Rekursiv Listen Erstellen

26.02.2020

L1) Schreibe eine rekursive Funktion woerter(*n*), die alle Wörter der Länge *n* ausgibt, die nur aus den Buchstaben A und B bestehen.

woerter(3) = [‘AAA’, ‘AAB’, ‘ABA’, ‘ABB’, ‘BAA’, ‘BAB’, ‘BBA’, ‘BBB’]

L2) Schreibe eine rekursive Funktion gute_woerter(*n*), die alle Wörter der Länge *n* ausgibt, die nur aus den Buchstaben A und B bestehen, aber keine aufeinanderfolgenden BB enthalten. (*Das ist doch schwer auszusprechen.*)

gute_woerter(3) = [‘AAA’, ‘AAB’, ‘ABA’, ‘BAA’, ‘BAB’]

L3) Schreibe eine rekursive Funktion schoene_woerter(*n*), die alle Wörter der Länge *n* ausgibt, die nur aus den Buchstaben A und B bestehen, und nicht mehr als zwei gleiche aufeinanderfolgende Buchstaben enthalten. (*Das passiert doch in natürlichen Sprachen nicht. Ausser “Schi**fff**ahrtsgesellschaft” :-)*)

schoene_woerter(3) = [‘AAB’, ‘ABA’, ‘ABB’, ‘BAA’, ‘BAB’, ‘BBA’]

L4) Schreibe eine rekursive Funktion woerter(*n, k*), die alle Wörter der Länge *n* ausgibt, die nur aus den Buchstaben A und B bestehen, und genau *k*-Mal ‘B’ enthalten.

woerter(4, 3) = [‘ABBB’, ‘BABB’, ‘BBAB’, ‘BBBA’]

L5) Schreibe eine rekursive Funktion untermengen(*liste*), die alle Untermengen der Liste *liste* ausgibt.

untermengen([1, 2, 3]) = [[], [1], [2], [3], [1, 2], [1, 3], [2, 3], [1, 2, 3]]

L6) Schreibe eine rekursive Funktion untermengen(*liste*, *k*), die alle Untermengen der Liste *liste* ausgibt, die genau *k* Elemente haben.

untermengen([‘A’, ‘B’, ‘C’, ‘D’], 2) = [[‘A’, ‘B’], [‘A’, ‘C’], [‘A’, ‘D’], [‘B’, ‘C’], [‘B’, ‘D’], [‘C’, ‘D’]]

L7) Schreibe eine rekursive Funktion ordnungen(*liste*), die alle mögliche Ordnungen der Liste *liste* ausgibt. (*Alle Elemente der Liste sind unterschiedlich.*)

ordnungen(['Salat', 'Wurst', 'Kuchen']) = [['Salat', 'Wurst', 'Kuchen'], ['Salat', 'Kuchen', 'Wurst'], ['Wurst', 'Salat', 'Kuchen'], ['Wurst', 'Kuchen', 'Salat'], ['Kuchen', 'Salat', 'Wurst'], ['Kuchen', 'Wurst', 'Salat']]

L8) Schreibe eine rekursive Funktion anagramme(*wort*), die alle Anagramme des Wortes *wort* ausgibt.

anagramme(‘kiwi’) = ['iikw', 'iiwk', 'ikiw', 'ikwi', 'iwik', 'iwki', 'kiiw', 'kiwi', 'kwii', 'wiik', 'wiki', 'wkii']

Rekursion und Backtracking (für Fortgeschrittene)

B1) Schreibe eine Funktion damen(*n*), die eine Konfiguration von *n* Damen auf einem (*n* x *n*)-Schachbrett ausgibt, wobei sich keine zwei der Damen schlagen können.

> damen(8)

```
.......D
...D....
D.......
..D.....
.....D..
.D......
......D.
....D...
```

B2) Schreibe eine Funktion, die **alle** solche Konfigurationen ausgibt.

B3) Löse das Schlangenwürfel-Puzzle.