

Aufgabe 4.0 Wechsel der Entwicklungsumgebung (Pflicht)

In der Datei `vs2010.pdf` ist beschrieben, wie Visual Studio 2010 beim ersten Start zu konfigurieren ist, wie man Projekte anlegt, und was die am häufigsten auftretenden Fehlermeldungen bedeuten. Beachte insbesondere den Hinweis zu `scanf` auf Seite 3!

Aufgabe 4.1 Unsere ersten Schleifen (Pflicht)

Diskutiere die Funktionen in `schleifen.c` mit Deinem Betreuer. Insbesondere solltest Du bei den Zählschleifen folgende Fragen beantworten können:

- Wie oft wird der Schleifenrumpf betreten?
- Welchen Wert hat der Schleifenzähler, wenn der Schleifenrumpf das erste Mal betreten wird?
- Welchen Wert hat der Schleifenzähler, wenn der Schleifenrumpf das letzte Mal betreten wird?
- Welchen Wert hat der Schleifenzähler, wenn die Schleifenbedingung `false` ist?

Aufgabe 4.2 Quadratzahlen (Pflicht)

Schreibe ein Programm, das den Benutzer nach einer Obergrenze fragt und anschließend sämtliche Quadratzahlen bis zu dieser Obergrenze auf die Konsole schreibt:

Obergrenze? 10

```
1 * 1 = 1
2 * 2 = 4
3 * 3 = 9
4 * 4 = 16
5 * 5 = 25
6 * 6 = 36
7 * 7 = 49
8 * 8 = 64
9 * 9 = 81
10 * 10 = 100
```

Bonusaufgabe: Schau dir die Differenzen der aufeinanderfolgenden Quadratzahlen an: 3, 5, 7, 9, 11... Fällt Dir ein Muster auf? Schaffst Du es, basierend auf diesem Muster, die Liste der Quadratzahlen zu generieren, ohne den Multiplikationsoperator zu verwenden?

Aufgabe 4.3 Multiplikationstabelle (Pflicht)

Schreibe ein Programm, das die Multiplikationstabelle mit Hilfe von verschachtelten Zählschleifen auf die Konsole schreibt:

		1	2	3	4	5	6	7	8	9	10
----	+	----	----	----	----	----	----	----	----	----	----
1		1	2	3	4	5	6	7	8	9	10
2		2	4	6	8	10	12	14	16	18	20
3		3	6	9	12	15	18	21	24	27	30
4		4	8	12	16	20	24	28	32	36	40
5		5	10	15	20	25	30	35	40	45	50
6		6	12	18	24	30	36	42	48	54	60
7		7	14	21	28	35	42	49	56	63	70
8		8	16	24	32	40	48	56	64	72	80
9		9	18	27	36	45	54	63	72	81	90
10		10	20	30	40	50	60	70	80	90	100

Als Platzhalter für `printf` bietet sich für diesen Zweck `%3d` an, d.h. 3 Dezimalziffern. Dann sind die Zahlen schön gleichmäßig rechtsbündig ausgerichtet.

Aufgabe 4.4 Pseudo-Zufallszahlen (Pflicht)

Das folgende Beispiel zeigt die idiomatische Erzeugung von Pseudo-Zufallszahlen in C:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int i, wuerfel_wurf;
    /* Zufallszahlengenerator EINMALIG initialisieren */
    srand((unsigned)time(0));
    for (i = 0; i < 10; ++i)
    {
        wuerfel_wurf = 1 + rand() % 6;
        printf("Der Wuerfel zeigt eine %d.\n", wuerfel_wurf);
    }
}
```

Tippe das Programm ab und probiere es aus.

Ein beliebter Anfängerfehler ist es, `srand` mehr als einmal aufzurufen, zum Beispiel innerhalb der Schleife unmittelbar vor dem Aufruf von `rand`. Was passiert dann?

Was passiert, wenn man `srand` überhaupt nicht aufruft?

Was passiert, wenn man `srand(42)` oder `srand(deine_lieblingszahl)` aufruft?

Aufgabe 4.5 Zahlenraten (Kür)

Implementiere das Spiel „Zahlenraten“. Das Spiel könnte z.B. wie folgt ablaufen:

Herzlich willkommen beim fröhlichen Zahlenraten! Ich habe mir eine Zahl zwischen 1 und 100 ausgedacht. Ihre Aufgabe ist es, diese Zahl in möglichst wenigen Schritten zu raten.

Ihr Tipp? 42

42 ist zu klein!

Ihr Tipp? 70

70 ist zu groß!

Ihr Tipp? 80

Ich hatte Ihnen doch bereits gesagt, dass 70 zu groß ist!
Trottel...

Ihr Tipp? 50

50 ist zu klein!

Ihr Tipp? 49

Ich hatte Ihnen doch bereits gesagt, dass 50 zu klein ist!
Trottel...

Ihr Tipp? 60

Glückwunsch, Sie haben die Zahl mit 6 Versuchen erraten!

Aufgabe 4.6 Zahlenraten mit vertauschten Rollen (Kür)

Implementiere eine Variante von Zahlenraten mit vertauschten Rollen.

Der Computer soll also die Zahl des Benutzers mit möglichst wenigen Schritten erraten.

Auf keinen Fall sollte der Benutzer am Anfang aufgefordert werden, die Zahl einzugeben!

Der Computer schreibt Tipps auf die Konsole, und der Benutzer muss wahrheitsgemäß antworten, ob der Tipp korrekt ist, oder ob die getippte Zahl zu klein oder zu groß ist.