

## Theoretische Informatik II Übungen

**Aufgabe 25.** Benutzen Sie `define-syntax` und schreiben sie ein Macro `swap`, so dass nach dem Aufruf `(swap a b)` mit zwei Variablen `a` und `b` die Werte von `a` und `b` vertauscht sind.

**Aufgabe 26.** Benutzen Sie `define-syntax` und schreiben sie ein Macro `sort`, so dass nach dem Aufruf `(sort a b)` mit zwei Variablen `a` und `b` der Werte von `a` kleiner gleich dem Wert von `b` ist. Erweitern sie dann das Macro, so dass auch `(sort)`, `(sort a)` und `(sort a b c)` möglich ist. Verwenden Sie rekursive Macro Aufrufe.

**Aufgabe 27.** Benutzen Sie `define-syntax` und schreiben sie ein Macro `while`, so dass nach dem Aufruf `(while Bedingung Ausdruck ...)` zuerst die Bedingung ausgewertet wird und dann, falls die Bedingung wahr ist, die Ausdrücke die dem Test folgen. Dannach wiederholt sich das ganze. Hinweis: Man kann dieses Makro unter Verwendung einer `do` Schleife schreiben (einfach) oder mittels eines “named let” und Endrekursion (etwas eleganter aber nicht wesentlich kürzer). Vorsicht: Ein rekursiver Macroaufruf funktioniert hier nicht!

**Aufgabe 28.** Benutzen Sie `define-syntax` und schreiben sie ein Macro `for`, so dass `(for variable from Ausdruck to Ausdruck do Ausdruck ...)` eine “gewöhnliche” `for`-Schleife ergibt. Funktioniert damit der folgende Code?

```
(define from 1)
(define to 5)
(for from from from to to do
  (display for)
  (newline)
)
```