Lab Practice Week 2

You need to show working versions of your answers to all questions to your tutor. Your tutor will expect to see them by your next session.

What to submit: your answers to exercises 1, 2 and 3.

Note: even though you only need to submit those exercises mentioned above, you should attempt all exercises in each lab practice to help broaden your understanding; this may also help with your assignment work.

Do all the programs in NetBeans IDE.

NOTE: Include internal documentation in your code, if you are not sure, read chapter 2.4 in your textbook and talk to your lecturer

---

The following pseudo-code describes a menu-driven algorithm:

```
loop
   ask the user to input one of the characters a, b, c, d, e, q
   read in a character from the keyboard
   if the character is
      'a' output your name and your tutor's name (hard-coded)
      'b' input 3 double numbers x, y, z and output the largest
          and the smallest of the three numbers
      'c' input 2 integer numbers m and n, and display all the
          numbers between m and n (both inclusive) with five
          numbers per line (note that the last line may have less
          than 5 numbers). At the end, display the sum of all the
          odd numbers between m and n (both inclusive)
      'd' input 3 integer numbers representing the sides of a
          triangle and display the numbers together with a
          message indicating whether or not the numbers form a
          triangle (Note: for the numbers to form a triangle, the
          sum of any two sides must be greater than the third
          side)
      'e' input an integer number n and determine whether or not
          n is a prime number (Note: a number is a prime number
          if it is greater than 1 and has no divisors other than
          1 and itself)
      'q' exit from the loop
      other: output an error message
   end if
end loop
```

Exercises:

Before starting the exercises, make sure you have read the relevant material.

1. Implement the above pseudo-code in a Java program using a **while** loop and a **switch-case** statement. The program should be well structured, and the task performed under each option (at least options 'b' to 'e') should be implemented as a separate method.

   NOTE: The Scanner class does not have a method to input a character. In order to read a character from the keyboard, use one of the following methods (after declaring the Scanner object):

   ```
   static Scanner kb = new Scanner(System.in);

   1. char ch = kb.nextLine().charAt(0); OR
   2. char ch = kb.nextLine().toLowerCase().charAt(0);
   ```

   where kb is a Scanner class object.

   The second method above also converts the input to lowercase, which is often useful. Though these methods allow the user to input more than one character on the input line, the rest of the line (after capturing the first character with charAt(0)) is discarded.

   If you also want to ignore the leading spaces before the first character then use:


   ```
   1. char ch = kb.nextLine().trim().charAt(0); OR
   2. char ch = kb.nextLine().trim().toLowerCase().charAt(0);
   ```

2. Write a program to extend question 1, by adding an option 'f' to your menu program. If this option is chosen, the program should ask the user to enter 10 integers. The program should store each integer into an array (as each one is entered). Once the array contains the 10 integers, the program should output:
   a) The sum of the 10 numbers entered
   b) The average of the 10 numbers entered
   c) The highest number entered
   d) The lowest number entered

3. Write a program that gets a string from the user and reports whether or not there are repeated characters in it.

4. Write a program that gets a string from the user and swaps the first and last characters.