1) (10 points) Implement an insertion sort for sorting an array of doubles in increasing order. Do this as follows:
   a) Write a method called `insertItem` that has 2 parameters. The first parameter is an array of doubles called `list` and the second is an integer called `index`. Assume that the first `index - 1` elements of the array are sorted before the method is called. When the method returns, the element at position `index` has been inserted in the array in its correct position so that the first `index` elements are correctly sorted.
   b) Use the `insertItem` method to write a method called `insertionSort` that takes one parameter, an array of doubles, that sorts the array in increasing order.
   c) If the method you wrote takes 1 second to sort an array of size 10,000, how long would you expect it to take to sort and array of size 1,000,000? Show how you got your answer.

2) (10 points) Suppose `student` is a `Student` and `studentList` is an array of `Student`, where `Student` is a class that implements `Comparable`. Write a code segment that returns the position (index) of the `student` in the array, or -1 if it is not in the array. Assume that the array is already sorted in increasing order. Use a binary search if you can. Otherwise, for half credit, you can use a linear search.

3) (10 points) Draw an accurate schematic of the program variables after the following has been executed.
   ```java
   int x = 10;
   int y = 11;
   int z = 12;
   int[] a = {2,4,6,8};
   int[] b = {1,3,5};
   int[] c;
   int[] d;
   c = b;
   d = a;
   a[2] = y;
   b[1] = 16;
   a = new int[2];
   a[0] = z;
   a[1] = z + 1;
   x = d[2];
   y = c[1];
   ```

4) (10 points)
   a) Write a method that takes two parameters, a `char` and an `int`. It prints the first parameter the number of times given by the second parameter.
   b) Use the method from part a) to write a code segment to output a triangular pattern of asterisks like the one below. The first row has no leading blanks and the last row contains 2 asterisks. Each row has two fewer asterisks than the preceding row. The variable `rows` contains the number of rows. For `rows = 5` the pattern looks like:
      
      **********
      *******
      *****
      ****
      **

5) (5 points) Given the following declaration:
   ```java
   String x[10];
   ```
   Assume that the array has been filled appropriately. Write a code segment to interchange the element with index 8 with the one with index 2. Declare any additional variables that you need to use.

6) (10 points)
   a) Write a method that takes one integer parameters and returns `true` if the integer is divisible by 7 and false otherwise.
   b) Write a code segment that declares a 2-dimensional array of `doubles` with 100 rows and 200 columns. Now fill the array so that the element at row `r` and column `c` contains 1 if `r+c` is divisible by 7 and -1 otherwise. Use your method from part a). Note that you can do part b) even if you cannot do part a).
7) (20 points)
   a) (8 points) Write a class called Employee to hold an employee record. An employee will have a last name, a
      first name, an age (in years, an integer) and a weight (in kilograms which might have a fractional part). Write
      the constructor and accessor methods for the fields.
   b) (2 points) Write a code segment to declare an appropriate variable and set it to a new Employee with your
      name, an age of 25 and a weight of 60.3 kilograms.
   c) (5 points) Explain in detail (with words, not code) what changes would be made to have this class implement
      Comparable in which the comparison is by weight.
   d) (5 points) Show the implementation of any new methods that are required for part c).

8) (5 points) Given the following declarations, find the value indicated.
   String s = new String("abcdefghij");
   String t = new String("def");
   String u = new String("deg");
   String v = new String("def");

   a) s.substring(3,5);
   b) s.indexOf(t);
   c) s.indexOf(u);
   d) t == v;
   e) t.equals(v);

9) (10 points)
   a) Write a method, findSmallest, that takes a single array of integers as a parameter and returns the smallest
      value in the array. If the array contains no elements, return -1;
   b) Write a method, findSecondSmallest, that takes a single array of integers as a parameter and returns the
      second smallest value in the array. If the array contains fewer than 2 elements, return -1.
      Note: If the array is sorted, the first function returns the element at index 0 and the second function returns the element
      at index 1, even if both values are identical. You may not do this problem by sorting the array since the methods are not
      allowed to modify the array. Part a) is easy, but part b) is considerably harder.

10) (10 points) The tables below show an array of 8 integers. Fill in the first table with a trace of a selection sort applied to
    this array. Fill in the second table with a trace of an insertion sort applied to this array. In each case the sort will sort the
    element in ascending (increasing) order. Each line in the table should be filled in with the values of the array after one
    pass through the main loop of the sort. There may be more lines in the table than needed.
    An additional copy of the tables appear on the last page if you make a mistake and want to start over.
    
    | Selection Sort Trace | Insertion Sort Trace |
    |-----------------------|----------------------|
    | 17, 11, 31, 23, 14, 32, 19, 43 | 17, 11, 31, 23, 14, 32, 19, 43 |