

Name: Musterlösung

Aufgabe 4 — Programmierung — 22 Punkte

Gegeben sei der Datentyp für einen Binärbaum

```
struct BinBaum {  
    int inhalt;  
    BinBaum *l,*r;  
};
```

Die Höhe eines Blattes ist bestimmt durch die Anzahl der Knoten auf dem Weg vom Blatt zur Wurzel. Wir wollen hier nun eine C++-Funktion

```
int minho(BinBaum * root)
```

schreiben, um in einem Binärbaum die minimale Höhe eines Blattes zu bestimmen. Der Parameter `root` zeigt dabei auf die Wurzel des Baums.

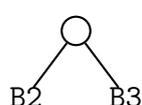
Beispiele: Der linke Baum hat eine minimale Höhe von 2 (das Blatt im rechten Unterbaum der Wurzel), der rechte Baum hat eine minimale Höhe von 4 (das einzige Blatt im Binärbaum)!



- a) (6 Punkte) Seien die Werte $m1=\text{minho}(B1)$, $m2=\text{minho}(B2)$ und $m3=\text{minho}(B3)$ für die drei Binärbäume $B1$, $B2$ und $B3$ bereits bestimmt. Geben Sie an, welchen Wert die Funktion `minho()` in folgenden 3 Fällen errechnet:

die minimale Höhe beträgt 1

 die minimale Höhe beträgt $1+m1$

 die minimale Höhe beträgt $1+\min(m2,m3)$

- b) (16 Punkte) Schreiben Sie nun (nur) die Funktion `minho()`. Funktionen zur Erzeugung von Binärbäumen oder `int main()` sind nicht erforderlich.

```
int minho(BinBaum * root) {

    if (root==0) return 0; // leerer Baum
    // if (root->l==0 && root->r==0) return 1; else
    // Die Abfrage auf das Blatt kann entfallen wegen Abfrage auf root==0
    if (root->l!=0 && root->r==0) return 1+minho(root->l);
    else if (root->l==0 && root->r!=0) return 1+minho(root->r);
    else {
        int ml=minho(root->l);
        int mr=minho(root->r);
        if (ml<mr) return 1+ml; else return 1+mr; // bei Blatt = 1+0
    }

}
```