Coq (2) - Questões

Crie um ficheiro Coq para desenvolver as provas das propriedades abaixo indicadas. Pode usar táticas automáticas.

Considere as seguintes declarações em Coq.

```
Set Implicit Arguments.

Require Import List.

Inductive In (A:Type) (y:A) : list A -> Prop :=
 | InHead : forall (xs:list A), In y (cons y xs)
 | InTail : forall (x:A) (xs:list A), In y xs -> In y (cons x xs).

Inductive Prefix (A:Type) : list A -> list A -> Prop :=
  | PreNil : forall (l:list A), Prefix nil l
  | PreCons : forall (x:A) (l1 l2:list A), Prefix l1 l2 -> Prefix (x::l1) (x::l2).

Inductive SubList (A:Type) : list A -> list A -> Prop :=
  | SLnil : forall (l:list A), SubList nil l
  | SLcons1 : forall (x:A) (l1 l2:list A), SubList l1 l2 -> SubList (x::l1) (x::l2)
  | SLcons2 : forall (x:A) (l1 l2:list A), SubList l1 l2 -> SubList l1 (x::l2).
```

1. Prove as seguintes propriedades:
   (a) `SubList (5::3::nil)  (5::7::3::4::nil)`
   (b) `forall (A:Type) (l:list A),  SubList l l`
   (c) `forall (A B:Type) (f:A->B) (l1 l2:list A), SubList l1 l2 -> SubList (map f l1) (map f l2)`
   (d) `forall (A:Type) (x:A) (l : list A), In x l -> exists l1, exists l2, l = l1 ++ (x::l2)`

2. Defina a função recursiva `drop` que dado um número natural n e uma lista l, retira os n primeiros elementos de l. Prove que:
   (a) `drop 2 (5::7::3::4::nil) = 3::4::nil`
   (b) `forall (A:Type) (n:nat) (l:list A), SubList (drop n l) l`

3. Defina indutivamente o predicado `Sorted` sobre listas de números naturais. Prove que:
   (a) `forall (x y:nat) (l:list nat), x<=y -> (Sorted (y::l)) -> Sorted (x::l)`
   (b) `forall (x y:nat) (l:list nat), (In y l) /\ (Sorted (x::l)) -> x <= y`

4. Prove que relação `Prefix` é uma relação de ordem. Isto é:
   (a) `forall (A:Type) (l:list A), Prefix l l`
   (b) `forall (A:Type) (l1 l2 l3:list A), Prefix l1 l2 /\ Prefix l2 l3 -> Prefix l1 l3`
   (c) `forall (A:Type) (l1 l2:list A), Prefix l1 l2 /\ Prefix l2 l1 -> l1 = l2`