

Aufgabe 1. Installiere einen Compiler auf deinem Computer und kompiliere ein Hallo-Welt-Programm. Informationen dazu und Cygwin findest du im Skript auf dem USB-Stick oder der Webseite des Kurses:

<http://www.ah-effect.net/>

Aufgabe 2. Was machen folgende drei Algorithmen?

Algorithmus 1

Eingabe ist eine ganze Zahl $c \in \mathbb{N}$. Ausgabe ist entweder »Ja« oder »Nein«.
Schritte:

- Setze n auf 2.
 - Wiederhole ...
 - Falls $n > \sqrt{c}$, so gib »Ja« aus und beende das Programm.
 - Falls n ein Teiler von c ist, so gib »Nein« aus und beende das Programm.
 - Erhöhe n um eins.
-

Algorithmus 2

Eingabe sind die ganzen Zahlen $a, b \in \mathbb{N}$. Ausgabe ist eine ganze Zahl $k \in \mathbb{N}$.
Schritte:

- Wiederhole ...
 - Falls $a = 0$ ist, gib b aus und beende das Programm.
 - Falls $b = 0$ ist, gib a aus und beende das Programm.
 - Falls $a > b$ ist, setze a auf $a - b$.
 - Falls obige Bedingung nicht erfüllt ist, setze b auf $b - a$.
-

Algorithmus 3

Eingabe ist eine reelle nicht-negative Zahl $a \in \mathbb{R}_0^+$. Ausgabe ist eine reelle Zahl $x \in \mathbb{R}$.

Schritte:

- Setze x auf 2 und y auf 1.
 - Wiederhole ...
 - Falls $|x - y| \leq 10^{-10}$ ist, gib x aus und beende das Programm.
 - Setze x auf y .
 - Setze y auf $\frac{1}{2}(x + \frac{a}{x})$.
-

Aufgabe 3. Wettbewerb: Gegeben ist folgender Programmrumppf:

```
1      #include <stdio.h>
2      int main(int argc, char **argv) {
3          int x = 2;
4          /* dein Code hier */
5          printf("%i\n", x);
6          return 0;
7      }
```

Füge an der markierten Stelle C-Code ein, sodass der Wert von $2^{(3^3)}$ ausgegeben wird. Erlaubt sind aber nur die Zeichen

`x + - * / = ;`

Versuche durch geschicktes Multiplizieren und Nutzung von Variablen mit möglichst wenig Zeichen auszukommen. Zeilenumbrüche und Leerzeichen können nach belieben verwendet werden, da sie vom Compiler ignoriert werden.

Diese Art ein Programm zu schreiben nennt man »Code Golf« und ist ein witziger Zeitvertreib.

Aufgabe 4. Schreibe ein Programm, das den Wert der folgenden Funktion ausgibt (für eine fest in den Quellcode geschriebene `int`-Variable):

$$f(n) = \begin{cases} \frac{n}{2} & \text{wenn } n \text{ gerade} \\ \frac{n+1}{2} & \text{wenn } n \text{ ungerade} \end{cases}$$

Und das geht natürlich nur mit Wissen aus der Vorlesung.