Bonus Assignment I

Uğur Uysal 150140012

CountingSundays1.hs for questions 1 & 2

CountingSundays2.hs for question 3 -> tail recursion version of sundays1

CountingSundays3.hs for question 4 -> sundays2 function is also tail recursive

Question 2

sundays' function calls dayOfWeek for the number of per month in the given year for the first day

```
sundays' 1900 1 calls dayOfWeek 1900 1 1
sundays' 1900 2 calls dayOfWeek 1900 2 1
..
..
sundays' 1900 12 calls dayOfWeek 1900 12 1
sundays' 1901 1 calls dayOfWeek 1901 1 1
```

then return sum of them and terminates when the year exceeds the end

Actually we dont need `rest = sundays' nextY nextM` expression, in tail recursive version of sundays1, I didn't use something like that and in my opinion it is more readable.

Without 'rest' statement, it looks like that.

```
sundays1 :: Integer -> Integer -> Integer
sundays1 start end = sundays' 0 start 1
    where
    sundays' :: Integer -> Integer -> Integer -> Integer
    sundays' acc year month
        | year > end                  = acc
        | dayOfWeek year month 1 == 1 = sundays' (acc+1) nextY nextM
        | otherwise                   = sundays'  acc    nextY nextM
        where
            nextM = if month == 12 then 1 else month + 1
            nextY = if month == 12 then year + 1 else year
```

Question 5

Number of days in any continous 400 (303 non-leap years and 97 leap year) years can be calculated as $303*365 + 97 * 366$ which is 146097, and mod of 146097 respect to 7 equal to zero. So number of weeks are integer value and we can say that all days(sunday, thursday etc.) have same frequency. So all of day are equally possible.