Interim Test Feedback

- Spaces around all operators
- Use good variable names:
  - number = x
  - boolean = b
  - identifier = v
  - operator = op
  - expression = e
  - expressions = es
  - condition = c
  - then = t
  - else = e
  - arguments = as
  - function = f
- Don't add loads of spaces to line things up, add a few if that lines it up
- Can do PM in where clauses
  - `where Boolean b = eval c env`
- Don't use `concat` if you need to remove duplicates (use `foldr union []` instead)
- If you see the same complex idea repeated multiple times, write it out once in a helper function and use the helper, to simplify things
- Don't use length!
- If you want to create functions of the form `isNumber` and `fromNumber`, you should instead be using PM
- If they provide functions, you don't need to use them
- When going through each possibility for a data type, if multiple possibilities inputted into the function give the same output, put them at the end using `_`, e.g:
  - `isWellFormed' _ = []`
- When adding to an environment, you must add to the front!
- Use `xs@(x : xs')`, rather than recreating `(x : xs')`
- Use sequence:
  - `sequence :: (Traversable t, Monad m) => t (m a) -> m (t a)`

Before you make your own function, think about the problem in its general form, and see if you can use a built in function.