

# Algorithmen 1

Informatik 1M

# Was ist ein Algorithmus?

- **Definition/Charakterisierung: Ein Algorithmus...**
  1. ...ist eine **Abfolge eindeutiger Handlungsanweisung** für die Lösung von Problemen.
    - *eindeutig* bedeutet, dass jeder Schritt zu 100% klar ist und keinen Interpretationsspielraum lässt.
  2. ...verarbeitet eine bestimmte **Eingabe (Input) Schritt für Schritt in eine bestimmte Ausgabe (Output)**.
  3. ...kann in **verschiedenen Sprachen** formuliert werden.  
(Menschliche Sprachen oder Programmiersprachen).
- **Analogie: Ein Algorithmus ist wie ein (sehr klares) Kochrezept.**

# Beispiel "Caramel-Bonbon"

- **Rezept in menschlicher Sprache:**

100 g Zucker und 2 EL Wasser in Pfanne geben. Bei mittlerer bis hoher Hitze unter ständigem Rühren erwärmen, bis der Zucker goldbraun ist. Das Caramel auf geölte Alufolie geben und auskühlen lassen.

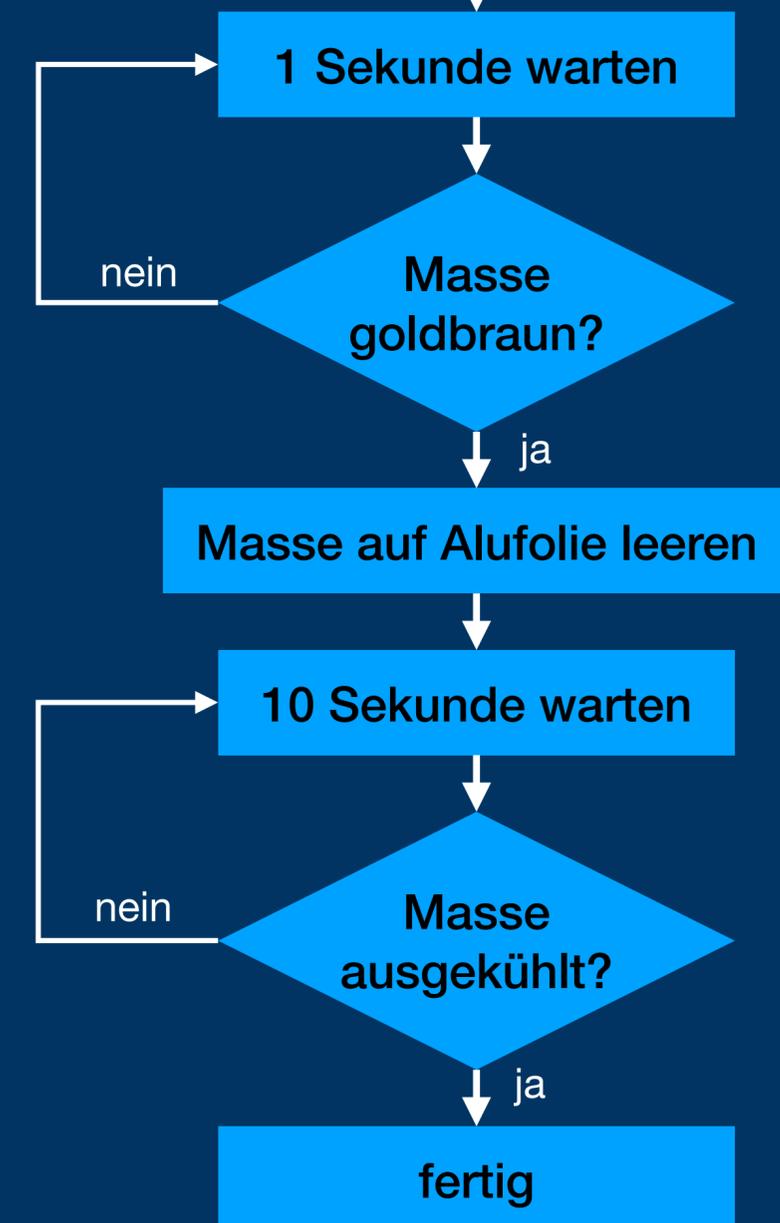
- **Rezept in Pseudocode:**

```
1.pfanne.input(sugar(100,g),water(2,es))
2.platte.on(.75)
3.while not(topf.inhalt.getcolor() == "goldbraun")
4.    warte(1000,ms)
5.alufolie.input(topf.inhalt ())
6.platte.off()
7.while not(alufolie.inhalt.gettempdeg() < 20)
8.    warte(10000,ms)
```

- **Rezept als Flussdiagramm:**

100 g Zucker und 2 EL Wasser in Pfanne geben.

Pfanne auf Herd stellen: Mittlere bis hohe Hitze.



# Beispiele von Algorithmen in der Informatik

- **Prüfen, ob x eine Primzahl ist**

**Eingabe:** Irgendeine ganze Zahl.

**Ausgabe:** Resultat (ja/nein; true/false; 1/0).

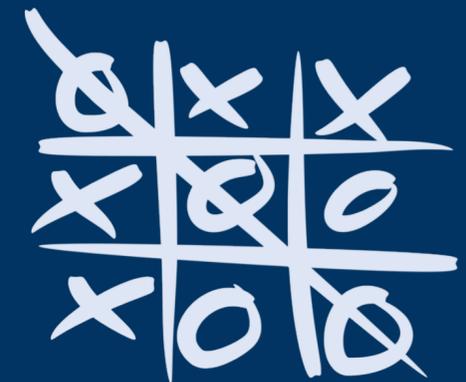
Solche Algorithmen werden z.B. in Verschlüsselungsverfahren verwendet.

- **Einfache Spiele wie TicTacToe**

Mensch gegen Computer: Computer spielt nach einem Algorithmus:

**Eingabe:** Spielzüge des Menschen.

**Ausgabe:** Spielzüge des Computers



- **Liste mit Zahlen sortieren**

**Eingabe:** [42, 3, 7]

**Ausgabe:** [3, 7, 42]

# Aufgabe A1: Wasserhahn (2er-Gruppen)

- **Ablauf:**

1. Jede Person nimmt ein **Blatt Papier** (ja Papier, nicht Laptop).
2. **Notiere für Partner:in in normaler Sprache einen Algorithmus**, der sie von der aktuellen Position (wahrscheinlich am Pult sitzend) zum Wasserhahn im Schulzimmer bringt.
3. Nimm den Algorithmus deiner Partner:in und befolge diesen **genau so, wie er da steht**. Führt dich der Algorithmus zum Ziel?

- **Beachtet die “Definition” des Algorithmus:**

- “...ist eine Abfolge eindeutiger Handlungsanweisung für die **Lösung von Problemen.**” —> **Man muss zum Ziel kommen!**
- “*eindeutig* bedeutet [...] keinen Interpretationsspielraum...”
  - “Drehe dich nach links” ist nicht eindeutig
  - “Drehe dich um 90 Grad nach links” ist eindeutig

- **Zusatzaufgabe:** Der Algorithmus soll von *jedem beliebigen Startpunkt* im Schulzimmer funktionieren. Passe deinen Algorithmus entsprechend an.

## Regeln

- Person kann nur gerade aus und 50 cm weit sehen. „**Schau wo der Wasserhahn ist und geh' dorthin**“ geht also nicht!
- Person hat kein Gefühl für Distanzen > 50 cm. „**Laufe 2 m nach vorne**“ ist also unzulässig.
- Person hat Gefühl für Winkel (weiss also z.B. was ein 90°-Winkel ist).
- Augen sind einzige Sensoren (Hände usw. dürfen nicht verwendet werden z.B. um Wand abzutasten).
- Tische sind wie Wände, werden von den Sensoren (Augen) also erkannt.

# Aufgabe A1: Wasserhahn (2er-Gruppen)

- **Mögliche Lösung:**

1. Stehe auf.

2. Drehe dich um 90 Grad nach rechts.

3. Laufe 50 cm vorwärts.

4. Hast du eine Wand vor dir?

1. ja: Drehe dich um 90 Grad nach links

2. nein: —> gehe zu Schritt 3.

5. Laufe 50 cm vorwärts.

6. Siehst du den Wasserhahn?

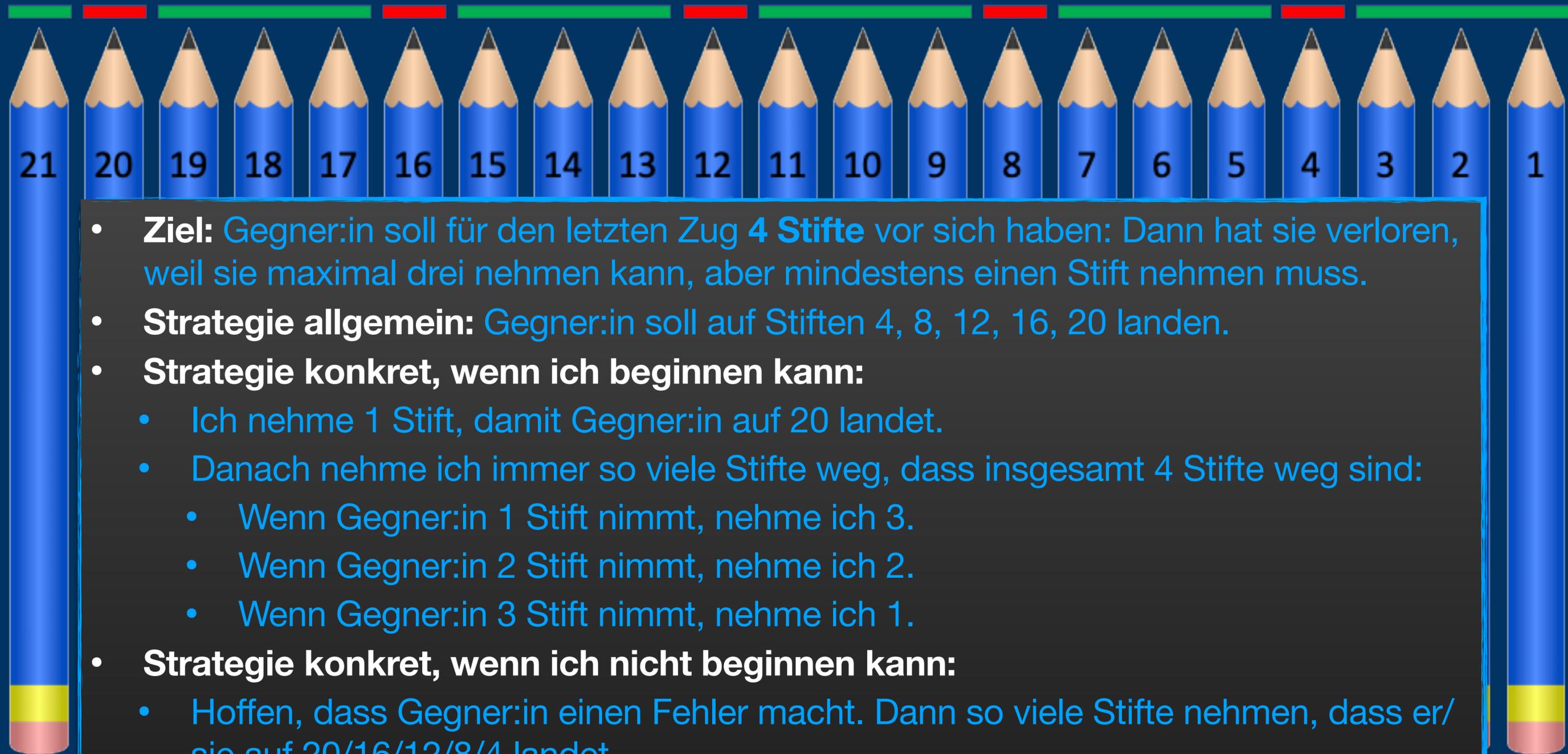
1. ja: DU BIST AM ZIEL

2. nein: Gehe zu Schritt 5.

# Aufgabe A2: Subtraction Game

- **Siehe Wiki**
- **Ablauf (2-er Gruppen)**
  1. Spielt das Spiel mit euren eigenen Stiften mehrere Male gegeneinander und **findet die optimale Spielstrategie.**
  2. **Formuliert einen Algorithmus in menschlicher Sprache, der diese Strategie umsetzt.**

# Subtraction Game – Lösung



- **Ziel:** Gegner:in soll für den letzten Zug **4 Stifte** vor sich haben: Dann hat sie verloren, weil sie maximal drei nehmen kann, aber mindestens einen Stift nehmen muss.
- **Strategie allgemein:** Gegner:in soll auf Stiften 4, 8, 12, 16, 20 landen.
- **Strategie konkret, wenn ich beginnen kann:**
  - Ich nehme 1 Stift, damit Gegner:in auf 20 landet.
  - Danach nehme ich immer so viele Stifte weg, dass insgesamt 4 Stifte weg sind:
    - Wenn Gegner:in 1 Stift nimmt, nehme ich 3.
    - Wenn Gegner:in 2 Stifte nimmt, nehme ich 2.
    - Wenn Gegner:in 3 Stifte nimmt, nehme ich 1.
- **Strategie konkret, wenn ich nicht beginnen kann:**
  - Hoffen, dass Gegner:in einen Fehler macht. Dann so viele Stifte nehmen, dass er/sie auf 20/16/12/8/4 landet.

# Struktogramme

# Erinnerung: Was ist ein Algorithmus?

- **Definition/Charakterisierung:**
  1. Ein Algorithmus ist eine **Abfolge eindeutiger Handlungsanweisung** für die Lösung von Problemen.
    - *eindeutig* bedeutet, dass jeder Schritt zu 100% klar ist und keinen Interpretationsspielraum lässt.
  2. Ein Algorithmus verarbeitet eine bestimmte **Eingabe (Input) Schritt für Schritt in eine bestimmte Ausgabe (Output)**.
  3. Ein Algorithmus kann in **verschiedenen Sprachen** formuliert werden. (Menschliche Sprachen oder Programmiersprachen).
- **Struktogramm: Ähnlich wie Flussdiagramm, aber kompakter und näher an der Struktur von Programmiersprachen...**

# Arten von Elementen in Struktogrammen

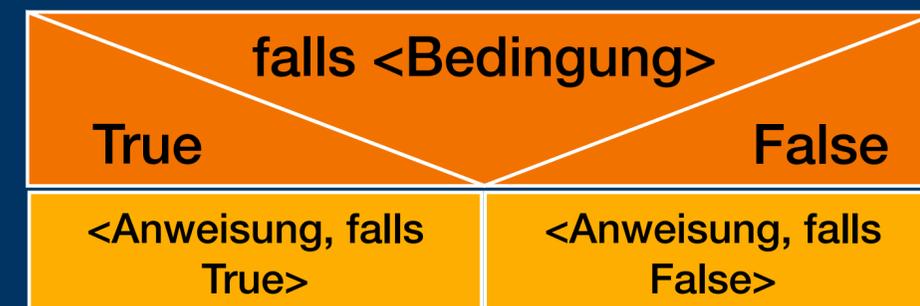
## 1. Einfache Anweisung



## 2. Schleifen



## 3. Verzweigung



# Beispiel Wasserhahn

## 1. Einfache Anweisung

Stehe auf

Gehe 0.5 m nach vorne

Drehe dich um 90 ° nach links

## 2. Schleifen

Solange keine Wand vor dir

Gehe 0.5 m nach vorne

## 3. Verzweigung

Falls Wasserhahn

True

vor dir

False

Du bist am Ziel

Gehe 0.5 m nach  
vorne

# Aufgaben B

- Siehe Wiki
- **Tipp für Aufgabe B3-1:**
  - An der Stelle, an der die Spieler:in am Zug ist: Verwende eine Anweisung wie "*Spieler:in nimmt 1-3 Stifte*" oder "*Spieler:in nimmt x Stifte*".
- **Tipps für Aufgabe B3-2 (Challenge):**
  - An der Stelle, an der die Spieler:in am Zug ist: Verwende die Anweisung: "*Spieler:in nimmt x Stifte, es bleiben N Stifte*".
  - Verwende die Anweisung: "*r = Rest der Ganzzahldivision N / 4*".
  - Falls  $r == 0$  ist: Verwende die Anweisung "Nimm random 1-3 Stifte. Es bleiben N Stifte".
  - Ansonsten soll der Computer  $r$  Stifte nehmen, damit die Spieler:in auf 4/8/12/16 landet.

# Aufgaben B – Lösungen

- **Beispiel-Lösung B1 (Wasserhahn revisited)**

Stehe auf

Drehe dich um 90 Grad nach rechts

Solange keine Wand vor dir ...

Gehe 0.5m nach vorne

Solange Wasserhahn nicht vor dir

Drehe dich um 90 Grad nach links

Gehe 0.5m nach vorne

Drehe dich um 90 Grad nach rechts

Falls Wasserhahn nicht  
läuft

True

False

Wasserhahn öffnen

Am Wasserhahn trinken

Wasserhahn schliessen

# Aufgaben B – Lösungen

- Lösung B2 (Karamell-Bonbons revisited)

Alufolie auf Kuchenblech legen, ölen.

Edelstahlpfanne mit 100g Zucker u. 2 EL  
Wasser auf Herd stellen

Temperatur auf mittlere-hohe Hitze  
(Stufe 3 von 4)

Solange Topfinhalt nicht goldbraun

1s warten

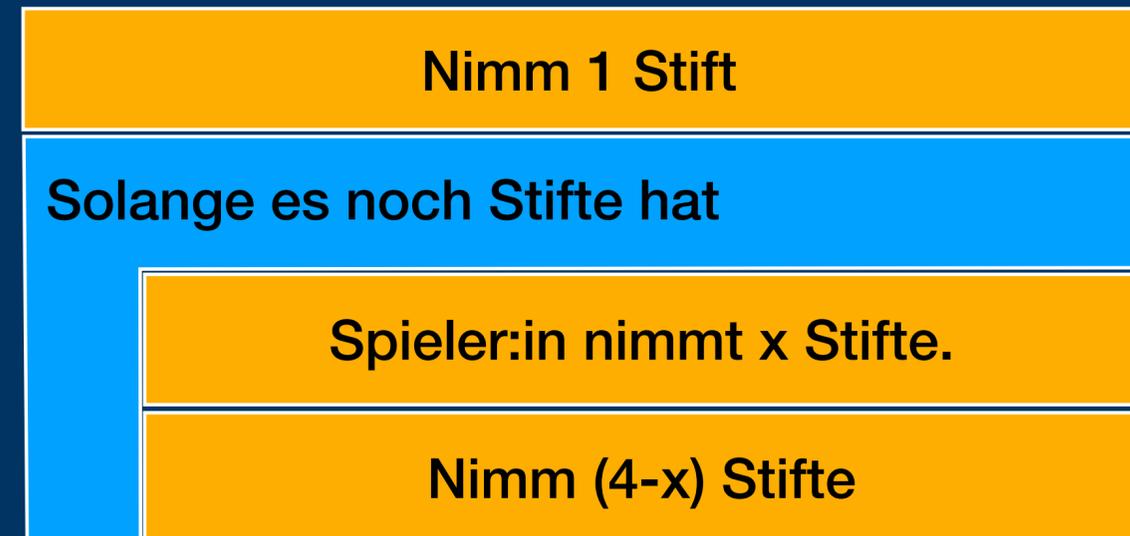
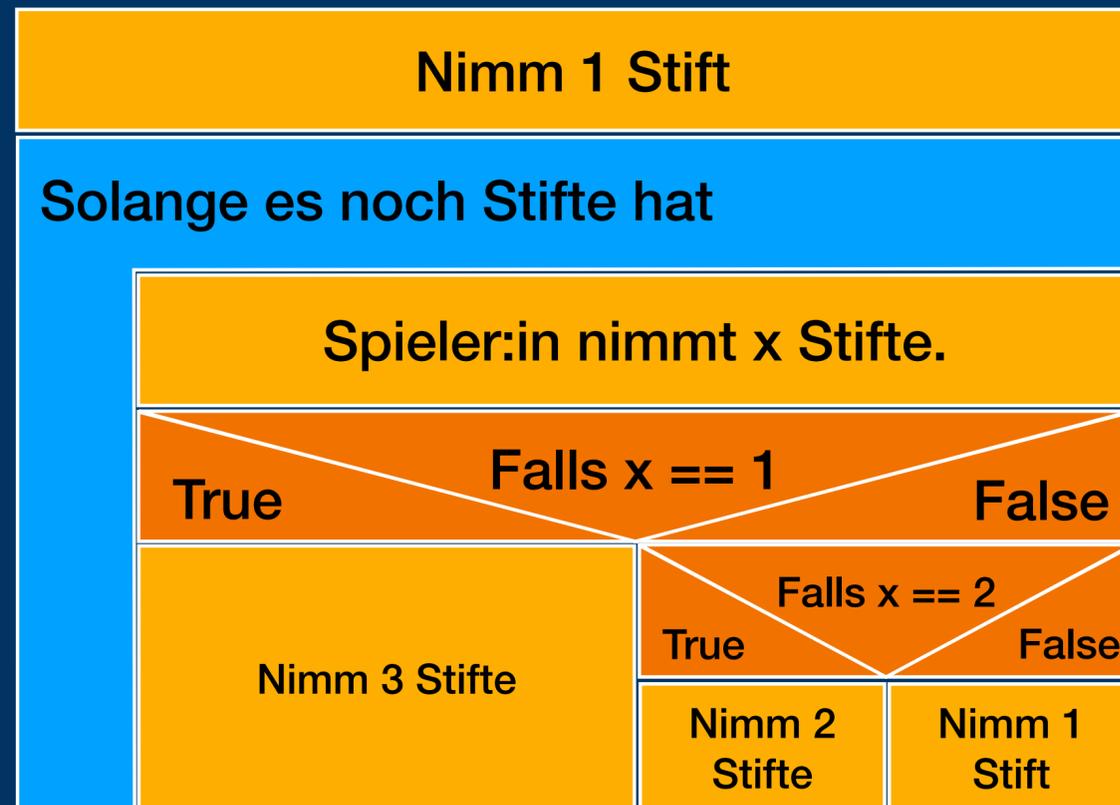
Topfinhalt auf Alufolie leeren, Herd  
ausschalten

Solange Masse nicht ausgekühlt

10s warten

# Aufgaben B – Lösungen

- Lösung B3 – 1: **Subtraction Game**, Computer beginnt (und gewinnt)
- Lösungsvariante 1:
- Lösungsvariante 2:



# Aufgaben B – Lösungen

- Lösungen B3-2 und B3-3: [Siehe Wiki](#)

# Eingabe und Ausgabe in Struktogrammen

# Variablen, Eingabe und Ausgabe

## 1. Variablen...

- sind wie Behälter (Speicher), in denen Zahlen, Buchstaben oder Texte gespeichert werden.
- können irgendeinen Namen haben, z.B.: 'x', 'i', 'meine\_variable', 'zahl\_1', 'input1', 'num\_of\_pens\_left' etc. (keine Abstände).
- '=' ist ein Zuweisungsoperator; damit kann ich einer Variable einen Wert zuweisen)
- ('==' ist ein Vergleichsoperator; damit kann ich eine Variable mit einem bestimmten Wert vergleichen)

i = 42

True Falls x == 1 False

## 2. Eingabe:

- Runde Klammer gibt Art des Inputs an (Zahl, Buchstabe, Text)

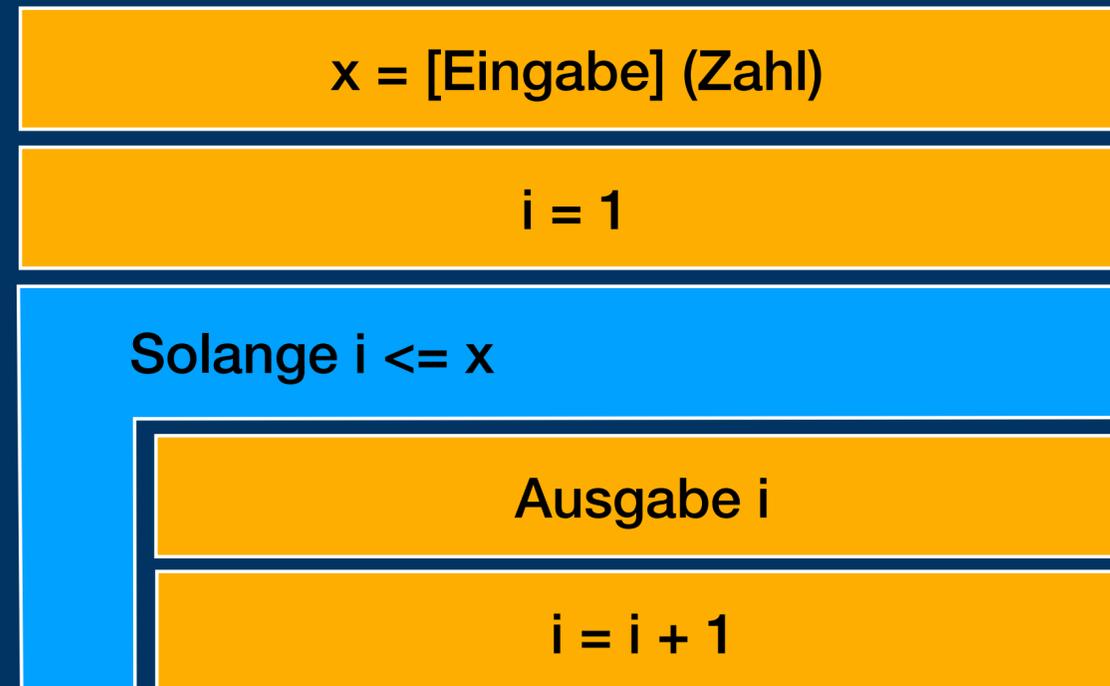
x = [Eingabe] (Zahl)

## 3. Ausgabe:

Ausgabe y

# Beispiel Zählen

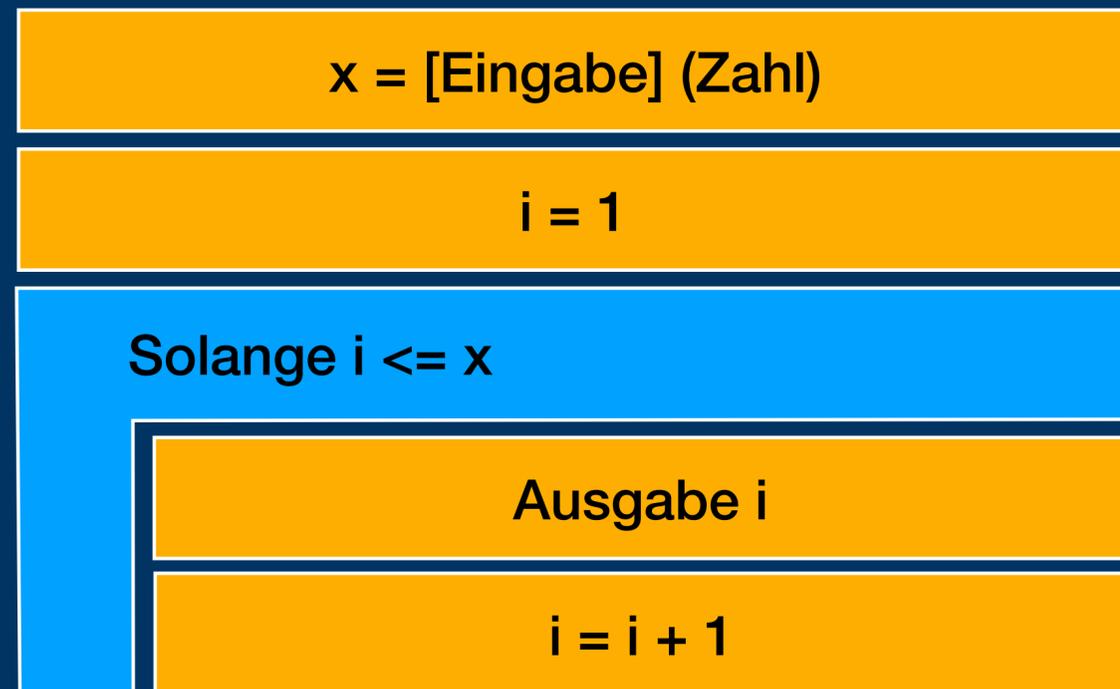
- Zahl eingeben (z.B. 3)
- Algorithmus zählt von 1 auf diese Zahl hoch und gibt alle Zahlen aus.



' $\leq$ ' ist (wie ' $==$ ') ein Vergleichsoperator: Er vergleicht, ob der linke Wert kleiner oder gleich gross ist, wie der rechte Wert.

# Evaluieren eines Algorithmus

1. Struktogramm für ein Konkretes Beispiel (z.B. Eingabe = 3) Schritt für Schritt durchgehen und dabei
2. alle Werte in einer Tabelle notieren.
  - siehe Regeln auf Wiki



Zeile	Bedingungen	x	i	Ausgabe
1		3		
2			1	
3	$i \leq x?$ True			
4				1
5			2	
6	$i \leq x?$ True			
7				2
8			3	
9	$i \leq x?$ True			
10				3
11			4	
12	$i \leq x?$ False			

# Aufgaben C

- Siehe [Wiki](#)