

Aufgabe 1. Berechne die Summe der ersten n ungeraden Zahlen mit einer `while`-Schleife. Wie ist die Ausgabe für $n = 1, \dots, 15$? Was fällt dir auf? Könnte man dies nun also effizienter programmieren?

Aufgabe 2. Das folgende Programm sollte die Summe der ersten n Zahlen berechnen. Allerdings enthält es 6 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 }
```

Aufgabe 3.

- Implementiere den Primzahltest (Algorithmus 1) von gestern.
- Schreibe ein Programm, das die nächste Primzahl nach einer gegebenen Zahl N ausgibt. Was ist die nächste Primzahl nach 20000, 30000 und 40000?

Aufgabe 4. 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 Wurzelalgorithmus. Dies ist übrigens der dritte Algorithmus von gestern.

Aufgabe 5. Implementiere den Algorithmus 2 von gestern, welcher den größten gemeinsamen Teiler zweier Zahlen berechnet.

Aufgabe 6. Implementiere den Cosinus über seine Reihendarstellung mit einer `while`-Schleife. Die Reihendarstellung des Cosinus ist gegeben durch:

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

Hinweis: Es ist nicht sinnvoll, jedes Mal den gesamten Summanden neu zu berechnen. Warum? Was ist der bessere Weg?

