FACHRICHTUNG 6.1
MATHEMATIK

Dr. M. Riplinger
Dipl. Math. Y. Johann

UNIVERSITÄT DES SAARLANDES

4. Übung zu „Modellierung/Programmierung“
Wintersemester 2012/2013

Aufgabe 1 Skalar–, Kreuz–, und Spatprodukt — 6 Punkte

Schreiben Sie Funktionen , die für Vektoren $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^3$ die folgenden Produkte berechnen:

a) das Skalarprodukt

$$\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \sum_{i=1}^{3} a_i \, b_i,$$

b) das Kreuzprodukt

$$\boldsymbol{a} \times \boldsymbol{b} = \begin{pmatrix} a_2 \, b_3 - a_3 \, b_2 \\ a_3 \, b_1 - a_1 \, b_3 \\ a_1 \, b_2 - a_2 \, b_1 \end{pmatrix},$$

c) das Spatprodukt

$$s = \langle (\boldsymbol{a} \times \boldsymbol{b}), \boldsymbol{c} \rangle.$$

Testen Sie Ihre Funktionen in einem Hauptprogramm, in dem Sie drei beliebige Vektoren einlesen können und das die Ergebnisse auf dem Bildschirm ausgibt.

Hinweis: Verwenden Sie in den Funktionsdeklarationen Pointer-Typen und keine statischen Felder, z.B. `Skalar(double *a, double *b)`.

Aufgabe 2 Zeichen und Strings — 3 + 2 = 5 Punkte

a) Schreiben Sie eine Funktion, die eine Zeichenkette einliest und diese darauf testet, ob es sich um ein *Palindrom* handelt. Ein *Palindrom* ist ein Wort, das vorwärts wie rückwärts gelesen dasselbe Wort ergibt. Die Funktion soll keine Unterschiede zwischen Groß- und Kleinschreibung machen und nur Zeichenketten der maximalen Länge 50 einlesen.

Hinweis: In `<ctype.h>` gibt es die Funktion `int toupper(int c)`, die Kleinbuchstaben in Großbuchstaben umwandelt. Ist dies nicht möglich, liefert die Funktion das Zeichen unverändert zurück. Die Funktion `strcmp()` vergleicht zwei Zeichenketten miteinander. Einzelne Zeichen können nur über == bzw. `!=` miteinander verglichen werden.

b) Testen Sie die Funktion in einem Hauptprogramm, in dem die Zeichenkette vom Benutzer mit Hilfe von `fgets()` eingelesen wird.

Hinweis: Wird keine Zeichenkette maximaler Länge eingelesen, muss das Stringende erzwungen werden.

Aufgabe 3 Pointer und Adressoperator 3 Punkte

Schreiben Sie ein Programm, das die folgenden Definitionen von Variablen und die geforderten Anweisungen enthält:

- Definition einer Variablen `a` vom Typ `double` und eines Pointers `point` vom Typ `double *`,
- Ausgabe der Adressen der Variablen `a` und des Pointers `point`,
- Zuweisung der Adresse von `a` an den Pointer `point`,
- Zuweisung des Wertes 5 an die Variable `a`,
- Ausgabe der Werte des Pointers `point` und der Variable `a`,
- Ausgabe des Wertes des Objekts, auf das der Pointer `point` zeigt, mit Hilfe des Inhaltsoperators,
- Zuweisung des Wertes π an die Variable `a` durch Zugriff auf den Inhalt der Zeigervariablen,
- Ausgabe der Variablen `a`.

Aufgabe 4 Dynamische Speicherplatzreservierung 6 Punkte

Schreiben Sie eine Funktion `initHilbertMatrix()`, die eine quadratische Matrix $A = (a_{i,j}) \in \mathbb{R}^{N \times N}$ mit den Werten

$$a_{i,j} = \frac{1}{i + j - 1}, \quad i, j = 1, \ldots, N$$

initialisiert. Die Dimension N soll vom Benutzer erfragt werden. Die Funktion hat die Deklaration `void initHilbertMatrix(double **matrix, int N)`. Reservieren Sie zunächst im Hauptprogramm den benötigten Speicherplatz. Geben Sie die Matrix auf dem Bildschirm aus. Am Ende des Hauptprogramms soll der reservierte Speicherplatz wieder freigegeben werden.

Hinweis: Verwenden Sie zur Reservierung und Freigabe des Speicherplatzes und zur Ausgabe der Matrix auf den Bildschirm die in der Vorlesung vorgestellten Funktionen.

Aufgabe 5 * Klausur 2011 - Zeiger

Was ist nach Ausführung der folgenden Anweisungen jeweils der Wert der Variablen `x` und `y`?

a)
```
int x = 3, y = 4;
int *p;

p = &x;
*p = *p + y;
```

b)
```
int x = 3, y = 4;
int *p1, *p2;

p1 = &x;
p2 = &y;
*p1 = *p1 + y;
p1 = p2;
*p1 = *p1 + y;
```

c)
```
int x = 3, y = 4;
int *p1 = &x, *p2 = &y;
int **pp = &p1;

(**pp)--;
*pp = p2;
(**pp)++;
```

Antworten:

a) `x =` _____ `, y =` _____

b) `x =` _____ `, y =` _____

c) `x =` _____ `, y =` _____

Abgabe bis zum 19. Dezember 2012, 19.00 Uhr per E-Mail an Ihren Bremser.