

## Klausur

Dauer: 90 Minuten

28. März 2018, 10:15 bis 11:45

- Alle folgenden Fragen und sämtlicher Programmcode beziehen sich auf die in der Vorlesung ausgiebig diskutierte Programmiersprache C.
- Erlaubte Ausrüstung/Hilfsmittel: Stift, Papier.
- Bücher, Ordner, handgeschriebene Aufzeichnungen, elektronische Geräte sind verboten.
- Bitte beantworte jede Aufgabe auf einer gesonderten DinA4-Seite und gib auf jeder verwendeten Seite Deinen Namen und Matrikelnummer an.
- Bitte gib dieses Deckblatt mit Deinen Aufgaben ab. Die Aufgabenblätter kannst Du behalten.
- Es können insgesamt 40 Punkte erzielt werden.
- Die Klausur ist mit 20 oder mehr Punkten bestanden.

Vollständiger Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

AUFGABE	PUNKTE
1	/5
2	/4
3	/5
4	/5
5	/5
6	/5
7	/5
8	/6
SUMME:	/40

Bestanden:  Ja  Nein

## Aufgabe 1

(5 Pkt.)

Gib für jede der folgenden Aussagen an, ob sie richtig oder falsch ist. Falls eine Aussage falsch ist, begründe dies entsprechend.

- Man kann in einem C-Programm einer konstanten Variablen (z.B. vom Typ `const int`) einen neuen Wert zuweisen. Der Compiler gibt jedoch eine Warnung aus.
- Weist man einer Variablen vom Typ `unsigned int` einen negativen Wert zu, scheitert das Kompilieren, das heißt, der Compiler bricht mit einer Fehlermeldung ab.
- Eine innerhalb einer Funktion definierte statische Variable (z.B. vom Typ `static int`) ist äquivalent zu einer konstanten Variable (z.B. vom Typ `const int`).
- Mit einem `break` statement kann man ein C-Programm beenden.
- Der Datentyp `const int * const` dient für konstante Zeiger auf konstante `int`-Variablen.

## Aufgabe 2

(4 Pkt.)

Vervollständige die folgende Funktion, die das Produkt einer  $n \times n$ -Matrix `A` mit einem `n`-dimensionalen Vektor `a` berechnet und in `b` speichert.

```
void multiply(double A[n][n], double a[n], double b[n]) { /* ... */ }
```

`n` ist dabei eine geeignet initialisierte globale Variable vom Typ `const int`, z.B. `const int n = 6;` .

## Aufgabe 3

(5 Pkt.)

Gegeben sind folgende Programmzeilen.

```
double a[3][2] = {{0.0, 1.0}, {2.0, 3.0}, {4.0, 5.0}};  
double *b = a[1];
```

Gib an, welchen Datentyp die folgenden Ausdrücke besitzen. Gib im Fall von einem Zeiger den Inhalt des Speichers an, auf den der Zeiger verweist, d.h. den Wert der dort gespeicherten `double`-Zahl. Gib ansonsten den Wert des Ausdrucks an.

- `a[1][1]`
- `a[2]`
- `b[1]`
- `b + 1`
- `*(&a[1][1]) - 1`

## Aufgabe 4

(5 Pkt.)

Schreibe ein vollständiges C-Programm, welches aus einer im gleichen Verzeichnis liegenden Datei `messdaten.dat` 100 Gleitkommazahlen einliest und deren Mittelwert berechnet und am Bildschirm ausgibt. Der Inhalt der Datei `messdaten.dat` kann z.B. wie folgt aussehen (... steht für weitere 94 Zahlen).

```
3.14  
1.0  
4.999  
-4.5  
+3.2  
...  
12345.6
```

## Aufgabe 5

(5 Pkt.)

Das folgende Programm soll die  $3 \times 3$ -Identitätsmatrix in einem zur Laufzeit dynamisch reservierten 9-elementigen `double`-Array speichern und in einfach lesbarer Form am Bildschirm ausgeben.

- (i) In welchen vier Programmzeilen hat der Programmierer Fehler begangen, die unerwünschte und unter Umständen katastrophale Auswirkungen haben können? Erläutere die Fehler im Detail und skizziere mögliche Auswirkungen. Gib einen entsprechenden korrigierten Programmteil an.
- (ii) Erweitere den Programmcode an geeigneter Stelle so, dass der dynamisch reservierte Speicher möglichst frühzeitig wieder freigegeben wird, das Programm aber nach wie vor fehlerfrei arbeitet.

```
1      #include <stdio.h>
2      #include <stdlib.h>
3
4      void superindex(int j, int k) {
5          return j * n + k;
6      }
7
8      int main(void) {
9          const int n = 3;
10         double a = (double *)malloc(n*n*sizeof(double));
11         if (a == NULL) {
12             printf("ERROR: malloc failed! Aborting...\n");
13             exit(1);
14         }
15         int j, k;
16         for(j = 0; j < n; j++) {
17             for(k = 0; k < n; k++) {
18                 if(j == k)
19                     a[superindex(j, k)] = 1.0;
20                 else
21                     a[superindex(j, k)] = 0.0;
22             }
23         }
24         for(j = 0; j < n; j++) {
25             for(k = 0; k < n; k++)
26                 printf("%.5f  ", a[superindex(j, k)]);
27             printf("\n");
28         }
29         return 0;
30     }
```

## Aufgabe 6

(5 Pkt.)

Definiere eine Struktur `struct c_number`, die zwei `double`-Variablen `re` (entspricht dem Realteil) und `im` (entspricht dem Imaginärteil) erhält. Schreibe die folgenden drei Funktionen:

- Eine Funktion mit Namen `conjugate`, die eine Variable vom Typ `struct c_number` erhält und die zugehörige komplex konjugierte Zahl berechnet.
- Eine Funktion mit Namen `multiply`, die zwei Variablen vom Typ `struct c_number` erhält und das Produkt der beiden Zahlen berechnet.
- Eine Funktion mit Namen `divide`, die zwei Variablen vom Typ `struct c_number` erhält und den Quotienten der beiden Zahlen berechnet.

Die Funktionen sollen das Ergebnis jeweils als `struct c_number` zurückgeben. Achte bei der Division auf problematische Spezialfälle und stattete Deine Funktion mit entsprechenden Fehlerabfragen aus.

## Aufgabe 7

(5 Pkt.)

Vervollständige die folgende Funktion, die das  $n$ -elementige Array  $a$ , in dem Gleitkommazahlen vom Typ `double` gespeichert sind, in aufsteigender Ordnung sortiert.

```
void sort(int n, double *a) { /* ... */ }
```

Erläutere den von Dir verwendeten Sortieralgorithmus in einigen wenigen Sätzen.

## Aufgabe 8

(6 Pkt.)

Gegeben sind folgende Programmzeilen.

```
int a = 12;  
double b = 3.0;  
double *c = &b;
```

- (i) Welchen Wert liefert der arithmetische Ausdruck `*c + 1`? Erläutere in diesem Zusammenhang das Konzept des Vorrangs von Operatoren.
- (ii) Welchen Wert liefert der arithmetische Ausdruck `5.0 - 3.0 + 2.0`? Erläutere in diesem Zusammenhang das Konzept der Linksassoziativität.
- (iii) Welche Werte liefern die folgenden arithmetischen Ausdrücke? Gib zusätzlich jeweils den Datentyp des Ausdrucks an.
  - (a) `(int)(a + 1.5)`
  - (b) `a + *c`
  - (c) `(double)(1 / 3)`
  - (d) `(b == 5.0) + a`