

Lösung Übung 1

infT

1.

Welche Ausgabe liefert das folgende Programm?

```
#include <stdio.h>
int main() {
    printf("\nDas Ergebnis von\n12 durch 4 ist %i.", 3);
    return 0;
}
```

Lösung:

*Das Ergebnis von
12 durch 4 ist 3.*

2.

Ergänzen Sie die folgende `printf`-Anweisung, damit auf dem Bildschirm die darunter dargestellte Ausgabe erscheint.

```
#include <stdio.h>
int main() {
    printf("\nDieser Text\nenthalt\nviele\nZeilenumbrueche.");
    return 0;
}
```

*Dieser Text
enthalt
viele
Zeilenumbrueche.*

3.

Welches Ergebnis liefert dieses Programm?

```
#include <stdio.h>
int main() {
    printf("4 + 6 / 2 = %i\n", 4 + 6 / 2);
    return 0;
}
```

Lösung:

4 + (6 / 2) = 7 // Operatorenprioritaeten!

4.

Welche `include`-Dateien werden für die Funktionen `strlen` (ermittelt die Länge einer Zeichenkette) und `getchar` (liest Zeichen von der Tastatur) benötigt? Benützen Sie zur Lösung die Doku zu den C-Libraries.

Lösung:

Eingabe von `man strlen` bzw. `man getchar` (ohne Funktionsklammern) auf der Linux-Konsole liefert:

`#include <string.h>` bzw. `#include <stdio.h>`

Diese Info kann auch in der Datei `libc_manual.html` gefunden werden oder in der *GNU C Library*, Abschnitt *Character Input*.

5.

Was ist an diesem Programm falsch?

```
#include <stdio.h>

int main() {
    float celsius;

    PRINTF("\nBitte Grad Celsius eingeben: ");
    SCANF("%f", &celsius);
    PRINTF("\n%.1f Grad Celsius entsprechen ", celsius);
    PRINTF("%.1f Grad Fahrenheit.\a", celsius * 9 / 5 + 32);
    return 0;
}
```

Lösung:

C unterscheidet zwischen Gross- und Kleinschreibung: es muss `printf` und `scanf` heissen.

6.

Dieses Programm berechnet die Anzahl der Herzschläge seit Ihrer Geburt. Was muss im Programm geändert werden, damit die Gesamtzahl der Herzschläge ohne Nachkommastellen ausgegeben wird?

```
#include <stdio.h>

int main() {
    float schlaege, alter;

    printf("\n\t\tHerzschlaege\n");
    printf("\nHerzschlaege pro Minute: ");
    scanf("%f", &schlaege);
    printf("Alter in Jahren: ");
    scanf("%f", &alter);
    printf("\nIhr Herz hat seit Ihrer Geburt ");
    printf("%.0f ", schlaege * 60 * 24 * 365.25 * alter);    \\ Aenderung
    printf("mal geschlagen.");
    return 0;
}
```

7.

Schreiben Sie ein Programm, das die nötige Energie eines Fahrradfahrers je 10 km errechnet. Als Eingabe benötigt das Programm die Energie des Fahrers für die totale Anzahl gefahrener Kilometer sowie die gefahrenen Kilometer selbst.

Lösung:

```
int main() {
    float kmTotal, energieTotal;

    printf("\n\t\tIhr Energiebedarf fuer 10km Fahrrad fahren");
    printf("\nWieviele Kilometer sind Sie mit dem Fahrrad gefahren? ");
    scanf("%f", &kmTotal);
    printf("Wie gross war dazu der Energiebedarf in Wh? ");
    scanf("%f", &energieTotal);
    printf(
        "Ihr Energiebedarf pro 10km betraegt ca. %.2fWh",
        energieTotal / kmTotal * 10);
    return 0;
}
```

8.

Welche der folgenden Zuweisungen sind falsch?

- a) `summe = summe + 10` ok
- b) `x * x = X * 5` falsch, links kein Ausdruck erlaubt
- c) `5a = anfang + ende` falsch, Variablenname beginnt mit Ziffer
- d) `wert = eingabe + alterWert * 2` ok
- e) `gBetrag = betrag * 1.14` ok

9.

Warum liefert das folgende Programm unter Umständen ein falsches Resultat?

```
#include <stdio.h>

int main() {
    double summe, zahl;

    summe = 0.0;           // fehlende Initialisierung ergänzt
    zahl = 10;
    summe = summe + zahl;
    summe = summe + zahl;
    printf("Ergebnis = %f", summe);
    return 0;
}
```

10.

Was bewirken die folgenden Ausdrücke?

- a) `(x < 0) ? -x : x`

Lösung:

Das Ergebnis weist stets einen positiven Wert auf oder 0.

- b) `(celsius < -273.15) ? -273.15 : celsius`

Lösung:

Das Ergebnis kann nicht unter den absoluten Nullpunkt zu liegen kommen.

11.

Gegeben ist das folgende Programm. Versuchen Sie vorerst, das Programm zu verstehen. Bringen Sie es dann zur Ausführung und prüfen Sie, ob Ihre Voraussage richtig war.

```
#include <stdio.h>

int main() {
    int nextChar;           Deklarieren einer Variablen für Character. Beachten Sie den
                           type int, wie mit getchar nötig
    int numLines = 0;      Deklarieren und initialisieren einer Variablen zum Zählen
                           von Zeilenendzeichen.
    while ((nextChar = getchar()) != EOF) {
        if (nextChar == '\n') {
            ++numLines;
        }
    }
    printf("\n%d lines read.\n", numLines);  Anzahl gefundener Zeilenendzeichen
                                             ausgeben.
}
```

