Problem 3.1

Let's consider a set J, if |J|= n, then J has 2^n subsets.

Let P(n) be the predicate "A set with cardinality n has 2^n subsets.

Consider the case at 0

P(0) is true, because the set with cardinality 0 (the empty set) has 1 subset (itself) and $2^0 = 1$.

Now assume P(k) is true

Prove P(k) :

A set with k elements has 2^k subsets,

To prove k+1 is true

which essentially means that we have to prove that a set with k+1 elements has 2^{k+1} subsets.

Assume that for an arbitrary k, any set with cardinality k has subsets.
Let T be a set such that |T| = k+1.

Now the elements of T:T are = { $elem_1, elem_2, \ldots elem_k, elem_{k+1}$ }

Let S = { $elem_1, elem_2, \ldots elem_k, elem_{k+1}$ }
Then |S|=k, so S has 2^k subsets, according to the inductive hypothesis.
Consider that T = S ∪ $elem_{k+1}$, so every subset of S is also a subset of T.

Any subset of T either contains the element $elem_{k+1}$, or it doesn't contain $elem_{k+1}$.

If a subset of T doesn't contain $elem_{k+1}$, then it is also a subset of S, and there are 2^k of those subsets.

On the other hand, if a subset of T does contain the element $elem_{k+1}$, then that subset is formed by including $elem_{k+1}$ in one of the 2k subsets of S, so T has 2^k subsets containing $elem_{k+1}$.
We have shown that T has 2^k subsets containing $elem_{k+1}$, and another 2^k subsets not containing $elem_{k+1}$, so the total number of subsets of T is

$$2^k + 2^k = (2)\ 2^k = 2^{k+1}$$

Problem 3.2

a.) R = {(a, b)|a, b ∈ Z ∧ a ≠ b} (The numbers a and b are different.)

Since a and b are not equal to each other, they can't be (a,a) or (b,b) so they aren't reflexive. Additionally, it can't be transitive since there is no element c such that the condition satisfies. For example, (4,5,4), a≠b, b≠c, and a=c, which means it is not transitive since we found a counter example.

b) $R = \{(a, b) | a, b \in Z \land |a - b| \leq 3\}$

(The absolute difference of the numbers a and b is less than or equal to 3.)

Firstly, the relation is reflexive since for any ordered pair (a,a) or (b,b) the difference between the two sets will be always be 0 which is less than three and always satisfies the condition
Let's look at the case for symmetry. Let's take case (4,6) we get 2 and (6,4) we still get 2 which is less than 3 and hence symmetric.
And it is not transitive, let's look at a counter example . (-3,0,3) now a=b , b=c however a≠c and hence not transitive.

c)
$R = \{(a, b) | a, b \in Z \land (a \bmod 10) = (b \bmod 10)\}$
(The last digit of the decimal representation of the numbers a and b is the same.)

The above relation is reflexive because if we take any (a,a) or (b,b) the last digit will always be the same since we are dividing by ten, the decimal is the last digit which is constant, It is symmetrical let's take another example that satisfies the condition (42,52) now if we do (b,a) i.e (52,42), it will still satisfy the condition i.e 2=2.It also transitive of the same logic take (38,48,68), a=b, b=c, and a=c thus transitive.

Problem 3.3

A.

```
isPrime :: Integer -> Bool
isPrime 2=True
isPrime x= length([a|a <- [2..(div x 2)] , (mod x a) ==0])==0
checkBoolean xs =
 if all (==True) xs then True
 else False
```

So essentially what I did was I first set the case for 2 , set it to true, and then if that number or list or whatever the input is is divisible by any number uptill that number then it will show that the number is not a prime(false)So for e.g. if you have 23 as your input it will check number from 2,3… till11 if it is divisible or not, in this case it is not.

b.

```
rotate :: Int -> [a] -> [a]
rotate n [] = []
rotate n xs= take( length xs) (drop n (cycle xs))
circle :: [a] -> [[a]]
circle []= []
circle a = [rotate n a| n <- [0.. length (a) -1]]
```

```
isPrime :: Integer -> Bool
isPrime 2= True
isPrime x= length([a|a <- [2..(div x 2)], (mod x a) ==0])==0
checkifitisboolean xs =
 if all (== True) xs then True
else False
isCircPrime :: Integer -> Bool
isCircPrime n = checkifitisboolean(map isPrime ( map ( read :: String
-> Integer) (circle( show n))))
```

Basically,what's happening here is I used previous functions and hint, what was new is the map read string and show. The input will go into a string and will be read as an integer and then the show n will switch it the other way around i.e the integer will be converted to a list or whatevs.