Python – Leap Year

Purpose

This lab was designed to teach you how to read data from a user, process the data via function calls and output the result.

Description

Write a Python function is_leap_year that takes as input the year as a parameter and returns True if year (an integer) is a leap year according to the Gregorian calendar and False otherwise.

```
# leap year algorithm

if (year is not divisible by 4) then (it is a common year)
else if (year is not divisible by 100) then (it is a leap year)
else if (year is not divisible by 400) then (it is a common year)
else (it is a leap year)[1]

# the logic above can be done w/ branching structure or in one line with Boolean logic
```

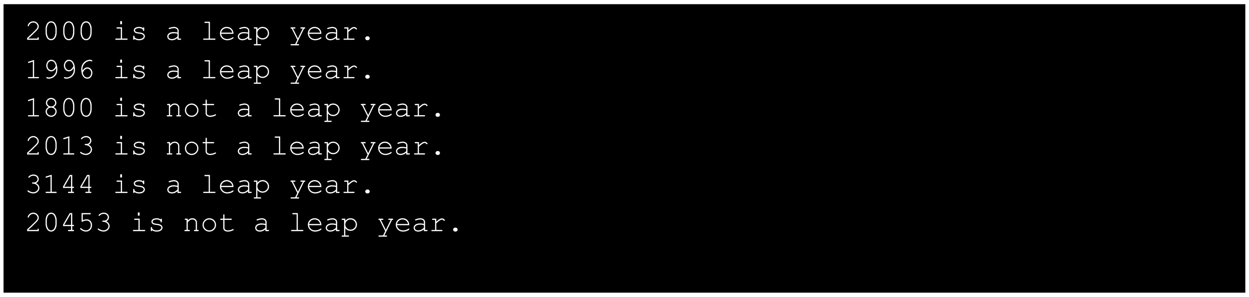
Program Shell

leap_year.py provided for you

Sample Data

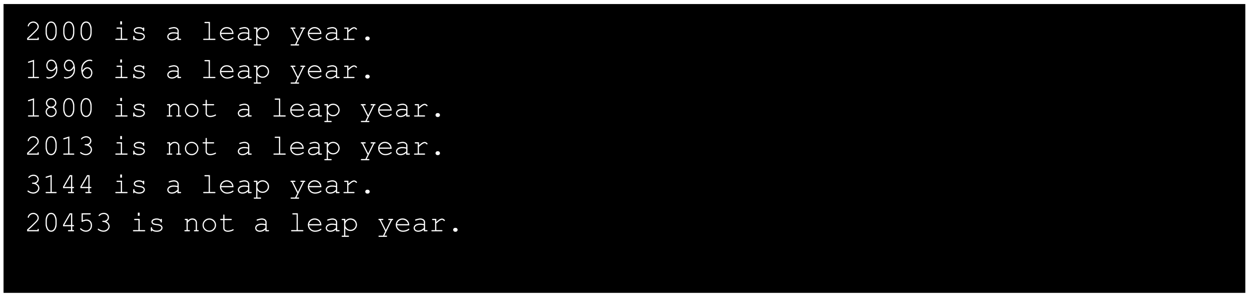
```
2000
1996
1800
2013
3144
20453
```

Sample Execution



```
2000 is a leap year.
1996 is a leap year.
1800 is not a leap year.
2013 is not a leap year.
3144 is a leap year.
20453 is not a leap year.
```

[1] Determining which years are a leap year can get a little more complicated if you go too far back in time. Visit The History of Leap Year and Getting the Date Right for more information.



```
2000 is a leap year.
1996 is a leap year.
1800 is not a leap year.
2013 is not a leap year.
3144 is a leap year.
20453 is not a leap year.
```

[1] Determining which years are a leap year can get a little more complicated if you go too far back in time. Visit The History of Leap Year and Getting the Date Right for more information.