Java exercises
Arrays and String

Exercise 1

Most Frequent Element in an Array

The purpose of this exercise is to write a program that identifies the most frequently occurring number in an array of integers. For example, for the following array:
{2, 7, 5, 6, 7, 1, 6, 2, 1, 7}
your program is supposed to find that the most occurring number is 7 and that its frequency is 3.

Execution Example

It is mandatory that your code respects the following formatting:

```
The most frequent number in the table is:
7 (3 x)
```

There is a new line after the ":" symbol.
Your program must work for any array of integers given as input. You can however suppose that the arrays used for testing are always non-empty. Note that in some cases, in a given array there might be more than one number which occur the most. In such cases, your program must output the number that appears first in the array. E.g., for array tab1 = {2, 7, 5, 6, 7, 1, 6, 2, 1, 7, 6}, 6and 7 are the most occurring numbers (both appear 3 times), but as 7 appears first in the array, the program will only keep 7 as the most frequently occurring element and will display:

```
The most frequent number in the table is:
7 (3 x)
```

Exercise 2 — Encryption

Julius Caesar was using a very simple encoding system, which consists of replacing every letter of a message by a letter placed several ranks after in the alphabetical order. For example, for a shifting of 4, A becomes E, B becomes F, up to Z which becomes D.
Here we want to apply this technique to encode one string. For this you will write a program which applies the processing described below.

Encoding

The given program for this exercise provides you with a string ALPHABET containing all the letters of the Latin alphabet, in lower case.
It asks the user to enter a string s.
The purpose of this exercise is to construct a new string toEncode as follows:

toEncode contains, following their initial order, all the characters of s which are present in ALPHABET or that correspond to the space character (' '). All the other characters are ignored.

The string toEncode will be encoded using the technique of Julius Caesar. It has to be displayed. If it does not contain any character, a message indicating that the string to encode is empty will be displayed.
As long as the string to encode is non-empty, your program should encode it with a fixed shifting given by the provided constant SHIFT.
Encode the string toEncode by replacing every character with the character placed SHIFT letters further in the string ALPHABET, and this in a circular way ('z' will be replaced by 'd' for example). The spaces will be maintained as they are. They are not object of any encoding but will stay present at their position.
This encoded string will be displayed.
The execution of your program should strictly conform to the following examples:

```
Please enter a string of characters:
run away
The initial chain was: 'run away'
The string to encode is: 'run away'
The coded string is: 'jycid qererxw'
Please enter a string of characters:
did you see my 3 cats and 2 dogs?
The initial chain was: 'did you see my 3 cats and my 2 dogs
The string to encode is: 'did you see my cats and my dogs'
The coded string is: 'ezidzsyw zy qiw glexw ix qiw glmirw'
Please enter a string of characters:
93589 () çç% & + = 12AD
The initial string was: '93589 () çç% & + = 12AD'
The string to encode is empty.
```

You can start your program like below:

```
static final String ALPHABET = "abcdefghijklmnopqrstuvwxyz";
    static final int SHIFT = 4;

    static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {

        System.out.println("Please insert a string of characters : ");
        String s = scanner.nextLine();

        // the string to encode
        String toEncode = "";
        // the coded string
        String chainCoded = "";

        /*****************************************
         *****************************************/

            // write your program in this between

        /*****************************************
         *****************************************/

        System.out.format("the initial chain was : '%s'\n", s);

        if (toEncode.isEmpty()) {
            System.out.println("the string to encode is empty.\n");
        } else {
            System.out.format("the string to encode is : '%s'\n", toEncode);
            System.out.format("the coded string is : '%s'\n", chainCode);
        }
    }
```