
Familienname:

Vorname:

Matrikelnummer:

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

Gesamtpunktzahl:

Schriftlicher Test 1 (90 Minuten)
VU Einführung ins Programmieren für TM

08. Mai 2009

Aufgabe 1 (2 Punkte). Schreiben Sie einen Struktur-Datentyp `Cdouble` zur Speicherung von komplexen Zahlen $x \in \mathbb{C}$. In der Struktur sollen der Realteil `real` und der Imaginärteil `imag` gespeichert werden. Diese Struktur soll auch in den übrigen Aufgaben des Tests verwendet werden.

Lösung zu Aufgabe 1.

Aufgabe 2 (2 Punkte). Schreiben Sie eine Funktion `newCdouble`, die eine neue komplexe Zahl allokiert und initialisiert.

Lösung zu Aufgabe 2.

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

Lösung zu Aufgabe 3.

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

Lösung zu Aufgabe 4.

Hinweis: In den folgenden Aufgaben dürfen Sie — zusätzlich zu den von Ihnen programmierten — folgende Funktionen verwenden:

- `void setCimag(Cdouble* c, double imag)`
- `double getCimag(Cdouble* c)`

Aufgabe 5 (3 Punkte). Schreiben Sie eine Funktion `Cmultiply`, die zwei komplexe Zahlen multipliziert und das Ergebnis entsprechend zurückgibt.

Lösung zu Aufgabe 5.

Hinweis: In den folgenden Aufgaben dürfen Sie — zusätzlich zu den von Ihnen programmierten — folgende Funktionen verwenden:

- `Cdouble* delCdouble(Cdouble* c)`
- `Cdouble* Cadd(Cdouble* c, Cdouble* d)`

Aufgabe 6 (3 Punkte). Schreiben Sie einen Strukturdatentyp `Cpol` zur Speicherung von Polynomen $p(x) = \sum_{j=0}^n a_j x^j$ mit komplexen Koeffizienten. In der Struktur sollen neben dem Grad $n \in \mathbb{N}$ auch die $(n + 1)$ Koeffizienten $a_j \in \mathbb{C}$ gespeichert werden. Diese Struktur soll auch in den übrigen Aufgaben des Tests verwendet werden.

Lösung zu Aufgabe 6.

Aufgabe 7 (4 Punkte). Schreiben Sie eine Funktion `newCpol`, die für gegebenen Grad $n \in \mathbb{N}$ ein neues Polynom allokiert und initialisiert.

Lösung zu Aufgabe 7.

Aufgabe 8 (4 Punkte). Schreiben Sie eine Funktion `delCpol`, die den Speicher für ein dynamisch allokiertes Polynom freigibt und den NULL-Pointer zurückgibt.

Lösung zu Aufgabe 8.

Aufgabe 9 (2 Punkte). Schreiben Sie eine Funktion `setCpolCoeff`, die für einen gegebenen Index j und eine komplexe Zahl $c \in \mathbb{C}$ dem Koeffizienten des Polynoms den Wert $a_j = c$ durch explizite Zuweisung von Real- und Imaginärteil zuweist.

Lösung zu Aufgabe 9.

Hinweis: In der folgenden Aufgabe dürfen Sie — zusätzlich zu den von Ihnen programmierten — folgende Funktionen verwenden:

- `Cdouble* getCpolCoeff(Cpol* p,int j)`
- `int getCpolDeg(Cpol* p)`

Aufgabe 10 (8 Punkte). Schreiben Sie eine Funktion `sumCpol`, die für zwei Polynome $p(x) = \sum_{j=0}^m a_j x^j$ und $q(x) = \sum_{k=0}^n b_k x^k$ die Summe $p + q$ berechnet. — Beachten Sie, dass p und q verschiedenen Grad haben können. Überlegen Sie sich zunächst, welchen Grad das Polynom $p + q$ hat!

Lösung zu Aufgabe 10.