Aufgabe 1. Implementiere die Signumsfunktion `sgn(x)`, den Absolutbetrag `betrag(x)`, den Cosinus `cos(x)`, den euklidischen Algorithmus zur Berechnung des `ggt(x)` und den Heronalgorithmus `wurzel(x)` als Funktionen.

Aufgabe 2. 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:

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

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 ohne rekursiven Aufruf.

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

Aufgabe 3. 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. Dies ist eine echte Verallgemeinerung zu oben, da dort $y \in \mathbb{N}$ vorausgesetzt war.

- 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.