
Familienname:

Vorname:

Matrikelnummer:

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

Gesamtpunktzahl:

Schriftlicher Test zu C (90 Minuten)
VU Einführung ins Programmieren für TM (WS 2009/10)

26. November 2009

Aufgabe 1 (2 Punkte). Schreiben Sie einen Struktur-Datentyp `CDouble` zur Speicherung komplexer Zahlen $z \in \mathbb{C}$. In der Struktur sollen Realteil `real` und Imaginärteil `imag` gespeichert werden. Verwenden Sie diesen Datentyp in allen folgenden Aufgaben.

Lösung zu Aufgabe 1.

Folgende Zugriffsfunktionen stehen für den Datentyp `CDouble` zur Verfügung:

- `CDouble* newCDouble(double real, double imag)`
- `CDouble* delCDouble(CDouble* z)`
- `double getCDoubleReal(CDouble* z)`
- `double getCDoubleImag(CDouble* z)`
- `void setCDoubleReal(CDouble* z, double real)`
- `void setCDoubleImag(CDouble* z, double imag)`

Verwenden Sie diese Signaturen bzw. Funktionen in den folgenden Aufgaben.

Aufgabe 2 (2 Punkte). Schreiben Sie die Funktion `newCDouble`, die eine neue komplexe Zahl allokiert und mit gegebenen Werten für Real- und Imaginärteil initialisiert.

Lösung zu Aufgabe 2.

Aufgabe 3 (1 Punkt). Schreiben Sie die Funktion `setCDoubleReal`, die den Realteil einer komplexen Zahl zuweist.

Lösung zu Aufgabe 3.

Aufgabe 4 (1 Punkt). Schreiben Sie die Funktion `getCDoubleReal`, die den Realteil einer komplexen Zahl zurückgibt.

Lösung zu Aufgabe 4.

Aufgabe 5 (2 Punkte). Schreiben Sie eine Funktion `CDoubleAbs`, die die euklidische Länge

$$|z| := \sqrt{a^2 + b^2}$$

einer komplexen Zahl $z = a + bi \in \mathbb{C}$ mit $a, b \in \mathbb{R}$ zurückgibt. Verwenden Sie die dabei `sqrt`.

Lösung zu Aufgabe 5.

Aufgabe 6 (3 Punkte). Schreiben Sie einen Strukturdatentyp `CMatrix` zur Speicherung von Matrizen $A \in \mathbb{C}^{m \times n}$ mit komplexwertigen Koeffizienten A_{ij} . In der Struktur sollen neben den Dimensionen $m, n \in \mathbb{N}$ die mn Koeffizienten $A_{ij} \in \mathbb{C}$ in Form eines dynamischen Vektors vom Typ `CDouble**` gespeichert werden. Verwenden Sie `CMatrix` in allen folgenden Aufgaben.

Lösung zu Aufgabe 6.

Folgende Zugriffsfunktionen stehen für den Datentyp `CMatrix` zur Verfügung:

- `CMatrix* newCMatrix(int m, int n)`
- `CMatrix* delCMatrix(CMatrix* A)`
- `int getCMatrixM(CMatrix* A)`
- `int getCMatrixN(CMatrix* A)`
- `CDouble* getCMatrixCoeff(CMatrix* A, int i, int j)`
- `void setCMatrixCoeff(CMatrix* A, int i, int j, double real, double imag)`

Verwenden Sie diese Signaturen bzw. Funktionen in den folgenden Aufgaben, wobei mathematische Indizierung $i = 1, \dots, m$ sowie $j = 1, \dots, n$ für $A \in \mathbb{C}^{m \times n}$ verwendet werde.

Aufgabe 7 (4 Punkte). Schreiben Sie eine Funktion `newMatrix`, die für gegebene Dimensionen $m, n \in \mathbb{N}$ eine Matrix $A \in \mathbb{C}^{m \times n}$ allokiert und alle A_{ij} mit 0 initialisiert.

Lösung zu Aufgabe 7.

Aufgabe 8 (4 Punkte). Schreiben Sie eine Funktion `delCMatrix`, die den Speicher für eine Matrix $A \in \mathbb{C}^{m \times n}$ freigibt und den NULL-Pointer zurückgibt.

Lösung zu Aufgabe 8.

Aufgabe 9 (2 Punkte). Schreiben Sie eine Funktion `setCMatrixCoeff`, die für eine Matrix $A \in \mathbb{C}^{m \times n}$, gegebene Indizes $i, j \in \mathbb{N}$ und einen Wert $z \in \mathbb{C}$ die Zuweisung $A_{ij} = z$ vornimmt. Beachten Sie die mathematische Indizierung $i = 1, \dots, m$ sowie $j = 1, \dots, n$ für $A \in \mathbb{C}^{m \times n}$.

Lösung zu Aufgabe 9.

Aufgabe 10 (9 Punkte). Schreiben Sie eine Funktion `normCMatrix`, die für eine gegebene Matrix $A \in \mathbb{C}^{m \times n}$ die Spaltensummennorm

$$\|A\|_S = \max_{j=1, \dots, n} \sum_{i=1}^m |A_{ij}|$$

berechnet und zurückgibt. Beachten Sie die mathematische Indizierung $i = 1, \dots, m$ sowie $j = 1, \dots, n$ für $A \in \mathbb{C}^{m \times n}$.

Lösung zu Aufgabe 10.