**Algorithms & Data Structures**

**Lab 2 : Week beginning 23rd February, 2023**

Here, we will write code for 2 encryption algorithms:

1. Row/Column Transposition
2. Caesar Cipher Encryption

**1. Row/Column Transposition**

Row/column encryption is one type of encryption.

To implement this, you don’t need to write the text into an array.

Instead read the characters from the plaintext string, skipping numColumns (number of columns) between each character.

In a class called Encrypter, write a method with header as follows:

public static String encryptRowColumn(String plaintext, int numColumns)

Use StringBuilder class to build ciphertext and not String concatenation.

**Pseudocode:**

Remove spaces from plaintext

Convert plaintext to uppercase

Calculate numRows required for plaintext

If plaintext doesn’t fit exactly, pad it with Xs

String: ciphertext = ""

For col = 0 To numColumns - 1

Integer: index = col1

For row = 0 To numRows - 1

ciphertext = ciphertext + plaintext[index]

index += numColumns

Next row

Next col

Note: plaintext[index] in this pseudocode is like stuff we would use to indicate an element of an array. Here it is used to specify a character in a String. What method would you use to do that?

Test the code in main() method of another class.

To decipher a message in a program, notice that decoding a message that was originally written in an array that has R rows and C columns is the same as encrypting a message with an array that has C rows and R columns.

Continued......

Write another methods with header as follows:

public static String decryptRowColumn(String cipherText, int numColumns)

This method should call the encryptRowColumn() method.

**2. Caesar Cipher Encryption**

Write a method with header as follows:

public static String encryptCaesarCipher(String plaintext, int shift)

Assume plaintext only contains upper-case letters (A-Z) or spaces and shift is a positive integer. Spaces should not be changed.

Use StringBuilder **not** concatenation to build ciphertext – check why it is more efficient to use StringBuilder

**Pseudocode**

IF shift >= 26 //note >= (if shift equals 26 what happens?)

????

ciphertext = empty string

FOR i from 0 to plaintext.length-1

IF char at position i of plaintext is not a space

ch = encrypt char at position i of plaintext – how is this done?

concatenate ch to ciphertext

ELSE

concatenate a space to ciphertext

return ciphertext

Note:

Here you could also remove spaces from plaintext and convert plaintext to uppercase like you did with row column transposition.

With char variables, you can use numeric operators ++, --, +=, -=

e.g.

char ch = 'D';  
ch++;  
System.*out*.println("Expected: E, Actual: " + ch);  
ch += 2;  
System.*out*.println("Expected: G, Actual: " + ch);  
ch -= 3;  
System.*out*.println("Expected: D, Actual: " + ch);