

Aufgabe 5.1

Geben Sie den für `int matrix[2][3]` benötigten Speicherplatz in Byte an, wenn `sizeof(int)` als 4 angenommen wird. An welcher Bytenummer beginnt das Element `matrix[i][j]` relativ zum Beginn des Arrays? Überprüfen Sie Ihre Berechnungen mit einem kleinen Programm!

Aufgabe 5.2

Gegeben ist das Template `Array2d` aus der Vorlesung (`array2d.h`). Erweitern Sie dieses um die folgenden 2 Funktionalitäten:

- Eine Methode `mmult` zur Matrix-Multiplikation. Dabei müssen die Arrays bzgl. ihrer Dimensionen auf 'Kompatibilität' geprüft werden (z.B. mit einem `assert`). Die Methode gibt das Ergebnis-Array zurück!
- Einen Konstruktor, der ein statisches, zweidimensionales C-Array übergeben bekommt, und daraus ein `Array2d` erstellt mit den Dimensionen und Werten des C-Arrays.

```
#include "array2d.h"

int main() {

    int a[3][4] = { {1,2,3,4},
                    {2,6,1,8},
                    {5,1,2,1} };

    int b[4][1] = { 1,9,11,5 };

    Array2d<int> A(a);
    Array2d<int> B(b);
    Array2d<int> C = mmult(A,B);
    printArray(C); // 72 107 41
}
```

Der nebenstehende Code sollte dann funktionieren!

Hinweis: Wenn Sie ein C-Array als Referenz übergeben, bleiben alle Dimensionen des Arrays auch in der aufgerufenen Methode erhalten (auch die erste)!

Aufgabe 5.3

In der Vorlesung wurde gezeigt, eine einfache verkettete Liste für den Datentyp `int` realisiert werden kann (ab Folie 116). Erweitern Sie diesen Ansatz so, dass die Liste den generischen Datentyp `T` verwaltet. Dazu müssen Sie die Funktionen zur Bearbeitung der Liste als Template-Funktionen formulieren. Zusätzlich soll die `insert`-Funktion einen dritten Parameter bekommen, der als Funktionszeiger eine Vergleichsmethode übergibt, die zum sortierten Einfügen in die Liste notwendig ist. Wenn dieser Parameter nicht angegeben wird, soll in aufsteigender Reihenfolge sortiert werden. Schreiben Sie ein Hauptprogramm, das die neue Funktionalität testet!

Aufgabe 5.4

Schreiben Sie eine Funktion `void leerzeichenEntfernen(char* s)`, die alle Leerzeichen im char-Array `s` entfernt. Zum Beispiel soll aus "a bb ccc d" die Zeichenkette "abbcccd" werden. Verwenden Sie dabei Zeiger.

Aufgabe 5.5

Schreiben Sie ein Programm, das Dateien, deren Namen in der Kommandozeile angegeben werden, auf der Standardausgabe ausgibt. Zum Beispiel könnte der Befehl

```
prog datei1.cpp XXX datei2.cpp
```

dazu führen, dass die Dateien `datei1.cpp` und `datei2.cpp` auf dem Bildschirm angezeigt werden und dass es eine Fehlermeldung "Datei XXX nicht gefunden!" gibt.

Aufgabe 5.6

Schreiben Sie ein Programm, das alle in einer Datei vorkommenden Namen ausgibt. Ein Name ist dabei so definiert: Er beginnt mit einem Buchstaben, anschließend folgen beliebig viele Buchstaben und Ziffern. Der Unterstrich zählt auch als Buchstabe.

Hinweis: In `<cctype>` finden Sie verschiedene Hilfsmethoden zur Überprüfung von `char`-Werten!

Zusatzaufgabe 5.7

Schreiben Sie zur Ausgabe von dreidimensionalen Arrays eine Template-Funktion `tabellenausgabe3D(Feldtyp T, size_t n)` entsprechend dem Muster für zwei Dimensionen (siehe Vorlesungsfolien).