

Aufgabe 1. Im folgenden Programm finden sich 5 Fehler. Welche?

```
1 #include <iostream>
2 #include <vector>
3
4 /* Liefert das Maximum einer Menge von nichtnegativen Zahlen.
5    Wenn keine Zahlen gegeben werden, ist die Ausgabe 0. */
6 unsigned maximum(const std::vector<unsigned> & zahlen)
7 {
8     unsigned result = 0;
9
10    for(unsigned i = 0; i < zahlen.size; ++i)
11    {
12        if(zahlen[i] > result)
13        {
14            result = zahlen[i];
15        }
16    }
17    return result;
18 }
19
20 int main()
21 {
22     std::vector<int> eintraege;
23
24     std::cout << "Geben Sie nicht-negative Zahlen ein. " << std::endl;
25     std::cout << "Mit einer negativen Zahl koennen Sie die Eingabe beenden" << std::endl;
26
27     do
28     {
29         int eingabe;
30         std::cout << "Naechster Eintrag bitte. ";
31         std::cin >> eingabe;
32         if(eingabe >= 0)
33         {
34             eintraege.push(eingabe);
35         }
36     } while(eingabe >= 0)
37
38     if(eintraege.size() > 0)
39     {
40         int groesste_zahl = maximum(eintraege);
41         std::cout << "Das Maximum ist: " << groesste_zahl << std::endl;
42     }
43     return 0;
44 }
```

Aufgabe 2. Schreiben Sie eine Funktion, die einen Vektor von positiven ganzen Zahlen per `const`-Referenz übergeben bekommt und die Einträge des Vektors der Größe nach sortiert auf den Bildschirm ausgibt.

Wie Sie die Sortierung bestimmen, ist Ihnen überlassen. Im einfachsten Fall finden Sie jeweils, indem Sie den Vektor durchlaufen, das kleinste Element, dann das zweitkleinste usw.

Aufgabe 3. Implementieren Sie die Riemannsche Zeta-Funktion für $s \in \mathbb{R}$:

$$\zeta(s) := \sum_{k=1}^{\infty} \frac{1}{k^s}$$

Benutzen Sie dazu Ihre Lösung von Aufgabe 2 von Zettel 4. Testen Sie die Funktion für einige Werte $s \in]1, 3[$. Für Werte $s \leq 1$ gilt $\zeta(s) = \infty$.

Aufgabe 4. Implementieren Sie (unter Benutzung von `std::vector`) einen Algorithmus, der bis zu einer gegebenen Zahl $n \in \mathbb{N}$ alle Primzahlen bestimmt. Gehen Sie dabei wie folgt vor: Sie betrachten nacheinander alle Zahlen von 1 bis n und merken sich für alle Vielfachen der gerade betrachteten Zahl, dass sie keine Primzahlen sind.

Vergleichen Sie Ihr Programm auch mit einer Lösung des Problems, die darin besteht, dass man die Zahlen bis n jeweils einzeln darauf testet, ob sie Primzahlen sind (wie Sie es für Aufgabe 2 (a) von Zettel 2 implementiert haben).