```
(** **** Exercise: 3 stars, standard (or_distributes_over_and)  *)
Theorem or_distributes_over_and : forall P Q R : Prop,
  P \/ (Q /\ R) <-> (P \/ Q) /\ (P \/ R).
Proof.
  intros. split.
  - intros[H1 | [H2 H3]].
    + split. left. apply H1. left. apply H1.
    + split. right. apply H2. right. apply H3.
  - intros [[H | H] [T | T]].
    + left. apply T.
    +  left. apply H.
    + left. apply T.
    + right. split. apply H. apply T.
Qed.
(** [] *)
```

```
Theorem In_app_iff : forall A l l' (a:A),
  In a (l++l') <-> In a l \/ In a l'.
Proof.
  intros A l. induction l as [|a' l' IH].
  - split.
    + unfold In. intros. right. apply H.
    + unfold In. intros [H | H]. destruct H. apply H.
  - simpl. split.
    + intros [H | H].
          left. left. apply H.
          apply IH in H. destruct H.
             left. right. apply H.
             right. apply H.
    + intros [[H|H] |H].
         left. apply H.
         right. apply IH. left. apply H.
         right. apply IH. right. apply H.
Qed.
```

```
Theorem orb_true_iff : forall b1 b2,
  b1 || b2 = true <-> b1 = true \/ b2 = true.
Proof.
  intros. split.
  - intros H. destruct b1.
    + left. reflexivity.
    + destruct b2.
      * right. reflexivity.
      * inversion H.
  - intros [H1 | H2].
    + rewrite H1. reflexivity.
    + rewrite H2. destruct b1.
      * reflexivity.
      * reflexivity.
Qed.
(** [] *)
```

```
Theorem not_exists_dist :
  excluded_middle ->
  forall (X:Type) (P : X -> Prop),
    ~ (exists x, ~ P x) -> (forall x, P x).
Proof.
  unfold excluded_middle. intros.
  destruct (H (P x)) as [HP | NP].
  - apply HP.
  - exfalso. apply H0.
    exists x. apply NP.
Qed.
```