
Familienname:

Vorname:

Matrikelnummer:

Email:

Aufgabe 1 (2 Punkte):
Aufgabe 2 (2 Punkte):
Aufgabe 3 (2 Punkte):
Aufgabe 4 (3 Punkte):
Aufgabe 5 (2 Punkte):
Aufgabe 6 (4 Punkte):
Aufgabe 7 (3 Punkte):
Aufgabe 8 (2 Punkte):
Aufgabe 9 (2 Punkte):
Aufgabe 10 (4 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

03. März 2014 (Wintersemester 2013/14)

Aufgabe 1 (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. **Hinweis:** Arbeiten Sie in den Aufgaben 2–6 mit dem Strukturdatentyp `Matrix` aus Aufgabe 1.

Aufgabe 2 (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 ℓ bestimmt, sodass bei spaltenweiser Speicherung der Matrix $A_{jk} = a_\ell$ gilt. Begründen Sie Ihre Formel!

Aufgabe 3 (2 Punkte). Schreiben Sie eine C-Funktion `getMatrixAjk`, die den Wert des Koeffizienten A_{jk} einer Matrix $A \in \mathbb{R}^{m \times n}$ zurückgibt.

Aufgabe 4 (3 Punkte). Schreiben Sie eine C-Funktion `newMatrix`, die eine dynamische Matrix $A \in \mathbb{R}^{m \times n}$ allokiert und initialisiert.

Aufgabe 5 (2 Punkte). Schreiben Sie eine C-Funktion `delMatrix`, die eine dynamische Matrix $A \in \mathbb{R}^{m \times n}$ freigibt.

Aufgabe 6 (4 Punkte). Schreiben Sie eine C-Funktion `snorm`, die die Spaltensummennorm

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

einer dynamischen Matrix $A \in \mathbb{R}^{m \times n}$ berechnet und zurückgibt. **Hinweis.** Sie dürfen die Existenz von (sinnvollen) Zugriffsfunktionen `getMatrixM` und `getMatrixN` voraussetzen.

Aufgabe 7 (3 Punkte). Was versteht man unter Aufwand? Welchen Aufwand hat Ihre Implementierung der Spaltensummennorm? Erklären Sie in diesem Zusammenhang auch die Landau- $\mathcal{O}(\cdot)$ -Notation.

Aufgabe 8 (2 Punkte). Was sind die Bestandteile einer Gleitkommazahl? Wie berechnet sich daraus der Wert der Gleitkommazahl? Was ist eine normalisierte Gleitkommazahl?

Aufgabe 9 (2 Punkte). Was sind Lifetime und Scope einer Variable?

Aufgabe 10 (4 Punkte). Was ist der Output der folgenden rekursiven C-Funktion

```
int function(int n, int k) {
    int val = 0;
    if (k == 1) {
        val = n;
    }
    else if (k == n) {
        val = 1;
    }
    else {
        val = function(n-1,k) + function(n-1,k-1);
    }
    printf("%d %d %d\n",n,k,val); /* Hier ist der Output */
    return val;
}
```

wenn sie mit

```
result = function(5,2);
```

aufgerufen wird? Was macht die Funktion?

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 1×1 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 `getAjk` und `setAjk`, 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 `isUpperTriangular`, die überprüft, ob eine Matrix $A \in \mathbb{R}^{m \times n}$ eine obere Dreiecksmatrix ist, d.h. es gilt $m = n$ und $A_{jk} = 0$ für alle $j, k = 0, \dots, n - 1$ mit $j > k$. Die Rückgabe sei vom Datentyp `bool`, d.h. `true` falls A obere Dreiecksmatrix ist und `false` sonst. **Hinweis.** Verwenden Sie die Klasse `Matrix`. Sie dürfen die Existenz aller benötigten Zugriffsmethoden der Klasse `Matrix` voraussetzen.

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. **Hinweis.** Sie dürfen die Existenz aller benötigten Zugriffsmethoden der Klasse `Matrix` voraussetzen.