

Zeichencodierung

Prüfungsvorbereitung

Aufgabe 1

Welchen dezimalen Wert muss man zu jedem ASCII-Code eines Grossbuchstabens addieren, um den ASCII-Code des entsprechenden Kleinbuchstabens zu erhalten?

Aufgabe 1

Wähle zum Beispiel den Buchstaben A:

Zeichen	hexadezimal	dezimal
A	0x41	$4 \cdot 16 + 1 \cdot 1 = 65$
a	0x61	$6 \cdot 16 + 1 \cdot 1 = 97$

Aufgabe 1

Wähle zum Beispiel den Buchstaben A:

Zeichen	hexadezimal	dezimal
A	0x41	$4 \cdot 16 + 1 \cdot 1 = 65$
a	0x61	$6 \cdot 16 + 1 \cdot 1 = 97$

Die Differenz beträgt $97 - 65 = 32$

Somit muss man 32 zum ASCII-Code eines Grossbuchstabens addieren, um den entsprechenden Kleinbuchstaben zu erhalten.

Aufgabe 2

Stelle den Text

Aua!

binär im ASCII-Code dar.

Aufgabe 2

Mit Hilfe der ASCII-Tabelle erhalten wir:

A	u	a	!
0x41	0x75	0x61	0x21
0100 0001	0111 0101	0110 0001	0010 0001

Aufgabe 3

Stelle den Violinschlüssel (U+1D11E) binär in der UTF-32-Codierung dar.

Aufgabe 3

Umrechnung der hexadezimalen Nummer in die binäre Darstellung:

0x1D11E $\Rightarrow$ 0001 1101 0001 0001 1110

Aufgabe 3

Umrechnung der hexadezimalen Nummer in die binäre Darstellung:

0x1D11E $\Rightarrow$ 0001 1101 0001 0001 1110

Links mit Nullen auffüllen, bis das Zeichen 32 Bit lang ist:

0000 0000 0000 0001 1101 0001 0001 1110

Aufgabe 4

Aus wie vielen Zeichen besteht der folgende UTF-8-Bitstrom?

11101111 10000000 10111111 01011010 11011010 10100011

Aufgabe 4

- 1. Zeichen 11101111 10000000 10111111
- 2. Zeichen 01011010
- 3. Zeichen 11011010 10100011

Also besteht der Bitstrom aus drei Zeichen.

Aufgabe 5

Welche hexadezimale Nummer hat das Unicode-Zeichen mit der folgenden binären UTF-8-Darstellung?

11100001 10001110 10101000

Aufgabe 5

11100001 10001110 10101000

Aufgabe 5

11100001 10001110 10101000

Codierungsinformation entfernen:

Aufgabe 5

11100001 10001110 10101000

Codierungsinformation entfernen:

0001 001110 101000

Aufgabe 5

11100001 10001110 10101000

Codierungsinformation entfernen:

0001 001110 101000

Bits von rechts in 4er-Gruppen zusammenfassen und ins
Hexadezimalsystem umrechnen:

Aufgabe 5

11100001 10001110 10101000

Codierungsinformation entfernen:

0001 001110 101000

Bits von rechts in 4er-Gruppen zusammenfassen und ins
Hexadezimalsystem umrechnen:

0001	0011	1010	1000
1	3	A	8

Aufgabe 5

11100001 10001110 10101000

Codierungsinformation entfernen:

0001 001110 101000

Bits von rechts in 4er-Gruppen zusammenfassen und ins Hexadezimalsystem umrechnen:

0001	0011	1010	1000
1	3	A	8

Das Zeichen hat die Nummer 0x13A8 bzw. U+13A8

Aufgabe 5

11100001 10001110 10101000

Codierungsinformation entfernen:

0001 001110 101000

Bits von rechts in 4er-Gruppen zusammenfassen und ins Hexadezimalsystem umrechnen:

0001	0011	1010	1000
1	3	A	8

Das Zeichen hat die Nummer 0x13A8 bzw. U+13A8

Nebenbei: 0x13A8 repräsentiert den Cherokee-Buchstaben GE.

Aufgabe 6

Bestimme die UTF-8-Darstellung der Herz-Dame mit der Unicode-Nummer U+1F0BD in binärer Form.

Aufgabe 6

0x1F0BD in Binärform:

Aufgabe 6

0x1F0BD in Binärform:

1	F	0	B	D
0001	1111	0000	1011	1101

Aufgabe 6

0x1F0BD in Binärform:

1	F	0	B	D
0001	1111	0000	1011	1101

Führenden Nullen sind (vorläufig) überflüssig und werden entfernt:

1 1111 0000 1011 1101

Somit müssen 17 Bits codiert werden. Wir erinnern uns:

Aufgabe 6

0x1F0BD in Binärform:

1	F	0	B	D
0001	1111	0000	1011	1101

Führenden Nullen sind (vorläufig) überflüssig und werden entfernt:

1 1111 0000 1011 1101

Somit müssen 17 Bits codiert werden. Wir erinnern uns:

0XXXXXXX	7 Bit
110XXXXX 10XXXXXX	8–11 Bit
1110XXXX 10XXXXXX 10XXXXXX	12–16 Bit
11110XXX 10XXXXXX 10XXXXXX 10XXXXXX	17–21 Bit

Aufgabe 6

0x1F0BD in Binärform:

1	F	0	B	D
0001	1111	0000	1011	1101

Führenden Nullen sind (vorläufig) überflüssig und werden entfernt:

1 1111 0000 1011 1101

Somit müssen 17 Bits codiert werden. Wir erinnern uns:

0XXXXXXX	7 Bit
110XXXXX 10XXXXXX	8–11 Bit
1110XXXX 10XXXXXX 10XXXXXX	12–16 Bit
11110XXX 10XXXXXX 10XXXXXX 10XXXXXX	17–21 Bit

Die 17 Bits müssen von rechts nach links an die freien Positionen (X) des Schemas für 17–21 Bit gesetzt werden. Fehlende Bits links werden wieder mit Nullen aufgefüllt.

11110000 10011111 10000010 10111101

Aufgabe 7

Vergleiche die Codierungen UTF-8 und UTF-32 in Bezug auf die folgenden Aspekte

- (a) Speicherbedarf für ein Zeichen
- (b) Gleichbehandlung der Zeichen verschiedener Sprachen
- (c) Codierungsaufwand

Aufgabe 7

(a) *Speicherbedarf für ein Zeichen:*

Aufgabe 7

(a) *Speicherbedarf für ein Zeichen:*

UTF-8-codierte Zeichen haben einen variablen Speicherbedarf von 1–4 Byte

Aufgabe 7

(a) *Speicherbedarf für ein Zeichen:*

UTF-8-codierte Zeichen haben einen variablen Speicherbedarf von 1–4 Byte

UTF-32-codierte Zeichen werden benötigen immer 4 Byte Speicherplatz.

Aufgabe 7

(a) *Speicherbedarf für ein Zeichen:*

UTF-8-codierte Zeichen haben einen variablen Speicherbedarf von 1–4 Byte

UTF-32-codierte Zeichen werden benötigen immer 4 Byte Speicherplatz.

(b) *Gleichbehandlung der Zeichen verschiedener Sprachen:*

Aufgabe 7

(a) *Speicherbedarf für ein Zeichen:*

UTF-8-codierte Zeichen haben einen variablen Speicherbedarf von 1–4 Byte

UTF-32-codierte Zeichen werden benötigen immer 4 Byte Speicherplatz.

(b) *Gleichbehandlung der Zeichen verschiedener Sprachen:*

Sprachen mit hohen Zeichennummern (Chinesisch) benötigen in UTF-8 mehr Speicherplatz als Sprachen mit tiefen Zeilennummern (Englisch).

Aufgabe 7

(a) *Speicherbedarf für ein Zeichen:*

UTF-8-codierte Zeichen haben einen variablen Speicherbedarf von 1–4 Byte

UTF-32-codierte Zeichen werden benötigen immer 4 Byte Speicherplatz.

(b) *Gleichbehandlung der Zeichen verschiedener Sprachen:*

Sprachen mit hohen Zeichennummern (Chinesisch) benötigen in UTF-8 mehr Speicherplatz als Sprachen mit tiefen Zeilennummern (Englisch).

Die UTF-32-Codierung behandelt alle Sprachen gleich, denn jedes Zeichen braucht 4 Byte Speicherplatz.

Aufgabe 7

(a) *Speicherbedarf für ein Zeichen:*

UTF-8-codierte Zeichen haben einen variablen Speicherbedarf von 1–4 Byte

UTF-32-codierte Zeichen werden benötigen immer 4 Byte Speicherplatz.

(b) *Gleichbehandlung der Zeichen verschiedener Sprachen:*

Sprachen mit hohen Zeichennummern (Chinesisch) benötigen in UTF-8 mehr Speicherplatz als Sprachen mit tiefen Zeilennummern (Englisch).

Die UTF-32-Codierung behandelt alle Sprachen gleich, denn jedes Zeichen braucht 4 Byte Speicherplatz.

(c) *Codierungsaufwand:*

Aufgabe 7

(a) *Speicherbedarf für ein Zeichen:*

UTF-8-codierte Zeichen haben einen variablen Speicherbedarf von 1–4 Byte

UTF-32-codierte Zeichen werden benötigen immer 4 Byte Speicherplatz.

(b) *Gleichbehandlung der Zeichen verschiedener Sprachen:*

Sprachen mit hohen Zeichennummern (Chinesisch) benötigen in UTF-8 mehr Speicherplatz als Sprachen mit tiefen Zeilennummern (Englisch).

Die UTF-32-Codierung behandelt alle Sprachen gleich, denn jedes Zeichen braucht 4 Byte Speicherplatz.

(c) *Codierungsaufwand:*

Der Codierungsaufwand für UTF-8 ist grösser als für UTF-32.

Aufgabe 8

Gib den ASCII-Text, der durch den folgenden Binärcode gegeben ist, möglichst exakt wieder.

00110101 00101011 00110011 00001010 00111101 00111000

Aufgabe 8

bin	hex	ASCII
00110101	0x35	5
00101011	0x2B	+
00110011	0x33	3
00001010	0x0A	<LF>
00111101	0x3D	=
00111000	0x38	8

⇒

$$\begin{array}{l} 5+3 \\ =8 \end{array}$$

Von den ASCII-Steuerzeichen müssen nur diejenigen „erkannt“ werden, die eine Zeilenschaltung codieren. Falls der Text auf einem Windows-Computer geschrieben wurde, sind dies <CR> und <LF> und auf Apple- und Unix-Systemen ist das <LF>. Diese Informationen stehen aber auf der ausgeteilten Tabelle.