



Freiwilliges Programmieren 2 Lösungsvorschlag

In diesem Freiwilligen Programmieren werden Wiederholungsaufgaben zu den Themen 3 und 4 des Vorkurses behandelt. Für jede Teilaufgabe soll ein C-Programm erstellt und dazu die in Worten beschriebenen Anweisungen in C-Anweisungen umformuliert werden. Dabei ist jeder Satz in genau eine C-Anweisung zu überführen - hiervon ausgenommen ist die Generierung von Zufallszahlen.

Erstellen Sie für jede Teilaufgabe jeweils eine C-Datei mit einer eigenen `main`-Funktion. Kompilieren Sie Ihre Programme mit den Compilerschaltern `-ansi` `-pedantic` `-Wall` `-Wextra` und führen Sie sie aus (jeweils über ein Kommandozeilen-Programm).

Aufgabe 2.1 (*Bedingungen*)

a)

- Deklarieren Sie eine `int`-Variable `a`, generieren Sie eine ganze Zufallszahl und weisen Sie `a` als Wert den Rest bei ganzzahliger Division der Zufallszahl durch 2000 zu.
- Erstellen Sie eine Bedingung für folgende Aussage und geben Sie deren Wert aus:
`a` hat 2 oder 3 Stellen

Lösung:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main(void)
6  {
7      int a;
8      srand(time(NULL));
9      a = rand() % 2000;
10     printf("%i\n", (a >= 10 && a <= 999));
11     return 0;
12 }
```

b)

- Deklarieren Sie eine `char`-Variable `b`, generieren Sie eine ganze Zufallszahl und weisen Sie `b` als Wert den Rest bei ganzzahliger Division der Zufallszahl durch 128 zu.
- Erstellen Sie eine Bedingung für folgende Aussage ohne dafür eine Bibliotheksfunktion zu benutzen und geben Sie deren Wert aus:
`b` ist kein lateinischer Kleinbuchstabe

Lösung:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main(void)
6 {
7     char b;
8     srand(time(NULL));
9     b = rand() % 128;
10    printf("%i\n", (b < 'a' || b > 'z'));
11    return 0;
12 }
```

Aufgabe 2.2 (Fallunterscheidungen)

a)

- Deklarieren Sie eine **int**-Variable **a**, generieren Sie eine ganze Zufallszahl und weisen Sie **a** als Wert den Rest bei ganzzahliger Division der Zufallszahl durch 40000 zu.
- Falls **a** kleiner als oder gleich 20000 ist, machen Sie Folgendes:
 - Falls **a** durch 5 teilbar ist, geben Sie 0 aus.
 - In allen anderen Fällen geben Sie 1 aus.Sonst geben Sie 2 aus.
- Geben Sie in einer neuen Zeile **The end!** aus.

Lösung:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main(void)
6 {
7     int a;
8     srand(time(NULL));
9     a = rand() % 40000;
10    printf("%i\n", a);
11    if (a <= 20000) {
12        if (a % 5 == 0)
13            printf("0");
14        else
15            printf("1");
16    } else {
17        printf ("2");
18    }
19    printf("\nThe end!\n");
20    return 0;
21 }
```

b)

- Deklarieren Sie zwei **int**-Variablen **b** und **c** und weisen ihnen jeweils eine Zufallszahl als Wert zu.
- Bestimmen Sie das Minimum von **b** und **c** und geben Sie dieses aus.

Lösung:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
```

```

5  int main(void)
6  {
7      int b, c;
8      srand(time(NULL));
9      b = rand();
10     c = rand();
11     if (b < c)
12         printf("%i\n", b)
13     else
14         printf("%i\n", c);
15     return 0;
16 }

```

Aufgabe 2.3 (Dezimalzahlen)

a)

- Deklarieren Sie eine **double**-Variable **a**.
- Weisen Sie ihr den Wert 1234e-5 zu.
- Geben Sie den Wert von **a** in Festkommenschreibweise aus.
- Geben Sie den Wert von **a** in Fließkommenschreibweise aus.

Lösung:

```

1  #include <stdio.h>
2
3  int main(void)
4  {
5      double a = 1234e-5;
6      printf("%f\n", a);
7      printf("%e\n", a);
8      return 0;
9  }

```

b) Geben Sie folgende Werte jeweils in einer eigenen Zeile in Fließkommenschreibweise aus:

- die dritte Potenz von 5.0
- den Tangens von 1.5
- den natürlichen Logarithmus von 12345.0
- den Logarithmus zur Basis 10 von 4321.0
- den Wert der Exponentialfunktion angewendet auf 11.0
- den kleinsten ganzzahligen Wert, der nicht kleiner ist als das Quadrat von 2.5
- den größten ganzzahligen Wert, der nicht größer ist als der Logarithmus zur Basis 10 von 9999.0

Lösung:

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main(void)
5  {
6      printf("pow(5.0, 3.0) = %e\n", pow(5.0, 3.0));
7      printf("tan(1.5) = %e\n", tan(1.5));
8      printf("log(12345.0) = %e\n", log(12345.0));
9      printf("log10(4321.0) = %e\n", log10(4321.0));
10     printf("exp(11.0) = %e\n", exp(11));
11     printf("ceil(pow(2.5, 2.0)) = %e\n", ceil(pow(2.5, 2.0)));
12     printf("floor(log10(9999.0)) = %e\n", floor(log10(9999.0)));
13     return 0;
14 }

```

Aufgabe 2.4 (Typumwandlung, Rundung und Overflow)

a)

- Deklarieren Sie eine **double**-Variable **a** und weisen Sie ihr den Wert **DBL_MAX** zu.
- Geben Sie den Wert von **a** aus.
- Addieren Sie **DBL_MAX** auf den Wert von **a**.
- Geben Sie wiederum den Wert von **a** aus.

Lösung:

```
1  #include <stdio.h>
2  #include <float.h>
3
4  int main(void)
5  {
6      double a = DBL_MAX;
7      printf("%e\n", a);
8      a += DBL_MAX;
9      printf("%e\n", a);
10     return 0;
11 }
```

b)

- Deklarieren Sie eine **double**-Variable **b**.
- Deklarieren Sie eine **int**-Variable **c**.
- Weisen Sie **b** den Wert 2.5 zu.
- Weisen Sie **c** den Wert 2.5 zu.
- Geben Sie den Wert beider Variablen getrennt durch ein Leerzeichen aus.
- Multiplizieren Sie beide Variablen jeweils mit 5.
- Geben Sie den Wert beider Variablen getrennt durch ein Leerzeichen aus.

Lösung:

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      double b;
6      int c;
7      b = 2.5;
8      c = (int) 2.5;
9      printf("%f %i\n", b, c);
10     b *= 5;
11     c *= 5;
12     printf("%f %i\n", b, c);
13     return 0;
14 }
```

Aufgabe 2.5 (Wiederholungen)

a)

- Deklarieren Sie eine **int**-Variable **a**, generieren Sie eine ganze Zufallszahl und weisen Sie **a** als Wert den Rest bei ganzzahliger Division der Zufallszahl durch 10 zu.

- Geben Sie mit einer **while**-Schleife alle Quadratwurzeln zwischen 1 und **a** jeweils in Fließkommadarstellung mit drei Nachkommastellen sowie getrennt durch ein Leerzeichen aus.

Die Ausgabe sieht für **a==8** so aus:

1.000e+00 1.414e+00 1.732e+00 2.000e+00 2.236e+00 2.449e+00 2.646e+00 2.828e+00

Lösung:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  #include <math.h>
5
6  int main(void)
7  {
8      int a, i = 1;
9      srand(time(NULL));
10     a = rand() % 10;
11     while (i <= a)
12         printf("%.3e ", sqrt(i++));
13     return 0;
14 }
```

b)

- Deklarieren Sie eine **int**-Variable **n**, generieren Sie eine ganze Zufallszahl und weisen Sie **n** als Wert den Rest bei ganzzahliger Division der Zufallszahl durch 15 zu.
- Berechnen Sie mit einer **for**-Schleife die Zahl 3^n , indem Sie das Zwischenergebnis in einer weitere Variable speichern, die zu Beginn mit dem Wert 1 initialisiert ist und in jedem Schleifendurchlauf die Zahl 3 hinzumultipliziert wird.
- Geben Sie zeilenweise das Zwischenergebnis des Produkts nach jedem Schleifendurchlauf aus.

Die Ausgabe sieht für **n==7** so aus:

3
9
27
81
243
729
2187

Lösung:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main(void)
6  {
7      int n, a = 1, i;
8      srand(time(NULL));
9      n = rand() % 15;
10     for (i = 0; i <= n; ++i) {
11         a *= 3;
12         printf("%i\n", a);
13     }
14     return 0;
15 }
```