Claim: For all lists `l1` and `l2`, `rev_append l1 l2` = `rev_append' l1 l2`.
Proof: By induction on `l1`.

- **Base case**: `l1` = `[]`.

  In this case, we have:

`rev_append l1 l2`	= `rev_append [] l2`	by assumption
	= `l2`	by definition of `rev_append`
	= `append [] l2`	by definition of `append`
	= `append (rev []) l2`	by definition of `rev`
	= `append (rev l1) l2`	by assumption
	= `rev_append' l1 l2`.	by definition of `rev_append'`

- **Step**: `l1` = `h::t` for some element `h` and list `t`.

  Inductive hypothesis: For all lists `l2`, `rev_append t l2` = `rev_append' t l2`.

  In this case, we have:

`rev_append l1 l2`	= `rev_append (h::t) l2`	by assumption
	= `rev_append t (h::l2)`	by definition of `rev_append`
	= `rev_append' t (h::l2)`	by the inductive hypothesis
	= `append (rev t) (h::l2)`	by definition of `rev_append'`
($\star$)	= `append (rev t) (h::(append [] l2))`	by definition of `append`
	= `append (rev t) (append [h] l2)`	by definition of `append`
	= `append (append (rev t) [h]) l2`	by associativity of `append`
	= `append (rev (h::t)) l2`	by definition of `rev`
	= `rev_append' (h::t) l2`	by definition of `rev_append'`
	= `rev_append' l1 l2`.	by assumption

  Note: skipping the step marked ($\star$) will not lose any marks.