



mitp

Peter
Prinz

2. Auflage

C Das Übungsbuch

Testfragen und Aufgaben mit Lösungen

- h) 0101
- i) 0xAL
- 2.2 Schreiben Sie die Anweisungen, die exakt die folgenden Ausgaben erzeugen.
- Hinweis:* Verwenden Sie Escape-Sequenzen zur Ausgabe von Sonderzeichen oder Steuerzeichen.
- a)

```
"It is impossible to make anything foolproof"
    "because fools are so ingenious (Murphy)."
```

wobei die Einrückung mit zwei horizontalen Tabs erfolgen soll.

- b)
- ```
Datei nicht gefunden: "C:\docs\Sprueche.doc"
```
- und einen Ton, um die Aufmerksamkeit des Benutzers zu wecken.
- 2.3 Nicht alle der folgenden Variablendefinitionen sind korrekt. Welche Fehler liegen vor?
- a) `int INT = 0X100;`
- b) `unsigned char code = 300;`
- c) `short zeichen = '\\';`
- d) `int 1i = 0, 2i = -1;`
- e) `short gültig = 40000;`
- f) `float Result = 1234.56789;`
- g) `long long goto = 10000;`
- h) `long file = "MeineBilder";`
- i) `double top-left = 10.5;`
- j) `long double size = 706*975;`
- 2.4 Schreiben Sie ein C-Programm, das zwei Variablen für Gleitpunktzahlen initialisiert und ihre Werte am Bildschirm anzeigt. Anschließend berechnet das Programm die Summe, die Differenz, das Produkt und den Quotienten beider Zahlen und zeigt die Ergebnisse an.
- 2.5 Bestimmen Sie die Ausgabe des folgenden C-Programms, ohne das Programm auszuführen.

*Hinweis:* Eine ASCII-Code Tabelle ist hilfreich.

```
#include <stdio.h>
int main()
{
 char c = 'A';
 printf("%c %d\n", c, c);
 c = c + 1;
 printf("%c %d\n", c, c);
 c = c + ('a' - 'A');
 printf("%c %d\n", c, c);
 return 0;
}
```

- 2.6 Schreiben Sie ein C-Programm, das die Größe des Speicherplatzes und den Wertebereich der Datentypen `char` und `int` anzeigt. Verwenden Sie die Konstanten `CHAR_MIN`, `CHAR_MAX`, `INT_MIN` und `INT_MAX`, die den kleinsten und größten möglichen Wert des jeweiligen Typs darstellen. Diese Konstanten sind in der Header-Datei `limits.h` definiert.

*Hinweis:* Der Operator `sizeof` liefert die Größe eines Typs in Anzahl Byte.

- 2.7 Die Header-Datei `float.h` definiert die Konstanten, die die Wertebereiche und Genauigkeiten der Gleitpunkttypen beschreiben. Für den Typ `float` sind das die Konstanten `FLT_MAX`, `FLT_MIN` und `FLT_DIG`. Sie stellen den größten Wert, den kleinsten positiven Wert und die Genauigkeit dar. Die entsprechenden Konstanten für den Typ `double` sind `DBL_MAX`, `DBL_MIN` und `DBL_DIG`. Schreiben Sie ein C-Programm, das den Wert dieser Konstanten anzeigt.

*Hinweis:* Zeigen Sie den größten und kleinsten Wert in exponentieller Schreibweise an. Verwenden Sie dazu das Formatelement `%E`.

- 2.8 Bestimmen und korrigieren Sie die vier Fehler in jedem der folgenden Programme.

a)

```
#include <stdio.h>
int main()
{ // Hier stimmt einiges nicht!
 double a;
 b = a + 10;
 printf("Ergebnis: %d/n", b);
 return 0;
}
```

b)

```
#include <stdio.h>
int main()
{
 char z1 = "?", z2 = 0x100, z3 = 0101;
 printf("Die drei Zeichen: %c %c\n", z1 z2 z3);
 return 0;
}
```

- 2.9 In einem C-Programm ist zu einem Brutto-Betrag der zugehörige Netto-Betrag und die enthaltene Mehrwertsteuer zu berechnen.

Definieren Sie die Variablen `brutto`, `netto`, `mwst` und `mwst_satz`, wobei Sie die Variablen `brutto` und `mwst_satz` beispielsweise mit den Werten 500.00 und 0.19 initialisieren. Speichern Sie das Ergebnis in den Variablen `netto` und `mwst` und zeigen Sie es am Bildschirm an. Es gilt die Formel:

$$\text{netto} = \text{brutto} / (1 + \text{mwst\_satz})$$

- 2.10 Was gibt folgendes C-Programm auf dem Bildschirm aus?

```
#include <stdio.h>
void myFunction(void);
int a = 100;

int main()
{
 int b = 10;
 myFunction();
 printf("In main(): a = %d, b = %d\n", a, b);
 myFunction();
 return 0;
}

void myFunction()
{
 int b = 20;
 a = a + b;
 b = b + 10;
 printf("In myFunction(): a = %d, b = %d\n", a, b);
}
```

## Lösungen zu den Verständnisfragen

- 2.1 a) und b)
- 2.2 Richtig
- 2.3 b) und c)
- 2.4 long
- 2.5 c)
- 2.6 b)
- 2.7 Richtig
- 2.8 c)
- 2.9 a) und c)
- 2.10 0, 48, 0
- 2.11 Falsch ('A' ist der Zeichencode von A, also 65, und "A" ist ein String.)
- 2.12 3 (ein Byte für das String-Endezeichen \0)
- 2.13 Horizontaler Tabulator und Zeilenwechsel (new-line).
- 2.14 Richtig
- 2.15 a)
- 2.16 b) und c)
- 2.17 a) und b)
- 2.18 0
- 2.19 c)
- 2.20 a) und b)

## Lösungen zu den Aufgaben

- 2.1 a) int (Zeichencode von X)
- b) int (Zeichencode oktäl 33 = dezimal 27)
- c) float
- d) unsigned long
- e) int (hexadezimale Konstante)
- f) double

- g) double
- h) int (oktale Konstante)
- i) long (hexadezimale Konstante)

2.2 a)

```
printf("\nIt is impossible to make anything foolproof\n"
 "\t\tbecause fools are so ingenious (Murphy).\n");
```

b)

```
printf("Datei nicht gefunden: "
 "\\\"C:\\docs\\Sprueche.doc\\\"a\n");
```

- 2.3
- a) Korrekt (Groß- und Kleinschreibung wird unterschieden.)
  - b) Wert 300 zu groß. (Wertebereich von unsigned char: 0 bis 0xFF = 255.)
  - c) Korrekt (Code des Sonderzeichens \ in einer short-Variablen.)
  - d) Namen dürfen nicht mit einer Ziffer beginnen.
  - e) Unzulässiger Name und 40000 nicht im Wertebereich von short.
  - f) Der Typ float besitzt nur eine Genauigkeit von 6 Dezimalstellen.
  - g) goto ist ein Schlüsselwort.
  - h) Eine numerische Variable kann nicht mit einem String initialisiert werden.
  - i) Bindestriche sind in Namen nicht zulässig.
  - j) Korrekt (Initialisierung mit dem Ergebnis von 706\*975.)

2.4

```
/* -----
 * ex02_04.c
 * Zwei Variablen für Gleitpunktzahlen initialisiert und
 * anzeigen. Anschließend Summe, Differenz, Produkt und
 * Quotienten beider Zahlen berechnen und anzeigen.
 * -----
 */
#include <stdio.h>

int main()
{
 double zahl1 = 12.3, zahl2 = 78.9;
 double summe, differenz, produkt, quotient;
```