

Exercise 11

```
-- define
data Month = January | February
           | March | April
           | May | June
           | July | August
           | September | October
           | November | December
  deriving (Show)

-- implement
daysInMonth :: Month -> Integer
daysInMonth m = case m of
  January -> 31
  February -> 28
  March -> 31
  April -> 30
  May -> 31
  June -> 30
  July -> 31
  August -> 31
  September -> 30
  October -> 31
  November -> 30
  December -> 31

nextMonth :: Month -> Month
nextMonth m = case m of
  January -> February
  February -> March
  March -> April
  April -> May
  May -> June
  June -> July
  July -> August
  August -> September
  September -> October
  October -> November
  November -> December
  December -> January

nextDay :: Integer -> Month ->
         (Integer, Month)
nextDay d m =
  if d == daysInMonth m
  then (1, nextMonth m)
  else (d+1, m)
```