

1. Express the following list comprehension using the function map and filter:

```
f type: int -> int
[ f x | x <- xs, p x ]

map f (filter p xs)

filter p xs = [ x | x <- xs, p x ]
map f xs = [ f x | x <- xs ]
```

2. Redefine map f using foldr.

```
map f = foldr (\x xs -> f x : xs) []
```

3. Redefine filter p using foldr.

```
filter p = foldr (\x xs -> if p x then x : xs else xs) []
```