

## Übungsblatt 2

**Aufgabe 1.** Wahr oder falsch?

- (a) Das folgende LOOP-Programm terminiert nicht.

$x_1 := 5; \text{LOOP } x_1 \text{ DO } x_1 := x_1 + 1 \text{ END}$

- (b) Das folgende WHILE-Programm berechnet die Funktion  $f(x) = 1$ .

$\text{WHILE } x \neq 0 \text{ DO } x := x - 2; x := x + 1 \text{ END}$

**Lösung.**

(a) **Falsch.** Die LOOP-Schleife wird  $x_1 = 5$  mal ausgeführt; das veränderte  $x_1$  wird nicht erneut gelesen. Grundsätzlich terminieren LOOP-Programme immer.

(b) **Falsch.** Das Programm berechnet stattdessen die folgende Funktion.

$$f(x) = \begin{cases} 0 & (x = 0) \\ \perp & (x \neq 0) \end{cases}$$

**Aufgabe 2.** Schreiben Sie jeweils ein LOOP-Programm, das die Funktion berechnet.

- (a)  $f(x, y) = x + y$    (b)  $g(x, y) = x \cdot y$    (c)  $\max(x, y)$    (d)  $\min(x, y, z)$

**Lösung.** Bei den folgenden LOOP-Programmen verwenden wir die Variable  $x$  jeweils als Ausgabevariable. Außerdem werden wir schrittweise einige Kurzformen einführen, die in dieser (und nur in dieser) Form ohne weitere Erklärungen auch in der Klausur verwendet werden dürfen.

- (a) Um die Summe  $x + y$  zu berechnen, erhöhen Sie  $y$ -fach den Wert von  $x$  um eins. Es ergibt sich das folgende LOOP-Programm.

LOOP  $y$  Do  
     $x := x + 1$   
  END

Im Folgenden werden wir, zur besseren Lesbarkeit, die Kurzformen

$$x := y + z \quad \text{und} \quad x := y - z$$

verwenden. Hierbei sind  $x$ ,  $y$  und  $z$  nicht notwendigerweise verschieden;  $y$  und  $z$  dürfen dabei sogar Konstanten sein.

- (b) Um das Produkt  $x \cdot y$  zu berechnen, gehen wir analog zu Teil (a) vor. Unter Verwendung der dort eingeführten Kurzform erhalten wir das folgende Programm.

```

 $x' := x; x := 0;$ 
LOOP  $y$  DO
     $x := x + x'$ 
END

```

Im Folgenden verwenden wir ebenfalls die entsprechende Kurzform  $x := y \cdot z$ .

In den weiteren Aufgabenteilen benutzen wir auch die Kurzform IF  $x = 0$  THEN  $P$  END. Die zugehörige Langform ist durch das folgende Programmfragment gegeben, wobei  $h$  eine beliebige Variable ist, die im restlichen Programm *nicht* vorkommt.

```

 $h := 1;$ 
LOOP  $x$  DO
     $h := 0$ 
END;
LOOP  $h$  DO
     $P$ 
END

```

- (c) Für den Vergleich von  $x$  und  $y$  nutzen wir, zusätzlich zu den obigen Kurzformen, die *modifizierte* Subtraktion: Es gilt  $x - y = 0$  genau dann, wenn  $x \leq y$ .

```

 $h := x - y;$ 
IF  $h = 0$  THEN
     $x := y$ 
END

```

- (d) Wir gehen analog zu Teil (c) vor und erhalten das folgende LOOP-Programm.

```

 $h := y - x;$ 
IF  $h = 0$  THEN
     $x := y$ 
END;
 $h := z - x;$ 
IF  $h = 0$  THEN
     $x := z$ 
END

```

**Aufgabe 3.** Schreiben Sie jeweils ein WHILE-Programm mit folgenden Eigenschaften.

- (a) Auf Eingabe einer Zahl  $n \geq 1$  wird der dyadische Logarithmus  $\lceil \log_2 n \rceil$  ausgegeben.
- (b) Auf Eingabe einer Zahl  $n \geq 2$  wird der kleinste Primteiler  $p$  von  $n$  ausgegeben.

### Lösung.

- (a) Um den dyadischen Logarithmus  $k$  der Zahl  $n$  zu bestimmen, erhöhen wir, ausgehend von  $k = 0$ , solange die Zahl  $k$ , bis  $2^k \geq n$  gilt. Unter Verwendung der Kurzformen für Addition und Multiplikation ergibt sich das folgende Programm.

```

INPUT n; OUTPUT k
k := 0; p := 1; h := n - p;
WHILE h ≠ 0 Do
    k := k + 1; p := 2 · p; h := n - p
END

```

- (b) Der kleinste Primteiler  $p$  einer Zahl  $n \geq 2$  ist gleich dem kleinsten Teiler von  $n$ . Es genügt, ausgehend von  $p = 2$ , die Zahl  $p$  solange zu erhöhen, bis  $n \equiv 0 \pmod{p}$  gilt. Unter Verwendung der erlaubten Kurzformen ergibt sich das folgende Programm.

```

INPUT n; OUTPUT p
p := 1; m := 1;
WHILE m ≠ 0 Do
    p := p + 1; m := 0;
    LOOP n Do
        m := m + 1; h := p - m;
        IF h = 0 THEN m := 0 END
    END
END

```

**Aufgabe 4.** Bestimmen Sie jeweils die Funktion von  $\mathbb{N}^k$  nach  $\mathbb{N}$ , die durch das zugehörige Programm berechnet wird; die Ausgabevariable des Programms ist dabei stets  $x_1$ .

(a) INPUT $x_1, x_2$ $x_3 := x_2 - x_1;$ WHILE $x_3 \neq 0$ DO $x_3 := 1$ END; $x_1 := 2 \cdot x_1;$ $x_1 := x_1 + 3$	(b) INPUT $x_1, x_2$ $M_1:$ $x_3 := x_2 \cdot x_2;$ $x_4 := x_3 - 41;$ IF $x_4 = 0$ THEN $\quad$ GOTO $M_1$ ; $\quad$ GOTO $M_2$ ; $M_2:$ $x_1 := x_1 + x_3;$ $\quad$ HALT	(c) INPUT $x_1, x_2, x_3$ $x_2 := x_2 \cdot x_2;$ $x_4 := x_2 - x_3;$ IF $x_4 = 0$ THEN $x_2 := x_3$ END; $x_3 := x_2 - x_1;$ $x_1 := x_1 \cdot x_2;$ IF $x_3 = 0$ THEN $x_1 := 1;$ LOOP $x_2$ DO $x_1 := 2 \cdot x_1$ END END
---	---	---

**Lösung.** Die Funktionen, die durch die gegebenen Programme berechnet werden, sind:

$$(a) f(x_1, x_2) = \begin{cases} 2x_1 + 3 & (x_1 \geq x_2) \\ \perp & (x_1 < x_2) \end{cases}$$

$$(b) f(x_1, x_2) = \begin{cases} x_1 + x_2^2 & (x_2 \geq 7) \\ \perp & (x_2 < 7) \end{cases}$$

$$(c) f(x_1, x_2, x_3) = \begin{cases} 2^{g(x_2, x_3)} & (x_1 \geq g(x_2, x_3)) \\ x_1 \cdot g(x_2, x_3) & (x_1 < g(x_2, x_3)) \end{cases}, \text{ mit } g(x_2, x_3) = \max(x_2^2, x_3)$$