

---

**Familienname:**

**Vorname:**

**Matrikelnummer:**

**Studienkennzahl:**

**Email:**

---

Aufgabe 1 (2 Punkte):

Aufgabe 2 (4 Punkte):

Aufgabe 3 (1 Punkt):

Aufgabe 4 (3 Punkte):

Aufgabe 5 (4 Punkte):

Aufgabe 6 (8 Punkte):

Aufgabe 7 (8 Punkte):

---

Gesamtpunktzahl:

---

Note:

---

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

**01. März 2007**

---

**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  $a = (a_0, \dots, a_n) \in \mathbb{R}^{n+1}$ . Neben dem dynamischen Koeffizientenvektor soll auch der Grad  $n = \text{Grad}(p)$  Bestandteil der Struktur sein. — Diese Struktur soll auch in den folgenden Aufgaben verwendet werden.

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

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

**Aufgabe 4 (3 Punkte).** Schreiben sie eine Funktion `getPolynomialAj`, die den Koeffizienten  $a_j$  zurückgibt. Berücksichtigen Sie den Fall, dass ggf.  $j > \text{Grad}(p)$  gilt. In diesem Fall ist  $a_j = 0$ .

**Aufgabe 5 (4 Punkte).** Schreiben sie eine Funktion `setPolynomialAj`, die den Koeffizienten  $a_j$  eines Polynoms schreibt. Berücksichtigen Sie den Fall  $j > \text{Grad}(p)$ , bei dem der Grad von  $p$  erhöht werden muss. Vergessen Sie nicht, ggf. neue Koeffizienten zu initialisieren.

**Aufgabe 6 (8 Punkte).** Das Produkt  $r = pq$  zweier Polynome  $p(x) = \sum_{j=0}^n a_j x^j$  und  $q(x) = \sum_{k=0}^m b_k x^k$  ist ein Polynom  $r(x) = \sum_{\ell=0}^{m+n} c_\ell x^\ell$  vom Grad  $m+n$ . Die Koeffizienten lassen sich gerade in der Form  $c_\ell = \sum_{j+k=\ell} a_j b_k$  schreiben. Schreiben Sie eine Funktion `productPolynomial`, die das Polynom  $r$  berechnet und zurückgibt. Dabei sollen unnötige Rechenoperationen vermieden werden, d.h. die Funktion soll nur auf Koeffizienten  $a_j$  für  $j \leq n$  und  $b_k$  für  $k \leq m$  zugreifen. — Für Strukturzugriffe sollen die Funktionen aus den vorausgegangenen Aufgaben verwendet werden.

**Aufgabe 7 (8 Punkte).** Die  $k$ -te Ableitung  $p^{(k)}$  eines Polynoms  $p(x) = \sum_{j=0}^n a_j x^j$  ist wieder ein Polynom. Schreiben Sie eine Funktion `derivativePolynomial`, die zu gegebenem  $p$  und  $k \in \mathbb{N}$  die Ableitung  $p^{(k)}$  berechnet. Dabei soll möglichst wenig Speicherplatz verwendet werden. — Für Strukturzugriffe sollen die Funktionen aus den vorausgegangenen Aufgaben verwendet werden. Überlegen sie sich zunächst, welchen Grad  $p^{(k)}$  hat und wie sich die Koeffizienten von  $p^{(k)}$  berechnen lassen.