

## Kompaktkurs Einführung in die Programmierung Übungsblatt 12: Vermischtes (2)

### Lernziele

- Wiederholung und Erweiterung ausgewählter Themen des Kurses.
- Einlesen von Dateiinhalten.
- Doppelt verkettete Listen als Datenstruktur.

### 1) Konkatenation von Zeichenketten (P)

Schreiben Sie eine Funktion der Signatur

```
char *concat(char *s1, char *s2)
```

die die beiden übergebenen Zeichenketten aneinanderhängt. `concat("Eins", "Zwei")` sollte also die Zeichenkette "EinsZwei" liefern. Dazu soll `concat` zunächst Speicherplatz für eine neue Zeichenkette passender Größe reservieren (mit `new` bzw. `malloc`) und dann alle Zeichen von `s1` und `s2` hintereinander hineinkopieren. Vergessen Sie nicht das `'\0'`-Zeichen am Ende der zu erzeugenden Zeichenkette!

### 2) Klammern und Catalan-Zahlen (P)

- i) Als Folge natürlicher Zahlen treten die Catalan-Zahlen in zahlreichen Zählproblemen auf. Die  $n$ -te Catalan-Zahl  $C_n$  ist gegeben durch

$$C_n = \frac{1}{n+1} \binom{2n}{n} = \frac{(2n)!}{(n+1)!n!}, \quad \text{für } n \geq 0. \quad (1)$$

Eine Anwendung ist die Bestimmung der Anzahl aller korrekten Klammerungen für das Vorliegen von  $n$  Klammerpaaren. Hierfür ergibt sich  $C_3 = 5$  da es 5 korrekte Klammerungen für 3 Klammerpaare gibt:

`((()))` , `()(())` , `()()()` , `((()))` , `((()()))`

Schreiben Sie eine Funktion `catalan()` die die  $n$ -te Catalan-Zahl berechnet und ausgibt.

*Hinweis:* Um Überläufe im Wertebereich zu vermeiden denken Sie daran, dass  $(2n)!$  nicht explizit berechnet werden muss. Über Vereinfachung von (1) und der Verwendung von Typen mit größerem Wertebereich als `int` sollte auch  $C_{25}$  berechenbar sein.

- ii) In dieser Teilaufgabe sollen „Formeln“, die nur aus Klammern '(' und ')' aufgebaut sind, auf korrekte Klammerung überprüft werden, d.h. ob öffnende und schließende Klammern passend zugeordnet werden können.

Ein Beispiel für eine korrekte Formel:

`()(())(())()`

Zwei nicht korrekte Formeln

`((()` (zu viele öffnende Klammern)

`))(` (die zweite schließende Klammer „passt nicht“)

Schreiben Sie eine zusätzliche Funktion `klammern()` in ihr Hauptprogramm, welche eine in einer Zeichenkette gespeicherte Formel überprüft und eine entsprechende Meldung über die Korrektheit der Klammerung ausgibt.

### 3) Datei ausgeben (P)

Schreiben Sie ein Programm welches sich selbst bei Aufruf ausdrückt. Sie könnten hierzu, nachdem Sie sich mit den Funktionen vertraut gemacht haben, ihr Programm `ausgabe.cpp` mit dem Befehl  `fopen()` einlesen, sowie die Funktion `fgets()` und `fclose()` benutzen.

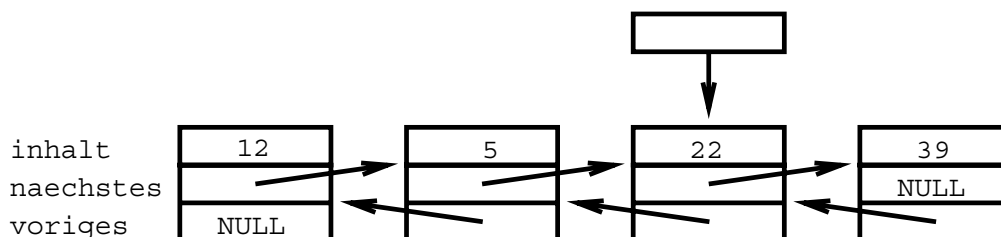
### 4) Zweifach verkettete Listen (P)

Eine zweifach verkettete Liste besteht aus Elementen mit

- dem Inhalt – hier ein `int inhalt`
- einem Zeiger `naechstes` auf den folgenden Listeneintrag und
- einem Zeiger `voriges` auf den vorhergehenden Listeneintrag.

Dabei ist im ersten Element einer Liste `voriges==NULL` und im letzten `naechstes==NULL`. Die Liste wird repräsentiert durch einen Zeiger auf ein beliebiges Element darin – also nicht notwendigerweise auf den Anfang.

Ein Beispiel:



- i) Schreiben Sie eine Funktion `kopiere`, die eine komplette Kopie einer solchen Liste erstellt, d.h. als Argument eine Liste hat und
- neuen Speicher reserviert,
  - die Inhalte kopiert,
  - die neuen Elemente untereinander richtig verknüpft und
  - einen Zeiger auf eines der neuen Listenelemente zurückliefert.
- ii) ★ Achten Sie wiederum auf korrekte Freigabe aller Daten des Freispeichers.