

## List Comprehensions

Rewrite the following functions using *list comprehensions*:

- `mapLC :: (a -> b) -> [a] -> [b]`, which takes a function and a list and applies the function to each element in the list (corresponds to the predefined function `map`).
- `filterLC :: (a -> Bool) -> [a] -> [a]`, which takes a predicate and a list and retains only the elements in the given list for which applying the predicate function returns `True`. The function should preserve the original order of the elements (corresponds to the predefined function `filter`).