A case study: a calendar

- Problem: Given a month and a year, construct a String that contains the corresponding calendar for that month,e.g. for 11/2003

```
      November 2003
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
```


Problem analysis

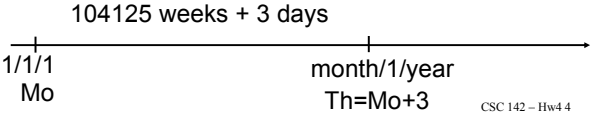
- Range for the year? year>=1 reasonable
- To compute the calendar, we need one reference point, e.g. 1/1/1 was a Monday.
- Leap year problem: a leap year is a year divisible by 4 and not 100 or divisible by 400.
- Practice on paper with some months and years
  - 6/2006, 3/1984, 11/2123

A first algorithm

- Input the month and year
  - Straightforward. Check the validity of the input (do this in CalendarDisplay)
- Find the day of the week for the first of the month (is it Monday?, Tuesday?…)
  - Obviously, we need to refine this step
- The calendar as a String
  - Write 1 under the day of the week corresponding to the first of the month and iterate from there

First day of the month

- We know that 1/1/1 is a Monday
- Compute the number of days between 1/1/1 and month/1/year
- Remove the full weeks. The number of days should be between 0 and 6.
- Add this number to Monday to find the day of the week of month/1/year



Number of days between 2 dates

- start date: 1/1/1

  end date: month/1/year
- Number of days is the sum of
  - # days from 1/1/1 to 12/31/(year-1)
  - # days from 1/1/year to month/1/year

Some details

- Use a method that gives the number of days of the month, given the month and the year
- Use a method to check if a year is a leap year
- # days from 1/1/1 to 12/31/(year-1)

```
for(y=1; y<year; y++){
        days+=365;
        if (isLeapYear(y)) days++;
 }
```

- # days from 1/1/year to month/1/year

```
for(m=1; m<month; m++)
    days+=numberOfDaysInMonth(m,year);
```