HW Assignment 03

Purpose:

- Array operations (1D arrays and 2D arrays)
- String operations
- Loop structures in Java

Each of the following task will be in one single .java file.

Task 1: Reverse An Array

Sample input:

```
int[] arr={1,2,3,4,5,6,7,8}
```

Sample output: after the manipulation, the above array should now look like

```
{8,7,6,5,4,3,2,1}
```

What you need to do

1. Write a main() function. Inside the main() function, prepare you test data, that is, an integer array to be reversed.

- You can either hard-coded an array, or
- Get the array from user input, or
- Use random number to generate the numbers in the array.
- Print out the Array before you reverse it.

After that, call the **reverse** function, defined as below.

2. Write a **reverse** function. The function's header looks like

   ```
   public static void reverse(int[] myArray)
   ```

   Note that the input or parameter is an array.

   - make sure that you algorithm will work for all array.
   - space efficiency: You algorithm should NOT make a whole copy of the array, and then read it backward and copy it again. You algorithm SHOULD operate *myArray* directly, swapping elements inside the array. You can use one integer value about as a temporary space to help you do the swapping.
   - The function has no return value. All the changes will just be applied to *myArray* directly. This is the so-called *side-effect* when you pass a parameter by reference.

3. Print the array after you reverse it.

Transpose a 2D array

Basically, to transpose a 2D array, you need to switch the element in row *i*, column *j*, with the element in row *j*, column *i*.

Sample input: you have a 2D array that looks like

```
1, 2, 3,
4, 5, 6,
7, 8, 9.
```

Sample output: after transposing, you have

```
1, 4, 7,
2, 5, 8,
3, 6, 9.
```

All the matrices you need to process are assumed to be square matrices. That is, they have the same number of rows and columns.

All other requirements will be similar to Task 1. Note that you still need consider space efficiency.

You core function will have the header

```
public static void transpose(int[][] myArray).
```

Palindrome

You can find the definition of palindrome on Wiki. You task here is to determine weather a string is a palindrome.

Note that you need to handle strings like

```
Was it a car or a cat I saw?
```

```
A man, a plan, a canal, Panama!
```

That is, you need to strip away the extra spaces and punctuation before you determine the string is a palindrome or not.

Same old structure:

- You need to have a main() function to prepare your data.
- You need to have a core function which does all the work, with header

  ```
  public static boolean isPalindrome(String aString)
  ```

- In your main(), output some message to tell the user whether a string is palindrome or not, once after you make a call to **isPalindrome()**.

Submission

- Total score 150. That is 50 points each.
- Each task should be in a .java file and has its own main() function.
- Submit you .java source file this time and all submissions after this one.
- Do not zip them.
- Due date: Mar 02, midnight.