CSE 130 (Wi 25) Handout

March 6, 2025

Question 1

Consider the datatype

```
data Tree a = Leaf | Node a (Tree a) (Tree a)
```

What should the type of `mapTree` be so that the function `showTree` behaves as shown below

```
-- >>> showTree (Node 2 (Node 1 Leaf Leaf) (Node 3 Leaf Leaf))
-- (Node "2" (Node "1" Leaf Leaf) (Node "3" Leaf Leaf))

sqrTree :: Tree Int -> Tree String
showTree t = mapTree (\n -> show n) t
```

and the function `sqrTree` behaves as shown below

```
-- >>> sqrTree (Node 2 (Node 1 Leaf Leaf) (Node 3 Leaf Leaf))
-- (Node 4 (Node 1 Leaf Leaf) (Node 9 Leaf Leaf))

sqrTree :: Tree Int -> Tree Int
sqrTree  t = mapTree (\n -> n ^ 2)  t
```

Question 2

Fill in the implementation of a function

```
combine :: [a] -> (a -> [b]) -> [b]

combine xs f = ____________________________________________________________________
               ____________________________________________________________________
               ____________________________________________________________________
               ____________________________________________________________________
               ____________________________________________________________________
```

So that when you are done we get the following behavior

```
q2 = combine [1,2,3] (\x ->
       combine ['a','b'] (\y ->
          [(x, y)]
       )
     )

-- >>> q2
-- [(1, 'a'), (1, 'b'), (2, 'a'), (2, 'b'), (3, 'a'), (3, 'b')]
```