```
Inductive ilist : nat -> Set :=
| Nil : ilist O
| Cons : forall n, A -> ilist n -> ilist (S n).


Definition hd' n (ls : ilist n) :=
  match ls in (ilist n) return (match n with O => unit | S _ => A end) with
    | Nil => tt
    | Cons _ h _ => h
  end.
```

```
data Vec (A : Set) : Nat -> Set where
  []    : Vec A zero
  _::_ : {n : Nat} -> A -> Vec A n -> Vec A (suc n)

head : {A : Set}{n : Nat} -> Vec A (suc n) -> A
head (x :: xs) = x
```