CS 212 **DUE: Tuesday, April 30 at 11:59 PM**
Spring 2019
Assignment 7
4/23/2019 **10 points**

Introduction: This assignment will give you practice working with relations.

This is an individual effort homework assignment. You must write up your solutions in LaTeX. Use the `a7.tex` template that I provide and be sure to replace each "Put your answer for ___ here." with your answers but leave everything else alone. Your solutions must be written up in a clear, concise and rigorous manner.

When you are done, zip up your .TEX file and corresponding .PDF file. Upload your .ZIP file to the **a7** dropbox on d2l. After you have uploaded the file, double-check to ensure your file was uploaded correctly. It is your responsibility to ensure your submission was done correctly. Assignments that are not uploaded correctly are worth 0 points.

Some Java code is given on the next page...

```
public static String tomorrow(int month, int day, int year) {
   if (day == getLastDayOfMonth(month, year)) {
      if (month == 12) {                              // 1
         year++;
         day = month = 1;
      }
      else {                                          // 2
         day = 1;
         month++;
      }
   }
   else                                               // 3
      day++;
   return month + "/" + day + "/" + year;
}

private static int getLastDayOfMonth(int month, int year) {
   switch (month)
   {
      case 4:
      case 6:
      case 9:
      case 11:
         return 30;                                   // 4
      case 2:
         if (isLeapYear(year))
            return 29;                                // 5
         else
            return 28;                                // 6
      default:
         return 31;                                   // 7
   }
}

private static boolean isLeapYear(int year) {
   if ((year % 4) == 0) {
      if ((year % 100) == 0)
         if ((year % 400) == 0)
            return true;                              // 8
         else
            return false;                             // 9
      else
         return true;                                 // 10
   }
   else
      return false;                                   // 11
}
```

The `tomorrow` method takes as its parameters a date, represented by three `int`s: `month`, `day` and `year`. Then, the `String`-representation of the next date is returned. Note that `tomorrow` is defined in terms of the helper methods `getLastDayOfMonth` and `isLeapYear`.

Is `tomorrow` correct? We could try every possible valid date, but this is overkill. Therefore, in order to study the correctness of `tomorrow`, we will define a special relation that will hopefully help us reduce the number of test cases to consider.

Let D be the set of all valid dates, starting with the date "1/1/0", where dates are assumed to consist of three integers: *m*onth, *d*ay and *y*ear. So, $D = \{$"1/1/0", "1/2/0", $\ldots\}$. The year of a valid date is any non-negative `int`eger value.

Assume that "$m_1/d_1/y_1$" and "$m_2/d_2/y_2$" are both elements of D. Define the relation R over $D \times D$, as follows: "$m_1/d_1/y_1$" and "$m_2/d_2/y_2$" are related if and only if `tomorrow` follows the same execution path when called on "$m_1/d_1/y_1$" as it does when called on "$m_2/d_2/y_2$".

For example...

...for the dates "7/6/1982" and "10/23/2009", `tomorrow` follows the same execution path, namely the path `7 3`. However, `tomorrow` does not follow the same execution paths for the dates "2/28/2012" and "2/28/2100". In fact, the former makes `tomorrow` follow the path `10 5 3`, whereas the latter makes `tomorrow` follow the path `9 6 2`

The questions begin on the next page.

1. (5 points) Show that R is an equivalence relation.

Equivalence relations have 3 properties, reflexivity, symmetry, and transitivity. We must prove each of these 3 properties in order to prove that R is an equivalence relation:

1. Reflexivity

   Let $d \in D$. We know that two dates are related if they follow the same execution path. This means that the dates are rolled over/processed in the same way. This means that if the method is given two equal dates, that is, if this method is given equal inputs, then the method should follow the same execution path. This means that $(\forall d \in D)((d, d) \in R)$ Therefore, R is reflexive.

2. Symmetry

   Let $d_1, d_2 \in D$. Now assume that $(d_1, d_2) \in R$. By the definition of R, (d_1, d_2) can only be in R if they *both* follow the same execution path. Therefore if d_1 has the same execution path as d_2, then d_2 has the same execution path as d_1. Therefore, if $(d_1, d_2) \in R$, then $(d_2, d_1) \in R$, which means that R is symmetric.

3. Transitivity

   Let $d_1, d_2, d_3 \in D$. Let's assume that $(d_1, d_2) \in R$ and $(d_2, d_3) \in R$. If d_1 and d_2 follow the same execution path P, and d_2 and d_3 also follow the same execution path P, then since d_1 and d_3 both follow the same execution path P, then $(d_1, d_3) \in R$. Therefore, R is transitive.

Since R is reflexive, symmetric, and transitive, then R is an equivalence relation. $\square$

2. (5 points) Since R is an equivalence relation, it partitions D into equivalence classes. In a very simplistic model of testing, we should test `tomorrow` on one representative from each equivalence class. The idea is that, since all the elements within an equivalence class are equivalent in the sense that they all force `tomorrow` down the same execution path, it doesn't pay to test more than one element from each equivalence class. So, we need to study the various equivalence classes.

   For example, the dates "7/6/1982" and "10/23/2009" are in the same equivalence class. Namely, they belong to the equivalence class comprised of valid dates that force `tomorrow` down the execution path `7 3`. However, rather than identify each equivalence class by a sequence of numbers, we will describe each equivalence class using a precise yet concise English statement. So, instead of `7 3`, we would say something like, "the set of dates that fall in the middle of a month that is comprised of 31 days".

   List at least five more equivalence classes and in a similar fashion to the one that I just gave.

   1. `7 1`: The set of all dates at the very end of a given year within the domain
   2. `7 2`: The set of all dates at the very end of the month of January, March, May, July, August, or October (the months containing exactly 31 days)
   3. `4 2`: The set of all dates at the very end of the month of April, June, September, or November (the months containing exactly 30 days)
   4. `4 3`: The set of all dates not at the very end of the month of April, June, September, or November (the months containing exactly 30 days)
   5. `8 5 2`: The set of all dates $2/29/y$ where y is a given year within the domain that is divisible by 400