

```

1  Zweierkomplement von Zahlen
2      1 = 00000001
3      -1 = 11111111
4
5  Vorgehen:
6  1. Alle Bit drehen
7      00000001
8  wird  11111110
9
10 2. 00000001 dazu addieren
11      11111110
12 +   00000001
13 =   11111111
14
15 Weitere Beispiele
16
17      1 00000001
18      11111110 + 00000001
19      -1 11111111
20
21      7 00000111
22      11111000 + 00000001
23      -7 11111001
24
25      6 00000110
26      11111001 + 00000001
27      -6 11111010
28
29      15 00001111
30      11110000 + 00000001
31      -15 11110001
32
33      16 00010000
34      11101111 + 00000001
35      -16 11110000
36
37      57 00111001
38      11000110 + 00000001
39      -57 11000111
40
41      126 01111110
42      10000001 + 00000001
43      -126 10000010
44
45      127 01111111
46      10000000 + 00000001
47      -127 10000001
48
49 Es geht auch umgekehrt:
50      -43 11010101
51      00101010 + 00000001
52      43 00101011
53
54 -----
55 Computer berechnen Zahlen binär.
56 Sie können dabei Bits rasch addieren und Bytes eine Stelle nach links oder
57 rechts schieben.
58 Das hilft ungemein, wenn der Computer die Grundrechenoperationen
59 durchführen muss:
60
61 Addition von binären Zahlen:
62 Bits Spalte um Spalte zusammenzählen und Uebertrag berücksichtigen:
63
64      1 00000001
65 + 3 00000011
66
67      10 Resultat Addition von Bit 0
68      10 Resultat Addition von Bit 1

```

```

66     = 4  00000100  (immer letztes Bit aus den Resultaten notieren)
67
68 Subtraktion mit einfachen Zahlen:
69     6  00000110
70     - 1  00000001
71           -1  Uebertrag aus Subtraktion von Bit 0
72     = 5  00000101
73
74 Wenn es ein negatives Resultat ergibt:
75     3  00000011
76     - 5  00000101
77           -1  Uebertrag von Bit 2
78           -1  Uebertrag von Bit 3
79           -1  Uebertrag von Bit 4
80           -1  Uebertrag von Bit 5
81           -1  Uebertrag von Bit 6
82     = -2  11111110
83
84 Subtraktion einer positiven Zahl als Addition des Zweierkomplements:
85 3-1 = 3+(Zweierkomplement von 1)
86
87     3  00000011
88   + -1  11111111
89     10000010
90
91 das 9te Bit (Bit ganz links) wird dabei ignoriert:
92     = 2  00000010
93
94 Subtraktion von negativen Zahl mittels Addition des Zweierkomplements:
95 3- (-1) = 4
96 3+ (Zweierkomplement von -1) = 4
97
98     3  00000011
99   + 1  00000001
100     = 4  00000100
101
102 Addition mit mehreren Zahlen und etwas schwierigerem Uebertrag:
103 3+3+3+3+3 = 18
104     00000011
105   + 00000011
106   + 00000011
107   + 00000011
108   + 00000011
109   + 00000011
110   +      110  Resultat von Addition Bit 0 ->6
111   +      111  Resultat von Addition Bit 1 ->7
112   +      10   Resultat von Addition Bit 2 ->2
113     10        Resultat von Addition Bit 3 ->2
114   00010010  (immer letztes Bit aus den Resultaten notieren)
115
116 ---
117 Multiplikation von binären Zahlen:
118 Was passiert wenn man mit einer Zweierpotenz multipliziert?
119     9  00001001
120   9*2 = 18 00010010
121   9*4 = 36 00100100
122
123 ==> Multiplikation mit 2: 1x alle Bits nach links schieben.
124 ==> Multiplikation mit 4: 2x alle Bits nach links schieben.
125
126 Betrachten wir eine etwas komplexere Rechnung:
127 9*13=117
128 Weg: Wir zerlegen den Faktors 13 in Additionen von 2er Potenzen.
129 Denn dann können wir diese 2er Potenzen für das links Schieben verwenden.
130 13= 8+4+1 = 2^3+2^2+2^0
131
132 somit:

```

```

133 9*13= 9*(8+4+1)= 9*8+9*4+1*9 =9*2^3+9*2^2+9*2^0
134                                     ^      ^      ^
135                                     3      2      0  2er Potenzen
136

```

```

137 9 =      00001001
138
139 + 00001001xxx  9 ->3 mal nach links schieben
140 +  00001001xx  9 ->2 mal nach links schieben
141 +   00001001   9 ->0 mal nach links schieben
142 = 00001110101
143 =   01110101
144 =      117
145

```

Ein zweites Beispiel:

```

147 7*11=77
148

```

```

149 7  = 00000111
150 11 = 00001011
151      ^  ^
152      3 10 2er Potenzen
153

```

```

154      00000111xxx  7 ->3 mal nach links schieben
155 +   00000111x   7 ->1 mal nach links schieben
156 +    00000111   7 ->0 mal nach links schieben
157 = 00001001001
158 =   01001101
159 =      77
160

```

Mit negativen Zahlen:

```

162 -7*-11=77
163

```

```

164 -7 = 11111001
165 -11 = 11110101
166      ^^^^ ^ ^
167      7654 2 0 2er Potenzen
168

```

```

169      11111001xxxxxxx  -7 -> 7 mal nach links
170 +   11111001xxxxxxx  -7 -> 6 mal nach links
171 +   11111001xxxxxxx  -7 -> 5 mal nach links
172 +   11111001xxxxxxx  -7 -> 4 mal nach links
173 +   11111001xxxxxxx  -7 -> 3 mal nach links
174 +   11111001xxxxxxx  -7 -> 2 mal nach links
175 +   11111001xxxxxxx  -7 -> 1 mal nach links
176 +   11111001xxxxxxx  -7 -> 0 mal nach links
177 +   11111001xxxxxxx  -7 -> 0 mal nach links
178 +   11111001xxxxxxx  -7 -> 0 mal nach links
179 +   11111001xxxxxxx  -7 -> 0 mal nach links
180 +   11111001xxxxxxx  -7 -> 0 mal nach links
181 +   11111001xxxxxxx  -7 -> 0 mal nach links
182 +   11111001xxxxxxx  -7 -> 0 mal nach links
183 = xxxxxxx01001101
184 =   01001101
185 =      77
186

```

---

Und nun die Division von binären Zahlen:

Beispiel:

```

190 16/4=4
191

```

```

192 4= 2^2
193 Ergo:statt nach links müssen wir zweimal nach rechts schieben.
194 16= 00010000
195    xx000100  2 mal nach rechts
196 = 00000100
197 =      4  korrekt :-)
198

```

Doch leider ist es nicht ganz so einfach.

```

200
201 107 / 7 = 15 Rest 2
202 107 / (4+2+1) = 15 Rest 2
203
204 Es wäre falsch jetzt "auszuklammern": 107/4+107/2+107/1 FALSCH!!
205 Denn den Nenner kann man nicht einfach "ausklammern".
206
207 Anderer Ansatz mit schriftlicher Division:
208 107 / 7 = 015 Rest 2
209 -0
210 --
211 10
212 - 7
213 --
214 37
215 35
216 --
217 2
218
219 Binär geht das genau gleich:
220 107 = 01101011
221 7 = 00000111
222
223 01101011 / 00000111 = 001111 Rest 010 = 15 Rest 2
224 -000
225 ---
226 110
227 -000
228 ---
229 1101
230 - 111
231 ---
232 1100
233 - 111
234 ---
235 1011
236 - 111
237 ---
238 1001
239 - 111
240 ---
241 010
242
243 Das heisst, binäre Division ist sehr aufwendig.
244
245 Und nun mit negativen Zahlen:
246 -107 / -7 = 15 Rest -2
247
248 Bei negativen Zahlen wird es noch schwieriger.
249 Man muss das Vorzeichen des Resultates voraus bestimmen.
250 Sodann die Rechnung mit positiven Zahlen machen
251 Sodann die Vorzeichen des Resultates korrigieren.
252
253 Wenn beide Zahlen negativ sind -> Resultat positiv, Rest negativ
254 Nun rechnen wir mit den positiven Zahlen:
255 107 = 01101011
256 7 = 00000111
257
258 01101011 / 00000111 = 001111 Rest 010 = 15 Rest 2
259
260 Und nun korrigieren wir das Resultat um die zuvor gemerkten Vorzeichen:
261 ==> 15 Rest -2
262
263 Das heisst für uns, binäre Division mit neagtiven Zahlen ist noch
aufwendiger.
264

```