

1)

```
mapandfilter :: (a -> b) -> (a -> Bool) -> [a] -> [b]
mapandfilter f p xs = map f (filter p xs)
```

2)

```
-- The foldr creates a temporary function that takes in x and xs
-- the temp function then applies f on x and then recurses it on the rest of xs
-- the [] is the base case that occurs when the input is an empty list and it tells
-- the function to return [] if the input is empty
map = foldr (\x xs : f x : xs) []
```

3)

```
-- The base case is [] where if there is an empty input list then return an empty
list
-- Otherwise, call p on x, if true then return x and recurse on xs. If it is not
true then
-- Just recurse on xs
filter = foldr (\x xs : if p x then x : xs else xs) []
```