FP-Aufgaben 07

Aufgabe 1

```
g :: (Num a, Eq a) => [a] -> a -> [a] -> a
```

Aufgabe 2

call-by-value

```
(\n -> mult n n) (1+2)
= mult (1+2) (1+2)
= mult 3 (1+2)
= mult 3 3
= 3 * 3 = (*) 3 3 = 9
```

call-by-name

```
(\n -> mult n n) (1+2)
= mult (1+2) (1+2)
= (mult (1+2)) (1+2)
= (\n -> (1+2) * n) (1+2)
= (1+2) * (1+2)
= (*) (1+2) (1+2)
= (*) 3 3 = 9
```

call-by-need

```
(\n -> mult n n) (1+2)
= (\n -> mult n n) 3
= mult 3 3
= 3 * 3 = 9
```

Aufgabe 3

(add 2 3)
= (add 2) 3
= (\x → inc (add (dec 2) x)) 3
= inc (add (dec 2) 3)
= (add (dec 2) 3) + 1
= (add (2-1) 3) + 1
= (add 1 3) + 1
= ((add 1) 3) + 1
= ((\x → inc (add (dec 1) x)) 3) + 1
= inc (add (dec 1) 3) + 1
= ((add (dec 1) 3) + 1) + 1) = (add (1+1) 3) + 1 + 1
= (add 0 3) + 1 + 1
= 3 + 1 + 1 = (+)((+) 3 1) 1 = (+) 4 1 = 5

(add' 2 3)
= (add' 2) 3
= (\m → add' (dec 2) (inc m)) 3
= add' (dec 2) (inc 3)
= add' (2-1) (3+1)
= add' 1 4
= (add' 1) 4
= (\m → add' (dec 1) (inc m)) 4
= add' (dec 1) (inc 4)
= add' (1-1) (4+1)
= add' 0 5
= 5