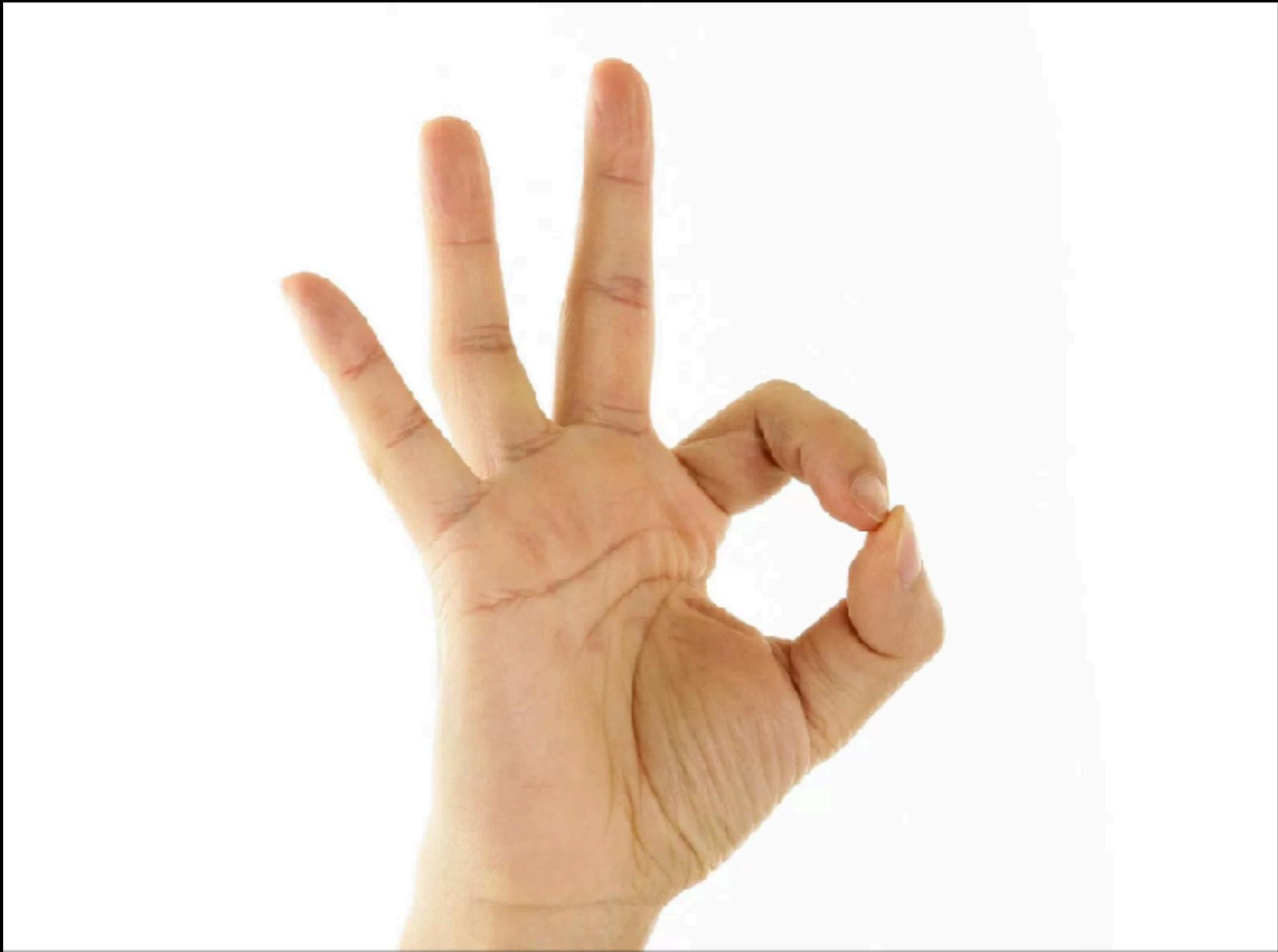
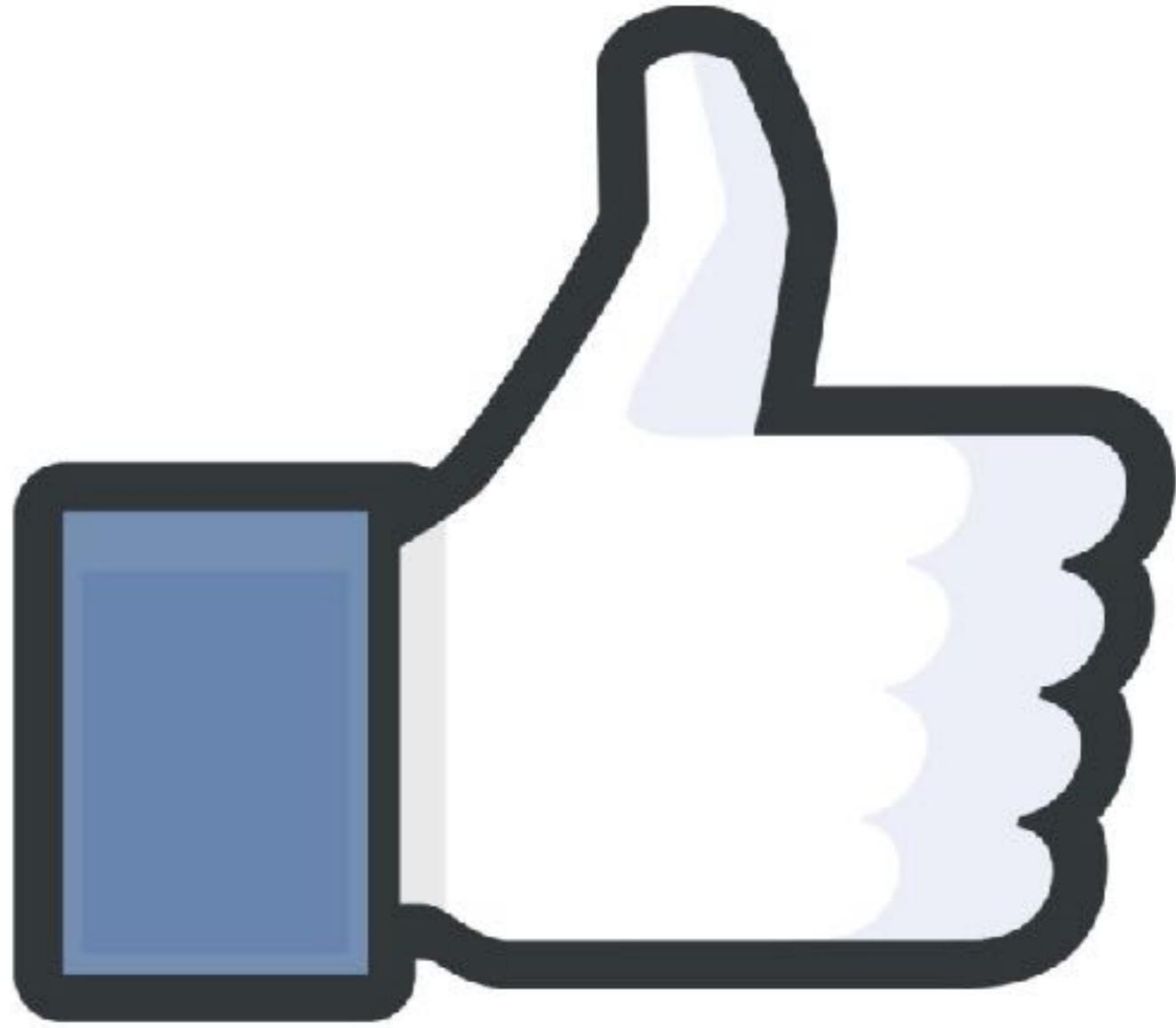


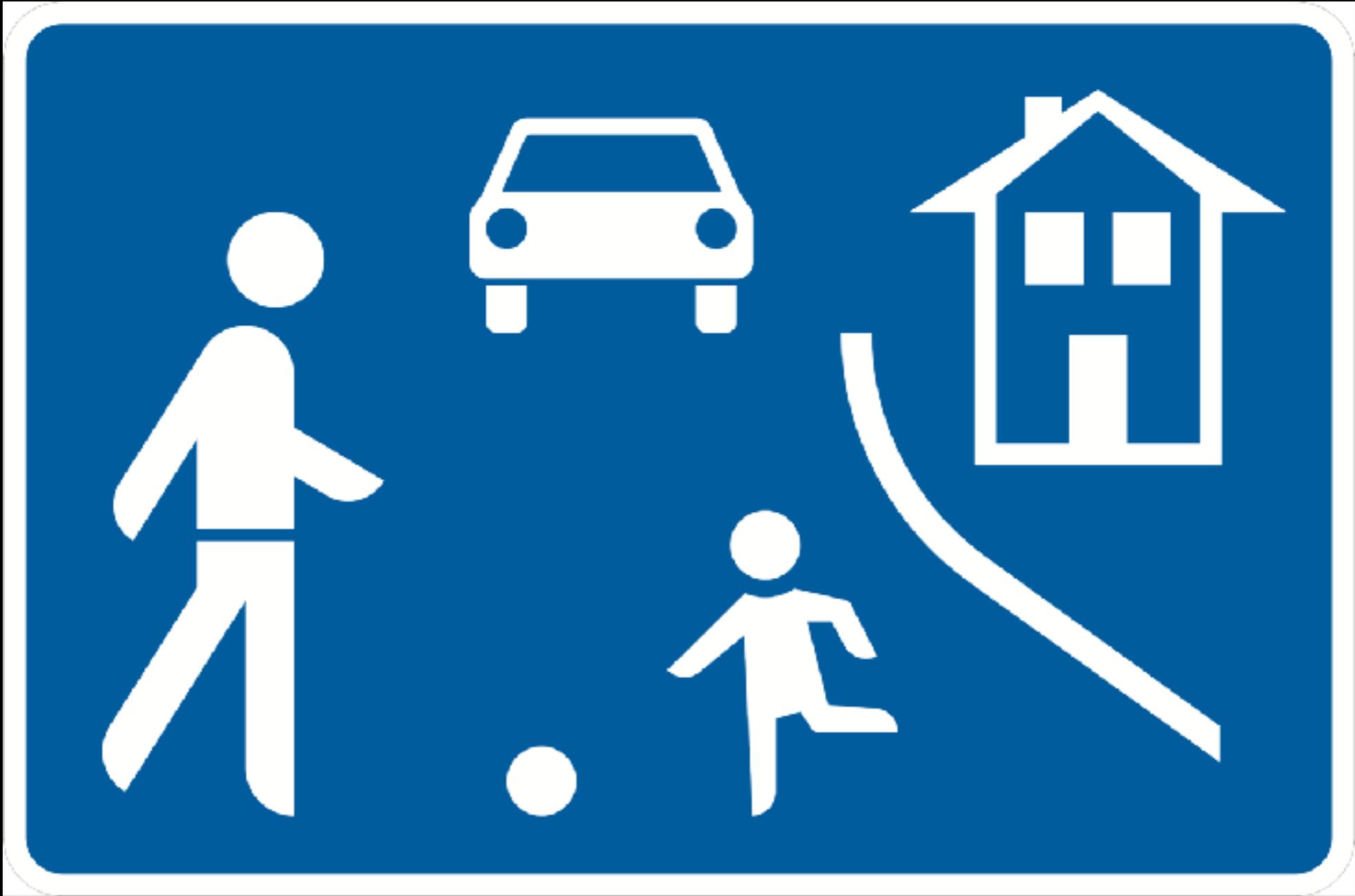
Informatik-AG Klasse 10
Schuljahr 18/19

25.9.18











MMXVIII

25.09.2018

9/25/2018

2018-09-25

1. Information

Informationen erhält man aus Wahrnehmungen, also von dem was wir sehen, hören, riechen, schmecken und fühlen. Wann erhält man eine Information?

Beispiel: Du fragst einen Freund, wie das Wetter morgen wird. Du hoffst auf eine Antwort wie: „Es wird schön!“ oder „Es wird regnerisch!“ Stattdessen erhältst du aber die Antwort: „39“.

Die Antwort ist für dich eigentlich keine Information.

Hättest du gefragt: „Wie alt ist deine Mutti?“, wäre „39“ korrekt, denn hier enthält die Antwort die gewünschte Information.

Der Informationsgehalt ist also stets von der Frage abhängig.

2. Daten

Nachrichten sind Angaben über einen Sachverhalt. Sie sind für uns stets erkennbar (in Wörtern, Bildern, Zahlen). Nachrichten enthalten Informationen und Informationen werden im Computer durch Daten dargestellt.

Daten sind demnach die computergerechte Form von Nachrichten. Daten sind also Träger von Informationen.

In der Elektronischen Datenverarbeitung (EDV) wird alles als Daten bezeichnet, was man für den Computer erkennbar speichern und darstellen kann.

3. Codierung

Wie werden Daten dargestellt?

Zwischen uns Menschen werden Informationen oft durch Zeichen übertragen (Schreiben, Lesen). Dazu wird das Alphabet benötigt, da alle Wörter aus Buchstaben zusammengesetzt werden.

Es gibt auch andere Möglichkeiten zur Informationsübertragung. Zum Beispiel die Farben Rot, Gelb und Grün bei der Verkehrsampel enthalten eine bestimmte Information.

Der Computer kann unsere Zeichenfolgen nicht lesen. Wir müssen unsere Sprache in eine Sprache übersetzen, die der Computer versteht. Dieses Übersetzen nennt man Codierung.

Der Code ist dabei der Schlüssel für die Lesbarkeit (Übersetzungsschlüssel). Und sowie die Quelle als auch der Empfänger müssen im Besitz dieses Schlüssels sein, damit die übermittelten Daten gelesen werden können.

4. Aufgabe

Überlege dir, welche Informationen es in deinem Alltag gibt und wie diese codiert werden.

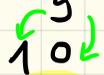
- Kopfnicken / - schütteln
- Uhr
- rot $\hat{=}$ warm, blau $\hat{=}$ kalt



- R 1 3 5
2 4
-
- Verkehrsschilder
- ✓ X

Dezimalsystem

0
1
2
3
4
5
6
7
8
9
10
11
⋮
19
20



Pentalsystem

0
1
2
3
4
10
11
12
13
14
20

Binärsystem

0
1
10
11
100
101
110
111
1000

$$11_2 = 3_{10}$$

Dezimalsystem

Das Dezimalsystem – oftmals auch Zehnersystem genannt – ist ein Stellenwertsystem zur Darstellung von Zahlen. Es verwendet die Basis 10. Das Dezimalsystem ist heute das weltweit verbreitetste Zahlensystem. Vermutlich hat das Dezimalsystem seinen Ursprung dem Umstand zu verdanken, dass der Mensch zehn Finger hat, welche man zum Zählen einsetzen kann. Im Zehnersystem kommen 10 Ziffern zum Einsatz: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Hat man von 0 bis 9 gezählt und möchte dies fortsetzen, dann beginnt man die folgenden Zahlen zusammen zu setzen. So folgt nach der 9 dann die 10, die 11, die 12 usw.

Haben wir beim Zählen beispielsweise an der Einerstelle schon die höchste Ziffer erreicht, so erhöhen wir die nächstgrößere Zehnerstelle und beginnen bei der Einerstelle wieder bei Null.

Wollen wir beispielsweise von der 99 auf die nächstgrößere Zahl, so sehen wir, dass an der Einerstelle bereits die höchste Ziffer steht, somit beginnen wir an dieser Stelle wieder bei 0 und erhöhen dafür die Zehnerstelle. Da an der Zehnerstelle jedoch auch bereits eine 9 steht, so beginnen wir auch hier wieder bei der 0 und erhöhen dafür die Hunderterstelle von 0 auf 1. Damit erhalten wir als Ergebnis die 100.

Binärsystem

Ein Computer rechnet mit Strom. hierbei kann er jedoch nur zwischen *Strom an* und *Strom aus* entscheiden bzw. zwischen 1 und 0. Ein Computer kann also lediglich mit 2 Ziffern anstatt mit 10 Ziffern rechnen!

Die Zählweise funktioniert aber ansonsten genau gleich wie im Dezimalsystem!

Dezimalsystem	Binärsystem
0	0 ₂
1	1 ₂
2	10 ₂
3	11 ₂
4	100 ₂
5	101 ₂

← im Binärsystem ist das bereits die höchste Ziffer...
 ← ... deshalb bekommt die Zahl bereits hier eine zweite Stelle!

Hinweis: Um Verwechslungen zu vermeiden schreibt man Zahlen im Binärsystem mit einer kleinen angehängten 2.

1. Aufgabe

Vervollständige die Tabelle:

Dezimalsystem	Binärsystem
0	0 ₂
1	1 ₂
2	10 ₂
3	11 ₂
4	100 ₂
5	101 ₂
6	110
7	111

Dezimalsystem	Binärsystem
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000

2. Aufgabe

Überlegt euch ein Rechenverfahren um auch ohne die obige Tabelle eine Zahl aus dem Binärsystem in das Dezimalsystem umzurechnen. Beschreibt euer Verfahren und testet es mit folgenden Zahlen: $10_2 = 2$, 101_2 , 1101_2 , 10101_2 , 100001_2 und 101111_2

$10111_2 = 23_{10}$	$101_2 = 5$
	$1101_2 = 13$
	$10101_2 = 21$
	$100001_2 = 33$
	$101111_2 = 47$

3. Aufgabe

Überlegt euch ein Rechenverfahren für den „Rückweg“ vom Dezimal- in das Binärsystem. Beschreibt euer Verfahren und testet es mit folgenden Zahlen: 17, 24, 28, 157 und 332

$17_{10} = 10001_2$	$24 = 11000_2$	
$24 = 1 \cdot 16 + 8$	$28 = 11100_2$	
$8 = 1 \cdot 8 + 0$	$157 = 10011101_2$	
$0 = 0 \cdot 4 + 0$	$332 = 101001100_2$	
$0 = 0 \cdot 2 + 0$		
$0 = 0 \cdot 1 + 0$		
		$157 = 1 \cdot 128 + 29$ $29 = 0 \cdot 64 + 29$ $29 = 0 \cdot 32 + 29$ $29 = 1 \cdot 16 + 13$ $13 = 1 \cdot 8 + 5$ $5 = 1 \cdot 4 + 1$ $1 = 0 \cdot 2 + 1$ $1 = 1 \cdot 1 + 0$

4. Aufgabe

Rechne in das jeweils andere Stellenwertsystem um.

- a) $11011110_2 = 222$ c) $11111101_2 = 253$ e) $13_{10} = 1101_2$ g) $254_{10} = 11111110_2$
 b) $00111111_2 = 63$ d) $01011010_2 = 90$ f) $96_{10} = 1100000_2$ h) $127_{10} = 1111111_2$

5. Aufgabe

Jede Stelle im Binärsystem wird durch ein Bit repräsentiert. Wie viele Bits benötigt man um folgende Zahlen darstellen zu können? Rechne die Zahlen ins Binärsystem um und versuche, einen Gesetzmäßigkeit für die Bitlänge zu finden.

- a) 1 → 1 Bit c) 8 → 4 Bits e) 32 → 6 Bits g) 260 → 9 Bits
 b) 7 → 3 Bits d) 31 → 5 Bits f) 120 → 7 Bits h) 4703 → 13 Bits

Welche Zahlen können mit 8 Bit = 1 Byte, 2 Byte, 4 Byte und 8 Byte dargestellt werden?

- 8 Bit → von 0 bis $255 = 2^8 - 1$
 16 Bit → von 0 bis $65535 = 2^{16} - 1$
 32 Bit → von 0 bis ~4 Mrd = $2^{32} - 1$
 64 Bit → von 0 bis ~18 Trl. = $2^{64} - 1$

	1	→	1 Bit
(2	→	2 Bit
(4	→	3 Bit
(8	→	4 Bit
	16	→	5 Bit
	32	→	6 Bit
	64	→	7
	128	→	8 Bit
	256	→	9 Bit

2^8 = 2^{n-1}

Dezimalsystem

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
:

Binärsystem

0
1
10
11
100
101
110
111
1000

Hexadezimalsystem

0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
10
11
:

Binärsystem: $1 \overset{16}{0} \overset{8}{0} \overset{4}{0} \overset{2}{1} \overset{1}{0} = 1 \cdot 16 + 1 \cdot 2 = 18$

Hexadezimalsystem: $4 \overset{16}{3} = 4 \cdot 16 + 3 \cdot 1 = 67$

$B \overset{16}{C} = 11 \cdot 16 + 12 \cdot 1 = 188$

Hexadezimalsystem

Im Hexadezimalsystem gibt es 16 Ziffern. Da wir in unserem „normalen“ Dezimalsystem jedoch nur 10 Ziffern kennen müssen wir weitere 6 „erfinden“. Hierfür nehmen wir Buchstaben:

- $A \hat{=} 10$
- $B \hat{=} 11$
- $C \hat{=} 12$
- $D \hat{=} 13$
- $E \hat{=} 14$
- $F \hat{=} 15$

Die Stellenwerte sind dabei:

$16^3 = 4096$	$16^2 = 256$	$16^1 = 16$	$16^0 = 1$
---------------	--------------	-------------	------------

Die Hexadezimale Zahl $3AC_{16}$ entspricht also der dezimalen Zahl $3 \cdot 256 + \underbrace{10}_{A} \cdot 16 + \underbrace{12}_{C} = 940$

1. Aufgabe

Rechne die folgenden hexadezimalen Zahlen ins Dezimalsystem um:

- | | |
|--|---|
| a) $C1_{16} = 12 \cdot 16 + 1 \cdot 1 = 193$ | e) $123_{16} = 1 \cdot 256 + 2 \cdot 16 + 3 \cdot 1 = 291$ |
| b) $F7_{16} = 15 \cdot 16 + 7 \cdot 1 = 247$ | f) $5AB_{16} = 5 \cdot 256 + 10 \cdot 16 + 11 \cdot 1 = 1451$ |
| c) $AAB_{16} = 10 \cdot 256 + 10 \cdot 16 + 11 \cdot 1 = 2731$ | g) $73A1_{16} = 7 \cdot 4096 + 3 \cdot 256 + 10 \cdot 16 + 1 \cdot 1 = 29601$ |
| d) $1FC_{16} = 1 \cdot 256 + 15 \cdot 16 + 12 \cdot 1 = 508$ | h) $13AF_{16} = 1 \cdot 4096 + 3 \cdot 256 + 10 \cdot 16 + 15 \cdot 1 = 5039$ |

2. Aufgabe

Rechne die folgenden dezimalen Zahlen ins Hexadezimalsystem um:

- | | |
|------------------|----------------------|
| a) 14 E | e) 532 214 |
| b) 27 1B | f) 1024 400 |
| c) 63 3F | g) 52012 CB2C |
| d) 139 8B | h) 65535 FFFF |

$$43 = 1 \cdot 32 + 11$$

$$11 = 0 \cdot 16 + 11$$

$$11 = 1 \cdot 8 + 3$$

$$3 = 0 \cdot 4 + 3$$

$$3 = 1 \cdot 2 + 1$$

$$1 = 1 \cdot 1 + 0$$

$$43 = 2 \cdot 16 + 11$$

$$11 = 11 \cdot 1 + 0$$

$$43_{10} = 2 B_{16}$$

$$43_{10} = 101011_2$$

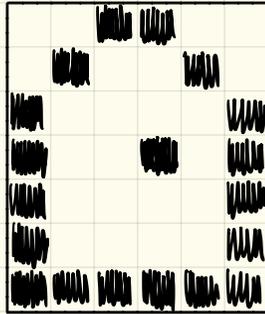
8 4 2 1 8 4 2 1 8 4 2 1
1 0 0 1 0 1 0 1 1 1 0 1

9 5 ①

2 C F

1 0 | 1 1 0 0 | 1 1 1 1

$$6 = 00000110$$



$$7 = 00000111$$

Breite

Höhe

00000110 | 00000111

001100010010100001100101100001

100001111111000000

Codierung: Bilder

1. Schwarz-Weiß-Pixelbilder

Einfache schwarz-weiß-Pixelbilder lassen sich sehr einfach darstellen. Hierbei muss lediglich angegeben werden, welche Pixel eingeschaltet (weiß) bzw. ausgeschaltet (schwarz) sind. Ein Bild kann so als lange Bit-Kette geschrieben werden.

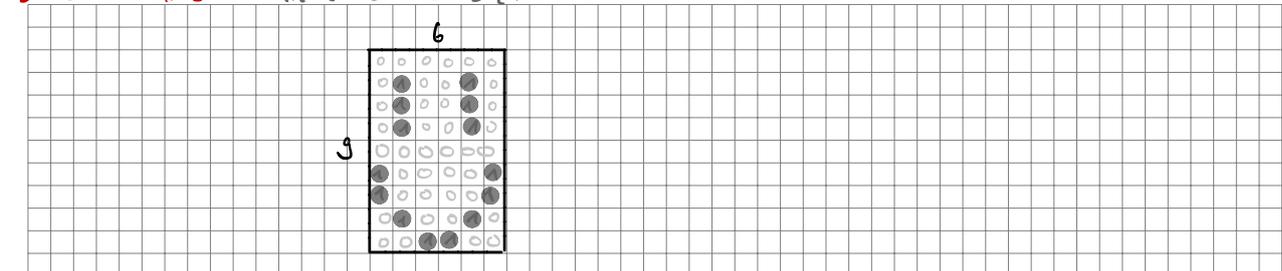
Damit der Computer jedoch weiß, wann eine neue Zeile beginnt, muss zunächst die Breite des Bildes angegeben werden. Damit ein Bild richtig abgespeichert werden kann und auch wieder richtig gelesen werden kann müssen wir uns ein eigenes Dateiformat definieren:

- Im ersten Byte der „Datei“ steht die Breite des Bildes
- Im zweiten Byte steht die Höhe des Bildes
- Anschließend folgen die Bits, die angeben, ob ein Pixel an oder aus ist.
- Ist die Länge der Bitkette kein Vielfaches von 8, so werden die restlichen Bits mit Nullen aufgefüllt.

2. Aufgabe

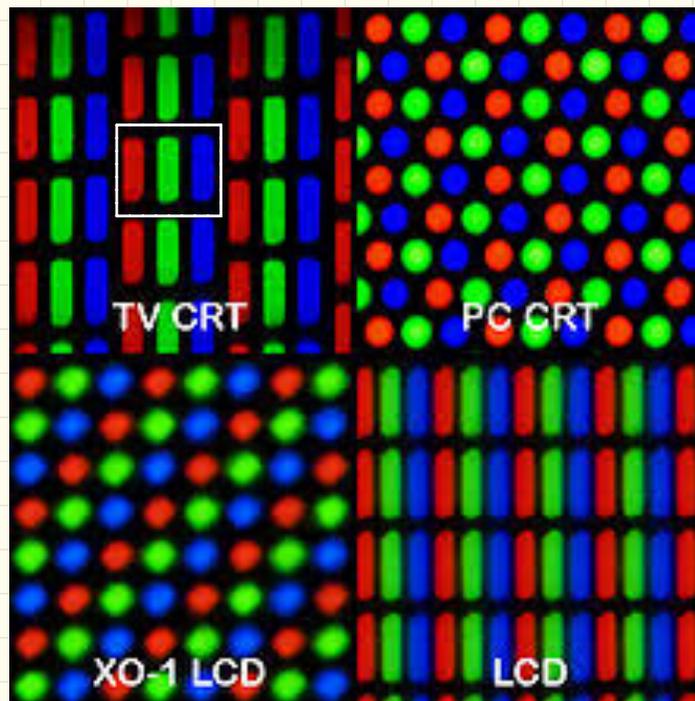
Entschlüsse mit obigen Angaben folgendes „Bild“:

00000110 00001001 00000001 00100100 10010010 00000010 00011000 01010010 00110000



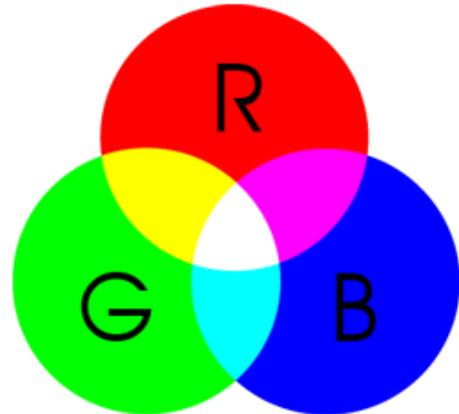
3. Aufgabe

Erstelle selbst ein Bild, codiere es in eine Bitfolge und gib es deinem Nachbarn zur Entschlüsselung.



4. Farbige Bilder

Auf Dauer werden Schwarz-Weiß-Bilder recht langweilig und es muss etwas Farbe ins Spiel kommen. Hier nutzen wir das RGB-System: Jeder Pixel besteht dabei aus 3 Farben rot, grün und blau und können folgendermaßen gemischt werden:



- Rot
- Grün
- Blau
- Rot + Grün = Gelb
- Rot + Blau = Magenta
- Blau + Grün = Cyan
- Rot + Grün + Blau = Weiß

Damit definieren wir ein neues Dateiformat:

- Im ersten Byte der „Datei“ steht die Breite des Bildes
- Im zweiten Byte steht die Höhe des Bildes
- Anschließend folgen die Pixel, dabei gehören immer 3 Bits zu einem Pixel:
 - das erste Bit gibt an, ob der rote Anteil dabei angeschaltet ist
 - das zweite Bit gibt den grünen Anteil an
 - das dritte Bit gibt den blauen Anteil an
 - z. B., eine Bitfolge 000 bedeutet schwarz, 001 bedeutet blau, 110 bedeutet gelb, 111 bedeutet weiß.
- Ist die Länge der Bitkette kein Vielfaches von 8, so werden die restlichen Bits mit Nullen aufgefüllt.

5. Aufgabe

6 Breit 6 Hoch

Entschlüsse mit obigen Angaben folgendes „Bild“:

00000110 00000110 00100000 00000000 01000111 11011011 10000001 10100100 11000000 01100100
 10110000 00011111 01101110 00001000 00000000 00010000

