Part A: Theory

1. Calculational Proof

Base Case: xs = '()

```
(append (append xs ys) zs)
 = { By assumption that xs = '() }
(append (append '() ys) zs)
 = { By append-nil law }
(append ys zs)
 = { By append-nil law (in reverse) }
(append '() (append ys zs))
 = { By xs = '() }
(append xs (append ys zs))
```

Inductive Case: xs = (cons w ws) [Non-empty list]

```
(append (append xs ys) zs)
 = { By xs = (cons w ws) }
(append (append (cons w ws) ys) zs)
 = { By append-cons law }
(append (cons w (append ws ys)) zs)
 = { By append-cons law }
(cons w (append (append ws ys) zs))
 = { By the inductive hypothesis, (append (append ws yw) zs) = (append ws (append yw zs)) }
(cons w (append ws (append yw zs)))
 = { By append-cons law (in reverse) }
(append (cons w ws) (append yw zs))
 = { By xs = (cons w ws) }
(append xs (append yw zs))
```

Thus, by calculations proof, `(append (append xs ys) zs) = (append xs (append ys zs))`