



Aufgabenblatt 5 Ausgabe: 15.11., Abgabe: 22.11. 24:00

Gruppe	
Name(n)	Matrikelnummer(n)

Aufgabe 5.1 (Punkte 5+5+5+5+5)

IEEE 754 Gleitkommazahlen: Konvertieren Sie die folgenden IEEE 754 Gleitkommadarstellungen (einfache Genauigkeit) in das Dezimalsystem, bzw. geben Sie einen passenden Wert an. Von den 23-bit der Mantisse sind hier nur die oberen 8-bit angegeben, alle anderen Stellen sind 0.

- s exponent mantisse*
- (a) 1 0000 0000 0000 0000
 - (b) 0 0000 0000 1010 0000
 - (c) 0 1000 0000 1010 0000
 - (d) 1 0111 1111 0101 0000
 - (e) 1 1111 1111 0101 0000

Aufgabe 5.2 (Punkte 10+10+10+10)

Arithmetische Operationen mit Gleitkommazahlen: Gegeben seien zwei IEEE 754 Zahlen A und B (32-bit). Sie sind wie in der ersten Aufgabe formatiert, als: $s\ eeee\ eeee\ mmmmm\ mmmmm$.

$$A = 0\ 1000\ 0011\ 0101\ 0000 \quad \text{und}$$
$$B = 1\ 1000\ 0010\ 0010\ 0000$$

Berechnen Sie ohne Umwandlung in das Dezimalsystem die folgenden Ausdrücke.¹ Alle Ergebnisse sollen wieder als IEEE 754 Zahlen (wie oben) dargestellt werden. Geben Sie dabei immer auch die einzelnen Rechenschritte an.

- (a) $A + B$
- (b) $A - B$
- (c) $A \cdot B$
- (d) $(A - B) / (A + B)$

¹Allerdings kann es helfen Kontrollrechnungen dezimal durchzuführen.

Aufgabe 5.3 (Punkte 10+10+5)

UTF-8 Zeichen: Entschlüsseln Sie mit Hilfe der Vorlesungsunterlagen den folgenden hexadezimal codierten Text.

```
55 54 46 2D 38 20 6E 75 74 7A 74 0D 0A 09 31 2D
62 79 74 65 3A 20 61 0D 0A 09 32 2D 62 79 74 65
3A 20 C3 A4 0D 0A 09 33 2D 62 79 74 65 3A 20 E2
82 BF 0D 0A 09 34 2D 62 79 74 65 3A 20 F0 9F 98
81 0D 0A 2E 2E 2E 0D 0A
```

- Wie sieht der Text aus? Notieren Sie dazu die Textdarstellung (mit Steuerzeichen).
- Erläutern Sie anhand der „*n*-byte“ Beispiele aus dem Text, wie aus der UTF-8 Codierung der Unicode-Wert berechnet wird.
- Was verrät Ihnen der Text über den Rechner mit dem er erstellt worden ist?

Aufgabe 5.4 (Punkte 10)

Base-64 Codierung: Wie in der Vorlesung skizziert, werden bei der Base-64 Codierung jeweils drei 8-bit Eingangswerte durch vier 6-bit Ausgangswerte ersetzt, die dann zur Datenübertragung in (7-bit) ASCII-Zeichen codiert werden.

Beschreiben Sie durch Logische- und Schiebe-Operationen, wie aus den drei Eingabezeichen $a_1 \dots a_3$, die vier 6-bit Ausgangswerte $b_1 \dots b_4$ berechnet werden. Vervollständigen Sie dazu die Ausdrücke `b..` im nachfolgenden Java-Code (Zeilen 3...6).

```
1  int a1, a2, a3;                // drei Zeichen, Wertebereich je 0..255
2
3  int b1 = ?
4  int b2 = ?
5  int b3 = ?
6  int b4 = ?
7
8  char[] base64table = new char[] { // Tabelle ersetzt  $b_i$  durch Zeichen
9    'A', 'B', ... 'Z',           // nicht vollständig ...
10   'a', 'b', ... 'z',
11   '0', '1', ... '9', '+', '/' };
12
13 String base64out =              // 4 Zeichen String als Ausgabe
14   base64table[ b1 ] +
15   base64table[ b2 ] +
16   base64table[ b3 ] +
17   base64table[ b4 ];
18 ...
```