Zweierkomplement

Die Zahl als Binärzahl darstellen, dabei eine genügend grosse Länge auswählen	$13_{10} \triangleq 00001101_2$
Das Vorzeichen wird gewechselt indem alle Bits umgekehrt (invertiert) werden	$13_{10} \triangleq 00001101_2$ $-13_{10} \triangleq 11110010_2$
Es wird zur Zahl 1 dazugerechnet	$-13_{10} \triangleq 11110011_2$

Spezielle Zahlen im 2er Komplement

-15 = 00001111 11110000 /invertieren 11110001 /+1	-127 = 01111111 10000000 /invertieren 10000001 /+1	0 = 00000000 11111111/invertieren 100000000 /+1
-128 = 10000000 01111111 /invertieren 10000000 /+1	-1 = 00000001 11111110 /invertieren 11111111 /+1	-2 = 00000010 11111101 /invertieren 11111110 /+1

Subtraktion 28-17 mit 5Bit

28=11100; 17=10001
K17 = 01110=>K17+1=>01111
(28) 11100
(2K17) + 01111
101011->Stellenüberlauf ignorierrt

Exzess mit 8 Bit

Dezimal	0	1	2	64	127	128	129	192	254	255
Binär	00000000	00000001	00000010	01000000	01111111	10000000	10000001	11000000	11111110	11111111
Einer K.	0	1	2	64	127	-127	-126	-63	-1	-0
Zweier K.	0	1	2	64	127	-128	-127	-64	-2	-1
Exzess	-128	-127	-126	-64	-1	0	1	64	126	127

Um von Exc. in Dec zu kommen muss +128 gerechnet werden

Hammingcode

Originalnachricht: 10110010

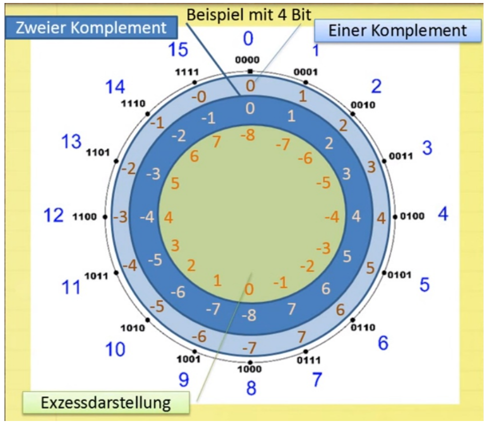
Füllen Sie diese Nachricht in das Raster ab, so dass das Raster versendet werden kann.

Nr.	12	11	10	9	8	7	6	5	4	3	2	1
Data	1	0	1	1	1	0	0	1	0	0	1	0

Stellenweises Addieren aller „1" Positionen ohne Übertrag:
Reihe 12 = 1100
Reihe 10 = 1010
Reihe 9 = 1001
Reihe 5 = 0101
Total = 1010

Empfänger: Addiert alle „1" Positionen

Reihe 12 = 1100
Reihe 10 = 1010
Reihe 9 = 1001
Reihe 8 = 1000
Reihe 5 = 0101
Reihe 2 = 0010
Total = 0000 Bei 0 Fehlerfrei sonst entspricht die Zahl der Falschen Stelle



Ziffer	BCD Code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
(leer)	1010
(leer)	1011
(leer)	1100
(leer)	1101
(leer)	1110
(leer)	1111

Fliesskommazahlen

Die Zahl 10.625 und – 10.625 im IEEE umwandeln

1) 10 umwandeln in Binär =>1010_b

2) 0.625 umwandeln in Binär =>1010_b

3) Resultat = 1010.1010

4) Zur Normalisierung muss das Komma in dieser Zahl nun um 3 Stellen nach links verschoben werden; Das entspricht einer Multiplikation mit 2^3

5) Resultat = 1.0101010 * 2^3

6)Mantisse = (1)0101010 Exponent 3_d=11_b

7) Bias auf den Exponenten addieren 127+3=130_d =>10000010_b

8) 10.625 => 0 10000010 (1)0101010 (Mantisse mit Hidden Bit in Klammer)

9) -10.625 => 1 10000010 (1)0101010 (Mantisse mit Hidden Bit in Klammer)

Das Prinzip der Biased-Darstellung für negative Zahlen:

Binär	Exponent	Wert	Berechnung Bias+Exp
00000000	-127	0	127-127
00000001	-126	1	127-126
00000010	-125	2	127-125
:	:		
01111101	-2	125	127-2
01111110	-1	126	127-1
01111111	0	127	127-0
10000000	1	128	127+1
10000001	2	129	127+2
10000010	3	130	127+3
:	:		
11111101	126	253	127+126
11111110	127	254	127+127
11111111	128	255	127+128

Exzessdarstellung

```
      + 65  =  0100 0001
EK    - 65  =  1011 1110
           +           1
           -------------
ZK    - 65  =  1011 1111

EX    - 65  =  0011 1111
```

IEEE 754 Gleitkommazahl einfache Genauigkeit (In Java elementarer Datentyp float / 32 Bit)

Vorzeichen	Exponent in Excess-Darstellung (Vorzeichen)	Mantisse (hidden bit)
0	1 0 0 0 0 0 1 0	1. 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
		$\frac{1}{2^0}$ $\frac{1}{2^1}$ $\frac{1}{2^2}$ $\frac{1}{2^3}$ $\frac{1}{2^4}$ $\frac{1}{2^5}$ $\frac{1}{2^6}$ $\frac{1}{2^7}$ $\frac{1}{2^8}$ $\frac{1}{2^9}$ … $\frac{1}{2^{23}}$
+	3	1.32812500000000000000000000000

$+\ 1.328125 \cdot 2^3 = 10.625000000$

Mantisse * Basis $^{\text{Exponent}}$

Umrechnung Kommazahl									
Wertigkeit	**8**	**4**	**2**	**1**		**1/2**	**1/4**	**1/8**	**1/16**
Bittkette	1	0	1	0	,	1	0	1	0
	8	+	2			0.5	+	0.125	
Dezimalwert	10				,	625			

	1 0 0 1	1 0 0 0	= 9 8
	0 0 1 0	0 0 1 1	= 2 3
	1 0 1 1	1 0 1 1	**Pseudo tetrade**
		0 1 1 0	Addition von 6
	1 1 1	1 1	
	1 1 0 0	0 0 0 1	**Pseudo tetrade**
	0 1 1 0		Addition von 6
1	1		
0 0 0 1	0 0 1 0	0 0 0 1	
1	2	1	