Computer Science CS134C (Fall 2018)
Duane A. Bailey
Laboratory 2
Computing the Age of the Moon

Objective. To construct a script that computes something non-trivial.

This week I'd like you to implement an algorithm to compute the phase of the moon as a python script. Like our day-of-the-week algorithm, this algorithm is simple enough so you could compute the age in your head, but complex enough so that you'd be motivated to have it available as a python function.

Method. We'll be implementing a method for determining the moon's age suitable for mental calculation. Like the day-of-week calculation, this method is due to John Conway.

Let's assume that you have a date, specified as a valid month, day, and year. For purposes of practicality, we'll limit the year to those in the range 1900 to 2099. Let's assume the month is specified as a value between 1 and 12 and the day is a value between 1 and 31. This algorithm computes the moon's *age* in days, a value between 0 and 29; when the age is zero, the moon is new, when it's 15, it's full.

The algorithm involves keeping a running sum, `s`, modulo 30.

1. Initialize the sum, `s`, to the sum of the day, the month, and 30. The final 30 is helpful to avoid negative modular arithmetic, later.
2. If the year's two leading "century digits" are 20, subtract 8 from `s`, otherwise subtract 4 from `s`.
3. Assuming that we have the year within the century (`yy`) a value between 0 and 99. Compute the distance, `d`, to the closest multiple of 19 (`m`). For example, 2018 yields -1, but 1999 gives +4. (Notice that 2000 gives 0, but the century correction avoids a discontinuity here.) This calculation is challenging to think about. Be careful to check your work!
4. To `d` "prepend" a ten's digit that is the value of `|d|` mod 3. Thus 2014, whose distance is -5, will have a final `d` of -25, 1999 generates 14, and 2000 gives 0. Notice that this part of the computation never changes during the year. The value of `d` for 2018 is -11 (or, if it's easier to remember, +19, mod 30).
5. Add `d` to `s` and compute the remainder, when divided by 30. That's the age of the moon.

Here are some examples you might use to verify your calculations:

- Moon Unit Zappa was born September 28, 1967. Think: $(9+28+30-4-9) \bmod 30 \equiv 24$
- Tally, the moon-colored dog was born on August 28, 2017.
  She calculates: $(8+28+30-8-22) \bmod 30 \equiv 6$
- Sheldon "Moon Pie" Cooper turned 21 on February 26, 2001: $(2+26+30-8+11) \bmod 30 \equiv 1$

What is to be done. Clone the project repository using

```
git clone ssh://22xyz@davey.cs.williams.edu/~cs134/22xyz/lab2.git ~/cs134/lab2
```

where your CS username replaces 22xyz.

Your job is to write, in a file called `phase.py`, a function, `moonAge(month,day,year)`, that computes the age of the moon associated with the date specified by `month`, `day`, and `year`.

You might imagine excercising such a function with the guarded code:

```
if __name__ == "__main__":
    month = int(input("Month? "))
    day = int(input("Day? "))
    year = int(input("Year (yyyy)? "))
    age = moonAge(month,day,year)
    print("On {}/{}/{} the moon's age is {}".format(month,day,year, age))
```

When your script is complete, you should be able to type:

```
-> python3 phase.py
Month? 9
Day? 17
Year (yyyy)? 2018
On 9/17/2018 the moon's age is 7.
```

You should also be able to test your code *interactively* with

```
-> python3
>>> from phase import moonAge
>>> moonAge(9,17,2018)
7
>>>
```

Be sure to understand the difference between excercising code in a script and testing functions interactively.

When you believe you're finished with `phase.py`, you use git to commit and push the file for grading. Completing this basic work will earn a good grade you can take pride in. To receive more credit will require more effort demonstrating more thought about the problem.

Going a bit further. Consider reporting the common western description of the moon's age:

age	description
0,1,29	new
2,3,4,5,6	waxing crescent
7,8	first quarter
9,10,11,12,13	waxing gibbous
14,15,16	full
17,18,19,20,21	waning gibbous
22,23,	third quarter
24,25,26,27,28	waning crescent

A solid program reports the moon's age together with its description:

```
-> python3 phase.py 9 18 2018
On 9/18/2018 the moon's age is 8, a first quarter moon.
```

With care, you can determine the description with a small number of strings and without using any `if` statements.

⋆