3A

AND

Input A	Input B	Output X
0	0	0
1	0	0
0	1	0
1	1	1

Ausgang ist wahr (1), nur wenn beide Eingänge wahr sind.

OR

Input A	Input B	Output X
0	0	0
1	0	1
0	1	1
1	1	1

Ausgang ist wahr (1), wenn mindestens einer der Eingänge wahr ist.

XOR

Input A	Input B	Output X
0	0	0
1	0	1
0	1	1
1	1	0

Ausgang ist wahr (1), wenn genau einer der Eingänge wahr ist, aber nicht beide.

NAND

Input A	Input B	Output X
0	0	1
1	0	1
0	1	1
1	1	0

Ausgang ist wahr (1), es sei denn beide Eingänge sind wahr.

NOR

Input A	Input B	Output X
0	0	1
1	0	0
0	1	0
1	1	0

Ausgang ist wahr (1), wenn beide Eingänge falsch sind.

XNOR

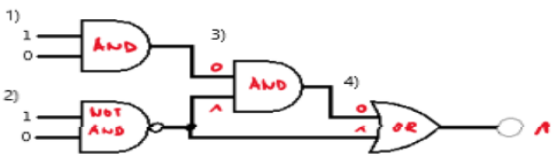
Input A	Input B	Output X
0	0	1
1	0	0
0	1	0
1	1	1

Ausgang ist wahr (1), wenn beide Eingänge gleich sind, entweder beide wahr oder beide falsch.

NOT

Input A	Output X
0	1
1	0

Ändert den Zustand des Eingangs; wenn Eingang wahr ist, wird der Ausgang falsch und umgekehrt.



4C

1)	EAN-Code:	2	2	1	1	5	6	4	5	6	6	6	6	
2)		2		1		5		4		6		6		24
3)			2		1		6		5		6		6	29
4)														24*3=72
5)														72+29=101
6)														110-101=1
	Code mit PZ	2	2	1	1	5	6	4	5	6	6	6	6	**9**

1) Ziffern notieren: Schreiben Sie die ersten 12 Ziffern des EAN-Codes auf. (Ohne die 13 Ziffer).
2) Addiere die Ziffern an den ungeraden Positionen
3) Addiere die Ziffern an den geraden Positionen
4) Multipliziere Summe aus 2) mit 3.
5) Addiere Ergebnis aus 4) und Summe 3).
6) Um die Prüfziffer zu ermitteln, berechnet man wie viel noch bis zur nächsten durch 10 teilbaren Zahl fehlt.

1B

Positionsbasierte: Die Position links/rechts gibt an, das eine Zahl ein vielfaches mehr «zählt», je nach Stelle.

Nicht positionsbasierte: Strichliste oder römischen Zahlen (Erweitert: warum verwenden diese so wenig: Versuche mal zwei römischen Zahlen zu addieren....)

Dezimal 10 Zahlen	Binär 2 Zahlen	Oktal 8 Zahlen	Hexadezimal 16 Zahlen
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13

Neben 1D Codes, welche mit Strichen umgesetzt sind, existieren auch 2D Codes welche mit Punkten dargestellt werden.

2A

$235 = 2*10^2 + 3*10^1 + 5*10^0$

Basis 10 durch 2 austauschen

$1110\ 1011 = 1*2^7 + 1*2^6 + 1*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 = 235$

235 : 2 = 117 Rest 1
117 : 2 = 58 Rest 1
58 : 2 = 29 Rest 0
29 : 2 = 14 Rest 1
14 : 2 = 7 Rest 0
7 : 2 = 3 Rest 1
3 : 2 = 1 Rest 1
1 : 2 = 0 Rest 1

→ 1110 1011

1. Schreibe die hexadezimale Zahl auf, und ordne den entsprechenden Dezimalwert jeder Hexadezimalziffer zu:
- 1A3 = (1 * 16^2) + (10 * 16^1) + (3 * 16^0)
2. Berechne die Werte für jede Potenz von 16:
- 16^2 = 256
- 16^1 = 16
- 16^0 = 1
3. Multipliziere die entsprechenden Ziffern mit ihren Potenzen von 16 und addiere die Ergebnisse:
- (1 * 256) + (10 * 16) + (3 * 1) = 256 + 160 + 3 = 419

- Beispielzahl: 11010011_2
- Zuerst wird die Zahl in Sequenzen aus 4 Bits aufgeteilt (sogenannte Nibbles)
- 1101 0011
- $1101_2 = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 13(D)$
- $0011_2 = 0 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 3$
- $11010011_2 = D3_{16}$

2)
1101
+ 101

10010

8)
35
+ 27

64

16)
A3
+ 2F

D2

Positive Zahl 5 in Binär: 0101
Einerkomplement von -5: 1010

Positive Zahl 5 in Binär: 0101
Zweierkomplement von -5: 1011

Offset Binary mit einem Offset von 3: 000 repräsentiert -3, 001 repräsentiert -2, ..., 111 repräsentiert 4.

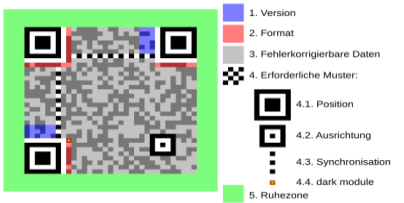
1.101 * 2^3 bedeutet 1.101 im Binär mit einem Exponenten von 3.

Um einen QR-Code zu generieren, braucht man:
-den Text, der codiert werden soll
-den gewünschten Grad der Fehlerkorrektur
Der grobe Ablauf ist dann:

1. Anhand der Länge des Textes und des Grades der Fehlerkorrektur bestimmt man, wie gross der QR-Code sein muss.
2. Man beginnt mit einer weissen Fläche, auf der nach und nach alle Elemente des QR-Codes dargestellt werden.
3. Die Erkennungsmuster, die nicht von dem Text abhängen, werden zuerst auf die Fläche gebracht. Das sind die Positionsmuster, die Ausrichtungsmuster und die Synchronisationslinien.
4. Aus dem Text wird eine Bitfolge generiert.
5. Zu der Text-Bitfolge wird eine weitere Bitfolge für die Fehlerkorrektur generiert.
6. Die Text-Bitfolge wird zusammen mit der Fehlerkorrektur-Bitfolge dort in das Symbol gezeichnet, wo noch Platz ist. Das geschieht von rechts nach links in Schlangenlinien.
7. Um zu erreichen, dass das Symbol ungefähr gleich viele schwarze und weisse Pixel enthält und um Muster zu vermeiden die das Einlesen erschweren, werden nacheinander acht verschiedene Masken über das Symbol gelegt. Die Maske, die das beste Ergebnis liefert, wird beibehalten.
8. Zum Schluss wird die Kennnummer der verwendeten Maske in das Symbol gezeichnet.

4D

1. **Versionsinformation:** Sagt aus, welche Version des QR-Codes verwendet wird.
2. **Datenformat:** Beschreibt, in welchem Format die Daten vorliegen.
3. **Datenteil:** Beinhaltet die eigentlichen Daten und Informationen für die Fehlerkorrektur.
4. **Orientierung und Ausrichtung:**
   - ***4.1. Eckenmuster:*** In drei von vier Ecken gibt es ein spezielles Muster, das hilft, den QR-Code zu erkennen und seine Orientierung zu bestimmen.
   - ***4.2. Zusatzmuster:*** Bei größeren QR-Codes kommen weitere Muster hinzu, die die Ausrichtung erleichtern.
   - ***4.3. Verbindungslinie:*** Zwischen den Hauptpositionsmarkierungen verläuft eine Linie, die aus abwechselnden Bits besteht. Sie definiert die Matrix des Codes.
   - ***4.4. Dunkles Modul:*** Ein Bit über der Format-Information, das immer dunkel ist und die dunkle Farbe des Codes repräsentiert.
5. ***Ruhezone:*** Ein weißer Rand um den Code herum, der ihn von anderen Informationen abgrenzt.



QR-Codes können im Gegensatz zu Strichcodes mehr Informationen enthalten und dürfen von der Dimension kleiner (und auch grösser) sein und können trotzdem gut gescannt werden.

Die möglichen Einsatzgebiete von QR-Codes sind enorm vielfältig. Anbei haben wir einige Beispiele aufgeführt:

- URLs z.B. Website Adressen aller Art (Landingpages, Downloads etc. Weiterleitungen z.B. dynamische QR-Codes)
- Produkteinformationen
- Anleitungen, Handbücher
- Musikstücke
- Und mehr...

6A

Verwendet denselben geheimen Schlüssel für Verschlüsselung und Entschlüsselung. Sender und Empfänger müssen den geheimen Schlüssel im Voraus teilen. Schnell und effizient, aber erfordert einen sicheren Schlüsselaustausch.

Stab-Chiffre (Skytale)

Die Stab-Chiffre ist eine relativ einfache Verschlüsselungsmethode, die zu den Substitutionstechniken gehört. Bei der Stab-Chiffre wird jede Buchstabe des Klartexts durch einen anderen Buchstaben ersetzt. Dieser Ersatz basiert auf einer festen Anzahl von Positionen, die jeden Buchstaben im Alphabet verschieben. Beispielsweise könnte der Buchstabe A durch den Buchstaben D ersetzt werden. Es handelt sich dabei um eine Form der Verschiebechiffre.

Ein Beispiel:

- Klartext: "VERSCHLUESSELUNG"
- Schlüssel: 3142
- Verschlüsselter Text: "VLSEUERSSNGHELC"

Ceasar-Chiffre

Die Cäsar-Chiffre ist eine spezielle Form der Stab-Chiffre und auch als Verschiebechiffre bekannt. Hierbei wird jeder Buchstabe im Klartext um eine feste Anzahl von Positionen im Alphabet verschoben. Der Schlüssel, der die Anzahl der Positionen angibt, wird oft als "Schlüssel" bezeichnet. Diese Chiffre ist einfach zu verstehen und zu implementieren, aber auch leicht zu brechen, da es nur 25 mögliche Schlüssel gibt. Zum Beispiel mit einer Verschiebung von 3:

- Klartext: "VERSCHLUESSELUNG"
- Verschlüsselter Text: "YHUFLJKXHVVDNHQJ"

Vignere-Chiffre

Die Vigenère-Chiffre ist eine verbesserte Version der Cäsar-Chiffre und fällt unter die polyalphabetischen Substitutionstechniken. Anstatt jeden Buchstaben um eine feste Anzahl von Positionen zu verschieben, verwendet die Vigenère-Chiffre einen Schlüsselwort, um die Verschiebung für jeden Buchstaben im Klartext anzugeben. Das Schlüsselwort wird wiederholt, um die Länge des Klartexts zu erreichen. Dies macht die Vigenère-Chiffre gegenüber der einfachen Cäsar-Chiffre widerstandsfähiger gegenüber Brute-Force-Angriffen.

Beispiel:

- Klartext: "VERSCHLUESSELUNG"
- Schlüssel: "KEY"
- Verschlüsselter Text: "VIRKBNYGOBKPWLVG"

Vernam-Chiffre

Die Vernam-Chiffre, auch als One-Time Pad bekannt, ist ein perfektes symmetrisches Verschlüsselungsverfahren, vorausgesetzt, der Schlüssel wird korrekt erzeugt und nur einmal verwendet. Jeder Buchstabe des Klartexts wird mit einem Buchstaben des zufällig generierten Schlüssels kombiniert. Da der Schlüssel so lang wie der Klartext ist und nur einmal verwendet wird, ist die Vernam-Chiffre gegenüber statistischen Angriffen und Brute-Force-Angriffen immun. Allerdings erfordert dies, dass beide Parteien den identischen Schlüssel im Voraus haben. Ein Beispiel mit einem Schlüsselstrom "K":

- Klartext: "VERSCHLUESSELUNG"
- Schlüsselstrom: "XYUJNSWHDCLTRQOL"
- Verschlüsselter Text: "ZFSJQSNFOTXVDFRJ"

7A

Symmetrische Verschlüsselung
DES (Data Encryption Standard)
Veralteter Verschlüsselungsalgorithmus mit 56-Bit-Schlüssel.
Blockverschlüsselung in 64-Bit-Blöcken.
AES (Advanced Encryption Standard)
Moderner Standard mit Schlüssellängen von 128, 192 und 256 Bit.
Blockverschlüsselung mit verschiedenen Rundenschlüsseln.

Asymmetrische Verschlüsselung
RSA (Rivest-Shamir-Adleman)
Basierend auf der Schwierigkeit des Faktorisierens großer Primzahlen.
Öffentlicher Schlüssel zum Verschlüsseln, privater Schlüssel zum Entschlüsseln.
Diffie-Hellman
Sicheres Protokoll für den Schlüsselaustausch. Parteien erzeugen gemeinsamen geheimen Schlüssel.
Hash-Verfahren (MD4, MD5, SHA)
Wandelt Daten in feste Hash-Werte um. MD4, MD5, SHA für Integritätsprüfungen und Signaturen.
Digitale Signaturen
Verwendet asymmetrische Verschlüsselung zur Authentifizierung. Sender verschlüsselt Hash-Wert der Daten mit privatem Schlüssel; Empfänger überprüft mit öffentlichem Schlüssel.

6B

Kerckhoffs Prinzip
Die Sicherheit hängt nicht davon ab, dass die Verschlüsselungsmethode geheim ist, sondern darauf, dass der Schlüssel geheim bleibt. Dadurch bleibt das System auch dann sicher wenn die Verschlüsselungsmethode bekannt wird.

Prinzip der asymmetrischen Verschlüsselung
Die asymmetrische Verschlüsselung, auch als Public-Key-Kryptographie bekannt, basiert auf der Verwendung von zwei Schlüsseln: einem öffentlichen Schlüssel (bekannt für alle), der zur Verschlüsselung verwendet wird, und einem privaten Schlüssel (geheim), der zur Entschlüsselung dient. Diese beiden Schlüssel sind mathematisch miteinander verknüpft, aber es ist praktisch unmöglich, den privaten Schlüssel aus dem öffentlichen Schlüssel abzuleiten. Das bedeutet, dass der öffentliche Schlüssel sicher verteilt werden kann, während der private Schlüssel geheim gehalten wird.
Das Verschlüsseln von Informationen erfolgt mit dem öffentlichen Schlüssel des Empfängers, und nur der Besitzer des zugehörigen privaten Schlüssels kann die verschlüsselten Daten entschlüsseln. Dies ermöglicht eine sichere Kommunikation, da der öffentliche Schlüssel nicht dazu verwendet werden kann, die Nachrichten zu entschlüsseln, die mit ihm verschlüsselt wurden.

- Modulo bestimmen
  - 37 (mod 5) = 2
  - 93 (mod 50) = 43
- Addition mit Modulo
  - 3+6 (mod 9) = 0
  - 17+23 (mod 9) = 4
- Subtraktion mit Modulo
  - 27–37 (mod 9) = 8
  - 8–9 (mod 7) = 6
- Multiplikation mit Modulo
  - 20 * 2 (mod 4) = 0
  - 3 * 7 (mod 6) = 3

7B

Die digitale Steganografie ist die Kunst der verborgenen Speicherung von Daten innerhalb von Daten. Sie bezeichnet Verfahren, die mithilfe von verschiedenen Techniken Daten in durch einen Computer zugänglichen Trägerdaten verbergen. Das Ziel der Steganografie besteht im Allgemeinen darin, Daten so gut zu verstecken, dass unbeabsichtigte Empfänger das steganografische Medium nicht verdächtigen, versteckte Daten zu enthalten. Hierbei wird das Ziel der Vertraulichkeit verfolgt wobei die glaubhafte Abstreitbarkeit hier eine weiter wichtige Rolle spielt.

Steganografie ist die Kunst, Informationen so zu verstecken, dass versteckte Botschaften nicht entdeckt werden können. Sie umfasst eine Vielzahl von geheimen Kommunikationsmethoden, die die Existenz der Nachricht selbst verbergen. Zu diesen Methoden gehören zum Beispiel digitale Signaturen. Steganografie und Kryptografie sind Verwandte in der Familie der Spionage-Kunsthandwerk: Die Kryptografie verschlüsselt eine Nachricht, so dass sie nicht verstanden werden kann, während die Steganografie die Nachricht versteckt, so dass sie nicht gesehen werden kann. Wenn eine verschlüsselte Nachricht abgefangen wird, weiß der Abfangjäger, dass der Text eine verschlüsselte Nachricht ist. Bei der Steganografie weiß der Abfangjäger jedoch möglicherweise nicht, dass eine versteckte Nachricht überhaupt existiert. Bei dem Begriff Steganografie geht es im Gegensatz zur Kryptologie, welches einen Oberbegriff für die zwei Disziplinen Kryptografie & Kryptoanalyse ist, nicht um die Wissenschaft der Verschlüsselung oder Entschlüsselung sondern um das Verbergen der Tatsache, dass überhaupt eine Botschaft übermittelt wird.

Trägerdaten: Bilddaten, Audiodaten, Textdaten, Dateisystem-Fragmentierung.
Bilder sind ein gutes Mittel, um Daten zu verbergen. Je detaillierter ein Bild ist, desto weniger Einschränkungen gibt es in Bezug darauf, wie viele Daten es verstecken kann, bevor es verdächtig wird. Neben Bilddateien als Träger gibt es jedoch auch weitere Modifikationsmöglichkeiten von verschiedenen anderen Trägerdaten wie zum Beispiel die Audiodatei oder die Fragmentierung eines Dateisystems. Das menschliche Auge reagiert gegenüber eines Bildrauschen erheblich unempfindlicher als z.B. gegenüber dem Audiorauschen. Das bedeutet, dass ein Bild stark beeinträchtigt werden kann, bevor die Veränderung als Störung wahrgenommen wird, somit ist die Steganografie auf Bilddaten verhältnismäßig einfacher.

7C

2. Generiere einen Schlüsselpaar:
•Öffne das Programm "Kleopatra", das mit GPG4Win geliefert wird.
•Wähle "Datei" > "Neuer Schlüssel" und folge den Anweisungen, um ein Schlüsselpaar zu generieren. Dies besteht aus einem öffentlichen und einem privaten Schlüssel.
3. Exportiere deinen öffentlichen Schlüssel:
•Wähle deinen erstellten Schlüssel in Kleopatra aus.
•Klicke mit der rechten Maustaste darauf und wähle "Exportieren". Speichere den öffentlichen Schlüssel an einem sicheren Ort.
4. Importiere den öffentlichen Schlüssel des Empfängers:
•Wenn du die E-Mail oder Datei für jemand anderen verschlüsseln möchtest, benötigst du den öffentlichen Schlüssel des Empfängers. Importiere diesen in Kleopatra.
5. Verschlüsselung von E-Mails:
•Verfasse deine E-Mail in deinem bevorzugten E-Mail-Programm (wie Outlook oder Thunderbird).
•Wähle die Option zur Verschlüsselung und signiere deine E-Mail. Das Verschlüsselungssymbol sollte erscheinen.
6. Verschlüsselung von Dateien:
•Klicke mit der rechten Maustaste auf die Datei, die du verschlüsseln möchtest.
•Wähle "Verschlüsseln" und wähle den Empfänger (falls du die Datei mit jemand anderem teilen möchtest).
7. Entschlüsselung von E-Mails und Dateien:
•Der Empfänger kann die verschlüsselte E-Mail oder Datei mit seinem privaten Schlüssel entschlüsseln, den er sicher aufbewahrt.
Beachte, dass sowohl Sender als auch Empfänger über GPG4Win und einen Schlüsselpaar verfügen müssen, um die Verschlüsselung erfolgreich durchzuführen. Es ist auch wichtig, die privaten Schlüssel sicher zu verwahren.