Counting Sundays

Given that 1 Jan 1900 was a Monday, how many Sundays fell on the first of the month during the 20th century (1 Jan 1901 to 31 Dec 2000)?[1]

Two solutions will be considered for this problem. The solutions (along with their JavaScript implementations) are explained on the following page:

https://www.xarg.org/puzzle/project-euler/problem-19/

Python implementations for these solutions can be found under the class files section.

1. Write a function "dayOfWeek" that, given a year, a month and a day, returns which day of the week the date is. Use Zeller's congruence. Note that Haskell's type system requires explicit type conversions as in:

```
t1 = floor (fromIntegral (13 * (m' + 1)) / 5.0)
```

2. Fill in the Haskell code below to calculate the result.

```
sundays1 :: Integer -> Integer -> Integer
sundays1 start end = sundays' ? ?
  where
    sundays' :: Integer -> Integer -> Integer
    sundays' y m
      | y > end   = ?
      | otherwise = if dayOfWeek y m 1 == 1 then rest + 1 else rest
      where
        nextY = ?
        nextM = ?
        rest = ?
```

   - What does the helper function (sundays') calculate?
   - What if you don't define a "rest" and use its expression where it's needed?

3. Write a tail recursive function of "sundays1".
4. Write the "leap" and "daysInMonth" functions as given in the Python source. Using these, implement "sundays2".
5. (math question) Is the number of weeks in 400 years an integer value? In other words, is the number of days in 400 years a multiple of 7? If so, what is the possibility that a certain day of a month (such as 1 Jan, or your birthday) is a Sunday (or some other day)? Are all days equally possible?

1 This exercise is taken from the Project Euler site: https://projecteuler.net/problem=19