Aufgabe 1. a) Implementiere für $x \in \mathbb{R}$ und $n \in \mathbb{N}$ eine Potenzfunktion $x^n =$ `power(x, n)` mit der Double-and-Add-Methode:

$$power(x,n) = \begin{cases} 1 & \text{wenn } n = 0 \\ x \cdot power\left(x^2, \frac{n-1}{2}\right) & \text{wenn } n \text{ ungerade} \\ power\left(x^2, \frac{n}{2}\right) & \text{wenn } n \text{ gerade} \end{cases}$$

zuerst mal rekursiv.

b) Implementiere eine Potenzfunktion `naiv_power(x, n)`, indem du eine Schleife von 1 bis n laufen lässt und bei jedem Durchlauf eine mit 1 Initialisierte Variable mit x multipliziert. Berechne $0,9999999999^{2000000000}$ einmal mit `power(x, n)` von oben und einmal mit `naiv_power(x, n)` (es sollte ca. $0,818731$ raus kommen).

c) * Implementiere die Double-and-Add-Methode iterativ, also ohne rekursiven Aufruf.

d) * Frage einen Tutor wie man Zeit messen kann und vergleiche die Laufzeiten der 3 Funktionen.

Aufgabe 2. Diese Aufgabe wird auf eine `power(x, y)`-Funktion führen, die für beliebige $x \in \mathbb{R}^+$ und $y \in \mathbb{R}$ den Wert von x^y berechnet. Du kannst mit dieser Funktion dein „mymath"-Modul um eine weitere Funktion erweitern.

- Implementiere die Exponential-Funktion `expo(x)`, die e^x mithilfe folgender Reihendarstellung:

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

- Implementiere eine Logarithmus-Funktion `logarithm(x)`, die $\ln(x)$ mithilfe folgender Reihedarstellung berechnet:

$$\ln(x) = 2 \cdot \sum_{k=0}^{\infty} \left(\frac{x-1}{x+1}\right)^{2k+1} \frac{1}{2k+1}$$

- Verwende die Formel

$$x^y = e^{y \cdot \ln(x)}$$

um `power(x, y)` zu bestimmen.

Aufgabe 3. Lege ein Modul `mymath.c` / `mymath.h` an, in dem du die bisher geschriebenen Funktionen auslagerst.

Aufgabe 4. 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. Andernfalls 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.

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

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

Tipp: Vielleicht kannst du die zuletzt implementierte Funktion `power(x, n)` dafür verwenden.

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 und die größere Lösung zurück gibt.