

Exercise 1.4: Programming Exercises

Expectations:

1. All code must be commented. Header comments with title, brief description, author, date. Only reasonable in-line code comments. It is not necessary to state the obvious. Using descriptive variables can cut down on the amount of redundant inline comments within your code. Joke commenting will result in joke grading, unless it is relevant or really, really funny (in my view, that is).
 2. Good programming style. Use indentation wisely. Use blank spaces where needed.
 3. Be consistent with your programming style.
 4. Use proper conventions:
 - filename, class name, method names, variable names
 5. Follow established Java programming standards
 6. Do not submit code to me until everything is complete.
-

1. Write a program that reverses the order of a one-dimensional array `a[]` of `String` values. Do not create another array to hold the result. (5)
2. What does the following code fragment print? (1)

```
int[] a = { 1, 2, 3 };
int[] b = { 1, 2, 3 };
System.out.println(a == b);
```
3. What does the following code fragment print? (1)

```
int[] a = { 1, 2, 3 };
int[] b = a;
System.out.println(a == b);
```
4. Explain the results of (2) and (3) above. (3)
5. Write a program that only prints the *transposition* (rows and columns changed) of a two-dimensional array. (5)

For example, the 2D array:

99	85	98
98	57	78
92	77	76
94	32	11
99	34	22

Would print the following 2D array:

99	98	92	94	99
85	57	77	32	34
98	78	76	11	22

6. Write a program to transpose a square two-dimensional array (the number of rows and columns are equal) in place without creating a second array. The program in 3 above just prints the elements.

But this time, create a program that replaces the transposed the values back into the original array. (5)

7. Write a program that multiplies two rectangular matrices that are not necessarily square. Note: For the dot product to be well-defined, the number of columns in the first matrix must be equal to the number of rows in the second matrix.

Print an error message if the dimensions do not satisfy this condition. (5)

8. Write a program to create a two-dimensional array `b[][]` that is a copy of an existing two-dimensional array `a[][]`, under each of the following assumptions:
- a) `a[][]` is square (there is an equal number of rows and columns)
 - b) `a[][]` is rectangular (the number of rows does not equal the number of columns)
 - c) `a[][]` may be ragged (each row in the two-dimensional array may have a different number of columns)

Square array: `a[5][5]`

Rectangular array: `a[5][10]`

Ragged array:

```
int[][] a = {
    { 1, 3},
    { 2, 5, 3, 6 },
    { 4, 6, 2 },
    { 6, 3, 8, 0, 5 }
};
```

Your solution for (b) should work for (a) and your solution to (c) should work for both (b) and (a), but your code should get progressively more generic.

Hint: A two-dimensional array is just an array of arrays. (15)

9. *Find a duplicate.* Given an `int` array of `N` elements, with each element having a value between 1 and `N`, write an algorithm to determine the number of duplicate values. You do not need to preserve the contents of the given array, but do not use an extra array.

Initialize the array with random numbers between 1 and `N` (You can use the `Math.random()` function and the code you created in Question 14 of Exercise 1.1). When testing to see if your code works, print out the values of your array before you do any calculations. (20)