```
import UW

Var (p q : Prop)

-- Proof of commutativity of AND
-- Demonstrates AndElim Asm and AndElim Goal proof rules
Def and_commute := Proof
    ⊢ p ∧ q → q ∧ p
  By ImplElim Goal
    Asm 1: p ∧ q
    ⊢ q ∧ p
  By AndElim Asm 1
    Asm 2: p
    Asm 3: q
  By AndElim Goal
    { Subproof 1: "lhs"
        ⊢ q
        QED Asm 3
      Subproof 2: "rhs"
        ⊢ p
        QED Asm 2
    }

-- Alternative proof of commutativity of AND.
-- Here we swap the order of AndElim Goal and AndElim Asm.
-- The proof works, but is longer than the first option.
-- By doing AndElim Goal first, we have to do AndElim Asm
-- in *both* of the subproofs that result from the AndElim Goal.
Def and_commute2 := Proof
    ⊢ p ∧ q → q ∧ p
  By ImplElim Goal
    Asm 1: p ∧ q
    ⊢ q ∧ p
    By AndElim Goal
    { Subproof 1: "lhs"
          ⊢ q
        By AndElim Asm 1
          Asm 1/1: p
          Asm 1/2: q
        QED Asm 1/2
      Subproof 2: "rhs"
          ⊢ p
        By AndElim Asm 1
          Asm 2/1: p
          Asm 2/2: q
        QED Asm 2/1
    }

-- Proof of commutativity of OR.
-- It follows a similar pattern to commutativity of AND.
Def or_commute := Proof
    ⊢ p ∨ q → q ∨ p
  By ImplElim Goal
```

```
    Asm 1: p ∨ q
  ⊢ q ∨ p
  By OrElim Asm 1
  { Subproof 1: "lhs"
        Asm 1/1: p
      By OrElimRight Goal
        ⊢ p
      QED Asm 1/1
    Subproof 2: "rhs"
        Asm 2/1: q
      By OrElimLeft Goal
        ⊢ q
      QED Asm 2/1
  }

-- Alternative proof *attempt* for commutativity of OR.
-- This proof doesn't work.
-- When we do OrElim Goal, we have to choose which goal (`p` or `q`) we want
-- to prove.  But, we don't know which goal we want to prove until after
-- we do OrElim Asm.
Def or_commute := Proof
  ⊢ p ∨ q → q ∨ p
  By ImplElim Goal
  Asm 1: p ∨ q
  ⊢ q ∨ p
  By OrElimLeft Goal
  ⊢ q
  By OrElim Asm 1
  { Subproof 1: "lhs"
        Asm 1/1: p
        -- stuck: have asm for p,
        -- but need to prove q
  }

/- Lesson learned: In the and_commute proof,
 - the order in which we did AndElim Asm vs
 - AndElim Goal affected the length of the proof,
 - but both orders worked (led to a successful proof).
 - In the or_commute proof, to complete the proof,
 - we had to do OrElim Asm before OrElim Goal.
 -/
```