

1. Klausur 11/II (A)

Dauer: 3 Schulstunden

Name: www.r-krell.de

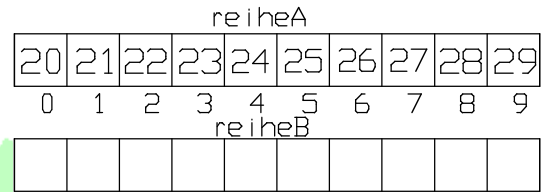
Hilfsmittel: normaler Taschenrechner

\* Achte auf sorgfältige Darstellung mit vollständigem, nachvollziehbarem Lösungsweg! \*

\* Kommentiere deine Programme! \*

1 Reihungen in Java

a) Die beiden Reihungen *reiheA* und *reiheB* sind jeweils für die Aufnahme von 10 ganzen Zahlen gedacht.



a1) Definiere und erzeuge beide Reihen

a2) Schreibe je ein kurzes Programmstück, das *reiheA* mit den Zahlen von 20 bis 29 füllt. Verwende dazu (1) *for* bzw. (2) *do..while*!

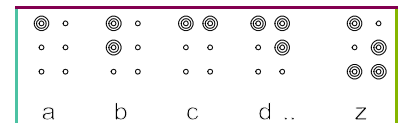
a3) Was macht `for (int i=0; i<10; i++) { reiheB[i] = reiheA[9-i]; }` ?

a4) Und was macht `for (int i=0; i<10; i++) { reiheA[9-i] = reiheA[i]; }` ?

b) Ein Parkhaus-Betreiber hat in einer Stadt 3 gleiche Parkhäuser (bekannt als Haus 0, Haus 1 und Haus 2) mit jeweils 5 Etagen (Stockwerknummern 0 bis 4) und jeweils 56 Parkplätzen pro Etage (Platznummern 0..55). An jedem Stellplatz registriert ein automatischer Sensor, ob der Platz frei ist oder nicht. Definiere eine geeignete Datenstruktur zur Aufnahme aller Informationen, nenne außerdem den einen Java-Befehl, um Platz 27 im 2. Stock des Hauses 1 auf frei zu setzen (der Sensor an diesem Platz spinnt leider) und schreibe schließlich noch eine Methode `public int freiPlätze()`, die die Gesamtzahl der freien Plätze (zusammen aus allen 3 Parkhäusern) zurück gibt.

2 Kodierungen

a) Bei der Blindenschrift bestehen alle Zeichen aus bis zu 6 erhabenen Punkten in einer 2x3-Matrix.



a1) Können außer den 26 Buchstaben *a* bis *z* auch die Umlaute *äöüß*, die Ziffern *0* bis *9* und die üblichen Satzzeichen `.,:?!'"/-+*=` dargestellt werden?

a2) In der Blindenschrift gibt es zwar einige Kurzschriftzeichen (z.B. ein Extra-Zeichen für *sch*), aber keine Großbuchstaben. Statt dessen zeigt ein vorangestelltes Großschriftzeichen an, dass der nächste Buchstabe eigentlich groß geschrieben werden müsste. Warum?

b) In Java belegt der String „Hi“ drei Byte. Notiere alle Bits!

c) Warum werden Dualzahlen im Computer platzverschwenderisch mit führenden Nullen gespeichert, d.h. 00000010 oder sogar 0000000000000010 statt einfach nur dual 10 für dezimal 2?

Ascii-Tabelle (Auszug)	1	1	1	1	1	1
	0	1	2	3	4	5
64	@	A	B	C	D	E
80	P	Q	R	S	T	U
96	'	a	b	c	d	e
		f	g	h	i	j
		k	l	m	n	o

(Beispiel:  $\wedge = 80 + 14 = 94$ )

d) Gib dezimal den Ganzzahlbereich an, der mit (1) 7 oder (2) *n* Bit dargestellt werden kann (Beisp.: 3 Bit für -4..+3)

e) Warum nimmt man in Kauf, dass eine negative Zahl mehr dargestellt werden kann als eine postive, d.h. bei 3-Bit-Zahlen -4..+3 statt z.B. -3..+4 ?

3 Rechnungen mit (Dual-)Zahlen

a) Gegeben sind die beiden Dualzahlen  $x = 29_{10} = 011101_2$  und  $y = -22_{10} = 101010_2$  (Zahlbereich -32 bis +31). Berechne schriftlich (dual auf 6 Bits, mit nachvollziehbarem Rechenweg, verwandle dann das Ergebnis in eine Dezimalzahl und schreibe, ob das Ergebnis richtig ist):

a1)  $x + y$

a2)  $x - y$

b) Berechne  $01110 * 01101$  dual und nenne das Ergebnis auch dezimal

c) Berechne  $0100111 : 01100$  mit bis zu 3 Dual-Nachkommastellen; führende Nullen können

weggelassen werden. Verwandle auch hier das Ergebnis in eine Dezimal(komma)zahl.

d) Warum reicht ein Addierwerk (zusammen mit einer Invertiermöglichkeit, d.h. parallelen NOT-Gattern) für die Hardware-Ausstattung eines Computers, der alle Zahlen-Rechnungen beherrscht?

4) Dezimalbrüche wie z.B. 0,6875 lassen sich nach folgendem Verfahren in Dualbrüche wie  $(0,1011)_2$  umwandeln: Starte mit  $a=0,6875$  und immer mit  $b=0,5$ . Wenn  $a \geq b$ , dann ziehe von  $a$  das  $b$  ab, speichere die Differenz in  $a$  und schreibe eine 1 hin, sonst schreibe eine 0 und ändere  $a$  nicht. Halbiere jetzt das  $b$  und prüfe wieder, ob  $a \geq b$  ist. Wenn  $a \geq b$ , dann ziehe wieder ... usw.

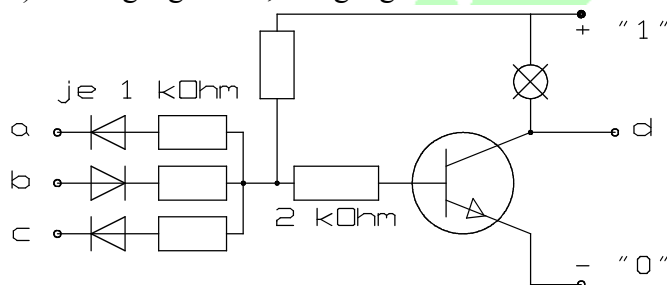
a) Schreibe eine vollständige Javamethode `public String dualbruch (double dezimalbruch)`, die einen Dezimalbruch zwischen 0 und 1 erhält und die (Nachkomma-)Bits der Dualdarstellung zurück gibt (Gelingt Java nicht, ist auch ein Struktogramm möglich).

b) Verwandle nach dem angegebenen Verfahren schriftlich/von Hand den Dezimalbruch 0,8 in einen Dualbruch. Brich nach 5 Nachkommstellen ab oder vorher, wenn eine Periode sichtbar wird!

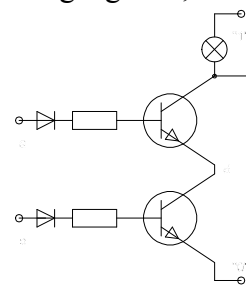
c) Versuche kurz zu erklären, warum das Verfahren wirklich die richtige Dualbruchdarstellung liefert.

5) Stelle die Wahrheitwerttabellen auf (und begründe jeweils die zweite Zeile mit zusätzlichem Text):

a) 3 Eingänge a b c, Ausgang d:



b) Eingänge a b; Ausgang c:



6) Die Wahrheitwerttabelle für die drei Eingänge a b c und einen Ausgang d enthält 8 Zeilen. Wenn die drei Bits der Eingänge als Dualdarstellung der Zahlen 0, 3, 4 und 5 aufgefasst werden können, hat der Ausgang d den Wert  $d=1$ , sonst ist  $d=0$ .

a) Notiere die Wahrheitwerttabelle und gib die DNF für d an.

b) Zeichne ein KV-Diagramm und vereinfache soweit möglich.

c) Zeichne die benötigte Hardware, um die vereinfachte Schaltung mit Logikgattern zu realisieren. Was/Wie viel wird gegenüber der nicht vereinfachten Schaltung (nach a)) gespart?

7) Analysiere die angegebene Schaltung (d.h. stelle eine Wahrheitwerttabelle für a, b, x, y auf!).

Kann das gleiche Ergebnis auch mit geringerem Hardware-Aufwand erreicht werden? Welche Schaltung ist schneller?

