Lab 3
(Arrays , ArrayLists, Iterators , And 2048 Helper)

CSC 172 (Data Structures and Algorithms)
University of Rochester
Due Date: 02/06 11:59 pm

Introduction

Every student must hand in their own work, but every student must list the name of their lab partner on all labs.

This lab has three sections. If you have a partner and you both end up using a portion of this code in your **Project 1**, you need to state that in the README file.

Objectives

Explore Array and ArrayList in more details. Learn to work with iterators. This lab has some elements common to Project 1 (Yes, that's intentional!).

Task 1

Implement a function `print2Darray(int[][] array)` to print a formatted 4x4 two dimensional integer array. When the array contains {{10, 15, 30, 40},{15, 5, 8, 2}, {20, 2, 4, 2},{1, 4, 5, 0}}, Your output should look like:

```
10          15          30          40

15          5           8           2

20          2           4           2

1           4           5           0
```

Now, implement another function `print2DList(ArrayList<ArrayList<Integer>> list)` to print a formatted 2D list.

Task 2

Implement a method/function that produces running sum

`runningSum2DArray(int[][] array, int dir)` across rows (left to right or right to left) or columns (top to bottom or bottom to top)

Input to the method: A 4x4 two dimensional *int* array and an integer (*1*, *2*, *3* or *4* for *left*, *right*, *up*, *down* respectively).

Output: The modified array after producing the running sums according to the direction.

For example: If the input to the method is the same as the earlier array, and if the direction is *2* (*right*), the output array would be printed as:

```
10          25          55          95

15          20          28          30

20          22          26          28

1           5           10          10
```

Now, implement another function `runningSum2DArrayList(ArrayList<ArrayList< Integer > > list, int dir)` that performs the same functionality.

Task 3 (Iterate over an ArrayList)

In Java, you can iterate an *ArrayList* in different ways. Write the following methods to print Integers in an ArrayList iterating in different ways:

1.
```
// Using basic while / for loop
        void printArrayListBasicLoop(ArrayList<Integer> al);
```

2.
```
// Using enhanced for loop (:)
        void printArrayListEnhancedLoop(ArrayList<Integer> al);
```

3.
```
// Using basic for loop with iterator
        void printArrayListForLoopListIterator(ArrayList<Integer> al);
```

4.
```
// Using basic while loop with iterator
        void  printArrayListWhileLoopListIterator(ArrayList<Integer>
        al);
```

Note: You can find related code from: https://memorynotfound.com/iterate-arraylist-java/. The purpose of this exercise is to make you familiar with iterators and how to use them.

Submission

Hand in the source code from this lab at the appropriate location on the Blackboard system at `learn.rochester.edu`. You should hand in a single zip (compressed archive) **`.zip`** containing your source code and README files, as described below. Your source code should include three Java files: `Lab3Task1.java, Lab3Task2.java, Lab3Task3.java`.

1. A plain text file named README that includes your contact information, your partner's name, a brief explanation of the lab (a one paragraph synopsis. Include information identifying what class and lab number your files represent.), and one sentence explaining the contents of any other files you hand in.

2. All source code files should contain author and partner identification in the comments at the top of the file. For Task 2, you can reuse methods from Task 1.

Testing

1. Ensure that you have three files named 'Lab3Task1.java' 'Lab3Task2.java' 'Lab3Task3.java'.

2. Ensure your program prints output in the format specified in the provided `*.ans` files attached to the post. Do not write any additional print statements.

* *Important note about input/output*: You can print the output on a file or on the standard output. Same applies to input. Explain the chosen method in your README file.

3. Create your zip directory (also used for submission) by selecting the necessary files (java files + README in this case) and compressing into your zip file. The file should be named [YourNetID]_Lab3.zip (replacing the filler with your actual NetID). There should be no other files or subdirectories in the zip folder - only those required by the lab.

Grading **(10 pts)**

Task 1:	2 pts
Task 2:	4 pts
Task 3:	4 pts

Notes:

All labs are open book. You can get code snippets from the internet if you want (make sure you cite those properly). But that is not the purpose. We want you to think about an algorithm, and then implement it together with your partner.