Computer Science 312
Principles of Programming Languages
Spring 2018
Assignment 4

Due: 11:59 p.m., Friday, 3/16/18

Overview

For this assignment, you will write a program in Pascal that works with dates.

Description

You must submit a single source file, `date.pas`, which contains all of your code. You may also want to name your program `dt` to avoid conflict with the Pascal `Date` function. In your file, you must provide the following procedures/functions with the given interfaces:

`procedure init_date (var dt : date_t; day : day_range; month : month_range; year : integer);`
- initializes date with the day, month, and year parameters

`procedure init_date1 (var dt : date_t);`
- initializes date with the current date
- declare variables for month, day, and year (of type **`word`**) and call
  `DeCodeDate (Date, year, month, day);`

`function date_equal (date1 : date_t; date2 : date_t) : boolean;`
- compares two dates and returns true if they're equal

`function date_less_than (date1 : date_t; date2 : date_t) : boolean;`
- compares two dates and returns true if date1 is less than date2

`function month_str (month : month_range) : string;`
- returns string name corresponding to month number

`procedure format_date (date : date_t; var ret_str : string);`
- formats a date into a 'month day, year' format (e.g. March 5, 2018)

`procedure next_day (var date : date_t);`
- increments the current date by one day Note: includes the following nested functions

`function leap_year (year : integer) : boolean;`
- returns true if year is a leap year

`function month_length (month: month_range; leap: boolean): day_range;`
- returns the number of days in month

In **main**, define three variables as follows:

```
d1, d2, d3 : date_t;
format_str : string;
```

Initialize **d1** with the init_date1 procedure. Initialize **d2** and **d3** using init_date with the values for December 31, 1999 and January 1, 2000, respectively.

Use the functions listed above for setting, incrementing, and comparing the dates, as indicated below. Use **writeln** for printing to the screen and add tags and blank lines to make the output more readable.

Your final output should look **exactly** like this:

```
d1: March 5, 2018
d2: December 31, 1999
d3: January 1, 2000

d1 < d2? FALSE
d2 < d3? TRUE

next day d2: January 1, 2000
d2 = d3? TRUE

next day d2: January 2, 2000
d2 < d3? FALSE
d2 = d3? FALSE
d2 > d3? TRUE

initialized d1 to February 28, 1529
next day d1: March 1, 1529

initialized d1 to February 28, 1460
next day d1: February 29, 1460

initialized d1 to February 28, 1700
next day d1: March 1, 1700

initialized d1 to February 28, 1600
next day d1: February 29, 1600
```

Use the online Pascal compiler at **`rextester.com`** to compile and run your code (be sure to change the language to Pascal on the drop-down menu). Maintain your code in a local text editor to ease saving and simply copy & paste to the online text area to compile and run.