Aufgabe 5: Bit-Operatoren

Es gibt verschiedene Bit-Operatoren in C. Diese wirken nicht auf den ganzen Wert einer Variablen, sonderen auf die einzelnen Bits. Es wird also z. B. beim Und-Operator (&) Bit1 der ersten Variable und Bit1 der zweiten Variablen "verUNDet".

```
wert1 = 10010010
wert2 = 01110110

      10010010
&     01110110
-------------
=    00010010
```

Hier die Wahrheitstabellen für die verschiedenen Bit-Operatoren (a = Bit aus erster Variabel, b = Bit aus zweiter Variable, e = Ergebnisbit):

AND (&) Syntax: `ergebnis = wert1 & wert2;`

a	b	e
0	0	0
0	1	0
1	0	0
1	1	1

OR (|) Syntax: `ergebnis = wert1 | wert2;`

a	b	e
0	0	0
0	1	1
1	0	1
1	1	1

XOR (~) Syntax: `ergebnis = wert1 ^ wert2;`

a	b	e
0	0	0
0	1	1
1	0	1
1	1	0

NOT (~) Syntax: `ergebnis = ~wert1;`

a	e
0	1
1	0

Das XOR (Exklusiv-Oder) hat eine besondere Eigenschaft:

```
wert1       = 01011010
schluessel  = 01100110

      01011010
^     01100110
--------------
=     00111100
```

Jetzt hat man den Wert nach dem XOR `00111100`. Wenn man nun wiederum mit dem erzeugten Wert und dem `schluessel` ein XOR durchführt kommt man wieder zu `wert1`.

```
      00111100
^     01100110
-------------
=     01011010
```

Diese XOR-Verschlüsselung ist ein relativ einfaches Verschlüsselungsverfahren.

Aufgabe:

Schreibe ein Programm:

1. Das eine Datei als Byte-Stream einliest.

2. Die Bytes mit XOR und einem von dir gewähltem Schlüssel verschlüsselt bzw. entschlüsselt.

3. Speichere diese Daten in einer neuen Datei.