

Aufgabe 1. Das folgende Programm sollte die Summe der ersten n Zahlen berechnen. Allerdings enthält es 5 Fehler. Finde sie alle!

```
1  /* Summe der ersten n Zahlen.
2   * (c) 2015 Clelia und Johannes */
3
4  #include <stdio.h>
5
6  int main () {
7      int n = 10;          /* Addiere bis zu dieser Zahl */
8      int i;
9      int summe;          /* speichert Zwischenergebnis */
10
11     i = 0;
12
13     while (i < n) {
14         summe += i        /* addiere ite Zahl auf summe */
15     }
16     printf ("Das Ergebnis ist %f.\n", summe);
17     return 0;
18 }
```

Was fällt dir auf, wenn du das Programm nach Korrektur ausführst? Könnte man diese Aufgabe nun also effizienter implementieren?

Aufgabe 2.

- a) Implementiere den Primzahltest (Algorithmus 1) von gestern.
- b) Schreibe ein Programm, dass jeweils die nächste Primzahl nach 20000, 30000 und 40000 findet.

Aufgabe 3. Für $a \in \mathbb{R}^+$ konvergiert die Folge (a_n) mit $a_0 = a$ und

$$a_{n+1} = \frac{1}{2} \left(a_n + \frac{a}{a_n} \right)$$

gegen \sqrt{a} . Implementiere damit einen Wurzellalgorithmus.

Aufgabe 4. Implementiere den Algorithmus 2 von gestern, welcher den größten gemeinsamen Teiler zweier Zahlen berechnet.

Aufgabe 5. Implementiere den Cosinus über seine Reihendarstellung mit einer for-Schleife. Du kannst die Formel bei Wikipedia nachschlagen, sie selbst entwickeln oder diese hier verwenden:

$$\cos(x) = \sum_{k=0}^{\infty} (-1)^k \cdot \frac{x^{2k}}{(2k)!}$$

Aufgabe 6. Schreibe ein Programm, um den Wert der Reihe

$$\sum_{k=1}^{\infty} \frac{1}{k^2}$$

zu berechnen. Er sollte $\frac{\pi^2}{6}$ sein. Wichtig ist, sich ein geeignetes Abbruchkriterium zu überlegen.