Familienname:

Vorname:

Matrikelnummer:

Studienkennzahl:

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

Gesamtpunktzahl:

Note:

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

04. Mai 2007

Aufgabe 1 (4 Punkte). Was sind die Bestandteile einer Gleitkommazahl des Zahlsystems $\mathbb{F}(b, p, e_{\min}, e_{\max})$, und wie ermittelt man aus diesen Bestandteilen den Wert der Gleitkommazahl?

Aufgabe 2 (2 Punkte). Was ist die größte Gleitkommazahl im IEEE-Gleitkommazahlsystem $\mathbb{F} = \mathbb{F}(2, 53, -1021, 1024)$? Was ist die kleinste positive denormalisierte Gleitkommazahl in $\mathbb{F}$?

Aufgabe 3 (2 Punkte). Was versteht man unter `Inf` und `NaN`?

Aufgabe 4 (2 Punkte). Schreiben Sie einen Strukturdatentyp `matrix` zur Speicherung von (oberen und unteren) Dreiecksmatrizen $A \in \mathbb{R}^{n \times n}$. In der Struktur sollen die Art der Dreiecksmatrix (`'L'` = untere, `'U'` = obere Dreiecksmatrix), die Dimension $n \in \mathbb{N}$ und die nichttrivialen Koeffizienten A_{jk} gespeichert werden. Zur Speicherung der nichttrivialen Koeffizienten verwenden man einen dynamischen Vektor der Länge $\frac{n(n+1)}{2} = \sum_{j=1}^{n} j$.

Erinnerung. Eine Matrix $L \in \mathbb{R}^{n \times n}$ ist **untere Dreiecksmatrix**, wenn gilt $L_{ij} = 0$ für $i < j$. Eine Matrix $U \in \mathbb{R}^{n \times n}$ ist **obere Dreiecksmatrix**, wenn gilt $U_{ij} = 0$ für $i > j$.

Aufgabe 5 (3 Punkte). Schreiben Sie eine Funktion `newMatrix`, die eine neue (untere oder obere) Dreiecksmatrix $A \in \mathbb{R}^{n \times n}$ allokiert und initialisiert. Neben der Dimension n muss also auch der Typ der Dreiecksmatrix übergeben werden.

Aufgabe 6 (1 Punkt). Schreiben Sie eine Funktion `delMatrix`, die eine dynamisch allokierte Dreiecksmatrix $A \in \mathbb{R}^{n \times n}$ freigibt.

Aufgabe 7 (6 Punkte). Schreiben Sie eine Funktion `getMatrixEntry`, die einen Eintrag A_{ij} einer Dreiecksmatrix $A \in \mathbb{R}^{n \times n}$ zurückliefert. Sie müssen nicht überprüfen, ob $i, j \in \{0, \ldots, n-1\}$ gilt. — Dabei werde eine untere Dreiecksmatrix zeilenweise im Speicher abgelegt, wobei Nulleinträge A_{ij} für $i < j$ nicht explizit gespeichert seien. Der Eintrag A_{ij} entspricht also dem $\frac{i(i+1)}{2} + j$-ten Eintrag im Koeffizientenvektor. Für eine obere Dreiecksmatrix sei ihre Transponierte U^T gespeichert, die wiederum eine untere Dreiecksmatrix ist. — Zum Auslesen der Dimension, verwenden Sie bitte eine Funktion `getMatrixN(A)`, die bereits vorimplementiert sei, d.h. Sie müssen diese Zugriffsfunktion nicht schreiben!

Aufgabe 8 (1 Punkt). Das Produkt $M = AB$ unterer Dreiecksmatrizen $A, B \in \mathbb{R}^{n\times n}$ ist wieder eine untere Dreiecksmatrix. Man beweise diese Aussage mathematisch, indem man die Formel $M_{ik} = \sum_{j=0}^{n-1} A_{ij}B_{jk}$ für die Matrix-Matrix-Multiplikation geeignet manipuliere.

Aufgabe 9 (8 Punkte). Schreiben Sie eine Funktion für die Matrix-Matrix-Multiplikation von unteren Dreiecksmatrizen, die es vermeidet, auf die (nicht-gespeicherten) Nulleinträge von unteren Dreiecksmatrizen zuzugreifen. — Verwenden Sie zum Schreibzugriff auf eine Dreiecksmatrix eine Funktion `setMatrixEntry(A,i,j,Aij)`, die bereits vorimplementiert sei, d.h. Sie müssen diese Zugriffsfunktion nicht schreiben!

Aufgabe 10 (1 Punkt). Folgern Sie aus Aufgabe 8, dass das Produkt $M = AB$ oberer Dreiecksmatrizen $A, B \in \mathbb{R}^{n\times n}$ eine obere Dreiecksmatrix ist. Wie könnte man basierend auf Aufgabe 9 (und der Speicherung gemäß Aufgabe 7) also eine Matrix-Matrix-Multiplikation für obere Dreiecksmatrizen realisieren?