LAB #3: ARRAYS, METHODS

1. Task: Write a Java program to solve the following problem using 1-dim arrays. Given a 1-dim array, create a second array that holds two copies of each value in the original array. Have the following methods:

- A method named `getInt` to get input a valid integer. The method is used to test for type errors in input.
- A method named `initRand` to fill the array with random values: the elements of the array should be random integers between 1 and 100 (declared as class constants).
- A method named `print` to print the elements of the array. Use `printf`.
- A method named `twice` that returns the second array holding two copies of each value in the original array

All methods should take as input parameters the array and its size (do not use `.length`). Make sure that all methods are reusable. Write a driver program to call these methods and get output similar to the sample output below (use a loop).

SAMPLE OUTPUT:

```
Start (y/n): y
How many elements/array: a
Not an integer! Try again!
How many elements in the list: 3.5
Not an integer! Try again!
How many elements in the list: 9
The original array is:
2 3 7 4 34 12 9 37 5
The second array is:
2 2 3 3 7 7 4 4 34 34 12 12 9 9 37 37 5 5
Continue(y/n): y
How many elements in the list: 5
The original array is:
1 37 49 3 72
The second array is:
1 1 37 37 49 49 3 3 72 72
Continue(y/n): n
```

2. Task: Write a Java program to solve the following problem using 2-dim arrays. Given a 2-dim array (matrix) containing only 1 or 0 values, determine if there are 2 rows `r1` and `r2` so that `r1` is the same as `r2` flipped. For example, if `r1` holds the values `0 1 1 0 0 1`, then `r2` should hold the values `1 0 0 1 1 0`.
Program requirements.

- Have (at least) the following methods:

  1. A method named `getInt` to get input a valid integer. The method is used to test for type errors in input.
  2. A method named `initRand` to fill the array with random values: the elements of the array should be random integers (0 or 1)
  3. A method named `print` to print the elements of the array. Use `printf`. Also, get the format from the sample output (row index in front of each row)

4. A method to find flipped rows (design decisions?)

- All methods processing the array should take as input parameters the array (use `.length`).
- Make sure that all methods are reusable!
- Write a driver program to call these methods and get output similar to the sample output below (use a sentinel-controlled loop)
- Work with dynamic arrays and get input from the user the row size/column size. Make sure you handle input validation.

SAMPLE OUTPUT:

```
Do you want to start(Y/N): y
How many rows: 423rt23
Not an integer! Try again! How many rows: 6.78
Not an integer! Try again! How many rows: 8
How many columns: 4
The matrix is:
  [0]    0    0    0    1
  [1]    1    1    1    0
  [2]    1    0    1    0
  [3]    1    1    0    1
  [4]    0    0    1    0
  [5]    1    1    0    1
  [6]    1    1    1    0
  [7]    0    0    1    0
Rows index 0 and 1 are flipped.
Rows index 0 and 6 are flipped.
Rows index 3 and 4 are flipped.
Rows index 3 and 7 are flipped.
Rows index 4 and 5 are flipped.
Rows index 5 and 7 are flipped.

Do you want to continue(Y/N): y
How many rows: 3
How many columns: fwef
Not an integer! Try again! How many columns: 8
The matrix is:
  [0]    1    1    1    1    0    0    0    1
  [1]    0    0    1    1    0    1    1    1
  [2]    0    1    0    0    1    1    1    1
No flipped rows in this matrix.

Do you want to continue(Y/N): y
How many rows: dsafdsf
Not an integer! Try again! How many rows: 7
How many columns: bgfgf
Not an integer! Try again! How many columns: 4
The matrix is:
  [0]    1    1    1    0
  [1]    0    1    0    0
  [2]    0    0    0    1
  [3]    1    0    1    1
  [4]    0    1    1    0
  [5]    1    1    0    1
  [6]    0    0    1    1
Rows index 0 and 2 are flipped.
Rows index 1 and 3 are flipped.

Do you want to continue(Y/N): n
```

Notes:
A. The lab will NOT be graded, but you have to submit good quality work in order to get credit.
B. The lab should be completed by the start of the next scheduled lab class. E-mail the **.java** files (attachments) to Rohan Patel (rpatel27@students.towson.edu)
Very important: Make sure that you have COSC 237.section, your name, and Lab#3 in the *Subject* box of your e-mail.
C. In case you have any problems, contact the instructor or the TA for assistance.