Foundations of Computer Science 2001 pl q1 LCP

Solution notes

`map f` returns [2, 4, 6, 8]. The shared reference, `x`, increases from 0 to 4 during evaluation; that is why the values increase so much.

`map g` returns [2, 3, 4, 5]. Each call generates a fresh reference that is set to 1, used and then discarded.

`map inc` returns [ref 5, ref 5, ref 5]. There are three separate references, each holding the value 5.

`filter` is bookwork (slide 1107):

```
fun filter p [] = []
  | filter p (x::xs) =
      if p x then x :: filter p xs
             else filter p xs;
```

Armed with `filter`, we can easily code Quicksort because the partition phase no longer requires declaring a local recursive function.

```
fun qs [] = []
  | qs [x] = [x]
  | qs (x::xs) = qs(filter (fn y => y<x) xs) @ [x] @
                 qs(filter (fn y => x<=y) xs);
```

Four marks are allocated to Quicksort because the code is more sophisticated than that of `filter`, despite the similarity in length.