Assignment overview

Students from the School of Business have been sneaking into the computer labs on the weekends! They have been having wild study parties, microwaving stinky fish dinners and popcorn, leaving their homework scribbled all over the whiteboards, and otherwise making a mess of things. One time they left a book in the mini-fridge!

In response, Stephanie is changing the keycode to the door of the lab. She needs to inform everyone about the new code, but if she simply emails it out, it will be intercepted by the interlopers, who will continue their meddlesome invasions.

Stephanie decides to obscure the new code using a clever method that we programmers can crack, but that will foil the efforts of the troublemakers. The message will include a LONG, scrambled list of keycodes. The new keycode is the smallest number that DOES NOT appear in the list.

You must write a program to determine the new keycode. There are multiple ways this could be done, but here is the way we are choosing to do it:

1. Sort the list of keycodes from Stephanie's message
2. Determine the smallest keycode that is missing from the sorted list

Both of these tasks can be done by Brute Force or Divide/Decrease-and-Conquer algorithms. You can earn partial credit for the Brute Force approach in both cases, but for full marks on the assignment, you must implement Divide/Decrease-and-Conquer algorithms for both parts.

Given code

Find_the_Keycode.java contains some sample Java code that will read a file containing a list of integers and save it into an array. You may use it or not, as you wish. If you have your own favourite way to read an array of integers from a data file, that is fine.

Sample input files

For your testing purposes there are several sample input files shared on Learning Hub. Each contains a different list of numbers and therefore each will give a different answer for the new keycode.

Part 1 (5 marks) – Sort the array

Implement a function that sorts an array of integers. The array should be the only argument, and the function should be of type `void`.

Full marks (5/5) – implement MergeSort.

Partial marks (2/5) – implement any of the other sorting algorithms we have studied: Bubble Sort, Selection Sort, Insertion Sort.

Part 2 (5 marks) – Find the keycode

Implement a function that finds the smallest missing number from a sorted array of integers. The primary function should take only the array as an argument, and will return the smallest missing number (an integer). You will also need to write a recursive "helper" function, which can have other arguments as needed. As inspiration/guidance for the recursive function, think about the algorithms we have seen for binary search and CountKeys.

Full marks (5/5) – implement a divide-and-conquer OR decrease-and-conquer algorithm

Partial marks (2/5) – implement a brute force algorithm

Part 3 (2 marks)

Include a main program that reads a data file and (after you do the processing for Parts 1 and 2) displays the new keycode that is defined by the data file. You may start with the code that is included on Learning Hub, but this is not a requirement.

Part 4 (3 marks)

You must use good programming style throughout your code, including:

- Header comments with at least the following information:
    - Your name, ID, and set
    - A short description of the program (a sentence is enough)
- A descriptive comment for any significant block of code such as a function or an important loop
- Comments for any other code that is "unusual" or otherwise interesting
- Meaningful variable names for important variables

Tips and tricks

You can treat the keycodes as ordinary integers during processing (sorting the list, and finding the smallest missing value). But whenever you *output* a keycode you should make sure it has enough leading zeroes so that it is the same length as all the other keycodes in the data file. For example, if keycodes are 4 characters, then the value 37 should be displayed as "0037". Here's one way to do that (the '4' determines the total length of the output integer):

```
System.out.println(String.format("%04d", keycode));
```

Suggestion: Since Brute Force algorithms are easier to implement, you might wish to implement one for either or both parts of this problem first, then the divide/decrease and conquer algorithms later. One bonus is that you will find out (more quickly) the correct answers from each one of the test data files. Another is that if you run out of time to complete the lab, you will still have earned partial credit.

Submission and marking

Due date: As shown on Learning Hub. Your last submission will be the one that counts. Late assignments will not be graded.

Submit: Your Java source code (file name not important)

Marking: This lab is worth 15 marks.