

## EXERCISE 2.

Elaborado por Victor López

---

### Problema.

Using `first_names.txt` (text file containing over five-thousand first names), begin by sorting it into alphabetical order. Then workout the alphabetical value for each name, multiply this value by its alphabetical position to obtain a name score.

Example: assuming sorted and COLIN to be the 938<sup>th</sup> name it is worth:  $3+15+12+9+14$ . So, the score would be  $938*53 = 49,714$ .

What is the total of all the name scores in the file?

### Paso a paso.

Para resolver este problema, seguí los siguientes pasos:

1. Leer el archivo de texto y almacenar los nombres en un array.
2. Ordenar los nombres en orden alfabético.
3. Calcular el valor alfabético de cada nombre.
4. Multiplicar el valor alfabético de cada nombre por su posición en el array.
5. Sumar los valores obtenidos en el paso anterior para obtener la puntuación total.

Con esto en mente, procedí a escribir el código siguiendo estos pasos:

1. Cree un método `readNamesFromFile` que reciba el nombre del archivo de texto como parámetro. Este método lee el archivo de texto y almacena los nombres en un array. En este método también eliminé las comillas que rodean a cada nombre en el archivo de texto y separé los nombres por comas para que cada nombre esté en su propia posición del array.
  2. Una vez que tengo el array de nombres, lo ordeno alfabéticamente utilizando el método `Arrays.sort`.
  3. Para calcular el valor alfabético de cada nombre, creé un método `calculateNameScore` que toma un nombre y su posición en el array como parámetros. Este método itera a través de cada letra del nombre y suma su valor alfabético, que es simplemente el valor ASCII de la letra menos 64. El resultado se multiplica por la posición del nombre en el array.
-

- 
4. Finalmente, itero a través del array de nombres y para cada nombre, llamo al método `calculateNameScore` para obtener su puntuación y luego sumo todas las puntuaciones para obtener la puntuación total.

*Total score of all the names in the file: **872835588***

## Código.

```
import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

import java.util.Arrays;

public class exercise2 {

    public static void main(String[] args) {

        String fileName = "first_names.txt"; // name of the file containing
the first names

        String[] names = readNamesFromFile(fileName); // read the names
from the file and store them in an array

        Arrays.sort(names); // sort the names in alphabetical order

        int totalScore = 0; // variable to store the total score

        for (int i = 0; i < names.length; i++) {

            int nameScore = calculateNameScore(names[i], i + 1); //
calculate the score for each name and add it to the total score

            totalScore += nameScore;

        }

    }

}
```

```

        System.out.println("Total score of all the names in the file: " +
totalScore);

    }

    /**
     * Reads the first names from the given file and returns them in an
array
     * @param fileName name of the file containing the first names
     * @return array of first names read from the file
     */
    private static String[] readNamesFromFile(String fileName) {

        String[] names = new String[0];

        try (BufferedReader br = new BufferedReader(new
FileReader(fileName))) {

            String line;

            while ((line = br.readLine()) != null) {

                // remove the quotes and split the line into names
separated by commas

                String[] splitNames = line.replace("\"", "").split(",");

                // add each name to the array

                names = Arrays.copyOf(names, names.length +
splitNames.length);

                System.arraycopy(splitNames, 0, names, names.length -
splitNames.length, splitNames.length);

            }

        } catch (IOException e) {

```

---

```
        e.printStackTrace();

    }

    return names;

}

/**
 * Calculates the score for the given name
 *
 * @param name name for which to calculate the score
 * @param position position of the name in the sorted array of names
 * @return score for the given name
 */
private static int calculateNameScore(String name, int position) {

    int score = 0;

    for (int i = 0; i < name.length(); i++) {

        char c = name.charAt(i);

        score += (int) c - 64; // calculate the alphabetical value of
each character

    }

    return score * position; // multiply the score by the position of
the name

}

}
```