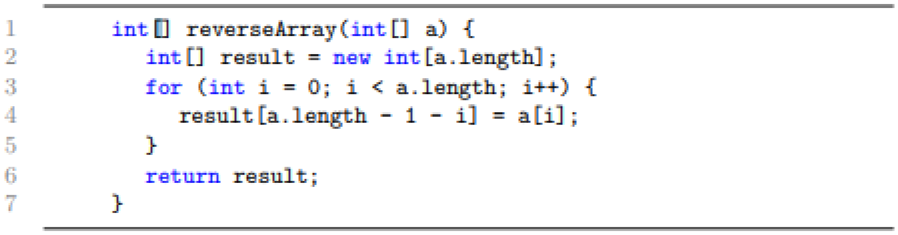
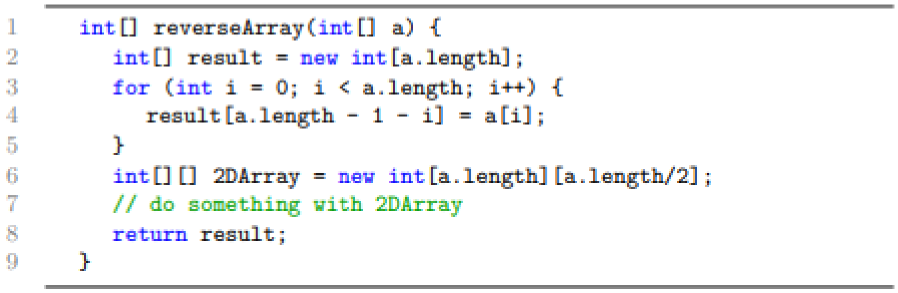
1. (5 points) Consider the following implementation of the method reverseArray(). Using Big O notation, what is the space complexity of this method? Justify your answer.



```
int[] reverseArray(int[] a) {
   int[] result = new int[a.length];
   for (int i = 0; i < a.length; i++) {
      result[a.length - 1 - i] = a[i];
   }
   return result;
}
```

   a. Considering the following implementation of the method reverseArray() above, using Big O notation, we can conclude that the space complexity of the method is O(N) in which the for loop takes in a user input in order to iterate through the for loop. Recursion also doesn't much affect the complexity in this case. It is merely a means to an end. The situation would be identical if an iterative search was conducted using a conventional loop design.

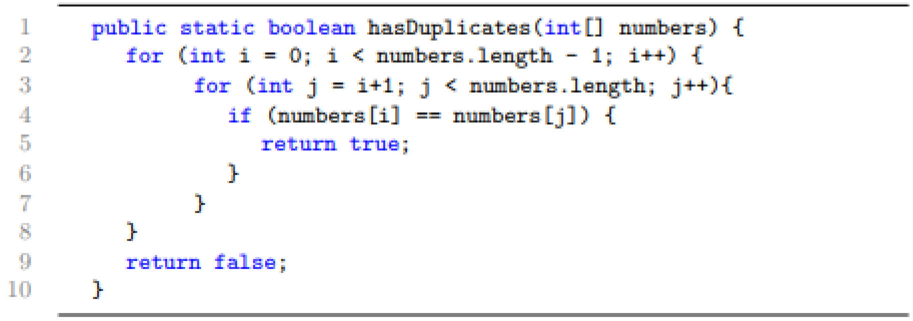
2. (5 points) Consider the following implementation. Using Big O notation, what is the space complexity of this method? Justify your answer.



```
int[] reverseArray(int[] a) {
   int[] result = new int[a.length];
   for (int i = 0; i < a.length; i++) {
      result[a.length - 1 - i] = a[i];
   }
   int[][] 2DArray = new int[a.length][a.length/2];
   // do something with 2DArray
   return result;
}
```

   a. Considering the following implementation of the method reverseArray() above, using Big O notation, we can conclude that the space complexity of the method is O(N^2) in which the for loop counts as O(N) and the instantiation of the array 2DArray also counts as O(N), and since it will take up a.length * a.length / 2, the space complexity will be O(N^2).

3. (3) (10 points) Consider the following implementation of the method hasDuplicates(). Using Big O notation, what is the worst-case running time of this method? Justify your answer.



```
public static boolean hasDuplicates(int[] numbers) {
   for (int i = 0; i < numbers.length - 1; i++) {
         for (int j = i+1; j < numbers.length; j++){
            if (numbers[i] == numbers[j]) {
               return true;
            }
         }
   }
   return false;
}
```

a. Considering the following implementation of the method hasDuplicates() above, using Big O notation, we can conclude that the space complexity is O(N^2), as there are 2 for loops that take in user input, as the outer for loop iterates N times and the inner loop iterates N times, leaving the total space complexity to N*N or O(N^2) times.