

Standardisierte Codes - Lösungsvorschläge

Der ASCII-Code

Aufgabe 1:

- a) Beschreibe den Aufbau des ASCII-Codes.
- b) Vergleiche den Aufbau des ASCII-Codes mit euren eigenen Codetabellen. Welche Gemeinsamkeiten und welche Unterschiede stellt ihr fest?

Antwort: Der ASCII-Code ist ein binärer Code der festen Länge 7. Je nachdem, wie der Code der Schüler*innen aufgebaut ist, werden sie feststellen, dass die Zeichen 0 und 1 für die Codierung verwendet werden und die einzelnen Codes mit sieben Stellen ggf. länger sind als ihre eigenen. Vielleicht erkennen sie sogar eine Systematik, insbesondere wenn sie mit Binärzahlen vertraut sind.

Aufgabe 2:

- a) Berechne, wie viele Zeichen sich mit 7 Bit codieren lassen.

Antwort: Da jede Stelle im Code zwei verschiedene Werte annehmen kann, kann man mit sieben Stellen $2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^7 = 128$ Zeichen codieren.

- b) Eine vollständige Codetabelle des ASCII-Codes findest du in der Datei *ASCII-Code.pdf*. Schau in der Codetabelle nach, welche der Zeichen aus Tabelle 2 im ASCII-Code berücksichtigt wurden. Wenn du eine binäre Codierung findest, trage sie in die Tabelle 2 ein.

Zeichen	ASCII-Code	Zeichen	ASCII-Code	Zeichen	ASCII-Code
!	0010 0001	ß	-	@	0100 0000
Ä	-	€	-	\$	0010 0100

Tabelle 1: Für welche Zeichen gibt es einen ASCII-Code?

- c) Stelle eine Vermutung auf, warum der ASCII-Code für einige der Zeichen in Tabelle 2 keine Codierung vorsieht.

Antwort: Bei dem ASCII-Code handelt es sich um einen amerikanischen Standard. Ä und ß sind Sonderzeichen der deutschen Sprache, die es im Englischen nicht gibt. Daher wurden diese Zeichen im ursprünglichen ASCII-Code nicht berücksichtigt. Aus dem gleichen Grunde wurde vermutlich das Dollarzeichen \$ für die amerikanische Währung berücksichtigt. Die Währung Euro mit dem Symbol € wurde zudem erst lange nach der Festlegung des ASCII-Codes eingeführt.

Erweiterungen des ASCII-Codes

Aufgabe 3:

- a) Berechne wie viele zusätzliche Zeichen sich mit einem weiteren Bit codieren lassen.

Antwort: Mit einem weiteren Bit lassen sich doppelt so viele Zeichen codieren, also können zu den bisherigen 128 Zeichen noch weitere 128 Zeichen codiert werden.

- b) Welche Probleme könnten unter Verwendung der verschiedenen Erweiterungen des ASCII-Codes auftreten, wenn ein Text sowohl Umlaute der deutschen Sprache als auch kyrillische Schriftzeichen enthalten soll?

Antwort: Der gleiche achtstellige Code würde für zwei unterschiedliche Zeichen stehen. Es müsste somit irgendwie kenntlich gemacht werden, welcher Code gerade verwendet wird. Ist

nicht bekannt, welche Erweiterung des ASCII-Codes gemeint ist, wird ggf. beim Decodieren ein falsches Zeichen dargestellt.

Interpretation des ASCII-Codes als Zahlen

Aufgabe 4:

- a) Trage den ASCII-Code für die Großbuchstaben C, K und Z und für das Leerzeichen in eine binäre Stellenwerttabelle ein. Berechne anschließend welcher Zahl der Code im Dezimalsystem entspricht, wenn man ihn als binäre Zahl interpretiert.

Antwort:

Stellenwert	$64 = 2^6$	$32 = 2^5$	$16 = 2^4$	$8 = 2^3$	$4 = 2^2$	$2 = 2^1$	$1 = 2^0$	Dezimalzahl
C	1	0	0	0	0	1	1	67
K	1	0	0	1	0	1	1	75
Z	1	0	1	1	0	1	0	90
Leerzeichen	0	1	0	0	0	0	0	32

C: $64 + 2 + 1 = 67$

K: $64 + 8 + 2 + 1 = 75$

Z: $64 + 16 + 8 + 2 = 90$

Leerzeichen: 32

- b) Wie kann man den 7-Bit ASCII-Code zu einem 8-Bit Code erweitern, ohne den Wert der Zahl zu verändern?

Antwort: Es wird eine führend Null ergänzt. Die Null darf nicht als letzte Stelle ergänzt werden, da sich dann die Werte der Stellen, die mit 1 belegt sind, ändern würden.

- c) Auch im Unicode soll sich der Zahlenwert der binären Codierung der Zeichen, die bereits im ASCII-Code enthalten sind, nicht verändern. Gib an, wie der 16-Bit Unicode des Großbuchstabens A lauten muss.

Antwort: Auch hier müssen entsprechend viele Nullen vorangestellt werden:

0000 0000 000100 0001

Aufgabe 5:

- a) Bestimme mit einem der beiden Verfahren die Darstellung der Zahlen 99 und 50. Welche Zeichen codieren diese Zahlen gemäß des ASCII-Codes?

Antwort: Verfahren 1 für die Zahl 99:

$$99 - 64 = 35 \rightarrow 35 - 32 = 3 \rightarrow 3 - 2 = 1$$

$$99 = 64 + 32 + 2 + 1 \rightarrow (99)_{10} = (110\ 0011)_2$$

Die 99 codiert somit das kleine c.

Verfahren 2 für die Zahl 50:

$$50 : 2 = 25 \text{ R } 0$$

$$25 : 2 = 12 \text{ R } 1$$

$$12 : 2 = 6 \text{ R } 0$$

$$6 : 2 = 3 \text{ R } 0$$

$$3 : 2 = 1 \text{ R } 1$$

$$1 : 2 = 0 \text{ R } 1$$

$$(50)_{10} = (11\ 0010)_2 \text{ Die 50 codiert somit die Ziffer 2.}$$

Hier sollte kurz thematisiert werden, dass die Ziffern 0 bis 9 nicht von den Dezimalzahlen 0 bis 9 bzw. den entsprechenden Binärzahlen codiert werden.

- b) Justus behauptet, ich muss mir nur merken, dass der Großbuchstabe A im ASCII-Code mit der Zahl 65 und der Kleinbuchstabe a mit der Zahl 97 codiert wird, dann kann ich mir die binäre Codierung aller Groß- und Kleinbuchstaben herleiten. Nimm Stellung.

Antwort: Da die Buchstaben im ASCII-Code fortlaufen durchnummeriert sind, kann von der 65 bzw. der 97 einfach im Alphabet weitergezählt werden. Die entsprechende Dezimalzahl kann dann in eine Binärzahl umgewandelt werden. Daraus ergibt sich dann der binäre ASCII-Code. Es reicht nicht, sich nur die 65 für das große A zu merken, da zwischen den Groß- und den Kleinbuchstaben noch sechs andere Zeichen liegen.

Mit diesem Trick gelingt es wahrscheinlich vielen Schüler*innen, die Aufgabe von Jakob in der Sendung *Klein gegen Groß* vom 29.02.2020 zu meistern, wenn ausreichen Zeit zur Verfügung steht.

Lizenz

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International Lizenz](#). Sie erlaubt Download und Weiterverteilung des vollständigen Werkes unter Nennung meines Namens, jedoch keinerlei Bearbeitung oder kommerzielle Nutzung.