**2D Arrays Lab 4 Secret Message**

In this lab you will write four methods for a class MessageCipher that encrypts (puts into a coded form) a message by changing the order of the characters in the message. The message cipher **fills** a two-dimensional array with single-character substrings of the original message in **row-major order**, encrypting the message by **retrieving** the single-character substrings in **column-major order**.

For example, the word "Surprise" can be encrypted using a 2-row, 4-column array as follows.



An incomplete implementation of the MessageCipher class is shown below.

import java.util.\*;  
public class MessageCipher {  
 private String[][] letterBlock;  
 private int numRows;  
 private int numCols;  
  
 public MessageCipher (int row, int col){  
 numRows=row;  
 numCols=col;  
 letterBlock = new String[row][col];  
 }  
  
 public void fillBlock(String str)

{ your code goes here }

public void printTable()

{ your code goes here }

public String [][] getTable()

{ your code goes here }

public String encryptBlock()

{ your code goes here }

public String encryptMessage(String message)  
 { your code goes here }

// There may be instance variables and methods that are not shown.

}

1. Write the method fillBlock that fills the two-dimensional array letterBlock with one-character strings from the string passed as parameter str. **The array must be filled in row-major order**—the first row is filled from left to right, then the second row is filled from left to right, and so on, until all rows are filled. If the length of the parameter str is smaller than the number of elements of the array, the string "A" is placed in each of the unfilled cells. If the length of str is larger than the number of elements in the array, the trailing characters are ignored.

If letterBlock has 3 rows and 5 columns and str is the string "Meet at midnight", the

resulting contents of letterBlock would be as shown in the following table.



For example, if letterBlock has 3 rows and 5 columns and str is the string "Meet at noon",

the resulting contents of letterBlock would be as shown in the following table.



2. Write the method printTable() that displays the filled 2D array.

The above example should look like:

**M e e t   
a t n o   
o n A A A**

3.Write the method String [][] getTable().

4. Write the method encryptBlock that extracts and concatenates the single-character substrings from letterBlock in **column-major order**.

In the example above, the returned string should look like: **Maoetne AtnA oA**

5. Write the method encryptMessage that encrypts its string parameter message. The method builds an encrypted version of message by repeatedly calling fillBlock with consecutive, nonoverlapping substrings of message and concatenating the results returned by a call to encryptBlock after each call to fillBlock. When all of message has been processed, the concatenated string is returned.

The following example shows the process carried out if letterBlock has 2 rows and 3 columns and

encryptMessage("Meet at midnight") is executed.

In this example, the method returns the string **"Mte eati dmnitgAhA".**

6. **Make sure to test all the methods as you go.** An **incomplete implementation** of the testing program is included below.

import java.util.\*;  
public class MessageCipher\_Tester  
{  
 public static void main (String [] args){  
 MessageCipher puzzle= new MessageCipher(3,5);  
 puzzle.fillBlock("Meet at midnight");  
 puzzle.printTable();  
 puzzle.fillBlock("Meet at noon");  
 puzzle.printTable();  
 System.out.println(puzzle.encryptBlock());   
 }

}