Isabelle/HOL Exercises
Lists

SNOC

Define a primitive recursive function *snoc* that appends an element at the *right* end of a list. Do not use `@` itself.

```
consts
  snoc :: "'a list => 'a => 'a list"

primrec
  "snoc []     a = [a]"
  "snoc (x#xs) a = x # snoc xs a"

lemma snoc_append: "snoc xs a = xs @ [a]"
  apply (induct "xs")
  apply auto
done
```

Prove the following theorem:

```
theorem rev_cons: "∀x. rev (x # xs) = snoc (rev xs) x"
  apply (induct "xs")
  apply (auto simp add: snoc_append)
done
```

Hint: you need to prove a suitable lemma first.