a) `max = fn : int list -> int`

die das Maximum einer Liste von Integer-Werten bestimmt. Sie dürfen als Vereinfachung annehmen, dass die Werte positiv sind.
Das Maximum der leeren Liste soll 0 sein.

```
fun max(nil) = 0
  | max(x::xs) = if x > max(xs) then x else max(xs);
```

Übungsblatt12 – Aufgabe1 b)

b) `nth = fn : int list * int -> int`

die das n-te Element einer Liste von Integer-Werten zurückgibt.
Bei einer leeren Liste soll 0 zurückgegeben werden.
Sie dürfen *nicht* die Funktion `List.nth` verwenden!

```
fun nth(nil, n) = 0
  | nth(x::xs, 0) = x
  | nth(x::xs, n) = nth(xs, n-1);
```

Übungsblatt12 – Aufgabe1 c)

c) `reverse = fn : 'a list -> 'a list`

die eine Liste von beliebigen Werten umdreht.
Das Ergebnis von z.B. `reverse([1, 2, 3]);` soll `[3, 2, 1]` sein.

```
fun reverse(nil) = nil
  | reverse(x::nil) = [x]
  | reverse(x::xs) = reverse(xs) @ [x];
```

Übungsblatt12 – Aufgabe1 d)

d) `insert = fn : 'a * 'a list * ('a * 'a -> bool) -> 'a list`

die einen Wert in eine Liste einfügt, und zwar genau an der Stelle, an der die ebenfalls übergebene Funktion zum ersten Mal `true` ergibt. Die übergebene Funktion vergleicht zwei Werte.
Hinweis: Mit `(op <)` können Sie die üblichen Vergleichs-Operatoren als Funktion verwenden.
Beispiel: Der Aufruf von `insert(3, [1,2,4,5], (op <));` soll `[1, 2, 3, 4, 5]` ergeben.

```
fun insert(x, nil, bef) = [x]
  | insert(x, y::ys, bef) = if bef(x,y)
                            then x::y::ys
                            else y::insert(x, ys, bef);
```

Übungsblatt12 – Aufgabe1 e)

e) `testInverse = fn : (''a -> ''b) * (''b -> ''a) * ''a * ''b -> bool`

die für zwei übergebene Funktionen $f : \alpha \to \beta$ und $g : \beta \to \alpha$ prüft, ob g die Umkehrfunktion von f ist, d.h. ob gilt $f = g^{-1}$ und $g = f^{-1}$.
Da die Prüfung nicht auf dem gesamten (unbekannten!) Definitionsbereich bzw. Wertebereich von f und g erfolgen kann, werden zusätzlich zwei Werte $x : \alpha$ und $y : \beta$ übergeben. *Ausschließlich* für diese beiden Werte soll die Prüfung erfolgen.

```
fun testInverse(f, g, x, y) = f(g(y))=y andalso g(f(x))=x;
```