Informatik

inf11 20.04.2023

Aufgabe 16
Nach Eingabe einer reellen Zahl **a** und einer natürlichen Zahl **n**, $n \geq 0$, berechnet der Algorithmus **POTENZ** den Wert $\mathbf{a^n}$ und gibt diesen aus.

a) Formuliere den Potenzierungsalgorithmus iterativ (wahlweise while- oder for-Schleife) als Struktogramm und Python-Programm.
Implementiere eine Zählvariable z, um die Anzahl der Schleifendurchläufe zu bestimmen und auszugeben.

b) Vergleiche den Potenzierungs-Algorithmus aus Aufgabe 13 mit dem Algorithmus aus 16.a) hinsichtlich der Anzahl der benötigten Schleifendurchläufe.

c) Wegen $\mathbf{a^n = a \cdot a^{n-1}}$ und $\mathbf{a^0 = 1}$ läßt sich die Potenz als rekursive Funktion **f** definieren, die jedem **n** den Wert $\mathbf{a^n}$ zuordnet:

Rekursionsanfang: $\mathbf{f(0) = 1}$
Rekursionsvorschrift: $\mathbf{f(n) = a \cdot f(n-1)}$ falls $n > 0$

Beispiel:
a = 7, n = 4

$f(4) = 7^4 = 7 \cdot 7^3 = 7 \cdot (7 \cdot 7^2) = 7 \cdot (7 \cdot (7 \cdot 7^1)) = 7 \cdot (7 \cdot (7 \cdot (7 \cdot 7^0))) = 7 \cdot (7 \cdot (7 \cdot (7 \cdot 1))) = 7 \cdot (7 \cdot (7 \cdot 7)) = 7 \cdot (7 \cdot 49) = 7 \cdot 343 = 2401$

Formuliere ein Python-Programm zur Berechnung von $\mathbf{a^n}$ mit rekursivem Funktionsaufruf!

Nachtrag: Lösungen zu **Aufgabe 13**

a)

```
# Potenz a^n iterativ (Aufgabe 13)

a = float(input('a = '))
n = int(input('n = '))

b = a
u = n
p = 1

while u > 0:
    if u % 2 != 0:
        u = u - 1
        p = p * b
    u = u / 2
    b = b * b

print()
print (a,'^',n,' =',p)
```

b) Trace für a = 2, n = 7

	a	n	b	u	p	u ungerade	u > 0
vor dem 1. SD	2	7	2	7	1	+	+
vor dem 2. SD	2	7	4	3	2	+	+
vor dem 3. SD	2	7	16	1	8	+	+
nach dem 3. SD	2	7	256	0	128	–	–

c) Trace für n = 18

	a	n	b	u	p	u ungerade	u > 0
vor dem 1. SD	a	18	a	18	1	–	+
vor dem 2. SD	a	18	a^2	9	1	+	+
vor dem 3. SD	a	18	a^4	4	a^2	–	+
vor dem 4. SD	a	18	a^8	2	a^2	–	+
vor dem 5. SD	a	18	a^{16}	1	a^2	+	+
nach dem 5. SD	a	18	a^{32}	0	a^{18}	–	–

Vermutung:
Für eine reelle Zahl **a** und eine natürliche Zahl **n**, $n \geq 0$, berechnet der Algorithmus die Potenz $\mathbf{a^n}$ und gibt deren Wert aus.

Ausblick:
Die vorgenannte Vermutung läßt sich auch streng beweisen; hierzu zeigt man, daß die Beziehung

$$\mathbf{p \cdot b^u = a^n}$$

vor und nach jedem Schleifendurchlauf erfüllt, also invariant gegenüber Schleifendurchläufen ist (eine solche Beziehung heißt Schleifeninvariante).

Da während jedem Schleifendurchlauf entweder u halbiert wird, falls u gerade ist, oder u – 1 halbiert wird, falls u ungerade ist, nimmt u nach endlich vielen Schleifendurchläufen den Wert 0 an, und der Algorithmus terminiert (u>0 ist False, falls u=0 ist).

Sobald u den Wert 0 annimmt, folgt wegen $\mathbf{b^0 = 1}$ aus der Schleifeninvariante:

$$\mathbf{p = a^n}$$