

# Übungsblatt 6

Prof. Dr. Frank Deinzer, Frank Ebner  
Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt  
Fakultät für Informatik & Wirtschaftsinformatik FHWS

Algorithmen und Datenstrukturen I  
Wintersemester 2019/2020

## Aufgabe 27 (4 Punkte) (\*\*)

Schreiben Sie eine Prozedur `zaehlen`, die eine beliebige ganze Zahl `start`, eine beliebige ganze Zahl `ende` und eine Zahl `n > 0` als Parameter übergeben bekommt. Als Ergebnis soll diese Prozedur die `n`-te Zahl aus diesem Bereich von `start` bis `ende` (jeweils inklusiv und gezählt ab `start`) liefern, die sowohl durch 3 als auch durch 7 teilbar ist. Falls es keine `n`-te Zahl gibt, die diese Forderung erfüllt, soll das Resultat 0 sein.

```
(define (zaehlen start ende n)
  ... )
```

### Beispiele:

```
(zaehlen 10 100 1)  →  21
(zaehlen 10 100 2)  →  42
(zaehlen 10 100 3)  →  63
```

## Aufgabe 28 (4 Punkte) (\*\*)

Schreiben Sie eine Prozedur `gleiche-ziffern`, die eine beliebige ganze Zahl `zahl` übergeben bekommt. Als Ergebnis soll diese Prozedur die erste Zahl zurückliefern, die  $\geq$  dem übergebenen Parameter ist und deren erste und letzte Ziffer identisch sind.

```
(define (gleiche-ziffern zahl)
  ... )
```

### Beispiele:

```
(gleiche-ziffern 123) → 131
(gleiche-ziffern 4567) → 4574
```

## Aufgabe 29 (5 Punkte) (\*)

Schreiben Sie eine Prozedur `konst-addierer`, die als Parameter eine Zahl `n` bekommt und als Resultat eine Prozedur liefert, die wiederum eine einzelne Zahl `a` als Parameter akzeptiert und auf diese Zahl `a` das bei der Prozedurkonstruktion angegebene `n` addiert.

```
(define (konst-addierer n)
  ...)
```

### Beispiele:

# Übungsblatt 6

Prof. Dr. Frank Deinzer, Frank Ebner  
Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt  
Fakultät für Informatik & Wirtschaftsinformatik FHWS

Algorithmen und Datenstrukturen I  
Wintersemester 2019/2020

```
(define plus1 (konst-addierer 1))  
(plus1 98) → 99  
  
(define plus10 (konst-addierer 10))  
(plus10 98) → 108
```

## Aufgabe 30 (8 Punkte) (\*)

Schreiben Sie eine Prozedur `konst-ggt`, die als Parameter eine Zahl `b` bekommt und als Ergebnis eine Prozedur liefert, die den größten gemeinsamen Teiler dieser Zahl `b` und einer Zahl, die als Parameter der generierten Prozedur angegeben wird, liefert.

```
(define (konst-ggt b)  
  ...)
```

### Beispiele:

```
(define ggt10 (konst-ggt 10))  
(ggt10 25) → 5 (da ggt(25,10)=5)  
(ggt10 27) → 1 (da ggt(27,10)=1)  
  
(define ggt987 (konst-ggt 987))  
(ggt987 762351) → 3  
(ggt987 98123746) → 7
```

## Aufgabe 31 (6 Punkte) (\*\*)

Schreiben Sie eine Prozedur `paar-operation`, die als Parameter `op` eine (zweistellige) Operation erhält. Diese Prozedur liefert als Ergebnis eine neue Prozedur, die als Parameter ein Paar bekommt und die Operation `op` auf den Links- und den Rechtsteil des Paares anwendet.

```
(define (paar-operation op)  
  ...)
```

### Beispiele:

```
(define paar=? (paar-operation =))  
(paar=? (cons 2 3)) → #f (da 2≠3)  
(paar=? (cons 3 3)) → #t (da 3=3)  
  
(define paar<? (paar-operation <))  
(paar<? (cons 2 3)) → #t (da 2<3)  
(paar<? (cons 3 3)) → #f (da 3=3)
```

# Übungsblatt 6

Prof. Dr. Frank Deinzer, Frank Ebner  
Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt  
Fakultät für Informatik & Wirtschaftsinformatik FHWS

Algorithmen und Datenstrukturen I  
Wintersemester 2019/2020

```
(define paar+ (paar-operation +))
```

```
(paar+ (cons 2 3))
```

→ 5 (da  $2+3=5$ )

```
(paar+ (cons 3 3))
```

→ 6 (da  $3+3=6$ )

### Aufgabe 32 (5 Punkte) (\*)

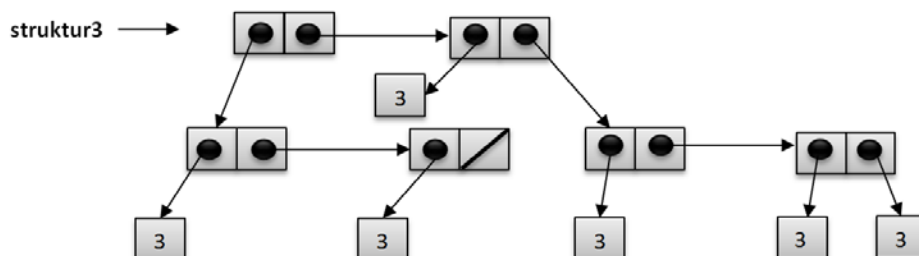
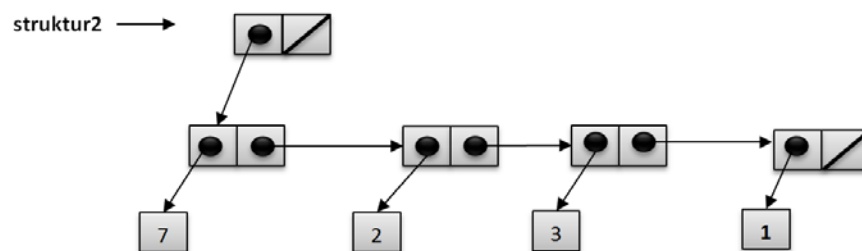
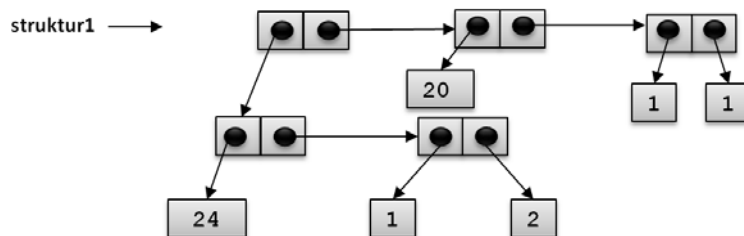
Erzeugen Sie die folgenden 4 Listen/Paar Strukturen mit Hilfe von list und cons, jeweils durch Definitionen in der Form

```
(define strukturl (...))
```

```
(define struktur2 (...))
```

```
(define struktur3 (...))
```

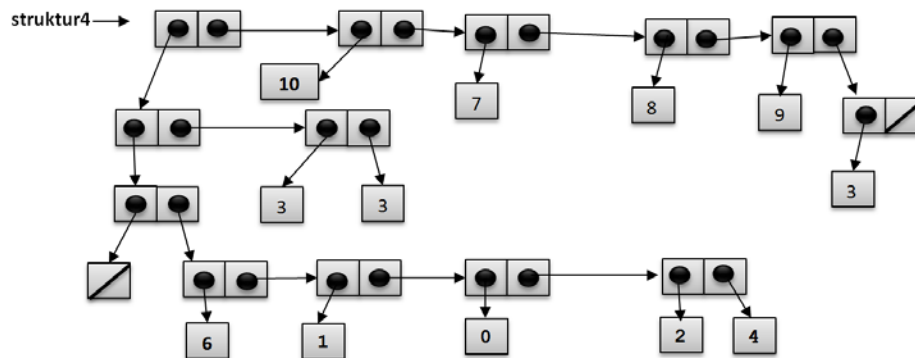
```
(define struktur4 (...))
```



# Übungsblatt 6

Prof. Dr. Frank Deinzer, Frank Ebner  
Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt  
Fakultät für Informatik & Wirtschaftsinformatik FHWS

Algorithmen und Datenstrukturen I  
Wintersemester 2019/2020



## Tafelübung T6 (\*\*)

Geben Sie für jede Eingabe das Ergebnis der Auswertung des angegebenen Ausdrucks an:

```
(define a 3)
```

```
(define b 5)
```

```
(define (funktion1 u v)
  ((lambda (a b) (+ a b u))
   (* 2 u v) (+ u v a)))
```

```
(define funktion2 (lambda (u v)
  (let ((a (* 2 u v)))
    (let ((b (+ u v a)))
      (+ a b u)))))
```

```
(funktion1 3 6)
```

```
(funktion1 17 23)
```

```
(funktion2 3 6)
```

```
(funktion2 17 23)
```

**Abgabe der Übung unter <https://algo.welearn.de> bis  
Montag, 02.12.2019, 23:59.**