9 - Sarah Zewge und Dominik Wille - Freitags

1 Rekursion

```
-- Aufgabe 1:
first (x:xs) = x
fibList_ x (a:b:l)
  | (x > 0)   = fibList_ (x-1) ([a+b] ++ [a] ++ [b] ++ l)
  | otherwise = [a] ++ [b] ++ l
fibList x = fibList_ x [0,1]
fastFib x = first(fibList x)

-- Aufgabe 2:
delMax l = delValue (maximum l) l
delValue a l = delValue_ a [] l
delValue_ a l (x:xs)
  | (a == x)  = l ++ xs
  | otherwise = delValue_ a (l ++ [x]) xs

selStep :: ([Int], [Int]) -> ([Int], [Int])
selStep (a, b) = ((delMax a), ([maximum a] ++ b))

selSort l = selSort_ (l, [])
selSort_ (a, b)
  | (a == []) = b
  | otherwise = selSort_ (selStep(a, b))

-- Aufgabe 3a:
subR x l = reflect (subL x (reflect l))
subL x l
  | (l == []) = []
  | otherwise = subL_ x l []

subL_ x (e:o) n
  | (x <= 0)    = n
  | (o == [])   = (n ++ [e])
  | otherwise   = subL_ (x-1) o (n ++ [e])

reflect l
  | (l == []) = []
  | otherwise = reflect_ l []
```

```
reflect_ (x:xs) l
  | (xs == []) = [x] ++ l
  | otherwise  = reflect_ xs ([x] ++ l)

maxList (a, b)
  | ((length a) < (length b)) = maxList_ a b
  | otherwise                 = maxList_ b a

maxList_ a b = (subR (length a) (selSort (a ++ (subL (length a) b))))
  ++ (subR ((length b) - (length a)) b)

-- Aufgabe 3b:
maxListOfLists a b
  | ((length a) < (length b)) = maxListOfLists_ a b
  | otherwise                 = maxListOfLists_ b a

maxListOfLists_ a b = (map maxList (zip a b)) ++ subR (length b - length a) b
```