

Übungen zur Vorlesung Grundlagen der Programmierung I Blatt 6

Aufgabe 1:

Welche der folgenden Programme berechnen die Funktion

$$f: \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z} \text{ mit}$$

$$f(m,n) = \begin{cases} m \cdot n, & \text{falls } n \geq 0, \\ \perp, & \text{sonst?} \end{cases}$$

<p>a) <u>def</u> m,n,p: Zahl; lies(m); lies(n); <u>solange</u> n≠0 <u>tue</u> p←p+m; n←n/2 <u>ende</u>; zeige(p) .</p>	<p>b) <u>def</u> a,b,c: Zahl; lies(a); lies(b); c←0; <u>solange</u> b≠0 <u>tue</u> c←a+c; b←b-1 <u>ende</u>; zeige(c) .</p>	<p>c) <u>def</u> m,n,p: Zahl; lies(m); lies(n); <u>wenn</u> n<0 <u>oder</u> m<0 <u>dann</u> zeige("⊥") <u>sonst</u> p←0; <u>solange</u> n≠0 <u>tue</u> p←-m+p; n←n-1 <u>ende</u>; zeige(p) <u>ende</u> .</p>
<p>d) <u>def</u> m,n,p: Zahl; lies(m); lies(n); p←0; <u>solange</u> n≥0 <u>tue</u> p←m+p; n←n-1 <u>ende</u>; zeige(p) .</p>	<p>e) <u>def</u> m,n,p: Zahl; lies(m); lies(n); p←0; <u>solange</u> n<0 <u>tue</u> p←-m+p; n←n-1 <u>ende</u>; <u>solange</u> n>0 <u>tue</u> p←p+m; n←n-1 <u>ende</u>; zeige(p) .</p>	

Aufgabe 2:

Bestimmen Sie die Semantik des folgenden Programms, indem Sie exakt die berechnete Funktion $f: \mathbb{Z} \rightarrow \mathbb{Z}$ angeben:

```

def p,n: Zahl;
lies(n);
p ← 1;
solange n≠1 tue
    p ← p*n;
    n ← n-1
ende;
zeige(p) .

```

Aufgabe 3:

Schreiben Sie eine Funktion in PRO mit einer Zahlenfolge als Parameter, die die gespiegelte Version der Zahlenfolge als Ergebnis liefert, die also folgende Funktion f berechnet:

$f: \mathbb{Z}^* \rightarrow \mathbb{Z}^*$ mit

$f([x_1, x_2, \dots, x_n]) = [x_n, x_{n-1}, \dots, x_1]$ für $n \geq 0$.

Zeigen Sie anschließend möglichst präzise, daß Ihre Funktion tatsächlich das Gewünschte leistet.

Aufgabe 4:

Schreiben Sie ein Programm in PRO, das eine Zahlenfolge einliest und diese aufsteigend sortiert wieder ausgibt. Das Programm soll folgende Strategie verwenden: Die Folge wird logisch in zwei Teilfolgen f_1 und f_2 zerlegt. f_1 besteht zunächst aus dem ersten, f_2 aus den restlichen Elementen der Ausgangsfolge. Nun wird fortlaufend das erste Element der Folge f_2 entfernt und an die bezüglich der Sortierung richtige Stelle in die Folge f_1 eingefügt. Zum Schluß ist f_2 leer und f_1 enthält das gesuchte Ergebnis. Diese Vorgehensweise nennt man *Sortieren durch Einfügen*.