



Aufgabe 1. Lege ein Modul `mymath.c` / `mymath.h` an, in dem du die bisher geschriebenen Funktionen auslagerst.

Wir brauchen im folgenden eine Potenzfunktion, die zwei Fließkommazahlen als Argumente akzeptiert. Falls du diese Funktion gestern geschrieben hast, sollte sie jetzt im `mymath`-Modul verfügbar sein. Diese Funktion wird aber vermutlich zu langsam sein, daher gibt es die funktion

```
double pow(double x, double y);
```

in der Systemheader `<math.h>`. Im Skript findest du im Anhang eine vollständige Referenz einiger Systembibliotheken.

Aufgabe 2. Implementiere die Riemann'sche Zeta-Funktion für $s \in \mathbb{N}$:

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

Aufgabe 3. Implementiere die Funktion, die den Inhalt zweier `int`-Variablen vertauscht.

Aufgabe 4. Schreibe Funktionen `square_to` und `root_to`, die einen `double`-Pointer entgegen nehmen, die dort stehende Variable quadrieren bzw. daraus die Wurzel ziehen und das Ergebnis sowohl zurück geben als auch an die gleiche Speicherstelle schreiben.

Aufgabe 5. Erweitere das „`mymath`“-Modul noch um eine Funktion, die zu den drei Koeffizienten $a, b, c \in \mathbb{R}$ einer quadratischen Gleichung

$$a \cdot x^2 + b \cdot x + c = 0$$

die Lösungen berechnet.

Tip: Eine Funktion kann nur einen Wert als Rückgabewert haben. Um mehr als einen Wert zurück zu geben, kann man Pointer verwenden.