

---

**Familienname:**

**Vorname:**

**Matrikelnummer:**

Aufgabe 1 (1 Punkte):  
Aufgabe 2 (3 Punkte):  
Aufgabe 3 (2 Punkte):  
Aufgabe 4 (2 Punkte):  
Aufgabe 5 (2 Punkte):  
Aufgabe 6 (3 Punkte):  
Aufgabe 7 (2 Punkte):  
Aufgabe 8 (4 Punkte):  
Aufgabe 9 (4 Punkte):  
Aufgabe 10 (3 Punkte):  
Aufgabe 11 (4 Punkte):  
Aufgabe 12 (4 Punkte):  
Aufgabe 13 (3 Punkte):  
Aufgabe 14 (3 Punkte):

---

Gesamtpunktzahl:

---

**Schriftlicher Test (120 Minuten)**  
**VU Einführung ins Programmieren für TM**

**23. Jänner 2014**

---

**Aufgabe 1 (1 Punkt).** Was ist eine rekursive Funktion?

**Aufgabe 2 (3 Punkte).** Schreiben Sie eine rekursive C-Funktion `binomial`, die mit Hilfe des Additionstheorems

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \quad \text{für } k, n \in \mathbb{N} \text{ mit } 2 \leq k < n$$

den Binomialkoeffizienten berechnet. Beachten Sie dabei

$$\binom{n}{1} = n \quad \text{und} \quad \binom{n}{n} = 1 \quad \text{für alle } n \geq 1.$$

**Aufgabe 3 (2 Punkte).** Was sind Lifetime und Scope einer Variable?

**Aufgabe 4 (2 Punkte).** Schreiben Sie einen C-Strukturdatentyp `Matrix` zur Speicherung einer dynamischen Matrix  $A \in \mathbb{R}^{m \times n}$ . In der Struktur sollen die Dimensionen  $m, n \in \mathbb{N}$  sowie die Koeffizienten  $A_{jk}$  in Form eines dynamischen Vektors  $a \in \mathbb{R}^{mn}$  gespeichert werden.

**Aufgabe 5 (2 Punkte).** Die Matrixeinträge der Matrix  $A \in \mathbb{R}^{m \times n}$  sollen spaltenweise im Vektor  $a \in \mathbb{R}^{mn}$  gespeichert werden. Leiten Sie eine Formel her, die für Indizes  $j, k$  den Index  $\ell$

bestimmt, sodass bei spaltenweiser Speicherung der Matrix  $A_{jk} = a_\ell$  gilt. Begründen Sie Ihre Formel!

**Aufgabe 6 (3 Punkte).** Schreiben Sie eine C-Funktion `newMatrix`, die eine dynamische Matrix  $A \in \mathbb{R}^{m \times n}$  allokiert und initialisiert.

**Aufgabe 7 (2 Punkte).** Schreiben Sie eine C-Funktion `delMatrix`, die eine dynamische Matrix  $A \in \mathbb{R}^{m \times n}$  freigibt.

**Aufgabe 8 (4 Punkte).** Die folgende C-Funktion soll den größten Koeffizienten einer dynamischen Matrix  $A \in \mathbb{R}^{m \times n}$  zurückgeben. Korrigieren Sie allfällige syntaktische und logische Fehler direkt im Code und begründen Sie kurz, was falsch ist.

```
void maxabs(Matrix* A) {
    int j,k;
    double max = 0;

    for (j=0; j<A.m; j++) {
        for (k=0; k<A.n; ++k) {
            if (A[j][k] > max); {
                max = A[j][k];
            }
        }
    }
    return max;
}
```

**Aufgabe 9 (4 Punkte).** Schreiben Sie eine C-Funktion `znorm`, die die Zeilensummennorm

$$\|A\| := \max_{j=0, \dots, m-1} \sum_{k=0}^{n-1} |A_{jk}|$$

einer dynamischen Matrix  $A \in \mathbb{R}^{m \times n}$  berechnet und zurückgibt.

**Aufgabe 10 (3 Punkte).** Was versteht man unter Aufwand? Welchen Aufwand hat Ihre Implementierung der Zeilensummennorm? Erklären Sie in diesem Zusammenhang auch die Landau- $\mathcal{O}(\cdot)$ -Notation.

**Aufgabe 11 (4 Punkte).** Schreiben Sie eine C++ Klasse `Matrix` zur Speicherung von dynamischen Matrizen  $A \in \mathbb{R}^{m \times n}$ . Die Koeffizienten sollen als `vector<double>` gespeichert werden. Die Klasse soll die folgenden Methoden enthalten:

- einen Standardkonstruktor, der eine  $1 \times 1$  Null-Matrix erzeugt;
- einen Konstruktor, der zu gegebenen Dimensionen  $m$  und  $n$  eine Matrix  $A \in \mathbb{R}^{m \times n}$  allokiert und initialisiert;
- einen Konstruktor, der einen `double`-Wert in eine Matrix  $A \in \mathbb{R}^{1 \times 1}$  konvertiert;
- Methoden `getM` und `getN`, die die Dimensionen  $m$  bzw.  $n$  zurückliefern;
- Methoden `get` und `set`, die Koeffizienten  $A_{jk}$  der Matrix auslesen bzw. zuweisen.

**Hinweis.** An dieser Stelle soll nur die Klassendefinition erfolgen, aber keine Funktionalität programmiert werden. Überlegen Sie auch wie Sie die Qualifier setzen.

**Aufgabe 12 (4 Punkte).** Implementieren Sie die drei Konstruktoren der Klasse `Matrix`.

**Aufgabe 13 (3 Punkte).** Schreiben Sie eine C++ Funktion `isSymmetric`, die überprüft, ob eine Matrix  $A \in \mathbb{R}^{m \times n}$  symmetrisch ist, d.h. es gilt  $m = n$  und  $A_{jk} = A_{kj}$  für alle  $j, k = 1, \dots, n$ . Die Rückgabe sei vom Datentyp `bool`, d.h. `true` falls  $A$  symmetrisch ist und `false` sonst.

**Aufgabe 14 (3 Punkte).** Implementieren Sie den Operator `+` für die Klasse `Matrix`, um zwei Matrizen  $A, B \in \mathbb{R}^{m \times n}$  zu addieren. Sie dürfen bei der Implementierung davon ausgehen, dass die Dimensionen der beiden Matrizen ident sind.