The problem

fun earlier x = ... other x ...
fun later x = ... other x ...
fun other x = earlier x ... later x ...

The above doesn't work because earlier and later don't have reference to the other binding!

Solutions?

If earlier and later aren't used anywhere else, then have other define them locally instead!

fun other x = let
  fun earlier x ...
  fun later x ...
in
  earlier x ... later x ...
end

Or... use mutual recursion!

In HW2, there are some problems where you want this

This is a common idiom in SML and other languages

Mutual recursion uses the and keyword

fun even 0 = true
| even n = odd (n - 1)
and odd 0 = false
| odd n = even (n - 1)

SML Docs

Useful skill to look at docs!

Don't want to reimplement something that's already in standard lib

Two of the most useful pages for this next homework:

http://sml-family.org/Basis/string.html
http://sml-family.org/Basis/list.html
Higher Order Functions

map([cook, eat, drink, sweets], cook)
=> [cook, eat, drink, sweets]

filter([cook, eat, drink, sweets], isVegetarian)
=> [eat, drink, sweets]

reduce([cook, eat, drink, sweets], eat)
=> eat