

Aufgabe 3.2: Matryoshka-Puppen iterativ

```
/* Öffnet die Puppen, entfernt die enthaltene Puppe und schließt die Puppen, bis die massive Puppe gefunden wurde. */
```

```
void nimmMassivePuppe(Puppe M) {  
    while (M ist nicht massiv) {  
        öffne M;  
        nimm Inhalt von M;  
        schließe M;  
        M = Inhalt von M;  
    }  
}
```

Beschreibung:

1. Puppe M wird geöffnet
2. Inhalt von Puppe M wird herausgenommen
3. Puppe M wird geschlossen
4. Der Vorgang wird mit dem Inhalt von Puppe M wiederholt, bis M eine massive Puppe ist

Aufgabe 3.4: Pi rekursiv

b)

Die in Aufgabe 2.6 implementierte Variante ist vorzuziehen, weil eine Rekursion hier verschiedene Nachteile mit sich bringt.

Bei der Variante aus Aufgabe 2.6 ist eine viel genauere Annäherung möglich, bis zu 2147483647 Schritte (Maximalwert für int).

Bei der rekursiven Methode ist schon nach 4913 Schritten Schluss, ab 4914 gibt es einen Stack overflow error.

Aufgabe 3.5: binäre Suche

a) Man legt ein sortiertes Feld zugrunde, und das gesuchte Element (Haus) wird mit dem jeweils mittleren Element des Feldes verglichen. Entspricht das gesuchte Element nicht dem verglichenen Element des Feldes, wird nur die jeweils linke oder rechte Hälfte des Feldes weiter betrachtet. Jeder Vergleich teilt den jeweiligen Suchraum in 2 Hälften, und zwar so lange bis

1. das gesuchte Element gefunden wurde oder
2. sich der Suchraum nicht mehr weiter unterteilen lässt.

Wir konzentrieren uns hier auf den Anfangsbuchstaben H, Ohne Kenntnis über die Verteilung der Anfangsbuchstaben im Wörterbuch zu haben.

A --- (mitte) ---- Z

Vergleich: $H == \text{(mitte)}$ → Fertig

oder

$H < \text{(mitte)}$ → weitersuchen von H in $(\text{mitte} - 1)$

oder

$H > \text{(mitte)}$ → weitersuchen in $(\text{mitte} + 1)$

b) keine Antwort