

Ergänzungsfach Informatik

16. September 2008, Klassen 4ACD

Vorname: _____ Name: _____

Erlaubte Hilfsmittel: **keine**.

Aufgabe 1

6 Punkte

Rechnen Sie folgende positiven Ganzzahlen ins 2er-, 10er- und 16er-System um.

- a) 99 (dezimal) b) AF (hexadezimal) c) 1001100 (binär)

Aufgabe 2

2 Punkte

Rechnen Sie folgende positive Ganzzahl vom 2er- ins 16er-System um:

10101111111111101111101011001110

Aufgabe 3

4 Punkte

a) Schreiben Sie $\frac{1}{10}$ als Binärbruch.

b) Schreiben Sie den Binärbruch 0.00001011 in der Form $2^n \cdot 1.xxxxxxxxx$ mit $n \in \mathbb{Z}$.

Aufgabe 4

6 Punkte

Erklären Sie die Begriffe "Little Endian" und "Big Endian". Geben Sie je ein für jedermann (auch Nicht-Informatiker) verständliches Beispiel aus dem Alltag an.

Aufgabe 5

7 Punkte

In dieser Aufgabe werden in 1 Byte vorzeichenbehaftete Ganzzahlen im Zweierkomplement dargestellt.

- a) Geben Sie die kleinste (Achtung: negativ!) und grösste mögliche Zahl sowohl binär als auch dezimal an.
b) Geben Sie -64 in Binärdarstellung an.
c) Berechnen Sie $23 \& 12$.
d) Berechnen Sie $23 | 12$.

Aufgabe 6

6 Punkte

Eine mögliche Methode, rgb-Werte in 16-Bit zu codieren, ist für Rot die untersten 5 Bits zu gebrauchen, für Grün die nächsten 6 Bits (sieht man am Besten) und für Blau die obersten 5 Bits zu gebrauchen.

- a) Schreiben Sie eine Formel, bzw. einen Code-Abschnitt, der den drei rgb-Werten, gespeichert in r , g , b , die **Zahl** x errechnet, die dem 16-Bit Farbcode entspricht.
b) Schreiben Sie einen Code-Abschnitt, der aus dem 16-Bit Farbwert x die rgb-Werte r , g , b berechnet.

Ergänzungsfach Informatik, Bits und Bytes, 21. September 2010

Vorname: _____ Name: _____

Erlaubte Hilfsmittel: TI-89 Zeit: 60 Minuten.

Aufgabe 1

2 Punkte

Geben Sie einen für Menschen ausführbaren Algorithmus zur Umrechnung einer Zahl im Zehnersystem in eine Zahl im Binärsystem an.

Aufgabe 2

3 Punkte

Erklären Sie in 3-4 Sätzen, was alles an Umrechnungen passiert (ohne diese im Detail zu beschreiben), wenn der Computer den Wert eines gespeicherten Bytes (z.B. binär 0010'0011) als Dezimalzahl auf dem Bildschirm für das menschliche Auge lesbar ausgibt.

Aufgabe 3

1 Punkte

Das Hexadezimalsystem (16er System) ist omnipresent in der systemnahen Informatik. Nennen Sie 2 Gründe in je 1-2 Sätzen warum das Hexadezimalsystem so praktisch ist.

Aufgabe 4

2 Punkte

Es sollen mit 1 Byte positive und negative Zahlen im Zweierkomplement dargestellt werden.

- Schreiben Sie die Binärdarstellung von -16 auf und dokumentieren Sie ganz kurz mit Formeln und/oder Stichworten Ihre Umrechnung.
- Nennen Sie einen Grund, warum das Zweierkomplement so praktisch ist und illustrieren Sie dies mit einem Beispiel.

Aufgabe 5

1 Punkt

Schreiben Sie $\frac{3}{7}$ als Binärbruch. Dokumentieren Sie Ihre Umrechnung (Rechnungen, Formeln, eventuell Stichworte).

Aufgabe 6

1 Punkt

Erklären Sie in 1-2 Sätzen, einer Skizze und Formeln die Prinzipien, wie im Computer gebrochene Zahlen (Gleitkommazahlen, bzw. floats) dargestellt werden.

Aufgabe 7

2 Punkte

Erklären Sie in 2-3 Sätzen den Unterschied und Zusammenhang zwischen Unicode und der UTF-8 Codierung.

Aufgabe 8

2 Punkte

In der UTF-8 Codierung wird die Information in 2 Bytes wie folgt codiert:

$$\mathbf{110}x_{10}'x_9x_8x_7x_6 \quad \mathbf{10}x_5x_4'x_3x_2x_1x_0,$$

wobei die fetten Bits Start- und Folgebyte kennzeichnen und die als x_i geschriebenen Bits informationstragend sind.

Sei das erste Byte als Ganzzahl in der Variable "a" und das zweite Byte als Ganzzahl in der Variable "b" gespeichert. Mittels bitweise logischen und shift Operationen geben Sie an, wie aus den Variablen "a" und "b" die Binärzahl $x_{10}x_9x_8'x_7x_6x_5x_4'x_3x_2x_1x_0$ in eine weitere Ganzzahlvariable "c" gespeichert werden kann.

Aufgabe 9

1 Punkt

Wie viele verschiedene Zeichen könnten in UTF-8 in zwei Bytes theoretisch codiert werden (siehe Aufgabe 8)? Wie viele davon werden praktisch höchstens benutzt und warum?