Chapter 7 Problems Part 2

12. What is an array traversal? Give an example of a problem that can be soled by traversing an array.

Traversing the array is simply going through every element in the array. A problem that can be solved by traversing an array is to find the maximum value in an array of stock prices.

13. Write code that uses a `for` loop to print each element of an array named `data` that contains 5 integers.

```
public class test {
    public static void main(String[] args) {
        int[] data = new int[]{1,2,3,4,5};
         for (int i=0; i<data.length;i++) {
             System.out.println("element [" + i + "] is " + data[i]);
         }
    }
}
```

14. What elements does the array `list` contain after the following code is executed?

```
{3,27,9,0,7,1}
```

17. Write a method called `allLess` that accepts two array of integers and returns `true` if each element in the array is less than the element at the same index in the second array. Your method should return `false` if the arrays are not the same length.

```
public class test {
    public static void main(String[] args) {
        int[] arrA = new int[]{1,3,3,4,5};
        int[] arrB = new int[]{1,2,3,4,5};
        System.out.println(allLess(arrA,arrB));
    }
    public static boolean allLess(int[] arrA,int[] arrB) {
        if (arrA.length < arrB.length || arrA.length > arrB.length) {
            return false;
        }
        for(int i = 0; i<arrA.length; i++) {
            if (arrA[i] == arrB[i]) {
                ;
            }
            else {
                return false;
            }
        }
        return true;
    }
}
```

19. `ReferenceMystery1`

```
2 [0, 0, 1, 0]
1 [0, 0, 1, 0]
3 [0, 0, 1, 1]
2 [0, 0, 1, 1]
```

20. `ReferenceMystery2`

```
2 [0, 1]
1 [0, 1]
1 [1, 2]
0 [1, 2]
```

21. Swapping method

```
public static void swapPairs(int[] list) {
    for (int i = 0; i < list.length - 1; i += 2) {
        swap(list, i, i + 1);
    }
}
```

22. `numbers`

```
[20, 30, 40, 50, 60, 70, 80, 90, 100, 100]
```

23. `numbers`

```
[10, 10, 10, 10, 10, 10, 10, 10, 10, 10]
```

24. `mystery`

```
[26, 19, 14, 11, 10]
[1, 4, 9, 16, 25]
```