```
fun append (xs,ys) =
    if xs=[]
    then ys
    else (hd xs)::append(tl xs,ys)

fun map (f,xs) =
    case xs of
        [] => []
      | x::xs' => (f x)::(map(f,xs'))

val a = map (increment, [4,8,12,16])
val b = map (hd, [[8,6],[7,5],[3,0,9]])
```

Programming Languages

Dan Grossman

Optional: More Macro Examples

More examples

See the code for macros that:

- A for loop for executing a body a fixed number of times
  - Shows a macro that purposely re-evaluates some expressions and not others

- Allow 0, 1, or 2 local bindings with fewer parens than **`let*`**
  - Shows a macro with multiple cases

- A re-implementation of **`let*`** in terms of **`let`**
  - Shows a macro taking any number of arguments
  - Shows a recursive macro