

```
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  
2013

*More Nested Patterns*

# Style

- Nested patterns can lead to very elegant, concise code
  - Avoid nested case expressions if nested patterns are simpler and avoid unnecessary branches or let-expressions
    - Example: **unzip3** and **nondecreasing**
  - A common idiom is matching against a tuple of datatypes to compare them
    - Examples: **zip3** and **multsign**
- Wildcards are good style: use them instead of variables when you do not need the data
  - Examples: **len** and **multsign**