Part I

Map
Squares

> squares [1,-2,3]
[1,4,9]

squares :: [Int] -> [Int]
squares xs = [ x*x | x <- xs ]

squares :: [Int] -> [Int]
squares [] = []
squares (x:xs) = x*x : squares xs
Ords

> ords "a2c3"
[97,50,99,51]

ords :: [Char] -> [Int]
ords xs = [ ord x | x <- xs ]

ords :: [Char] -> [Int]
ords [] = []
ords (x:xs) = ord x : ords xs
Map

map :: (a -> b) -> [a] -> [b]
map f xs = [ f x | x <- xs ]

map :: (a -> b) -> [a] -> [b]
map f [] = []
map f (x:xs) = f x : map f xs
Squares, revisited

\[
\begin{align*}
\text{\texttt{\textgreater squares \[1,-2,3\]}} \\
\text{[1,4,9]} \\
\text{squares :: [Int] \to [Int]} \\
\text{squares \ \texttt{xs} \ = \ \texttt{[ x*x \mid x <- xs ]}} \\
\text{squares :: [Int] \to [Int]} \\
\text{squares \ [] \ = \ []} \\
\text{squares (x:xs) \ = \ x*x : \ \text{\texttt{squares \ \texttt{xs}}}}
\end{align*}
\]

\[
\begin{align*}
squares :: \text{[Int]} \to \text{[Int]} \\
squares \ \texttt{xs} \ = \ \text{map \ \texttt{sqr} \ \texttt{xs}} \\
\quad \text{where} \\
\quad \texttt{sqr \ x} \ = \ x*x
\end{align*}
\]
Map—how it works

\[
\text{map} :: (a \to b) \to [a] \to [b] \\
\text{map } f \text{ } xs = [ f x \mid x \leftarrow xs ]
\]

\[\text{map} \text{ sqr } [1,2,3] = [ \text{ sqr } x \mid x \leftarrow [1,2,3] ] = [ \text{ sqr } 1 ] ++ [ \text{ sqr } 2 ] ++ [ \text{ sqr } 3] = [1, 4, 9]\]
Map—how it works

\[\text{map} :: (a \to b) \to [a] \to [b]\]
\[\text{map } f \; [] = []\]
\[\text{map } f \; (x:xs) = f \; x : \text{map } f \; xs\]

\[
\text{map} \; \text{sqr} \; [1,2,3]
= 
\text{map} \; \text{sqr} \; (1 : (2 : (3 : [])))
= 
\text{sqr} \; 1 : \text{map} \; \text{sqr} \; (2 : (3 : []))
= 
\text{sqr} \; 1 : (\text{sqr} \; 2 : \text{map} \; \text{sqr} \; (3 : []))
= 
\text{sqr} \; 1 : (\text{sqr} \; 2 : (\text{sqr} \; 3 : \text{map} \; \text{sqr} \; []))
= 
\text{sqr} \; 1 : (\text{sqr} \; 2 : (\text{sqr} \; 3 : []))
= 
1 : (4 : (9 : []))
= 
[1, 4, 9]
Ords, revisited

> ords "a2c3"
[97, 50, 99, 51]

ords :: [Char] -> [Int]
ords xs = [ ord x | x <- xs ]

ords :: [Char] -> [Int]
ords [] = []
ords (x:xs) = ord x : ords xs

ords :: [Char] -> [Int]
ords xs = map ord xs
Part II

Filter
Odds

> odds [1,2,3]
[1,3]

odds :: [Int] -> [Int]
odds xs = [ x | x <- xs, odd x]

odds :: [Int] -> [Int]
odds [] = []
odds (x:xs) | odd x = x : odds xs
| otherwise = odds xs
Digits

> digits "a2c3"
"23"

digits :: [Char] -> [Char]
digits xs = [ x | x <- xs, isDigit x ]

digits :: [Char] -> [Char]
digits [] = []
digits (x:xs) | isDigit x = x : digits xs
            | otherwise = digits xs
Filter

\[
\text{filter} :: (a \rightarrow \text{Bool}) \rightarrow [a] \rightarrow [a] \\
\text{filter} \; p \; \text{xs} = \{ \; x \mid x \leftarrow \text{xs}, \; p \; x \; \}
\]

\[
\text{filter} :: (a \rightarrow \text{Bool}) \rightarrow [a] \rightarrow [a] \\
\text{filter} \; p \; [] = [] \\
\text{filter} \; p \; (x:xs) \mid p \; x = x : \text{filter} \; p \; \text{xs} \\
\mid \text{otherwise} = \text{filter} \; p \; \text{xs}
\]
Odds, revisited

> odds [1,2,3]
[1,3]

odds :: [Int] -> [Int]
odds xs = [ x | x <- xs, odd x ]

odds :: [Int] -> [Int]
odds [] = []
odds (x:xs) | odd x = x : odds xs
| otherwise = odds xs

odds :: [Int] -> [Int]
odds xs = filter odd xs
Digits, revisited

> digits "a2c3"
"23"

digits :: [Char] -> [Char]
digits xs = [ x | x <- xs, isDigit x ]

digits :: [Char] -> [Char]
digits [] = []
digits (x:xs) | isDigit x = x : digits xs
| otherwise = digits xs

digits :: [Char] -> [Char]
digits xs = filter isDigit xs
Part III

Fold
Sum

> sum [1,2,3,4]
10

sum :: [Int] -> Int
sum [] = 0
sum (x:xs) = x + sum xs
Product

> product [1,2,3,4]
24

product :: [Int] -> Int
product [] = 1
product (x:xs) = x * product xs
Concatenate

> concat [[1,2,3],[4,5]]
[1,2,3,4,5]

> concat ["con","cat","en","ate"]
"concatenate"

concat :: [[a]] -> [a]
concat [] = []
concat (xs:xss) = xs ++ concat xss
And

> and [True, True, True]
True
> and [True, False, True]
False

and :: [Bool] -> Bool
and [] = True
and (x:xs) = x && and xs
Or

> or [False, False, False]
False
> or [False, True, False]
True

or :: [Bool] -> Bool
or [] = False
or (x:xs) = x || or xs
Foldr

foldr :: (a -> a -> a) -> a -> [a] -> a
foldr f v [] = v
foldr f v (x:xs) = f x (foldr f v xs)
Foldr, with infix notation

\[
\text{foldr} :: (a \to a \to a) \to a \to [a] \to a
\]

\[
\text{foldr } f \ v \ [] \quad = \quad v
\]

\[
\text{foldr } f \ v \ (x:xs) \quad = \quad x \ 'f' \ (\text{foldr } f \ v \ xs)
\]
Sum, revisited

> sum [1,2,3,4]
10

sum :: [Int] -> Int
sum [] = 0
sum (x:xs) = x + sum xs

Recall that (+) is the name of the addition function,
so x + y and (+) x y are equivalent.
Sum, Product, Concat, And, Or

\[
\begin{align*}
\text{sum} & \quad :: \ [\text{Int}] \rightarrow \text{Int} \\
\text{sum} \ \text{xs} & \quad = \ \text{foldr} \ (+) \ 0 \ \text{xs} \\
\text{product} & \quad :: \ [\text{Int}] \rightarrow \text{Int} \\
\text{product} \ \text{xs} & \quad = \ \text{foldr} \ (*) \ 1 \ \text{xs} \\
\text{concat} & \quad :: \ [[\text{a}]] \rightarrow \ [\text{a}] \\
\text{concat} \ \text{xs} & \quad = \ \text{foldr} \ (++) \ [] \ \text{xs} \\
\text{and} & \quad :: \ [\text{Bool}] \rightarrow \text{Bool} \\
\text{and} \ \text{xs} & \quad = \ \text{foldr} \ (\&\&) \ True \ \text{xs} \\
\text{or} & \quad :: \ [\text{Bool}] \rightarrow \text{Bool} \\
\text{or} \ \text{xs} & \quad = \ \text{foldr} \ (||) \ False \ \text{xs}
\end{align*}
\]
Sum—how it works

```haskell
sum :: [Int] -> Int
sum [] = 0
sum (x:xs) = x + sum xs
```

```
sum [1,2] =
    sum (1 : (2 : [])) =
    1 + sum (2 : []) =
    1 + (2 + sum []) =
    1 + (2 + 0) = 3
```
Sum—how it works, revisited

foldr :: (a -> a -> a) -> a -> [a] -> a
foldr f v [] = v
foldr f v (x:xs) = x ‘f’ (foldr f v xs)

sum :: [Int] -> Int
sum xs = foldr (+) 0 xs

sum [1,2] = foldr (+) 0 [1,2] = foldr (+) 0 (1 : (2 : [])) = 1 + (foldr (+) 0 (2 : [])) = 1 + (2 + (foldr (+) 0 [])) = 1 + (2 + 0) = 3
Part IV

Map, Filter, and Fold
All together now!
Sum of Squares of Odds

\[
\text{f :: [Int] -> Int} \\
f \text{xs} = \text{sum} \ (\text{squares} \ (\text{odds} \ \text{xs}))
\]

\[
\text{f :: [Int] -> Int} \\
f \text{xs} = \text{sum} \ [x^2 \mid x \leftarrow \text{xs}, \ \text{odd} \ x]
\]

\[
\text{f :: [Int] -> Int} \\
f \text{[]} = [] \\
f (x:xs) \\
\begin{cases} 
\text{odd} \ x & = (x^2) + f \ xs \\
\text{otherwise} & = f \ xs 
\end{cases}
\]

\[
\text{f :: [Int] -> Int} \\
f \text{xs} = \text{foldr} \ (+) \ 0 \ (\text{map} \ \text{sqr} \ (\text{filter} \ \text{odd} \ \text{xs})) \\
\text{where} \\
\text{sqr} \ x = x \times x
\]