

4.2.A Datenstrukturen/Zeiger und Funktionen – Arbeitsblatt

Aufgabe 1: Funktion `setzenull`

Die Funktion `setzenull()` soll die Variable, deren Adresse als Parameter übergeben wird, auf den Wert `0.0` setzen. Das folgende Programmstück soll also die Ausgabe `0.00000` haben:

```
1 #include <stdio.h>
2 int main(void)
3 {
4     double x=3.7;
5     setzenull(&x);
6     printf("%lf\n", x);
7     return 0;
8 }
```

- Wie lautet der vollständige Prototyp der Funktion?
- Bitte schreiben Sie die Funktionsdefinition!
- Testen Sie das Programm!

Aufgabe 2: Funktion `setzeauf`

Die Funktion `setzeauf()` soll die Variable, deren Adresse als erster Parameter übergeben wird, auf den Wert des zweiten Parameters setzen. Das folgende Programmstück soll also die Ausgabe `8.200000` haben:

```
1 #include <stdio.h>
2 int main(void)
3 {
4     double x=3.7;
5     setzeauf(&x, 8.2);
6     printf("%lf\n", x);
7     return 0;
8 }
```

- Wie lautet der vollständige Prototyp der Funktion?
- Bitte schreiben Sie die Funktionsdefinition!
- Testen Sie das Programm!

Aufgabe 3: Funktion `msetzenull`

Die Funktion `msetzenull()` soll die Variablen, deren Adressen als erster, zweiter, dritter und vierter Parameter übergeben werden, auf den Wert `0.0` setzen. Das folgende Programmstück soll also die Ausgabe `0 0 0 0` haben:

```
1 #include <stdio.h>
2 int main(void)
3 {
4     double w=3.7, x=4.1, y=-1.5, z=-2.2;
5     msetzenull(&w, &x, &y, &z);
6     printf("%lg %lg %lg %lg\n", w, x, y, z);
7     return 0;
8 }
```

- a) Wie lautet der vollständige Prototyp der Funktion?
- b) Bitte schreiben Sie die Funktionsdefinition!
- c) Testen Sie das Programm!

Aufgabe 4: Funktion `msetzeauf`

Die Funktion `msetzeauf()` soll die Variablen, deren Adressen als zweiter, dritter und vierter Parameter übergeben werden, auf den Wert des ersten Parameters setzen. Das folgende Programmstück soll also die Ausgabe `12 12 12` haben:

```
1 #include <stdio.h>
2 int main(void)
3 {
4     double w=12.0, x=4.1, y=-1.5, z=-2.2;
5     msetzeauf(w, &x, &y, &z);
6     printf("%lg %lg %lg\n", x, y, z);
7     return 0;
8 }
```

- a) Wie lautet der vollständige Prototyp der Funktion?
- b) Bitte schreiben Sie die Funktionsdefinition!
- c) Testen Sie das Programm!

Aufgabe 5: Funktion `verdoppel`

Die Funktion `verdoppel()` soll eine Zahl vom Typ `float` verdoppeln, so dass sie nach dem Funktionsaufruf doppelt so groß ist wie zuvor:

```
1 ...
2 float zahl=12.0;
3 verdoppel(&zahl);
4 printf("Neuer_Wert_der_Zahl: %f\n", zahl); /* muss 24.0 ergeben */
5 verdoppel(&zahl);
6 printf("Neuer_Wert_der_Zahl: %f\n", zahl); /* muss 48.0 ergeben */
```

- a) Wie lautet der Prototyp von `verdoppel()`?
- b) Bitte schreiben Sie die Definition von `verdoppel()`!

Aufgabe 6: Funktion `kugel`

Die Funktion `kugel` soll als ersten Parameter den Radius `r` einer Kugel bekommen. Die Funktion soll daraus das Volumen und die Oberfläche der Kugel berechnen. Der zweite und dritte Parameter sollen Adressen sein, an die die Funktion die errechneten Werte zurückschreibt:

```
1 #include <stdio.h>
2 #include <math.h> // fuer M_PI=3.14159...
3 int main(void)
4 {
5     double r=6283.; // Erdradius in km (ungefaehr)
6     double v, a;
7     kugel(r, &v, &a);
```

```
8   printf("Erdvolumen=%lg\km^3\n", v);
9   printf("Erdoberflaeche=%lg\km^2\n", a);
10  return 0;
11 }
```

a) Wie lautet der vollständige Prototyp der Funktion?

b) Schreiben Sie die Funktionsdefinition!

Aufgabe 7: Funktion **paket**

Die Funktion `paket()` soll als erste Parameter die Länge, Höhe und Breite eines Paketes erhalten. Die Funktion soll das Volumen und die Oberfläche des Paketes berechnen und per Zeiger an `main()` zurückgeben:

```
1 #include <stdio.h>
2 void paket (...);
3 int main(void)
4 {
5     double l=0.3, b=0.15, h=0.2;
6     double v, a;
7     paket (...);
8     printf("Volumen=%lfm^3\nOberflaeche=%lfm^2\n", v, a);
9     return 0;
10 }
11 void paket (...) {....}
```

a) Wie lautet der vollständige Prototyp der Funktion?

b) Wie wird die Funktion richtig aufgerufen?

c) Vervollständigen Sie bitte die Definition der Funktion!

Aufgabe 8: Funktion **swap**

Die Funktion `swap()` soll die Inhalte zweier `int`-Variablen vertauschen:

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int a=10, b=20;
5     printf("a=%i, b=%i\n", a, b);
6     swap(&a, &b);
7     printf("a=%i, b=%i\n", a, b);
8     return 0;
9 }
```

Das Programm soll die Ausgabe haben:

a=10, b=20

a=20, b=10

a) Wie lautet der vollständige Prototyp der Funktion?

b) Schreiben Sie bitte die Funktionsdefinition!

Aufgabe 9: Funktion `rot3`

Die Funktion `rot3()` soll die Inhalte dreier `int`-Variablen zyklisch vertauschen (wie in einem Schieberegister: `b` bekommt den alten Wert von `a`; `c` bekommt den alten Wert von `b` und `a` bekommt den alten Wert von `c`):

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int a=10, b=20, c=30;
5     printf("a=%i, b=%i, c=%i\n", a, b, c);
6     rot3(&a, &b, &c);
7     printf("a=%i, b=%i, c=%i\n", a, b, c);
8     return 0;
9 }
```

Das Programm soll die Ausgabe haben:

```
a=10, b=20, c=30
a=30, b=10, c=20
```

- a) Wie lautet der vollständige Prototyp der Funktion?
- b) Schreiben Sie bitte die Funktionsdefinition!

Aufgabe 10: Funktion `sort2`

Die Funktion `sort2()` soll zwei Zahlen vom Typ `int` so anordnen, dass nach dem Funktionsaufruf die kleinere Zahl vorn ist:

```
1 ...
2 int a=5, b=3; /* Hier ist a=5, b=3 */
3 sort2(...);
4 printf("a=%i, b=%i\n", a, b); /* Hier soll a=3, b=5 sein. */
```

Das heißt, die Zahlen sollen nur dann vertauscht werden, wenn die größere Zahl vorne ist.

- a) Wie muss `sort2()` hier aufgerufen werden?
- b) Wie lautet der Prototyp von `sort2()`?
- c) Bitte erstellen Sie die Definition von `sort2()`!

Aufgabe 11: Funktion `sort3`

Nach dem Vorbild von `sort2()` soll die neue Funktion `sort3()` drei Zahlen der Größe nach sortieren.

- a) Schreiben Sie bitte die Funktionsdefinition!
- b) Testen Sie die Funktion anhand der Werte-Kombinationen 1-2-3, 1-3-2, 2-1-3, 2-3-1, 3-1-2 und 3-2-1!

Aufgabe 12: Funktion `rchochpass`

Die Funktion `rchochpass()` soll die Dämpfung A und die Phasenverschiebung φ eines RC-Hochpasses (Abbildung 1) bei (vom Nutzer) eingegebenen Werten für f , R und C bestimmen.

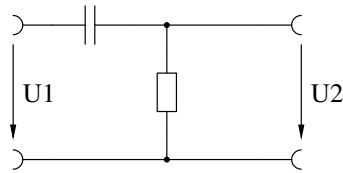


Abbildung 1: RC-Hochpass

```

1  double f=100e3;    // 100 kHz
2  double R=6.8e3;    // 6,8 kOhm
3  double C=234e-12;  // 234 pF
4  double A, phi;
5  rchochpass(f,R,C,&A,&phi);
6  printf("A=%lg\n", A);
7  printf("phi=%lg Grad\n", phi);

```

Das Programm soll die Ausgabe haben:

```

A    = 0,707
phi  = 45 Grad

```

- a) Wie lautet der Prototyp der Funktion?
- b) Schreiben Sie bitte die Funktionsdefinition!

Hinweise:

- $X_C = \frac{1}{2\pi f \cdot C}$
- $A = \frac{U_2}{U_1} = \frac{R}{\sqrt{R^2 + X_C^2}}$
- $\varphi = \arctan \frac{X_C}{R}$
- Die Funktion `atan()` berechnet den Winkel in rad, nicht in Altgrad. Das Ergebnis muss durch π geteilt und mit 180° multipliziert werden.
- Die Headerdatei `<math.h>` muss eingebunden werden.
- Das Programm muss mit der Option `-lm` compiliert werden.