Aufgabe	A1	A2	A3	A4	$\sum$
Punkte					

Aufgabe 1. Zahlendarstellung

a) Berechnung der Determinante einer 2×2 Matrix.

```
double determinante_double(double a, double b, double c, double d) {
    return a*d - b*c;
}

float determinante(float a, float b, float c, float d) {
    return a*d - b*c;
}
```

Berechnung der Determinante

Das Ergebnis bei Verwendung von `float` ist 10000 und damit nicht exakt. Das liegt an der zu geringen Größe eines `float`'s, der nur rund 7 Dezimalstellen exakt speichern kann, danach wird gerundet. Das führt in diesem Fall zum Verlust aller Nachkommastellen. Bei Verwendung des Datentyps `double` reichen die rund 16 Stellen aus, um das Ergebnis exakt darzustellen.

b) Assoziativität bei `float`s

```
float testAssoziativitaet() {
    for (int n = 6; n <= 14; n++) {
        // (a+b)+c
        float vers1 = (pow(10, n) + pow(-10, n)) + pow(10,-n);
        // a+(b+c)
        float vers2 = pow(10, n) + (pow(-10, n) + pow(10,-n));
        print(n, vers1, vers2, 0);
    }
}
```

Vergleich der zwei Versionen

Das Ergebnis ist bereits ab $n = 5$ nicht mehr assoziativ. Das liegt daran, dass bei $(a + b) + c$ zunächst $10^n - 10^n$ berechnet wird, das aber immer null ist und danach einfach 10^{-n} ausgewertet wird. Dabei kann die gesamte Präzision des `float`'s für die Nachkommastellen von 10^{-n} verwendet werden. Bei $a + (b + c)$ wird erst $-10^n + 10^{-n}$ berechnet. Hierbei reicht die Präzision nicht aus, um die Nachkommastellen darzustellen, da jetzt die Zahl bereits vor dem Komma sehr groß ist. Dadurch wird $-10^n + 10^{-n}$ auf -10^n gerundet und dann mit 10^n addiert, was dann null ergibt.

Aufgabe 2. Effektiver Zinssatz

Programm siehe `uebung4.cpp`

Die Ergebnisse erleiden durch Runden und den begrenzten Speicherplatz des `float`'s einige Ungenauigkeiten. Die Differenz zum Grenzwert wird wie erwartet immer geringer.

Die Genauigkeit der `double` Werte ist höher.

Aufgabe 3. Multiplikation im Zweierkomplement

Zeigen Sie, dass für $n \in \mathbb{N}$ mit

$$a, b, a \cdot b \in \left[-2^{n-1}, 2^{n-1} - 1\right].$$

für die Multiplikation im Zweierkomplement folgendes gilt:

$$d_n(a \cdot b) = s_n(d_n(a) \cdot d_n(b)).$$

Beweis. Seien $a, b \in \left[-2^{n-1}, 2^{n-1} - 1\right]$ mit $-2^{n-1} \leq a \cdot b \leq 2^{n-1} - 1$ beliebig.

Fallunterscheidung:

(i) $a, b \geq 0$. Dann ist $a \cdot b \geq 0$. Damit

$$s_n(d_n(a) \cdot d_n(b)) = s_n(a \cdot b) = a \cdot b = d_n(a \cdot b).$$

(ii) $a < 0, b > 0$ (analog $a > 0, b < 0$). Dann ist $a \cdot b < 0$.

$$\begin{aligned} s_n(d_n(a) \cdot d_n(b)) &= s_n((2^n - |a|) \cdot b) \\ &= s_n((2^n + a) \cdot b) \\ &= s_n(2^n \cdot b + a \cdot b) \\ &= s_n(2^n \cdot b - |a + b|) \\ &= 2^n - |a + b| \\ &= d_n(a \cdot b). \end{aligned}$$

(iii) $a < 0, b < 0$. Dann ist $a \cdot b > 0$.

$$\begin{aligned} s_n(d_n(a) \cdot d_n(b)) &= s_n((2^n - |a|)(2^n - |b|) \\ &= s_n((2^n \cdot 2^n + 2^n \cdot b + a \cdot 2^n + a \cdot b) \\ &= a \cdot b \\ &= d_n(a \cdot b). \end{aligned}$$

(iv) $a < 0, b = 0$ (analog $a = 0, b < 0$). Dann ist $a \cdot b = 0$.

$$s_n(d_n(a) \cdot d_n(b)) = s_n((2^n - |a|) \cdot 0) = s_n(0) = 0 = d_n(0) = d_n(a \cdot b).$$

$\square$