Question 3

1)

Assuming `fmap = <$>`,

```
(<$>) :: (a -> b) -> [a] -> [b]
_ <$> []     = []
f <$> (x:xs) = (f $ x) : (f <$> xs)
```

2)

```
pure :: a -> [a]
pure a = [a]

(<*>) :: [a -> b] -> [a] -> [b]
[] <*> _ = []
_ <*> [] = []
(f:fs) <*> (x:xs) = (f <$> pure x) ++ (fs <*> xs)
```

3)

Assuming `x = [x']`,

Note : `(\g -> g y) = ($ y)`,

```
x <*> pure y                          [ by assumption    ]
= [x'] <*> pure y                     [ apply pure       ]
= [x'] <*> [y]                        [ apply <*>        ]
= (x' <$> pure y) ++ ([] <*> [])      [ apply pure       ]
= (x' <$> [y]) ++ ([] <*> [])         [ apply <*>        ]
= (x' <$> [y]) ++ []                  [ apply ++         ]
= (x' <$> [y])                        [ apply <$>        ]
= (x' $ y) : (x' <$> [])              [ apply <$>        ]
= (x' $ y) : []                       [ apply (:)        ]
= [(x' $ y)]                          [ unapply ($ y)    ]
= [(($ y) x')]                        [ unapply <$>      ]
= [($ y)] <$> [x']                    [ unapply <*>      ]
= pure ($ y) <*> [x']                 [ note             ]
= pure (\g -> g y) <*> [x']           [ by assumption    ]
= pure (\g -> g y) <*> x
```