Familienname:	Aufgabe 1 (2 Punkte): Aufgabe 2 (4 Punkte): Aufgabe 3 (1 Punkt):
Vorname:	Aufgabe 4 (2 Punkte): Aufgabe 5 (6 Punkte): Aufgabe 6 (6 Punkte):
Matrikelnummer:	Aufgabe 7 (9 Punkte):
	Gesamtpunktzahl:
Email:	Note:

C-Nachtest (90 Minuten)
VU Einführung ins Programmieren für TM

01. Oktober 2008

Das Newton-Verfahren erlaubt die effiziente numerische Berechnung von Nullstellen von Polynomen $p(x) = \sum_{j=0}^{n} a_j x^j$. Dabei bezeichnet man $n \in \mathbb{N}$ als Grad von p, sofern $a_n \neq 0$ gilt. Im Folgenden sollen Polynome in Form des Koeffizientenvektors $a = (a_0, \dots, a_n) \in \mathbb{R}^{n+1}$ gespeichert werden. Der Test gliedert sich wie folgt:

- Definition und Zugriff auf einen Struktur-Datentyp zur Speicherung von Polynomen.
- Berechnung des Funktionswerts $p(x)$ eines Polynoms.
- Berechnung des Funktionswerts $p'(x)$ der Ableitung eines Polynoms.
- Realisierung des Newton-Verfahrens $x_{\ell+1} = x_\ell - p(x_\ell)/p'(x_\ell)$ zur Approximation einer Nullstelle von p.

Aufgabe 1 (2 Punkte). Schreiben Sie einen Strukturdatentyp `polynomial` zur Speicherung eines Polynoms $p(x) = \sum_{j=0}^{n} a_j x^j$ mit $a_n \neq 0$ in Form seines Koeffizientenvektors $a = (a_0, \ldots, a_n) \in \mathbb{R}^{n+1}$. Neben dem dynamischen Koeffizientenvektor soll auch der Grad $n = \mathrm{Grad}(p)$ Bestandteil der Struktur sein. — Diese Struktur soll auch in den folgenden Aufgaben verwendet werden.

Lösung Aufgabe 1.

Aufgabe 2 (4 Punkte). Schreiben Sie eine Funktion `allocPolynomial`, die ein Polynom vom Grad n allokiert und initialisiert.

Lösung Aufgabe 2.

Aufgabe 3 (1 Punkt). Schreiben Sie eine Funktion `getPolynomialDegree`, die den Grad $n = \mathrm{Grad}(p)$ eines Polynoms p zurückgibt. Verwenden Sie diese Funktion auch in den folgenden Aufgaben!

Lösung Aufgabe 3.

Aufgabe 4 (2 Punkte). Schreiben sie eine Funktion `getPolynomialAj`, die den Koeffizienten a_j zurückgibt. Berücksichtigen Sie den Fall, dass ggf. $j > \mathrm{Grad}(p)$ gilt. In diesem Fall ist $a_j = 0$.

Lösung Aufgabe 4.

Aufgabe 5 (6 Punkte). Schreiben Sie eine Funktion `evalPolynomial`, die für gegebenes Polynom p und Auswertungspunkt x den Funktionswert $p(x)$ berechnet. Die Funktion `pow` zur Berechnung von x^j soll *nicht* verwendet werden. Schreiben Sie die Funktion so, das die Berechnung möglichst effizient erfolgt. — Für Strukturzugriffe sollen die Funktionen aus den vorausgegangenen Aufgaben verwendet werden.

Lösung Aufgabe 5.

Aufgabe 6 (6 Punkte). Schreiben Sie eine Funktion `evalPolyDiff`, die für gegebenes Polynom p und Auswertungspunkt x den Funktionswert $p'(x)$ der Ableitung des Polynoms berechnet. Schreiben Sie die Funktion so, dass die Berechnung möglichst effizient erfolgt.

Lösung Aufgabe 6.

Aufgabe 7 (9 Punkte). Eine Möglichkeit eine Nullstelle eines Polynoms p zu Berechnen ist das Newtonverfahren. Ausgehend von einem Startwert x_0 definiert man induktiv eine Folge (x_n) wie folgt: Zu gegebenem x_k sei x_{k+1} die Nullstelle der Tangente an den Graphen von p im Punkt $(x_k, p(x_k))$, d.h. $x = x_{k+1}$ erfüllt $0 = p(x_k) + p'(x_k)(x - x_k)$. Auflösen nach x zeigt

$$x_{k+1} = x_k - p(x_k)/p'(x_k).$$

Man Schreibe eine Funktion `newtonPoly`, die zu gegebenen Polynom p, Startwert x_0 und Toleranz τ das Newtonverfahren durchführt, wobei die Iteration abgebrochen wird, falls entweder

$$|p'(x_n)| \leq \tau$$

oder

$$|p(x_n)| \leq \tau \quad \text{und} \quad |x_n - x_{n-1}| \leq \tau$$

gilt.

Lösung Aufgabe 7.