# Hands-on Experiment # 10 : Worksheet (2019)

Section\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

No more than 3 students per one submission of this worksheet.

Student ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Part A: Getting Familiar with Writing Recursive Methods

1. Consider the following recursive definition of “Logistic Map” and write a java program “LogisticMapRecursion.java”, where *double x(int n)* is a recursion method as follows:

Find the value of *f(n)* for all values of *n* listed in the table below. Please print the output with 4 decimal points.

|  |  |
| --- | --- |
| *n* | *f(n)* |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |

List all your source code below.

## Part B: Thinking Recursively

We will be writing recursive methods that operate on arrays. To do so, we imagine an array to consist of

* Head: a data in the first slot.
* Tail: an array containing all but the data in the first slot.

For example: {1,2,3,4,5} has head = 1, tail = {2,3,4,5}.

Operations that allow us to write arrays codes in recursive ways are written in class MyArrayUtil. Please read methods in class MyArrayUtil. (We will ignore loops in MyArrayUtil).

A class called RecursiveExercise (with main method to test your other methods) is given for this question. Then, inside RecursiveExercise.java, do the followings:

1. Write a recursive method **public** **static** **boolean** isIn(**int** x, **int**[] a) **throws** Exception
   1. This method returns true if x is stored inside array a, and false otherwise.
   2. If a is null, this method returns false.

List your source code below:

1. Write a recursive method **public** **static** **boolean** subArray(**int**[] a1, **int**[] a2) **throws** Exception .
   1. This method returns true if all data in a1 are also in a2 (the method returns true if a1 == null), and false otherwise.

For example:

subArray({1,3,4,5}, {0,1,2,3,4,5,6,7}) returns true.

subArray({1,3,4,5}, {1,2,3,5}) returns false.

List your source code below:

1. Write a recursive method **public** **static** **int**[] reverse(**int**[] ht) **throws** Exception .
   1. This method returns an array that orders data in reverse of ht. For example, if ht = {1,2,3,4,5}, then this method will return {5,4,3,2,1}.
   2. This method returns null if ht is null.

List your source code below:

1. Write a recursive method **public** **static** **int**[] shift(**int**[] ht, **int** n) **throws** Exception
   1. This method returns an array that stores the same data as ht, but the first n data are moved to the back of the array.

For example, shift({1,2,3,4,5}, 3) will return {4,5,1,2,3}.

shift({1,2,3,4,5}, 2) will return {3,4,5,1,2}.

* 1. If ht is null, this method returns null.

List your source code below:

## Part C: Experimenting with ArrayList

Read **ArrayList.docx** and complete the exercise.

Update the program “SchoolLottery.java” (Application in ArrayList) in order to obtain the list from file instead of keyboard. Also, a name in the list cannot be duplicated.

List your source code below:

Submit this worksheet (by only one member of the group) via <http://www.myCourseVille.com> (Assignments > Hands-on Experiment # 10) before noon of the day after your lecture.