Instructions

- The program described in sections I and II of this handout should be submitted through Sakai’s Assignment link by 5pm Monday as a single source code file.
- Program files submitted through the Sakai Assignment page must use the following naming format:
  - Lastname_Firstinitial_Lab_lab#.py
  - For example, for Lab #5, my file would be named: *Thomas_S_Lab_5.py*
- All programming assignments must follow the Style Guide and include meaningful comments.

I. On the Gregorian calendar a leap year has 366 days. A year is a **leap year** if it is divisible by 4, unless it is a century year (a multiple of 100) that is not divisible by 400. Examples: 1900 was not a leap year; 2000 was a leap year; 1776 was a leap year. Complete the following program that reads a year value greater than 1582 and determines whether or not that year is a leap year. The bulk of the program is given. Your task is to complete the program by completing the function named **isLeapYear**. You don’t need to make changes below the dashed line.

```
def isLeapYear( year ):
    “””returns True if year is a leap year, otherwise False”””
    # your changes go here




# ------------------------------------------------------------
# Get the input
year = eval(input("Type a year after 1582:"))

# Report the result
if isLeapYear( year ) == True:
    print(year, "is a leap year.")
else:      # can only get here if it’s not a leap year
    print(year, "is not a leap year.")
```

Scoring Rubric:
5 points – your function correctly determines if a year is a leap year
5 points – your function returns True/False as appropriate

II. The Easter holiday observed in Western countries is a *movable feast*, meaning that it occurs on different dates in different years. The First Council of Nicaea of 325AD established the date of Easter as the first Sunday after the full moon following the March equinox. That still doesn’t tell us the month and day. The month and day can be calculated by the following algorithm published in *Nature* in 1876. Note that “/” in this text is not “/” in Python.

This algorithm holds for any year in the Gregorian Calendar, which (of course) means years including and after 1583.

In the text below, / represents an integer division neglecting the remainder, while % is division keeping only the remainder. So 30/7=4 , and 30%7=2 .

```
 a=year%19
b=year/100
c=year%100
d=b/4
e=b%4
f=(b+8)/25
g=(b-f+1)/3
h=(19*a+b-d-g+15)%30
i=c/4
k=c%4
l=(32+2*e+2*i-h-k)%7
m=(a+11*h+22*l)/451
Easter Month =(h+l-7*m+114)/31
p=(h+l-7*m+114)%31
Easter Date=p+1     (date within Easter Month)
```

In this part of the lab you will add two functions to the program from part I of lab. The first function should take a year as a parameter and compute and return the month in which Easter falls in that year. The second function should take a year as a parameter and compute and return the day of the month in which Easter falls in that year. Add to the program an additional print() statement so that the program will report its results like the following examples. [NOTE: print(value, end=””) prevents Python from moving to a new line after it prints *value*]

```
The date of Easter in 2014 is 4 / 20 / 2014. 2014 is not a leap year.
The date of Easter in 1961 is 4 / 2 / 1961. 1961 is not a leap year.
The date of Easter in 2013 is 3 / 31 / 2013. 2013 is not a leap year.
The date of Easter in 2000 is 4 / 23 / 2000. 2000 is a leap year.
```

Be sure to include comments and appropriate documentation in your program. Upload your complete program to Sakai as Lastname_Initial_Lab_5.py

Scoring Rubric:
4 points – function correctly determines the month of Easter for a year
4 points – function correctly determines the day of the month of Easter for a year
4 points – prints out a well-formatted result in the format shown above
3 points – the functions you write are documented properly with comments