

---

---

---

---

---

---

---



# UM WAS GEHT'S IN INFORMATIK

## TECHNISCHE INFORMATIK

Netzwerk

Google

## PRAKTISCHE INFORMATIK

Programmierung

Script

## THEORETISCHE INFORMATIK

Mathematik

## ANGEWANDTE INFORMATIK

Excel

Bedienung

Computerspiele

Fotos

Word

PowerPoint

## Einführungs-„Test“

Klasse: \_\_\_\_\_

Name: \_\_\_\_\_

Kreuze jeweils die richtigen Antworten an. Achtung: es können mehrere Antworten richtig sein!

- |   |                                     |   |                                     |
|---|-------------------------------------|---|-------------------------------------|
| 1. Was ist ein <b>Bit</b>   |                                     | 7. Welche <b>Datenmenge</b> passt auf eine handelsübliche <b>CD</b> ? |                                     |
| a) die kleinste Informationseinheit   | <input checked="" type="checkbox"/> | a) 4,7 GB   | <input type="checkbox"/>            |
| b) Abkürzung für das Wort <b>Bitte</b>  | <input type="checkbox"/>            | b) 25 KB  | <input type="checkbox"/>            |
| 2. Welche Zustände kann ein <b>Bit</b> annehmen?  |                                     | c) 700 MB   | <input checked="" type="checkbox"/> |
| a) an oder aus  | <input checked="" type="checkbox"/> | 8. Was bedeutet <b>WWW</b> ?  |                                     |
| b) an und aus   | <input type="checkbox"/>            | a) Wie Wo Was   | <input type="checkbox"/>            |
| c) 0 oder 1   | <input checked="" type="checkbox"/> | b) World Wide Web   | <input checked="" type="checkbox"/> |
| 3. Ein <b>Byte</b> besteht heutzutage zumeist aus:  |                                     | c) Welt Weites Wissen   | <input type="checkbox"/>            |
| a) 4 Bit  | <input type="checkbox"/>            | 9. Was ist <u>kein</u> <b>Browser</b> ?                               |                                     |
| b) 8 Bit  | <input checked="" type="checkbox"/> | a) Google ( <i>Chrome</i> )   | <input checked="" type="checkbox"/> |
| c) 100 Bit  | <input type="checkbox"/>            | b) Internet-Explorer  | <input type="checkbox"/>            |
| 4. Ordne die folgenden Speicherkapazitäten nach ihrer Größe (1-6)! Beginne mit der kleinsten Einheit. |                                     | c) Excel  | <input checked="" type="checkbox"/> |
| a) 105 <del>k</del> <sup>B</sup> B  | <u>3</u>                            | d) Mozilla Firefox  | <input type="checkbox"/>            |
| b) 256 Byte   | <u>2</u>                            | 10. In welcher Reihenfolge verläuft eine typische PC-Sitzung (1-6)?   |                                     |
| c) 100 TB   | <u>6</u>                            | a) Betriebssystem startet   | <u>3</u>                            |
| d) 127 MB   | <u>5</u>                            | b) PC ausschalten   | <u>6</u>                            |
| e) 125 <del>k</del> <sup>B</sup> B  | <u>4</u>                            | c) BIOS prüft Hardware  | <u>2</u>                            |
| f) 256 Bit  | <u>1</u>                            | d) PC einschalten   | <u>1</u>                            |
| 5. Was gehört zur <b>Hardware</b> ?   |                                     | e) Anwendungen nutzen   | <u>4</u>                            |
| a) Maus   | <input checked="" type="checkbox"/> | f) Betriebssystem herunterfahren                                      | <u>5</u>                            |
| b) Betriebssystem   | <input type="checkbox"/>            | 11. Was ist <u>kein</u> <b>Betriebssystem</b> ?                       |                                     |
| c) Tastatur   | <input checked="" type="checkbox"/> | a) Windows 10   | <input type="checkbox"/>            |
| d) Drucker  | <input checked="" type="checkbox"/> | b) Powerpoint   | <input checked="" type="checkbox"/> |
| e) RAM  | <input checked="" type="checkbox"/> | c) Windows XP   | <input type="checkbox"/>            |
| 6. Was gehört zur <b>Software</b> ?   |                                     | d) Linux  | <input type="checkbox"/>            |
| a) RAM  | <input type="checkbox"/>            | e) Google   | <input checked="" type="checkbox"/> |
| b) Word   | <input checked="" type="checkbox"/> | f) Android  | <input type="checkbox"/>            |
| c) Internet-Explorer  | <input checked="" type="checkbox"/> |   |                                     |
| d) Prozessor  | <input type="checkbox"/>            |   |                                     |
| e) Windows 10   | <input checked="" type="checkbox"/> |   |                                     |

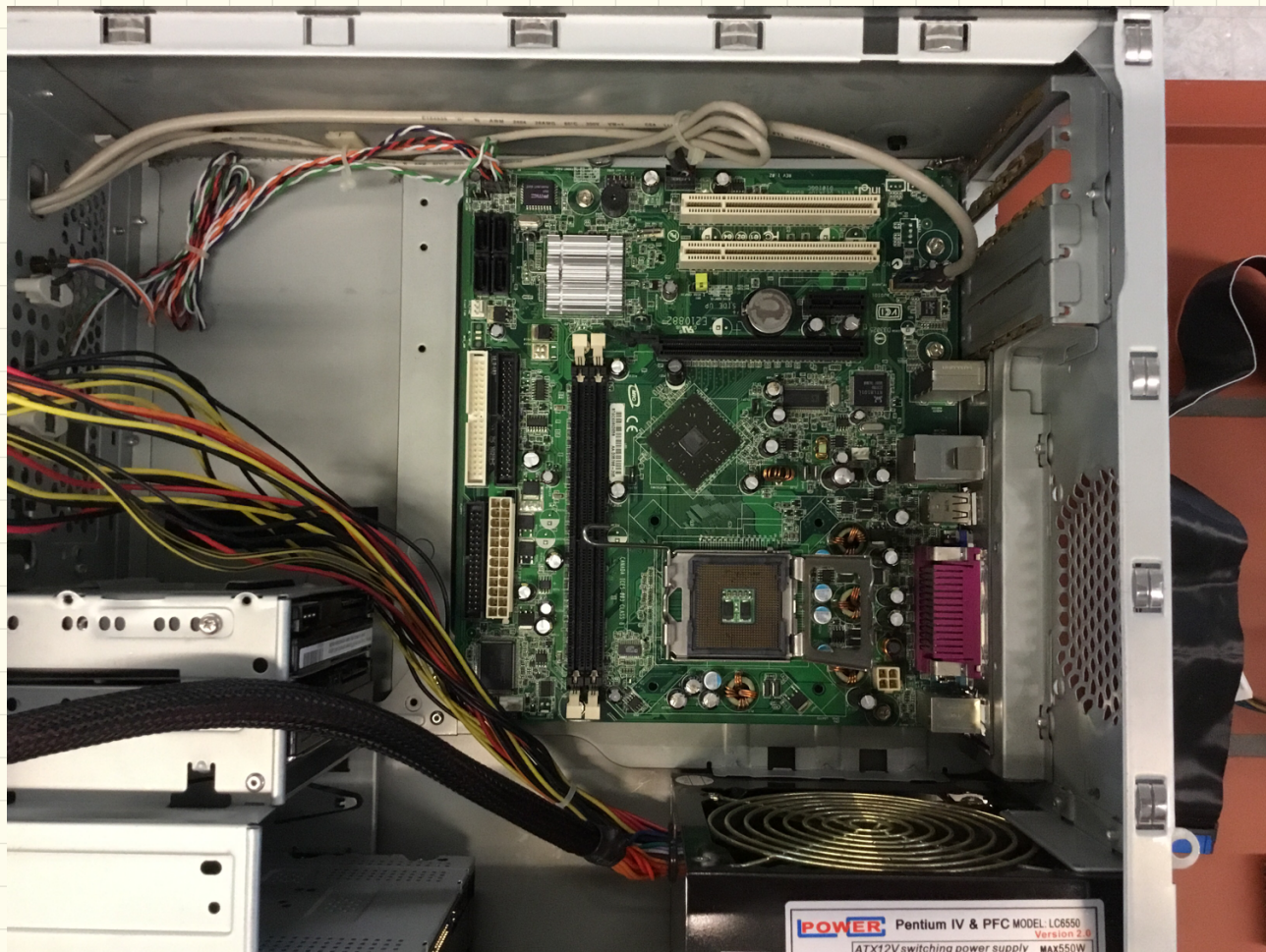
## Begriffe

W F Y H L P R R S J Q D X R L K K D C N  
 J P A U X C U E O E C H N V A N R A R T  
 L D R M F T E X B A T J L E O M E T K Q  
 B X P S A V O X G A U Y B J J B W E I Y  
 F V P T H A R D W A R E B N L H F N W C  
 U J S M S W S C N S C K G V P Q U S O F  
 M A T O A E W K G R O X Y N Z T A C N S  
 T O X N V O L V G B E F S R E F L H I L  
 X E T T A L P T S E F T T Q X M Y U N I  
 N S G H W W R C U C I Y U W D M D T T V  
 Q Q R B E E L A N B Z B V P A R E Z E K  
 M A U S C R P K B R I S B I M R S R R V  
 N Q H H Z U B T G L D U G R P O E O N B  
 Y M N W N W D O D X S R S R K A C M E R  
 M E C W S W W S A B U I V L D R H Y T A  
 R S H E E Y C N P R Q V S J B L Y V L M  
 K T T Q S H I Q I O D J D B P S Y F X Y  
 B E T R I E B S S Y S T E M B X Z K T Q  
 L A A R V G V F Q X C C E B P Y X K C A  
 G T M C J O R E K C U R D F X P B Z V I

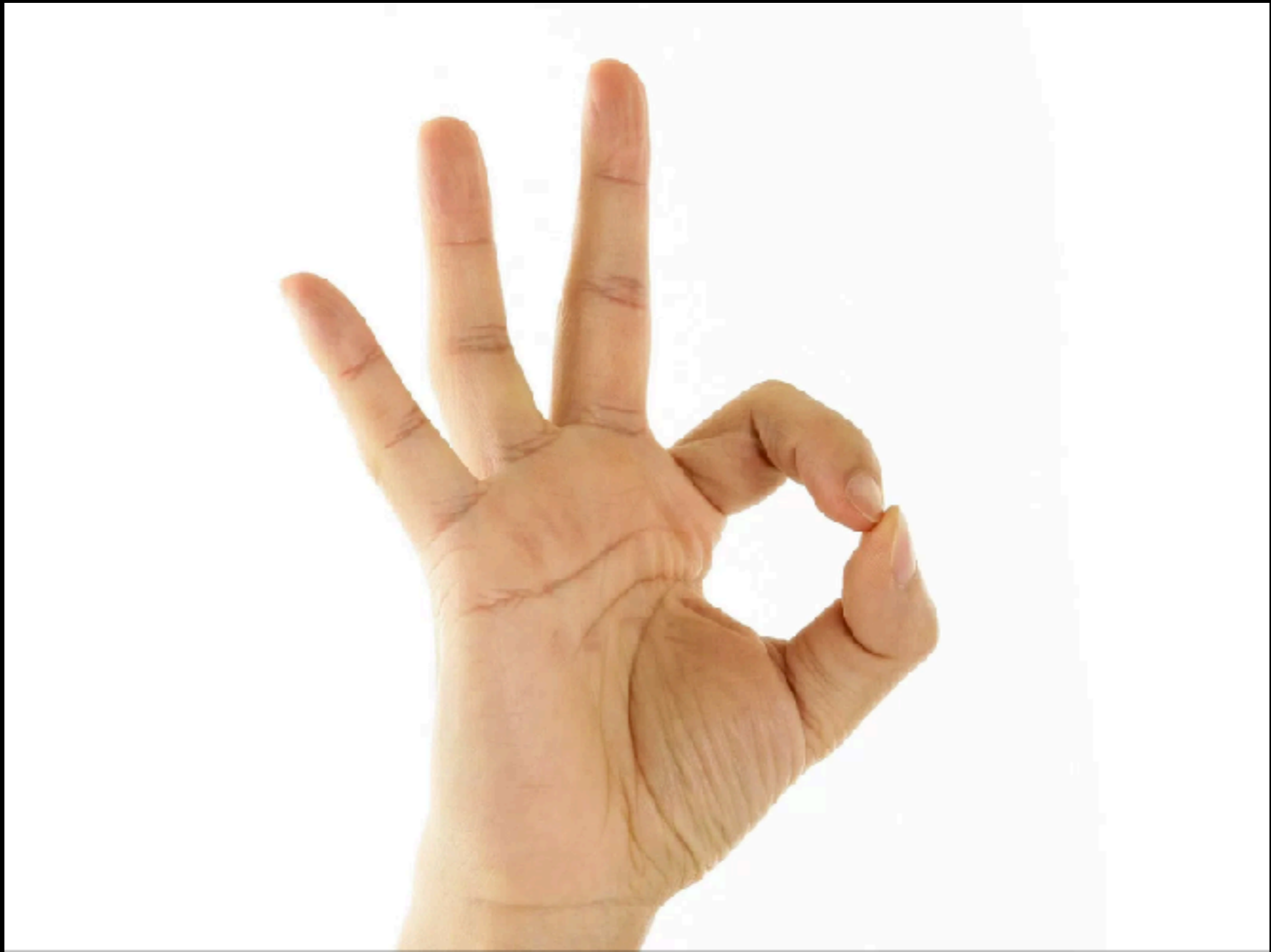
Suche folgende Begriffe:

- |                  |               |               |            |
|------------------|---------------|---------------|------------|
| • Betriebssystem | • Datenschutz | • Laufwerk    | • ROM      |
| • Bildschirm     | • Drucker     | • Maus        | • Software |
| • Bits           | • Festplatte  | • Motherboard | • Tastatur |
| • Bytes          | • Hardware    | • RAM         | • Virus    |
| • Computer       | • Internet    | • Rechner     |            |

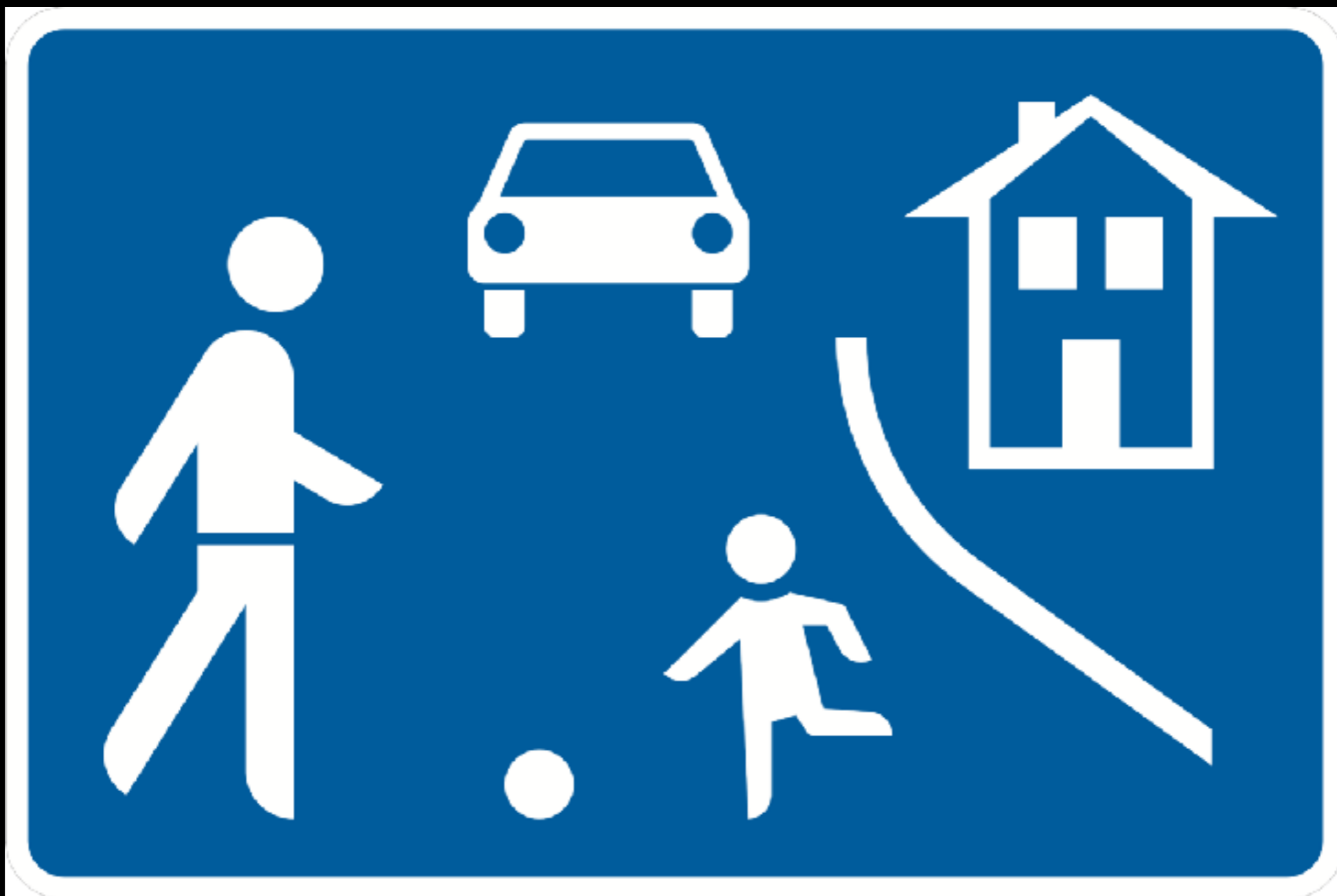
Die Begriffe sind waagrecht, senkrecht und diagonal versteckt und teilweise auch rückwärts!













MMXVIII

2/21/2018

**21.2.2018**

2018 - 02 - 21

## 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.

## 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.

## 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.

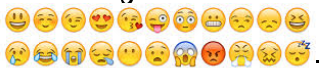

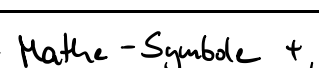

Der Computer kann unsere Zeichenfolgen nicht lesen. Wir müssen unsere Sprache in eine Sprache übersetzen, die der Computer versteht. Dieses Übersetzen nennt man Codierung.

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.

## 1. Aufgabe

Überlege dir, welche Informationen es in deinem Alltag gibt und wie diese codiert werden. Welche Dinge fallen dir ein, deren Bedeutung du sofort erkennst, ohne dass diese beschrieben sind?

*Beispiel:* Eine Ampel signalisiert mit den Farben rot, gelb und grün, ob man fahren/gehen darf oder stehen bleiben muss.

- Warnsymbole (Achtung Elektrozaun,...)	- Emoji :-)
- Ziehen / Drücken	
- Play ►, Stop ■, Pause	
- Wasserhahn: blau = kalt, rot = warm	
- Einschalttaste ⏻	- Mathe-Symbole +, -, ;, :
- Verkehrsschilder	

0  
 1  
 2  
 3  
 4  
 5  
 6  
 7  
 8  
 9  
 10  
 :  
 99  
 100

0  
 1  
 2  
 3  
 4  
 5  
 10  
 11  
 12  
 13  
 14  
 15  
 20

0  
 1  
 10  
 11  
 100  
 101  
 110  
 111  
 1000

## 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.

Wie du aus Mathematik sicherlich kennst können wir die Stellen auch benennen:

T	H	Z	E
6	3	1	2
		1	0
	1	0	0

Wir nennen die Stellen dann:

**E:** Einerstelle

**H:** Hunderterstelle

**Z:** Zehnerstelle

**T:** Tausenderstelle

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 <sub>2</sub>
1	1 <sub>2</sub>
2	10 <sub>2</sub>
3	11 <sub>2</sub>
4	100 <sub>2</sub>
5	101 <sub>2</sub>

← 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 <sub>2</sub>
1	1 <sub>2</sub>
2	10 <sub>2</sub>
3	11 <sub>2</sub>
4	100 <sub>2</sub>
5	101 <sub>2</sub>
6	110 <sub>2</sub>
7	111 <sub>2</sub>
8	1000 <sub>2</sub>
9	1001 <sub>2</sub>
10	1010 <sub>2</sub>
11	1011 <sub>2</sub>
12	1100 <sub>2</sub>
13	1101 <sub>2</sub>
14	1110 <sub>2</sub>
15	1111 <sub>2</sub>

Dezimalsystem	Binärsystem
16	10000 <sub>2</sub>
17	10001 <sub>2</sub>
18	10010 <sub>2</sub>
19	10011 <sub>2</sub>
20	10100 <sub>2</sub>
21	10101 <sub>2</sub>
22	10110 <sub>2</sub>
23	10111 <sub>2</sub>
24	11000 <sub>2</sub>
25	11001 <sub>2</sub>
26	11010 <sub>2</sub>
27	11011 <sub>2</sub>
28	11100 <sub>2</sub>
29	11101 <sub>2</sub>
30	11110 <sub>2</sub>
31	11111 <sub>2</sub>
32	100000 <sub>2</sub>

## 2. Aufgabe

Erinner dich nochmals an die Benennung der Stellen im Dezimalsystem (s. Vorderseite). Wie würdest du die Stellen im Binärsystem benennen?

F	E	D	C	B	A
1	1	0	1	0	1

Wir nennen die Stellen dann:

A: Einer stelle      C: Vier er stelle      E: Sechzehner stelle  
 B: Zweier stelle      D: Achter stelle      F: 32er stelle

## 3. Aufgabe

Überlegt euch zu zweit 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<sub>2</sub>, 101<sub>2</sub>, 1101<sub>2</sub>, 10101<sub>2</sub>, 100001<sub>2</sub> und 101111<sub>2</sub>

$$101001_2 = 41$$

$$4721 = 4 \cdot 1000 + 7 \cdot 100 + 2 \cdot 10 + 1 \cdot 1$$

$$11010_2 = 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 26_{10}$$

$\leftarrow$   $\leftarrow$   $\leftarrow$   $\leftarrow$   $\leftarrow$   
 Neu-Stelle    8er-Stelle    4er-Stelle    2er-Stelle    Einerstelle

32 16 8 4 2 1

$$1 \ 0 \ 0 \ 1 \ 0 \ 1_2 = 37$$

## Umrechnung Binärsystem → Dezimalsystem

Im Dezimalsystem können wir eine Zahl auch als Summe der Dezimalstellen schreiben:

$$6312 = 6 \cdot \underbrace{1000}_{=10 \cdot 10 \cdot 10} + 3 \cdot \underbrace{100}_{=10 \cdot 10} + 1 \cdot 10 + 2 \cdot 1$$

Da wir im Binärsystem jedoch nicht mit der Basis 10 sondern mit der Basis 2 arbeiten können wir die Binärzahlen auch wie folgt darstellen:

$$11010_2 = 1 \cdot \underbrace{16}_{=2 \cdot 2 \cdot 2 \cdot 2} + 1 \cdot \underbrace{8}_{=2 \cdot 2 \cdot 2} + 0 \cdot \underbrace{4}_{=2 \cdot 2} + 1 \cdot 2 + 0 \cdot 1 = 26_{10}$$

### 1. Aufgabe

Rechne die folgenden Zahlen ins Dezimalsystem um:

a)  $110_2 = 4 + 2 = \underline{6}$

b)  $1011_2 = 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = \underline{11}$

c)  $10101_2 = 16 + 4 + 1 = \underline{21}$

d)  $11000_2 = 16 + 8 = \underline{24}$

e)  $100101_2 = 32 + 4 + 1 = \underline{37}$

f)  $111010_2 = 32 + 16 + 8 + 2 = \underline{58}$

g)  $1000110_2 = 64 + 4 + 2 = \underline{70}$

h)  $1101011_2 = 64 + 32 + 8 + 2 + 1 = \underline{107}$

*Hinweis: genau wie man Zahlen im Dezimalsystem oft in 3er-Gruppen schreibt, so gruppiert man Zahlen im Binärsystem zur besseren Lesbarkeit in 4er-Gruppen.*

### 2. Aufgabe

Ein **Bit** ist die kleinste Informationseinheit im Computer und kann 0 oder 1 darstellen. Damit wird jede Stelle einer binären Zahl durch ein Bit repräsentiert.

Welches ist die größte Zahl, die mit

a) 2 Bit  $00_2 / 01_2 / 10_2 / 11_2$

b) 4 Bit  $1111_2 = \underline{15}$

c) 6 Bit  $111111_2 = \underline{63}$

d) 8 Bit  $11111111_2 = \underline{255}$

dargestellt werden kann?

$$41 = 0 \cdot 64 + 41$$

$$41 = 1 \cdot 32 + 9$$

$$9 = 0 \cdot 16 + 9$$

$$9 = 1 \cdot 8 + 1$$

$$1 = 0 \cdot 4 + 1$$

$$1 = 0 \cdot 2 + 1$$

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

101001

$$57 = 0 \cdot 64 + 57$$

$$57 = 1 \cdot 32 + 25$$

$$25 = 1 \cdot 16 + 9$$

$$9 = 1 \cdot 8 + 1$$

$$1 = 0 \cdot 4 + 1$$

$$1 = 0 \cdot 2 + 1$$

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

$111001_2$

## Umrechnung Dezimalsystem $\rightarrow$ Binärsystem

Umgekehrt müssen wir eine Dezimalzahl als Summe ausdrücken. Dazu schreiben wir zunächst die *2er-Potenzreihe* (rückwärts) auf: Wir beginnen rechts mit der 1 und multiplizieren die Zahl mit 2 bei jedem Schritt.

$$\dots \leftarrow 32 \leftarrow 16 \leftarrow 8 \leftarrow 4 \leftarrow 2 \leftarrow 1$$

Anschließend zerlegen wir die Dezimalzahl in die Summe, beispielsweise  $25_{10}$ :

$$25 = 0 \cdot 32 \quad + 25 = 1 \cdot 16 \quad + 9 = 1 \cdot 8 \quad + 1 = 0 \cdot 4 \quad + 1 = 0 \cdot 2 \quad + 1 = 1 \cdot 1 \quad + 0$$

und erhalten damit die Binärdarstellung:

$$25_{10} = 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 11001_2$$

### 3. Aufgabe

Rechne die folgenden Zahlen ins Binärsystem um:

a) 7  $= 111_2$

f) 63  $= 111111_2$

b) 13  $= 1101_2$

g) 71  $= 1000111_2$

c) 24  $= 11000_2$

h) 93  $= 1011101_2$

d) 36  $= 100100_2$

i) 103  $= 1100111_2$

e) 49  $= 110001_2$

j) 127  $= 1111111_2$

### 4. Aufgabe

Wie viele Stellen haben die folgenden Zahlen im Binärsystem?

a) 3  $= 11 \rightarrow 2$  Stellen

e) 15  $= 1111 \rightarrow 4$  Stellen

b) 4  $= 100 \rightarrow 3$  Stellen

f) 16  $= 10000 \rightarrow 5$  Stellen

c) 7  $= 111 \rightarrow 3$  Stellen

g) 31  $= 11111 \rightarrow 5$  Stellen

d) 8  $= 1000 \rightarrow 4$  Stellen

h) 32  $= 100000 \rightarrow 6$  Stellen

Erkennst du ein Muster? Wie viele binäre Stellen hätten dann die folgenden Zahlen?

a) 64  $\rightarrow 7$  Stellen

d) 512  $\rightarrow 10$  Stellen

b) 128  $\rightarrow 8$  Stellen

e) 1024  $\rightarrow 11$  Stellen

c) 256  $\rightarrow 9$  Stellen

f) 2048  $\rightarrow 12$  Stellen

$$= 128 +$$

$$= 64 +$$

$$= 32 +$$

$$= 16 +$$

$$13 = 1 \cdot 8 + 5$$

$$5 = 1 \cdot 4 + 1$$

$$1 = 0 \cdot 2 + 1$$

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

7.3.18

ASCII control characters		
00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters					
32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	\$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

Extended ASCII characters					
128	Ç	160	á	192	Ł
129	ü	161	í	193	ł
130	é	162	ó	194	Ť
131	â	163	ú	195	ť
132	ä	164	ñ	196	—
133	à	165	Ñ	197	†
134	ă	166	ª	198	‡
135	ç	167	º	199	Ã
136	ê	168	¿	200	Ł
137	ë	169	®	201	Œ
138	è	170	¬	202	℥
139	ï	171	½	203	℥
140	î	172	¼	204	℥
141	ì	173	¡	205	=
142	Ä	174	«	206	≠
143	Å	175	»	207	¤
144	É	176	⋯	208	ð
145	æ	177	⋮	209	Ð
146	Æ	178	■	210	Ê
147	ô	179	┆	211	Ë
148	ö	180	┆	212	È
149	ò	181	À	213	Ì
150	û	182	Â	214	Í
151	ù	183	À	215	Î
152	ÿ	184	©	216	Ï
153	Ö	185	≡	217	Ɔ
154	Ü	186	≡	218	┆
155	ø	187	┆	219	■
156	£	188	┆	220	■
157	Ø	189	¢	221	¡
158	×	190	¥	222	ì
159	ƒ	191	ƒ	223	■
				224	Ó
				225	ß
				226	Ô
				227	Ò
				228	ō
				229	Õ
				230	μ
				231	þ
				232	þ
				233	Ú
				234	Û
				235	Ü
				236	ý
				237	Ý
				238	—
				239	‘
				240	≡
				241	±
				242	≡
				243	¾
				244	¶
				245	§
				246	÷
				247	ˆ
				248	ˆ
				249	ˆ
				250	ˆ
				251	¹
				252	³
				253	²
				254	■
				255	nbsp

# Codierung: Text und Zeichen

## 1. Einführung: ASCII

Der **American Standard Code for Information Interchange** (ASCII) ist eine 7-Bit-Zeichencodierung und dient als Grundlage für spätere, auf mehr Bits basierende Kodierungen für Zeichensätze. Die druckbaren Zeichen umfassen das lateinische Alphabet in Groß- und Kleinschreibung, die zehn arabischen Ziffern sowie einige Interpunktions- und andere Sonderzeichen. Der Zeichenvorrat entspricht weitgehend dem einer Tastatur oder Schreibmaschine für die *englische Sprache*.

Um weitere Zeichen darstellen zu können, wurden von verschiedenen Gremien Erweiterungen für das Standardpaket entwickelt, welche dann 8 Bit verwendet und damit 256 Zeichen darstellen können. Die ersten 128 Zeichen sind dabei identisch zum Standard-Zeichensatz:

ASCII control characters			ASCII printable characters				Extended ASCII characters									
00	NULL	(Null character)	32	space	64	@	96	`	128	Ç	160	á	192	Ł	224	Ó
01	SOH	(Start of Header)	33	!	65	A	97	a	129	ü	161	í	193	ł	225	ô
02	STX	(Start of Text)	34	"	66	B	98	b	130	é	162	ó	194	Ł	226	Ô
03	ETX	(End of Text)	35	#	67	C	99	c	131	â	163	ú	195	ł	227	Õ
04	EOT	(End of Trans.)	36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	ö
05	ENQ	(Enquiry)	37	%	69	E	101	e	133	à	165	Ñ	197	†	229	Ö
06	ACK	(Acknowledgement)	38	&	70	F	102	f	134	â	166	ª	198	ä	230	µ
07	BEL	(Bell)	39	'	71	G	103	g	135	ç	167	º	199	Å	231	þ
08	BS	(Backspace)	40	(	72	H	104	h	136	ê	168	¿	200	Ł	232	þ
09	HT	(Horizontal Tab)	41	)	73	I	105	i	137	ë	169	®	201	Œ	233	Û
10	LF	(Line feed)	42	*	74	J	106	j	138	è	170	™	202	Œ	234	Ü
11	VT	(Vertical Tab)	43	+	75	K	107	k	139	ï	171	½	203	Œ	235	Ù
12	FF	(Form feed)	44	,	76	L	108	l	140	î	172	¼	204	Œ	236	Ý
13	CR	(Carriage return)	45	-	77	M	109	m	141	ï	173	ı	205	Œ	237	Ÿ
14	SO	(Shift Out)	46	.	78	N	110	n	142	Ä	174	«	206	Œ	238	—
15	SI	(Shift In)	47	/	79	O	111	o	143	Å	175	»	207	Œ	239	˙
16	DLE	(Data link escape)	48	0	80	P	112	p	144	É	176	§	208	ø	240	≡
17	DC1	(Device control 1)	49	1	81	Q	113	q	145	æ	177	§	209	ð	241	±
18	DC2	(Device control 2)	50	2	82	R	114	r	146	Æ	178	§	210	É	242	≡
19	DC3	(Device control 3)	51	3	83	S	115	s	147	ø	179	§	211	Ê	243	¾
20	DC4	(Device control 4)	52	4	84	T	116	t	148	ö	180	§	212	Ë	244	¶
21	NAK	(Negative acknowl.)	53	5	85	U	117	u	149	ò	181	À	213	ı	245	§
22	SYN	(Synchronous idle)	54	6	86	V	118	v	150	û	182	Á	214	ı	246	÷
23	ETB	(End of trans. block)	55	7	87	W	119	w	151	ü	183	Â	215	ı	247	°
24	CAN	(Cancel)	56	8	88	X	120	x	152	ÿ	184	©	216	ı	248	˙
25	EM	(End of medium)	57	9	89	Y	121	y	153	Ö	185	Œ	217	ı	249	˙
26	SUB	(Substitute)	58	:	90	Z	122	z	154	Ü	186	Œ	218	ı	250	˙
27	ESC	(Escape)	59	;	91	[	123	{	155	ø	187	Œ	219	ı	251	˙
28	FS	(File separator)	60	<	92	\	124		156	£	188	Œ	220	ı	252	˙
29	GS	(Group separator)	61	=	93	]	125	}	157	Ø	189	Œ	221	ı	253	˙
30	RS	(Record separator)	62	>	94	^	126	~	158	×	190	Œ	222	ı	254	˙
31	US	(Unit separator)	63	?	95	_			159	f	191	Œ	223	ı	255	nbsp
127	DEL	(Delete)														

## 2. Aufgabe

Entschlüssele folgende Nachricht:

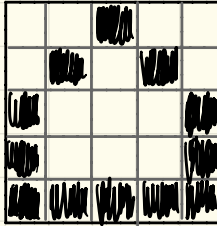
73 110 102 111 114 109 97 116 105 107 32 105 115 116 32 116 111 108 108 33  
 l h f o r m a t i k i s t t o l l o

## 3. Aufgabe

Schreibe deinem Sitznachbarn eine (kurze) Nachricht, indem du lediglich die ASCII-Codes verwendest.

## 4. Zusatzaufgabe

Warum gab es immer wieder Fehler in der Darstellung von Webseiten? Wie wurde das Problem umgangen?



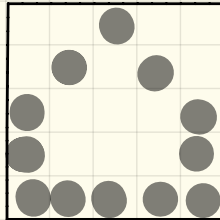
```

0 0 1 0 0
0 1 0 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1

```

$\underbrace{00000101}_{\text{Breite}}$ 
 $\underbrace{00000101}_{\text{Höhe}}$ 
0 0 1 0 0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0

5 = 1 0 1



# 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.

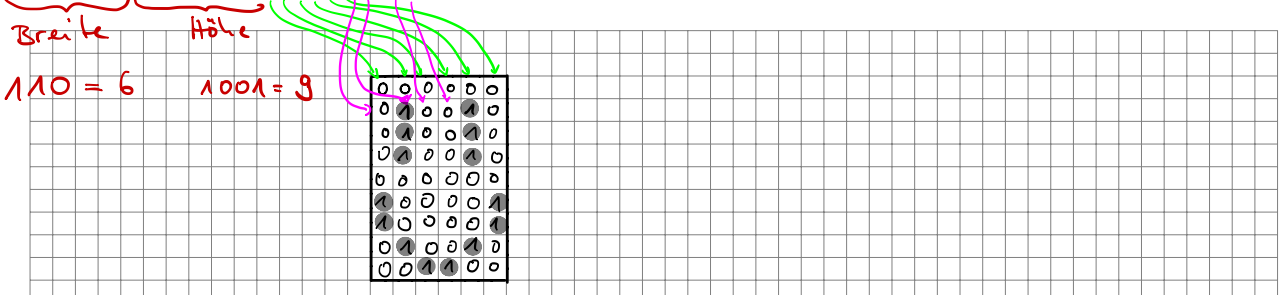
Damit der Computer jedoch weiß, wann eine neue Zeile beginnt, muss zunächst die Breite des Bildes angegeben werden. Damit ein Bild richtig abgespeichert werden kann und auch wieder richtig gelesen werden kann müssen wir uns ein eigenes Dateiformat definieren:

- 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

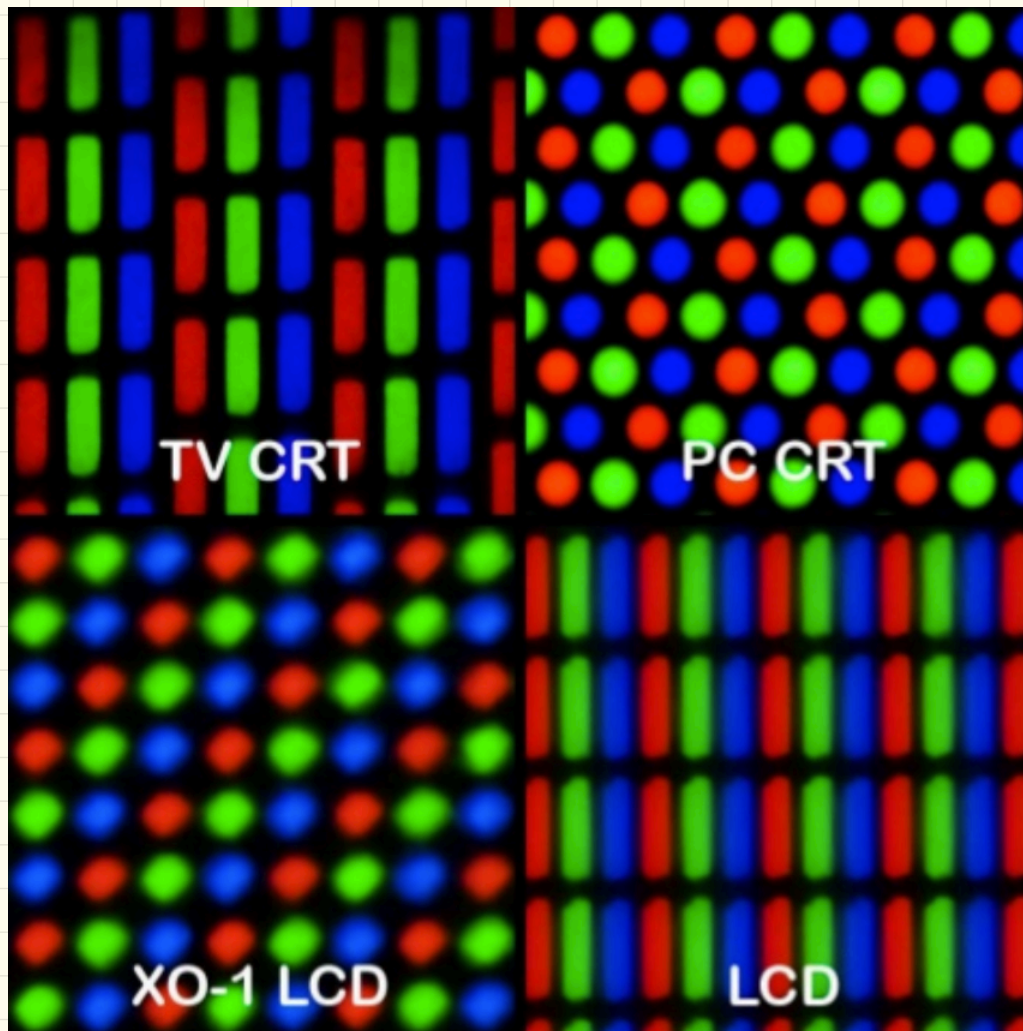
Entschlüsse mit obigen Angaben folgendes „Bild“:

00000110 00001001 00000001 00100100 10010010 00000010 00011000 01010010 00110000



## 3. Aufgabe

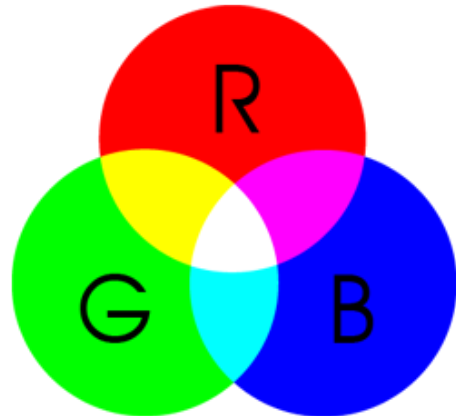
Erstelle selbst ein Bild, codiere es in eine Bitfolge und gib es deinem Nachbarn zur Entschlüsselung.



## 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



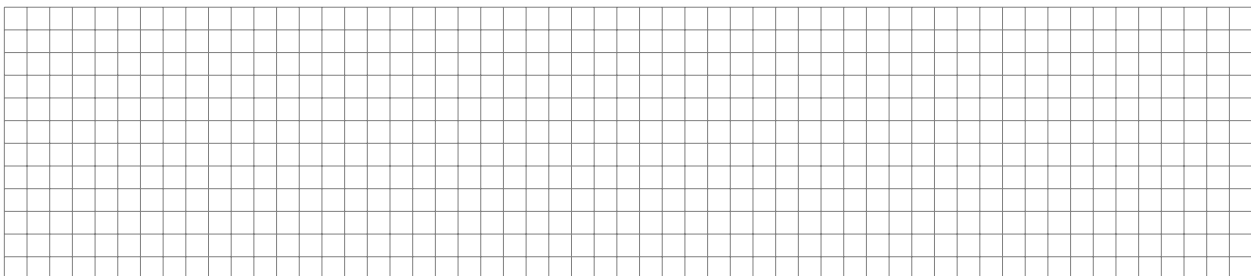
Damit definieren wir ein neues Dateiformat:

- 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

Entschlüsse mit obigen Angaben folgendes „Bild“:

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



# Übungen zur Klassenarbeit

## 1. Binärsystem → Dezimalsystem

Rechne die folgenden Zahlen aus dem Binärsystem in das Dezimalsystem um:

a)  $101_2$   $\overset{4}{1} \overset{2}{0} \overset{1}{1}$   $\overset{5}{5}$

b)  $11010_2$   $\overset{16}{1} \overset{8}{1} \overset{4}{0} \overset{2}{1} \overset{1}{0}$   $\overset{26}{26}$

c)  $100001_2$   $\overset{32}{1} \overset{16}{0} \overset{8}{0} \overset{4}{0} \overset{2}{0} \overset{1}{1}$   $\overset{33}{33}$

d)  $11001100_2$   $\overset{128}{1} \overset{64}{1} \overset{32}{0} \overset{16}{0} \overset{8}{1} \overset{4}{1} \overset{2}{0} \overset{1}{0}$   $\overset{204}{204}$

## 2. Dezimalsystem → Binärsystem

Rechne die folgenden Zahlen aus dem Dezimalsystem in das Binärsystem um:

a)  $9_{10}$   $\overset{1001}{1001}$

b)  $17_{10}$   $\overset{10001}{10001}$

c)  $99_{10}$   $\overset{256}{1} \overset{128}{1} \overset{64}{0} \overset{32}{0} \overset{16}{0} \overset{8}{0} \overset{4}{0} \overset{2}{0} \overset{1}{1}$

d)  $219_{10}$   $\overset{256}{1} \overset{128}{1} \overset{64}{0} \overset{32}{0} \overset{16}{1} \overset{8}{0} \overset{4}{0} \overset{2}{1} \overset{1}{1}$

## 3. Bildcodierung

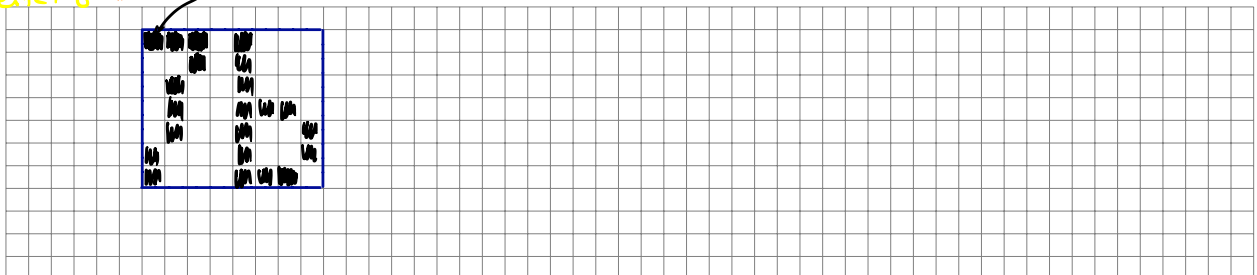
Wir definieren unser Bildformat wie folgt:

- 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.

Entschlüssele mit diesen Angaben folgendes „Bild“:

$00001000$   $00000111$   $01101000$   $00101000$   $01001000$   $01001110$   $01001001$   $10001001$   $10001110$

Breite: 8 Höhe: 7



## 4. Textcodierung

ASCII control characters			ASCII printable characters			Extended ASCII characters										
00	NULL	(Null character)	32	space	64	@	96	`	128	Ç	160	á	192	Ł	224	Ó
01	SOH	(Start of Header)	33	!	65	A	97	a	129	ü	161	í	193	ł	225	ô
02	STX	(Start of Text)	34	"	66	B	98	b	130	é	162	ó	194	Ł	226	õ
03	ETX	(End of Text)	35	#	67	C	99	c	131	â	163	ú	195	ł	227	ö
04	EOT	(End of Trans.)	36	\$	68	D	100	d	132	ä	164	ñ	196	Ł	228	÷
05	ENQ	(Enquiry)	37	%	69	E	101	e	133	à	165	Ñ	197	ł	229	ø
06	ACK	(Acknowledgement)	38	&	70	F	102	f	134	á	166	ª	198	Ł	230	µ
07	BEL	(Bell)	39	'	71	G	103	g	135	ç	167	º	199	Ł	231	þ
08	BS	(Backspace)	40	(	72	H	104	h	136	ê	168	¿	200	Ł	232	ß
09	HT	(Horizontal Tab)	41	)	73	I	105	i	137	ë	169	®	201	Ł	233	Ù
10	LF	(Line feed)	42	*	74	J	106	j	138	è	170	¬	202	Ł	234	Ú
11	VT	(Vertical Tab)	43	+	75	K	107	k	139	ï	171	½	203	Ł	235	Û
12	FF	(Form feed)	44	,	76	L	108	l	140	î	172	¼	204	Ł	236	Ý
13	CR	(Carriage return)	45	-	77	M	109	m	141	ï	173	⅓	205	Ł	237	Ÿ
14	SO	(Shift Out)	46	.	78	N	110	n	142	Ä	174	«	206	Ł	238	
15	SI	(Shift In)	47	/	79	O	111	o	143	Å	175	»	207	Ł	239	˘
16	DLE	(Data link escape)	48	0	80	P	112	p	144	É	176	⌘	208	Ł	240	≡
17	DC1	(Device control 1)	49	1	81	Q	113	q	145	æ	177	⌘	209	Ł	241	±
18	DC2	(Device control 2)	50	2	82	R	114	r	146	Æ	178	⌘	210	Ł	242	¼
19	DC3	(Device control 3)	51	3	83	S	115	s	147	ø	179	⌘	211	Ł	243	¾
20	DC4	(Device control 4)	52	4	84	T	116	t	148	ö	180	⌘	212	Ł	244	¶
21	NAK	(Negative acknowl.)	53	5	85	U	117	u	149	ò	181	Å	213	Ł	245	§
22	SYN	(Synchronous idle)	54	6	86	V	118	v	150	û	182	Ä	214	Ł	246	÷
23	ETB	(End of trans. block)	55	7	87	W	119	w	151	ù	183	Å	215	Ł	247	°
24	CAN	(Cancel)	56	8	88	X	120	x	152	ÿ	184	©	216	Ł	248	°
25	EM	(End of medium)	57	9	89	Y	121	y	153	Ö	185	⌘	217	Ł	249	°
26	SUB	(Substitute)	58	:	90	Z	122	z	154	Ü	186	⌘	218	Ł	250	°
27	ESC	(Escape)	59	;	91	[	123	{	155	ø	187	⌘	219	Ł	251	°
28	FS	(File separator)	60	<	92	\	124		156	£	188	⌘	220	Ł	252	°
29	GS	(Group separator)	61	=	93	]	125	}	157	Ø	189	¢	221	Ł	253	°
30	RS	(Record separator)	62	>	94	^	126	~	158	×	190	¥	222	Ł	254	■
31	US	(Unit separator)	63	?	95	_			159	ƒ	191	¬	223	Ł	255	nbsp
127	DEL	(Delete)														

Bringe die „Zeichenfolge“ in eine lesbare Form:

75 108 97 115 115 101 32 55 98

K l a s s e 7 b

## 5. Programmierung

Erkläre, was nebenstehendes Programm macht (schrittweise).

Skizziere das Muster, das dabei gezeichnet wird.

The Scratch script consists of the following blocks:

- Wenn angeklickt
- gehe zu x: 0 y: 0
- setze Richtung auf 90
- schalte Stift ein
- wische Malspuren weg
- setze Stiftfarbe auf
- setze Stiftstärke auf 2
- setze Schrittweite auf 8
- wiederhole 20 mal
  - gehe Schrittweite er-Schritt
  - drehe dich um 90 Grad
  - ändere Schrittweite um 10

Handwritten annotations and a sketch:

- gehe zur Mitte (pointing to 'gehe zu x: 0 y: 0')
- Blicke nach rechts (pointing to 'setze Richtung auf 90')
- Stift einschalten (pointing to 'schalte Stift ein' and 'wische Malspuren weg')
- Anfang: 8er-Schritt (pointing to 'setze Schrittweite auf 8')
- 20 mal (highlighted in yellow, pointing to the loop block)
- aktuelle schrittweite nach vorne gehen (pointing to 'gehe Schrittweite er-Schritt')
- drehen (pointing to 'drehe dich um 90 Grad')
- Schrittweite erhöhen (pointing to 'ändere Schrittweite um 10')

A red sketch of a square is shown next to the annotations.



# Klassenarbeit 10.01.2018

Name: \_\_\_\_\_ Punkte: \_\_\_\_\_/26P Note: \_\_\_\_\_ mündlich: \_\_\_\_\_

## 1. Binärsystem → Dezimalsystem (2P)

Rechne die folgenden Zahlen aus dem Binärsystem in das Dezimalsystem um:

a)  $1101_2$ 13b)  $10010_2$ 18c)  $11011_2$ 27d)  $11100010_2$ 226

## 2. Dezimalsystem → Binärsystem (2P)

Rechne die folgenden Zahlen aus dem Dezimalsystem in das Binärsystem um:

a)  $4_{10}$ 100b)  $27_{10}$ 11011c)  $65_{10}$ 1000001d)  $232_{10}$ 11101000

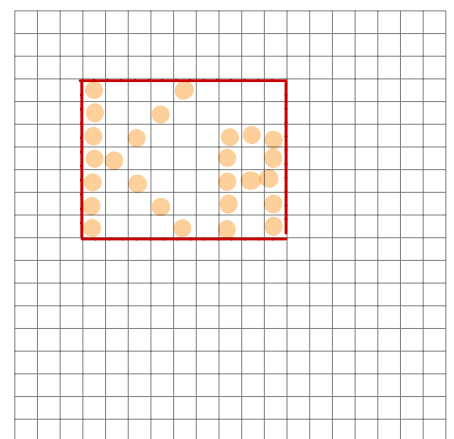
## 3. Bildcodierung (4P)

Entschlüssele folgende Bildcodierung und zeichne das Bild

- 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.

Entschlüssele mit diesen Angaben folgendes „Bild“:

00001001 00000111 10001000 01001000 00101000 11111000 01011010 00111100 10010110 00101010



## 4. Textcodierung (4P)

ASCII control characters			ASCII printable characters					Extended ASCII characters								
00	NULL	(Null character)	32	space	64	@	96	`	128	Ç	160	á	192	Ł	224	Ó
01	SOH	(Start of Header)	33	!	65	A	97	a	129	ü	161	í	193	ł	225	ô
02	STX	(Start of Text)	34	"	66	B	98	b	130	é	162	ó	194	Ł	226	Ö
03	ETX	(End of Text)	35	#	67	C	99	c	131	â	163	ú	195	ł	227	ò
04	EOT	(End of Trans.)	36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	õ
05	ENQ	(Enquiry)	37	%	69	E	101	e	133	à	165	Ñ	197	†	229	Ö
06	ACK	(Acknowledgement)	38	&	70	F	102	f	134	á	166	ª	198	‡	230	µ
07	BEL	(Bell)	39	'	71	G	103	g	135	ç	167	º	199	Ä	231	þ
08	BS	(Backspace)	40	(	72	H	104	h	136	ê	168	¿	200	Å	232	ÿ
09	HT	(Horizontal Tab)	41	)	73	I	105	i	137	ë	169	®	201	‡	233	Û
10	LF	(Line feed)	42	*	74	J	106	j	138	è	170	™	202	‡	234	Ü
11	VT	(Vertical Tab)	43	+	75	K	107	k	139	ï	171	½	203	‡	235	Ù
12	FF	(Form feed)	44	,	76	L	108	l	140	î	172	¾	204	‡	236	Ý
13	CR	(Carriage return)	45	-	77	M	109	m	141	ï	173	¿	205	‡	237	Ÿ
14	SO	(Shift Out)	46	.	78	N	110	n	142	Ä	174	«	206	‡	238	—
15	SI	(Shift In)	47	/	79	O	111	o	143	Å	175	»	207	‡	239	·
16	DLE	(Data link escape)	48	0	80	P	112	p	144	É	176	¸	208	‡	240	≡
17	DC1	(Device control 1)	49	1	81	Q	113	q	145	æ	177	¸	209	‡	241	±
18	DC2	(Device control 2)	50	2	82	R	114	r	146	Æ	178	¸	210	‡	242	±
19	DC3	(Device control 3)	51	3	83	S	115	s	147	ø	179	¸	211	‡	243	¼
20	DC4	(Device control 4)	52	4	84	T	116	t	148	ö	180	¸	212	‡	244	½
21	NAK	(Negative acknowl.)	53	5	85	U	117	u	149	ò	181	¸	213	‡	245	¾
22	SYN	(Synchronous idle)	54	6	86	V	118	v	150	û	182	¸	214	‡	246	÷
23	ETB	(End of trans. block)	55	7	87	W	119	w	151	ü	183	¸	215	‡	247	÷
24	CAN	(Cancel)	56	8	88	X	120	x	152	ý	184	¸	216	‡	248	÷
25	EM	(End of medium)	57	9	89	Y	121	y	153	ÿ	185	¸	217	‡	249	÷
26	SUB	(Substitute)	58	:	90	Z	122	z	154	Ü	186	¸	218	‡	250	÷
27	ESC	(Escape)	59	;	91	[	123	{	155	ø	187	¸	219	‡	251	÷
28	FS	(File separator)	60	<	92	\	124		156	£	188	¸	220	‡	252	÷
29	GS	(Group separator)	61	=	93	]	125	}	157	Ø	189	¸	221	‡	253	÷
30	RS	(Record separator)	62	>	94	^	126	~	158	×	190	¸	222	‡	254	÷
31	US	(Unit separator)	63	?	95	_			159	ƒ	191	¸	223	‡	255	nbsp
127	DEL	(Delete)														

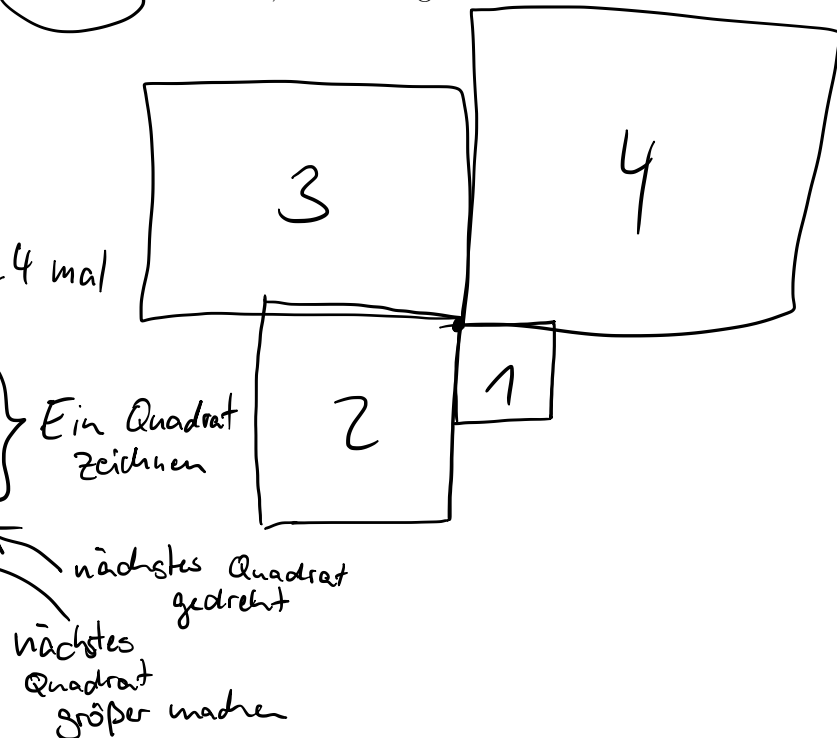
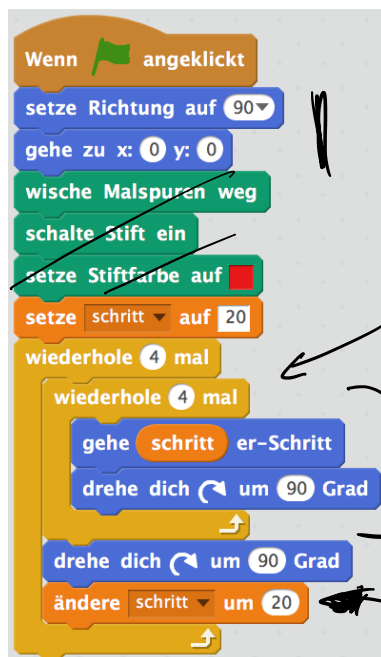
„Übersetze“ folgenden ASCII-codierten Text:

68 97 115 32 104 97 115 116 32 100 117 32 103 117 116 32 103 101 109 97 99 104 116 33

D a s h a s t d u g u t g e m a c h t !

## 5. Programmierung (4P)

Beschreibe in nebenstehendem Scratch-Programm die Schritte und skizziere das Muster, das dabei gezeichnet wird.



## 6. Datenübertragung im Internet (4P)

Arthur will eine Nachricht über WhatsApp an Beate schicken. Erkläre wie die Nachricht in vier Schritten von einem Smartphone zum anderen gesendet wird.

---

---

---

---

---

---

---

---

---

---

## 7. Datenaustausch (2P)

Du hast eine GFS gehalten und willst dein Handout deinen Klassenkameraden digital (also nicht in Papierform) zur Verfügung stellen. Welche Möglichkeit bietet Dir da das Schulnetzwerk? Gehe kurz auf die Begriffe „Client“ und „Server“ ein.

---

---

---

---

---

---

---

---

---

---

## 8. Verschlüsselung (2P)

Eine Schwachstelle bei der Konstruktion der Cäsar-Verschlüsselung ist die geringe Anzahl möglicher Schlüssel, da nur 26 Drehungen der Scheibe möglich sind.

- a) Wie kann man die Cäsar Drehscheibe modifizieren, um die Verschlüsselung sicherer zu machen? Wie wird diese Verschlüsselung dann genannt?
- b) Wie kann diese Verschlüsselung wiederum gebrochen werden? Erkläre dieses Verfahren kurz.

---

---

---

---

---

---

---

## 9. Fragen (2P)

Kreuze die richtige Antwort an. Es gibt immer nur eine Lösung.

- a) Welche Aussage über Cloud-Speicher stimmt nicht?
- ☐ Die Daten werden vor der Speicherung über das Internet übertragen.
  - ☒ Die Daten werden durch das Internet geschickt, bis sie wieder abgerufen werden.
  - ☐ Die Daten sind von jedem an das Internet angeschlossenen Rechner abrufbar.
  - ☐ Die Daten werden von einer kommerziellen Firma für den Anwender gespeichert.
- b) Was ist ein Router
- ☒ Ein Gerät, um Daten im Internet weiterzuschicken.
  - ☐ Ein Routenplaner für's Auto.
  - ☐ Ein Verkehrszeichen in England.
  - ☐ Ein besonders schnelles Datenpaket.
- c) Bei welchem Speichermedium hat man den schnellsten Zugriff?
- ☐ USB-Stick
  - ☐ Cloud-Speicher
  - ☒ lokale Festplatte (SSD)
  - ☐ auf dem Schulserver
- d) „Wer“ speichert die noch nicht ausgelieferten WhatsApp-Nachrichten?
- ☐ Das Handy des Empfängers speichert sie, auch wenn es nicht online ist.
  - ☐ Sie bewegen sich im Internet umher, bis sie zugestellt werden.
  - ☐ Das Handy des Absenders wartet bis der Empfänger online ist und schickt sie dann.
  - ☒ Der Server von Whatsapp.