

Aufgabe	A1	A2	A3	Σ
Punkte				

Aufgabe 1. a) Zur Basis 10 dargestellt:

$$(0.5731 \times 8^5)_8 = (5 \cdot 8^{-1} + 7 \cdot 8^{-2} + 3 \cdot 8^{-3} + 1 \cdot 8^{-4}) \cdot 8^5 = 24264 = 0.24264 \times 10^5.$$

b) Die Zahl $x_1 = (0.3)_{10} \in \mathbb{R}$ in der normierten Fließkommadarstellung $\mathbb{F}(2, 11, 2)$ ergibt

$$(0.10011001100 \times 2^1)_2.$$

Zurück in $\mathbb{F}(10, r, 1)$ ergibt

$$(0.2998046875 \times 10^0)_{10}.$$

Das heißt für $r = 1$, $r = 2$ und $r = 3$ ergibt natürliche Rundung jeweils $(0.3)_{10}$. Größere r ergeben andere Ergebnisse.

c) Die größte positive Zahl in $\mathbb{F}(4, 6, 2)$ ergibt sich im Dezimalsystem dargestellt als:

$$3 \cdot (4^{-1} + 4^{-2} + 4^{-3} + 4^{-4} + 4^{-5} + 4^{-6}) \cdot 4^{3 \cdot 4^0 + 3 \cdot 4^1} = 0.1073 \cdot 10^{10}.$$

Die kleinste negative Zahl in $\mathbb{F}(3, 7, 1)$ ergibt sich im Dezimalsystem dargestellt als:

$$-2 \cdot (3^{-1} + 3^{-2} + \dots + 3^{-7}) \cdot 3^{2 \cdot 3^0} \approx -0.8996 \cdot 10^1.$$

Damit folgt für den maximalen Abstand zweier Zahlen aus $\mathbb{F}(4, 6, 2)$ und $\mathbb{F}(3, 7, 1)$:

$$\max_{x_2, x_3} |x_2 - x_3| \approx 0.1073 \cdot 10^{10} + 0.8996 \cdot 10^1 \approx 0.1073 \cdot 10^{10}.$$

d) Die Aussage ist falsch. Gegenbeispiel: $\mathbb{F}(2, 2, 1)$. Die Menge der darstellbaren Zahlen ist hier, analog zur Vorlesung:

$$\mathbb{F}(2, 2, 1) = \left\{ -\frac{3}{2}, -1, -\frac{3}{4}, -\frac{1}{2}, -\frac{3}{8}, -\frac{1}{4}, 0, \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{3}{4}, 1, \frac{3}{2} \right\}.$$

Hier ist sofort zu sehen, dass beispielsweise für $x_4 = 1$ gilt:

$$\left| 1 - \frac{3}{4} \right| = \frac{1}{4} \neq \frac{1}{2} = \left| \frac{3}{2} - 1 \right|.$$

Aufgabe 2. Die ersten zehn Folgenglieder in Dezimaldarstellung (nicht normiert):

n	Aufrunden	gerades Runden
0	2.46	2.46
1	2.47	2.46
2	2.48	2.46
3	2.49	2.46
4	2.50	2.46
5	2.51	2.46
6	2.52	2.46
7	2.53	2.46
8	2.54	2.46
9	2.55	2.46
10	2.56	2.46

Beim Aufrunden ist deutlich zu sehen, dass, da nach jedem Rechenschritt die 4. Stelle auf 5 steht, jedes mal aufgerundet wird. Da zwei Rundungen pro Folgenglied stattfinden, führt das zu einer Erhöhung um $2 \cdot 0.005 = 0.01$ pro Folgenglied.

Beim geraden Runden wird zwar nach der ersten Operation aufgerundet, nach der zweiten jedoch nicht. Deshalb bleibt das Ergebnis exakt.

Aufgabe 3. Programm zum Test der Nulladdition:

```
1 #include<iostream>
2 #include<iomanip>
3
4 using namespace std;
5
6 int main() {
7     // set output width and precision
8     cout << setprecision(17);
9     cout << setw(10);
10
11     // float
12     cout << "Enter a float > ";
13     float x;
14     cin >> x;
15     x = x + 1;
16     cout << x << endl;
17
18     // double
19     cout << "Enter a double > ";
20     double y;
21     cin >> y;
22     y = y + 1;
23     cout << y << endl;
24 }
```

nulladdition.cpp

- a) Bei Verwendung von `float` muss x als $5 \cdot 10^{-8}$ gewählt werden, bei Verwendung von `double` muss x als 10^{-16} gewählt werden, damit exakt 1 zurückgegeben wird.
- b) Das ist bei weitem nicht die kleinste positive Zahl, die `float` bzw. `double` darstellen können, da im Exponenten noch weit mehr Stellen zur Verfügung stehen, um sehr viel kleinere positive Zahlen darstellen zu können.

Allerdings kann dies hier nicht genutzt werden, da durch die Addition mit 1 gerundet werden muss.