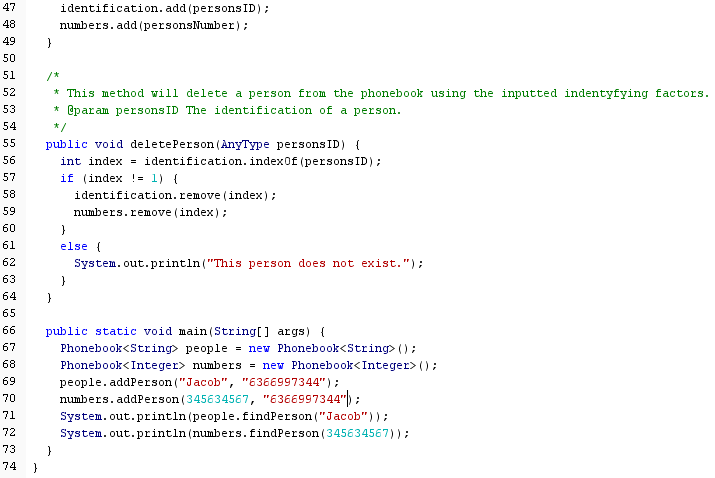
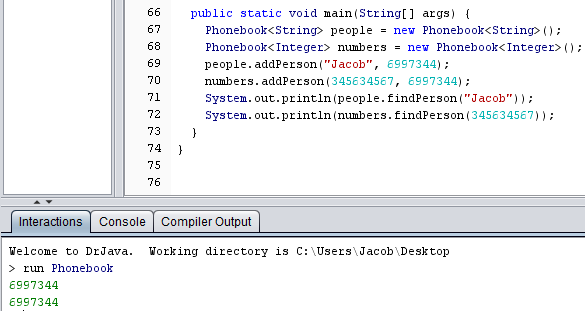
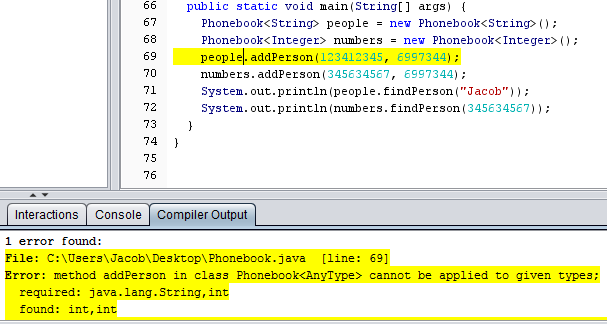
1)



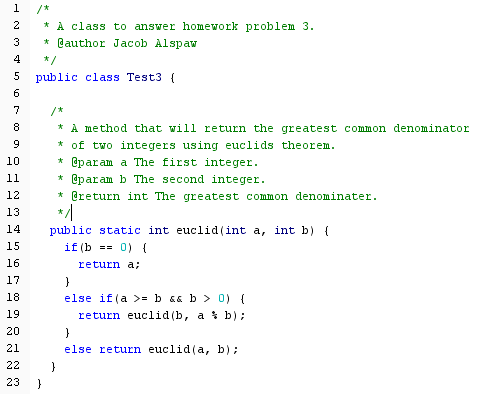
The output in the interaction pane is the correct inputted phone numbers.

The input for the phone book was of incorrect type and produced an error with this attempt.

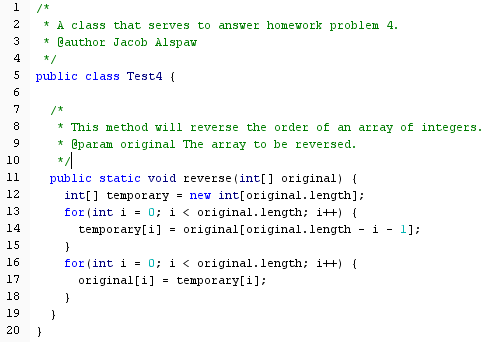


­­­2) Question two will not work as a substitute for the method we discussed in class. The method will never meet the base case requirements of a string length of zero. The minimum length will always be one when a char is found. The recursion will run infinitely, no matter where the char is in the string and, in doing so, will produce stack overflow errors.

3)



4)



5) f(N) = O(g(N)) is equivalent to f(N) < (constant)\*g(N) by the definition of Big-O.

g(N) = O(h(N)) is equivalent to g(N) < (constant)\*h(N) by the definition of Big-O.

Therefore, due to the transitive property, f(N) < (constant)\*h(N). And by definition of Big-O, is equivalent to f(N) = O(h(N)).

6) a. O(sqrt(n)) because sqrt(n) grows much faster.

b. O(log n) because of change of base.

c. O(n3) because n3 grows much faster.

7) a. Keeps track and returns the number of zeroes in an array. The Big-O bound of the worst-case running time is O(n). One for-loop will produce only a “n” number of iterations. The problem size is one iteration through the array. It is a linear size, so the length to the array is the length of problem size.

b. Returns the position of the first instance of a value of zero in an array. The Big-O bound of the worst-case running time is O(n). One for loop will produce only a “n” number of iterations. The problem size is again linear. However, do to variability of the array, the problem size will only be equal to or less than the size of the array.

c. Sorts an array in an order of smallest to largest values. The Big-O bound of the worst-case running time is O(n2). Two for loops running through a designated size will be represented as “n2” in a worst case scenario. The problem size is quadratic. The problem size for this case is the number of iterations for the for-loops multiplied by each other.