King Fahd University of Petroleum and Minerals

###### Information and Computer Science Department

**ICS 102: Introduction to Computer Programming**

**Second Semester 2017-2018 (Term 172)  
Homework No. 3  
[Posted: Tuesday 13th March 2018]  
[Due Date: Sunday 1st April 2018 @ 11:59 PM (Before Midnight)]**

**Submission Guidelines:**

Submit a zipped file containing the following files:

* Q1.java (Java source file) containing your answer to the programming question no. 1.
* Q2.java (Java source file) containing your answer to the programming question no. 2.
* Q3.java (Java source file) containing your answer to the programming question no. 3.

PLEASE DO NOT INCLUDE .class FILES IN YOUR SUBMISSION

The zipped file should be named as follows:

**HW\_3\_XXXXXXXXX\_YourFamilyName\_Lecture\_Section\_No.zip**

where:

XXXXXXXXX is your 9 digit KFUPM ID.

YourFamilyName is your family name

Lecture\_Section\_No is the number of your ICS 102 lecture section

Submission should be made through your ICS 102 Lecture section Blackboard course page under **HW\_3 Assignment** submission link.

**Important Notes:**

* **Cheating is taken seriously**. Any cheating attempt will result in an F grade in the course.
* **EACH STUDENT IS REQUIRED TO DO THE HOMEWORK ALONE**. COPYING FROM ANY SOURCE IS REGARDED AS CHEATING.
* **No late submissions are allowed**.
* **Submissions via email are not accepted and will be simply ignored**.
* Submission of the homework solution should be in a zipped filed with the format specified above. **Any different formatting/naming will result in reducing the total homework score by half!**
* **You must use proper indentation and meaningful variable names in your programs.**

**Question 01**: An n-digit positive integer **d1d2d3 . . . dn-1 dn** satisfies **property A** if the sum of the nth powers of its digits is equal to the number itself:

**d1d2d3 . . . dn-1 dn = d1n + d2n + d3n + . . . + dn-1 n + dnn**

Examples of integers satisfying property A:

**6 = 61**

**1634**  = **14 + 64 + 34 + 44**

**153 = 13 + 53 + 33**

Write a Java program consisting of the **main** method and a private static **boolean** method that takes an integer and it returns **true** if the integer satisfies **property A**; otherwise it returns **false**. The **main** method prompts for and reads an integer, it then determines whether the integer satisfies **property A** or not by using the **boolean** method.

Note:

* The **boolean** method must throw **java.lang.IllegalArgumentException** if it is passed an integer less than one.
* The **main** method must recover from both **java.util.InputMismatchException** and **java.lang.IllegalArgumentException** by looping until a valid value is entered.
* The behaviour of your program must be similar to the program runs below:

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

**Question 02:** A date in the form **dd mm yyyy** can be converted to a decimal form **yyyy.xxxxx** where the decimal part **.xxxxx** is calculated as  **(NumberOfDaysFromFirstJanuaryOfTheYearToTheDate – 1) / NumberOfDaysForTheYear**

Where **NumberOfDaysForTheYear** is **365.0** for a non-leap year, and **366.0** for a leap year

Examples: 1 1 2018 is 2018 + (1 – 1) / 365.0 = 2018.000000

3 2 2017 is 2017 + (34 – 1) / 365.0 = 2017.0904109

5 3 2020 is 2020 + (65 – 1) / 366.0 = 2020.17486338 [Note: 2020 is a leap year]

Write an interactive Java program to compute and display the decimal form of a date for a date that is input at the keyboard in the form **dd mm yyyy**.

Your program must check for input validity. It must display an appropriate error message and loop as long as the three inputs for date (day, month, and year) do not form a valid date. Your program must also recover from **InputMismatchException**.

In addition to the **main** method your program must use the following private static methods:

* A **void** method that displays the following message to the user:

"This program computes and displays the decimal form of a date input as: day month year."

* A **boolean** method that tests for input validity. The method returns **true** if the three inputs form a valid date; otherwise it returns **false**. This function must not have calls to **input** or **output** methods.
* A **boolean** method that returns **true** if its parameter year is a leap year; otherwise it returns **false** (A leap year is a year that has 366 days). This method must not have calls to **input** or **output** methods.
* An **int** method that returns the number of days from 1st January to the date. This method must not have calls to **input** or **output** methods.
* A method of type double that returns the decimal form of a date. This method must not have calls to **input** or **output** methods.

Note:

* A year is a leap year if it is divisible by 400; otherwise if it is divisible by 4 **and** not divisible by 100.
* Assume a year is between 1900 and 3000 inclusive, i.e., **1900 ≤ year ≤ 3000**
* Your program must be general.
* YOUR PROGRAM MUST NOT USE Java Date, Time, or Calendar APIs.
* Your program must behave as the sample program runs below.

Sample program runs:

|  |
| --- |
|  |
|  |
|  |

**Question 03**: Write a Java **main** method that calls a method **readArray** two times to read integer values in two integer arrays each of size input in the **readArray** method. Your **readArray** method must validate the size of an array by looping until a size that is not negative is entered. Assume the values read in each array are distinct [i.e., in each array there are no duplicate values. DON’T TEST FOR THIS] and that there are no **InputMismatchExceptions**. The **main** method then passes the two array references to a method **getSetUnion** which returns a reference to an integer array that contains the union of the two arrays. The **main** method finally displays the union.

Note:

* Your program must be general.
* The returned set union must have distinct elements.
* The method **getSetUnion** must not contain calls to input or output methods.
* The behaviour of your program must be similar to the program runs below:

|  |
| --- |
|  |
|  |
|  |
|  |
|  |