

# Algorithmen und Datenstrukturen

## 05. Übungsblatt

Zeitraum: 13.–17. November 2017

### Übung 1 (AGS 3.1.15)

Implementieren Sie rekursive Funktionen zur Bestimmung des  $n$ -ten Glieds nachstehender Folgen:

- (a) die Folge der Fibonacci-Zahlen,
- (b) die Folge  $G: \mathbb{N} \rightarrow \mathbb{N}$ , definiert durch  $G(0) = 0$ ,  $G(n) = n - G(G(n-1))$  für  $n > 0$  und
- (c) die Folgen  $F, M: \mathbb{N} \rightarrow \mathbb{N}$ , definiert durch

$$\begin{aligned} F(0) &= 1, & F(n) &= n - M(F(n-1)) \quad \text{für } n > 0, \\ M(0) &= 0, & M(n) &= n - F(M(n-1)) \quad \text{für } n > 0. \end{aligned}$$

### Übung 2 (AGS 3.1.16)

Schreiben Sie eine C-Funktion `swap` mit zwei Parametern, welche die Werte der aktuellen Parameter  $x$  und  $y$  vertauscht. Ist  $x$  zudem ungerade, so soll der Wert von  $y$  um eins erhöht werden.

### Übung 3 (AGS 4.22)

Gegeben sei folgendes C-Programm.

```
1  #include <stdio.h>
2
3  void g(int x, int* y);
4
5  void f(int* x, int y){
6  /* label1 */
7  while (*x < y){
8  *x = *x * 3;
9  /* label2 */
10 g(*x, &y); /* $1 */
11 }
12 }
13
14 void g(int x, int* y){
15 /* label3 */
16 if (*y < x){
17 *y = *y * 2;
18 /* label4 */
19 if (x > *y)
20 f(&x, *y); /* $2 */
21 }
22 /* label5 */
23 }
24
25 int main(){
26 int a, b;
27 a = 3;
28 b = 6;
29 /* label6 */
30 f(&a, b); /* $3 */
31 /* label7 */
32 printf("%d", a);
33 return 0;
34 }
```

- (a) Tragen Sie den Gültigkeitsbereich jedes Objektes in eine Tabelle ein. Nutzen Sie dazu die Zeilennummern.
- (b) Setzen Sie das folgende Speicherbelegungsprotokoll fort.

Haltepunkt	RM	1	2	3	4	5	6	7	8
<i>label6</i>	-	a 3	b 6						

### Zusatzaufgabe 1 (AGS 3.1.9 \*)

Gegeben sei die Ackermann-Funktion  $\text{ack}: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ . Implementieren Sie diese in C.

$$\begin{aligned} \text{ack}(0, y) &= y + 1 && (y \geq 0) \\ \text{ack}(x, 0) &= \text{ack}(x - 1, 1) && (x > 0) \\ \text{ack}(x, y) &= \text{ack}(x - 1, \text{ack}(x, y - 1)) && (x, y > 0) \end{aligned}$$

### Zusatzaufgabe 2 (AGS 4.18)

Gegeben sei folgendes C-Programm.

```

1  #include <stdio.h>           20  /* label4 */
2                               21  if (n < 0) {
3  void g(int n, int *p);      22    *p = 3;
4                               23  } else {
5  void f(int m, int *q) {     24    f(n, &x); /* $3 */
6  /* label1 */                25    *p = 2 * x;
7  if (m > 0) {                26  }
8    g(m - 1, q); /* $1 */     27  /* label5 */
9    /* label2 */              28  }
10   g(m - 2, &m); /* $2 */     29
11   *q = *q + m;              30  int main() {
12  } else {                    31  int x;
13   *q = 1;                   32  /* label6 */
14  }                           33  f(1, &x); /* $4 */
15  /* label3 */               34  printf("%d\n", x);
16  }                           35  /* label7 */
17                             36  return 0;
18  void g(int n, int *p) {     37  }
19  int x;

```

- (a) Geben Sie den Gültigkeitsbereich jedes Objektes des Programms an. Nutzen Sie dazu die Zeilennummern.
- (b) Setzen Sie das folgende Speicherbelegungsprotokoll fort.

Haltepunkt	RM	1	2	3	4	5	6	7	8	9
<i>label6</i>	-	x ?								