```
(*  apl-2013-06-24a.v
 *
 * Programmer: Mayer Goldberg, 2013
 *)

Require Import Setoid.

Axiom PNP: ∀ p : Prop, p ∨ ¬p.

Theorem DM1: ∀ (p q : Prop), ¬ (p ∨ q) ↔ ( ¬p ∧ ¬q).
Proof.
  intros p q.
  split.

  (*  Part 1: ~ (p \/ q) -> ~ p /\ ~ q *)
  intro H.
  unfold not in H.
  unfold not.
  split.
  intro Q; destruct H; left; exact Q.
  intro R; destruct H; right; exact R.

  (*  Part 2: ~ p /\ ~ q -> ~ (p \/ q) *)
  intro H.
  unfold not.
  intro Q.
  unfold not in H.
  destruct H as [H1 H2].
  destruct Q as [Q1 | Q2].
  apply (H1 Q1).
  apply (H2 Q2).
Qed.

Theorem DM2: ∀ (p q : Prop), ¬ (p ∧ q) ↔ ( ¬p ∨ ¬q).
  intros p q.
  unfold not.
  split.
  intro H.
  destruct (PNP p) as [Q1 | Q2].
  destruct (PNP q) as [R1 | R2].

  assert(L: p ∧ q).
  split; assumption.

  apply H in L.
  contradict H.
  contradict L.
  right; exact R2.
```

```
  left; exact Q2.
  intros H Q.
  destruct Q as [Q1 Q2].
  destruct H as [H1 | H2].
  apply (H1 Q1).
  apply (H2 Q2).
Qed.
```