Introduction to Programming - Lab Exercises for Week 16 (Trimester 2, week 3)

Some Exercises using Arrays

Some exercises with Arrays of String

1. Write a program which checks if a username which is entered exists in an array of users.

   Your program should create an array, for example:

   ```
   String[] theUsers = {"Anne", "Tom", "Dick", "Kate", "Harry"};
   ```

   It should then read in a name from the keyboard, and use a method to output a message which says whether the name entered is in the array. To help you, here is a method that would work for an array of **`int`**:

   ```
   public static void find(int[] list, int target) {
      for (int i=0; i<list.length; i++) {
            if (target == list[i]) {
                  TextIO.putln("Have found " + target +
                             " at index " + i);
            }
      }
   }
   ```

   The code can be improved - to see how, what would be the output if the same target were present twice in the array (in our username example we are going to assume that the same name cannot be present more than once)?

   Why does the code need to be changed (apart from replacing **`int`** with `String`, of course, in the parameters) for `String`?

2. Modify the program to replace your static method with a new one, named `indexOf()`, that takes the array and the name to check as parameters and returns the index position in the array at which the name occurs. The method should return -1 if the name is not present, and it should not output anything to the screen.

   Modify your main() method to call the new method to check whether the name is present and display a message that states whether the name is present.

3. Extend the username check program above.

Write a program which maintains an array of usernames. It should repeatedly show a menu with the following options:

- Read in a name and append it to the array (but do not allow duplicate names, each name should be unique)
- Read in a name and check whether it is in the array (as 2 above)
- Read in a name and delete it from the array if it exists. This should not leave a “hole” in the array – the names should continue to occupy consecutive positions in the array – but the order in which the names occur in the list is not important
- Allow the user to change their username by reading in a name and changing it to a new name if it exists
- Exit the program

The program should make appropriate use of methods (question 2 will already have provided you with one of these).

An exercise with a 2D array

4. Write a body for the method below that adds two matrices and returns the result.

```java
public static double[][] add(double[][] first, double[][] second)
```

Here is an example of matrix addition:

$$\begin{vmatrix} 0.5 & 1.0 & 0.0 \\ 0.0 & 0.5 & 1.0 \\ 1.0 & 1.0 & 1.5 \end{vmatrix} + \begin{vmatrix} 0.5 & 0.0 & 1.0 \\ 0.5 & 0.5 & 1.0 \\ 1.0 & 2.0 & 1.5 \end{vmatrix} = \begin{vmatrix} 1.0 & 1.0 & 1.0 \\ 0.5 & 1.0 & 2.0 \\ 2.0 & 3.0 & 3.0 \end{vmatrix}$$

That is, for each element [i,j] in the first matrix, you add its value to the corresponding element [i,j] in the second matrix. You can only add two matrices if they are the same size (in this case, 3x3), for this exercise you can assume that they are (but do not assume that they are always 3x3) and that the matrices are square or rectangular (i.e. that all the rows are the same length).

Write a second method that takes a matrix as a parameter and displays it row by row, with the numbers lined up in columns (each number should take up a six characters in . You can assume that the matrices that the method will be passed are not so large that a row of the matrix occupies more than one row on the display.

Write a main() method to create and initialize two matrices and call your method to add them, and then display the two matrices being added and their result. For example, the output might be:

```
0.50  1.00  0.00
0.00  0.50  1.00
1.00  1.00  1.50
 +
0.50  0.00  1.00
0.50  0.50  1.00
0.00  2.00  1.50
 =
1.00  1.00  1.00
0.50  1.00  2.00
2.00  3.00  3.00
```