Arnav Singh
Sheet 2#
Intro to CS

Problem 2.1
To prove if n is not divisible by 3 it is also not divisible by 15
15 can also be written as
15 = 3*5
Assume now that a number is not divisible by 3 and let it be denoted by 3' then by the transitivity of divisibility and using 15=3*5; then 15'=3'*5
Hence we have a contradiction if a natural number n is not divisible by 3 it will also not be divisible by 15.

Problem 2.2

Solve for P(1)
LHS:
$$(2(1)-1)^2 = \frac{(2*3)}{6}$$
1=1 LHS= RHS , hence P(1) is true

Now Assume P(m) is true,

$$\sum_{k=1}^{m}(2k-1)^2 = \frac{2m(2m-1)(2m+1)}{6}$$

Now show that n=m+1 is also true for any real number,

Our end result we want by plugging m+1 is
$$\frac{2(2m+1)(m+1)(2m+3)}{6}$$
Now starting from the LHS, The sum till the m+1th term can be represented by the last term m defined by the sigma notation plus the m+1 term

$$\sum_{k=1}^{m+1}(2k-1)^2 = \sum_{k=1}^{m}(2k-1)^2 + (2m+1)^2 \quad \text{Equation 1}$$

From the assumption the sum till the mth term can also be written as $\frac{2m(2m-1)(2m+1)}{6}$
substituting into equation 1 we get,

$$\sum_{k=1}^{m+1}(2k-1)^2 = \frac{2m(2m-1)(2m+1)}{6} + (2m+1)^2$$

Now solving the RHS and then factoring out (2m+1) and $\frac{1}{6}$ we get,
$$\frac{1(2m+1)\ [(2m)(2m-1)+6(2m+1)]}{6}$$

Arnav Singh
Sheet 2#
Intro to CS

Simplifying,

$$\frac{(2m+1)\ \ [\,4m^{\,2}+10m+6]}{6}$$

Solving the quadratic

$$\frac{(2m+1)\ \ [\,(4m+6)(m+1)]}{6}$$

Factoring out 2

$$\frac{2(2m+1)\,(2m+3)(m+1)}{6}$$

This is essentially the RHS stated above what we wanted to get, LHS=RHS

Since P(m+1) is true the assumption P(m) is also true and hence by the principle of mathematical induction P(m) is true for any m $\in$ N

Problem 2.3

a.

module Leapyear(is leapyear)
where
isLeapYear::int> Bool
isLeapYear n = (mod n 4==0) && (not(mod n 100==0) || (mod n 400==0)

b.

isLeapYear':: int ➝ Bool
isLeapYear' n| n 'rem' 400 ==false
isLeapYear' n| n 'rem' 100 == false
isLeapYear' n| n 'rem' 4 ==true
isLeapYear' n| otherwise ==false

The div function always returns a number whereas the mod returns a remainder , the difference is how in the explicitness. I have used the rem function this allows to check if the remainder is 0 or not.

Problem 2.4

a.

```
rotate :: Int -> [a] -> [a]
rotate _[]=[]
rotate 0 xs=xs
rotate n  (x:xs) = xs++[x]
```

Arnav Singh
Sheet 2#
Intro to CS

b.

```
circle :: [a] -> [[a]]
circle []=[]
circle list= rotate ( length list-1)list
where
      rotate :: Int -> [a] -> [a]
      rotate _[]=[]
      Rotate 0 list =list []
      rotate num list
        | num>0 = list: rotate (num-1)(tail list ++ [head list])
```