Exercises: up to Chapter 8

1. Write a program that contains an array holding the names of the days of the week (in whatever language you want) and prints them out.
2. Modify your program so that instead of printing out the days of the week, it asks the user for an integer in the range of 1 to 7 and looks up and prints the day corresponding go that number (you can decide if Sunday or Monday is the first day of the week if you pick English; opinions differ).
3. Modify your program so that it does not crash if a number is outside the range of the array with the days of the week - remember to use the length property of arrays to do this.
4. Write a program that creates an empty array of size 10, then reads in integers and stores them in the array but only if that value is not already in the array (that is, it ignores duplicates). The program should not let the user enter more numbers than the array can hold (but remember you cannot just ask them to enter 10 numbers because some may be duplicates).
5. Write a program that computes the average of the values stored in an array. Your program should not depend on a fixed size of array - that is, if the array of data is made bigger, the rest of the program should not need to be changed.
6. **Tricky** - Extend your average program so that if one number occurs more than once, the program prints out how many times.
7. Write a program that reverses the order of elements in an array - for example, if your array contained 1 3 7 5 9 2 then the output would be 2 9 5 7 3 1.
8. **Tricky but not a challenge** - A magic square is an n by n matrix filled with numbers between 1 and n2 such that the sum of all rows and columns, as well as the two diagonals, is the same: E.g.

2	7	6
9	5	1
4	3	8

   where all the rows, columns and both diagonals add up to 15. Write a program that reads in 16 numbers in the range of 1 to 16, and stores them in a 4 by 4 (2d) array - and then works out if they form a magic square.
9. **Challenge**. The first days of the months of 2016 are: January - Friday; February - Monday; March - Tuesday; April - Friday; May - Sunday; June - Wednesday; July - Friday; August - Monday; September; Thursday; October - Saturday; November - Tuesday; December - Thursday.
   Write a program that stores this data and the days of the week in an array; and when the user enters the name and day of a month, looks up which which day of the week corresponds with that day - you may find the `%` operator quite handy. (Hint: your array of the days that start each month should probably store the *numbers* and not the names of the days).
10. **Serious Challenge** - do the same but only rely on knowing that the 1st Jan 2016 is a Friday.
11. Write a program that reads in words and stores them in sorted order in an array or ArrayList (you choose - which do you think would be a better choice?).
12. Re-write your program to reverse an array so that it reverses an ArrayList instead - to quickly fill an ArrayList with 'test' data, try this:

```
import java.util.ArrayList;
import java.util.Arrays;

String[] example = {…};

ArrayList<String> = new ArrayList<>(Arrays.asList(example));
```

13. Write a program that reads in numbers and stores them in an ArrayList - every time a new number is added, the program outputs the position of the smallest and largest elements in the ArrayList (hint: you only need one loop to do this).
14. Write a program that reads in numbers and stores then in an ArrayList - the program does not allow duplicates. Hint: there are several ways to do this which do not need a loop - the order in which numbers are stored is not important.