# Exercise 14

```haskell
-- type of if value
class IfValue a where
  boolVal :: a -> Bool

instance IfValue Int where
  boolVal 0 = False
  boolVal _ = True

instance IfValue Integer where
  boolVal 0 = False
  boolVal _ = True

instance IfValue Double where
  boolVal 0 = False
  boolVal _ = True

instance IfValue Bool where
  boolVal = id

instance IfValue Char where
  boolVal '\NUL' = False
  boolVal _ = True

-- map for maybe
maybeMap :: (a -> b) -> Maybe a -> Maybe b
maybeMap _ Nothing = Nothing
maybeMap f (Just x) = Just (f x)

-- map for pair
pairMap1 :: (a -> a') ->
    (a, b) -> (a', b)
pairMap1 f (x, y) = (f x, y)
```