is another map.

In the inner map, the key is a character, this is the indicator letter of each column. The value is a character as well, this is the letter in the intersection of the related row and column. So, this map has 26 keys, a single character for each letter in the English alphabet. The inner map has also 26 keys, a single character for each letter in the English alphabet.

Consider the encryption example, the row indicator is “C” and the column indicator is “G”. In this case,

- the key of the outer map is “C”
- the value of the outer map is another map (which we call the inner map)
- the key of the inner map is “G”
- the value of the inner map is “I”

To be more specific, the inner map in this example forms as below:

KEY	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
VALUE	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B

Once the map is generated, the program will print it and ask for the inputs. There are two inputs in this program: *the text* and *the key*. Both inputs should be preprocessed by

- capitalizing the string and then
- removing every character that is not a letter.

After the preprocessing, if *the text* or *the key* has no letter left (or the input itself is an empty string), the program should terminate. If the inputs are proper, the program should print the preprocessed strings and then perform an encryption and decyrption process on these strings.

- The first operation is encryption. The encryption consists of generating the keystream and the ciphertext. In this operation, *the text* input should be treated as the plain text. Both the keystream and the cipher text should be printed.
- The second operation is decryption. The decryption consists of generating the keystream and the plaintext. In this operation, *the text* input should be treated as the cipher text. Both the keystream and the plain text should be printed.

In other words, the program both encrypts and decrypts the given text. Once both operation is done, the program terminates.

Class Structure and Programming Details

In this homework, you will receive a template for the program. In this template, the classes and methods are already defined. You are asked to fill these methods, making sure the program works correctly. You should obey the following instructions in order to receive full credit.

- You cannot add new classes.
- You cannot add new methods.
- You cannot change the variables of a class.
- If a method is provided with its code in the template, you cannot change it.
- You cannot change the parameters of the given classes (or you cannot add new ones).
- You cannot add new libraries, other than the ones provided in the template.
- You cannot use char arrays to define the map, you should use “set” (along with an “iterator”). This step will be described in detail in the Problem Session.
- You will also receive a makefile, you cannot corrupt it.
- You must add comments to each method.

Sample Screenshots

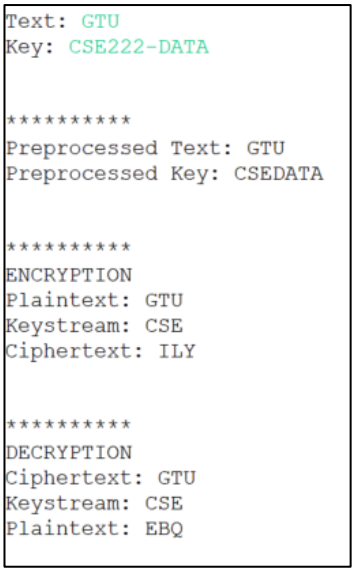
The program should work as seen in the examples below.

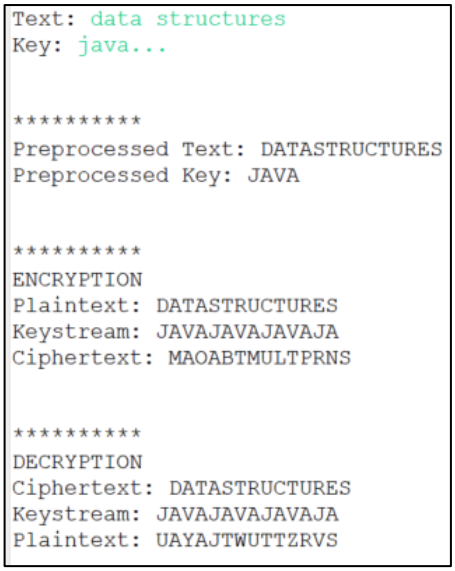
PS: Green texts are inputs, the rest is output. The screenshots after the first one are cut to leave the map out (for a better view).

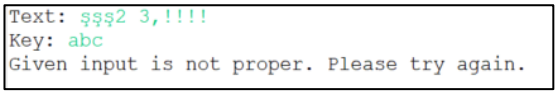
```
*** Viegenere Cipher ***


    [A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z]
    ------------------------------------------------------------------------------
A | [A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z]
B | [B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A]
C | [C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A, B]
D | [D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A, B, C]
E | [E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A, B, C, D]
F | [F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A, B, C, D, E]
G | [G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A, B, C, D, E, F]
H | [H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A, B, C, D, E, F, G]
I | [I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A, B, C, D, E, F, G, H]
J | [J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A, B, C, D, E, F, G, H, I]
K | [K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A, B, C, D, E, F, G, H, I, J]
L | [L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A, B, C, D, E, F, G, H, I, J, K]
M | [M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A, B, C, D, E, F, G, H, I, J, K, L]
N | [N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A, B, C, D, E, F, G, H, I, J, K, L, M]
O | [O, P, Q, R, S, T, U, V, W, X, Y, Z, A, B, C, D, E, F, G, H, I, J, K, L, M, N]
P | [P, Q, R, S, T, U, V, W, X, Y, Z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O]
Q | [Q, R, S, T, U, V, W, X, Y, Z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P]
R | [R, S, T, U, V, W, X, Y, Z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q]
S | [S, T, U, V, W, X, Y, Z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R]
T | [T, U, V, W, X, Y, Z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S]
U | [U, V, W, X, Y, Z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T]
V | [V, W, X, Y, Z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U]
W | [W, X, Y, Z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V]
X | [X, Y, Z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W]
Y | [Y, Z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X]
Z | [Z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y]

Text: alglor,ith m
Key: data3 456


**********
Preprocessed Text: ALGORITHM
Preprocessed Key: DATA


**********
ENCRYPTION
Plaintext: ALGORITHM
Keystream: DATADATAD
Ciphertext: DLZOUIMHP


**********
DECRYPTION
Ciphertext: ALGORITHM
Keystream: DATADATAD
Plaintext: XLNOOIAHJ
```

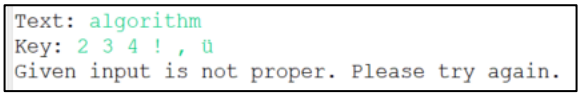
```
Text: cse
Key: GTU


**********
Preprocessed Text: CSE
Preprocessed Key: GTU


**********
ENCRYPTION
Plaintext: CSE
Keystream: GTU
Ciphertext: ILY


**********
DECRYPTION
Ciphertext: CSE
Keystream: GTU
Plaintext: WZK
```

```
Text: ILY
Key: CSE


**********
Preprocessed Text: ILY
Preprocessed Key: CSE


**********
ENCRYPTION
Plaintext: ILY
Keystream: CSE
Ciphertext: KDC


**********
DECRYPTION
Ciphertext: ILY
Keystream: CSE
Plaintext: GTU
```









General Information About the Homework:

- Cheating is not permitted. The students who cheat will receive NA from the course.
- You should submit your work as a single **zip** file.
- The Problem Session on April 30 will describe the homework in detail. If you have any questions, you should ask them in PS. Any other questions before/after the PS will be ignored unless there is a mistake or missing information in this PDF. In such a case, the announcement will be made on Teams, you are responsible for reading them in time.
- Late submission is not allowed. The due date will not be postponed.

Grading Details

Part	Grade
Preprocessing of the strings	20 points
Encryption	40 points
Decryption	40 points

You will not receive partial grading in any part.

Additionally, you must be careful with the following situations.

Behavior	Result
If your program doesn't work due to an error	You will lose 70 points.
If you use additional libraries	You will lose 20 points per library.
If you add a new method/class or change the structure of a given method/class	That method/class will be deleted. If your program works fine without it, nothing changes. If your program doesn't work without it (due to an error), you will lose 70 points.
If you don't use "set" and "iterator" as it will be described in the Problem Session	You will lose 40 points.
If you don't use "map" to define the table	You will lose 70 points.
If you don't add comments	You will lose 10 points.
If you hand-in your work in a format other than zip (you should submit it as a single zip file, the name of the zip file is not important in this homework)	You will lose 10 points.