



**Aufgabe 1.** Implementiere einige Funktionen um mit quadratischen Matrizen umzugehen:

- a) Eine Funktion, die Speicher für eine quadratische Matrix allokiert, eine um ihn freizugeben, eine um sie auszugeben, eine um sie aus einer Datei einzulesen und eine, um sie zur Einheitsmatrix zu initialisieren (das ist die Matrix mit 1en auf der Hauptdiagonalen und 0en sonst):

```
1 #include <stdio.h>
2
3 double **matrix_alloc(int n);
4 void     matrix_free(double **A, int n);
5 void     matrix_print(double **A, int n);
6 double **matrix_read(FILE *fp, int *n);
7 double **matrix_id(double **A, int n);
```

- b) Eine Funktion, die zwei solche Matrizen miteinander multipliziert und eine neue Matrix zurück gibt. Für zwei  $n \times n$ -Matrizen  $A = (a_{ij})$  und  $B = (b_{ij})$  ist  $A \cdot B = C = (c_{ij})$  über folgende Formel definiert:

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

- c) Eine Funktion, die die Spur einer Matrix ausrechnet: Das ist die Summe der Einträge auf der Hauptdiagonalen.
- d) Auf der Webseite des Kurses gibt es zwei Dateien mit Matrizen. Lese die beiden Matrizen ein, multipliziere sie, und bestimme die Spur des Ergebnisses. Wenn dein Ergebnis stimmt, bekommst du eine Dose Cola, wenn wir beeindruckt sind.

**Aufgabe 2.** Gegeben sei eine Datei, in der ausschließlich Zahlen stehen. In der ersten Zeile stehe eine natürliche Zahl, die angibt wie viele Zahlen noch folgen.

- a) Schreibe ein Programm, dass diese Datei einliest, die Zahlen sortiert und die Datei mit der sortierten Liste überschreibt.
- b) Modifiziere dein Programm nun so, dass in der ersten Zeile nicht mehr stehen muss, wie viele Zeilen noch folgen.