

Was ist ein Algorithmus?



MATHEMATIK  
macht  
FREU(N)DE

Als Kind hast du ein Verfahren gelernt, um jede natürliche Zahl als Produkt von Primfaktoren zu schreiben. Zum Beispiel:  $84 = 2 \cdot 2 \cdot 3 \cdot 7$

- 1) Dividiere so oft wie möglich durch 2 ohne Rest.
- 2) Dividiere so oft wie möglich durch 3 ohne Rest.
- 3) Dividiere so oft wie möglich durch 5 ohne Rest.
- 4) Setze mit den weiteren Primzahlen fort, bis das Ergebnis 1 ist.

84	2
42	2
21	3
7	7
1	

$\rightarrow 84 = 2 \cdot 2 \cdot 3 \cdot 7$

„Ein **Algorithmus** ist [...] eine wohldefinierte **Rechenvorschrift**, die eine Größe oder eine Menge von Größen als **Eingabe** verwendet und eine Größe oder eine Menge von Größen als **Ausgabe** erzeugt. Somit ist ein Algorithmus eine **Folge von Rechenschritten**, die die **Eingabe** in die **Ausgabe** umwandeln.“

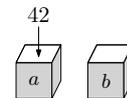
Quelle: Algorithmen - Eine Einführung, Cormen, Thomas H. / Leiserson, Charles E. / Rivest, Ronald / Stein, Clifford

Variablen & Wertzuweisungen



MATHEMATIK  
macht  
FREU(N)DE

Mit **Variablen** können wir Werte abspeichern. Dafür verwenden wir die folgende Schreibweise:



- |                                 |   |
|---------------------------------|---|
| 1: $a \leftarrow 42$            | In Zeile 1 erhält die Variable $a$ den Wert 42.                               |
| 2: $b \leftarrow a$             | In Zeile 2 erhält die Variable $b$ den aktuellen Wert von $a$ .               |
| 3: $a \leftarrow a + 2$         | In Zeile 3 erhält die Variable $a$ den aktuellen Wert von $a$ plus 2.         |
| 4: $b \leftarrow \frac{a+b}{2}$ | In Zeile 4 erhält die Variable $b$ das arithmetische Mittel von $a$ und $b$ . |

Zeile	1	2	3	4
$a$	42	42	44	44
$b$	-	42	42	43

In der Tabelle links siehst du welchen Wert die Variablen  $a$  und  $b$  in jeder Zeile nach Durchführung der Wertzuweisung haben.

Variablen & Wertzuweisungen



MATHEMATIK  
macht  
FREU(N)DE

Trage in die Tabelle jene Werte ein, die die Variablen jeweils nach Durchführung der Zeile haben.

- 1:  $x \leftarrow 23$
- 2:  $x \leftarrow x - 5$
- 3:  $y \leftarrow \frac{x}{2}$
- 4:  $x \leftarrow y - x$
- 5:  $y \leftarrow -x - 3$

Zeile	1	2	3	4	5
$x$	23	18	18	-9	-9
$y$	-	-	9	9	6

if-Abfrage



MATHEMATIK  
macht  
FREU(N)DE

In der Praxis hängen unsere Entscheidungen von der aktuellen Situation ab.

*Falls* es kalt ist, ziehe ich mich warm an. *Falls* es regnet, nehme ich einen Regenschirm.

Bei Algorithmen heißen solche bedingten Entscheidungen **if-Abfragen**. Zum Beispiel:

- |                                      |  |
|--------------------------------------|--|
| 1: $T \leftarrow 8$                  | In Zeile 3 wird überprüft, ob der Wert von $T$ höchstens 10 ist.   |
| 2: $K \leftarrow 5$                  | Falls diese Bedingung erfüllt ist, werden die eingerückten Befehle bis zum <b>end if</b> durchgeführt.               |
| 3: <b>if</b> $T \leq 10$ <b>then</b> | Falls sie <i>nicht</i> erfüllt ist, werden die eingerückten Befehle bis zum <b>end if</b> <i>nicht</i> durchgeführt. |
| 4: $K \leftarrow 10$                 |  |
| 5: $T \leftarrow T + 4$              |  |
| 6: <b>end if</b>                     |  |
| 7: $T \leftarrow 20$                 |  |

Zeile	1	2	3	4	5	6	7
$T$	8	8	8	8	12	12	20
$K$	-	5	5	10	10	10	10

Trage in die Tabelle jene Werte ein, die die Variablen jeweils nach Durchführung der Zeile haben.

```

1: x ← -5
2: y ← 3
3: if x · y < 0 then
4:   x ← x · y
5:   y ← x
6: end if
7: if x ≠ y then
8:   x ← 42
9: end if
    
```

Zeile	1	2	3	4	5	6	7	8	9
x	-5	-5	-5	-15	-15	-15	-15	-15	-15
y	-	3	3	3	-15	-15	-15	-15	-15

Bei einer Schularbeit hängt die Note von den erreichten Punkten ab.

Ein Programm soll die Punkteanzahl  $P$  in die entsprechende Ziffernote  $N$  (1, 2, 3, 4 oder 5) übersetzen.

Punkte	Note
30 oder mehr	Sehr gut
[26; 30[	Gut
[21; 26[	Befriedigend
[16; 21[	Genügend
weniger als 16	Nicht genügend

Wir können die Übersetzung mit fünf if-Abfragen lösen:

```

1: if P ≥ 30 then
2:   N ← 1
3: end if
4: if 26 ≤ P < 30 then
5:   N ← 2
6: end if
7: if 21 ≤ P < 26 then
8:   N ← 3
9: end if
10: if 16 ≤ P < 21 then
11:   N ← 4
12: end if
13: if P < 16 then
14:   N ← 5
15: end if
    
```

Es reicht aber auch eine einzige if – else if – else-Abfrage:

```

1: if P ≥ 30 then
2:   N ← 1
3: else if P ≥ 26 then
4:   N ← 2
5: else if P ≥ 21 then
6:   N ← 3
7: else if P ≥ 16 then
8:   N ← 4
9: else
10:  N ← 5
11: end if
    
```

Bei einer if – else if – else-Abfrage werden die Bedingungen von oben nach unten überprüft, bis zum *ersten Mal* eine Bedingung erfüllt ist.

Dann werden *ausschließlich* die Befehle von dieser *ersten* erfüllten Bedingung durchgeführt. Alle weiteren Befehle bis zum **end if** werden übersprungen.

Zeile 3 im Programm wird also nur dann erreicht, falls  $P \geq 30$  *nicht* gilt. Dann muss stattdessen  $P < 30$  gelten. Deshalb reicht es in Zeile 3 die Bedingung  $P \geq 26$  abzufragen.

Falls *keine einzige* Bedingung erfüllt ist, werden die Befehle von **else** durchgeführt.

1) Der Wert, den die Variable  $s$  nach der folgenden if – else if – else-Abfrage hat, hängt von den Werten der beiden Variablen  $a$  und  $b$  ab. Vervollständige die Tabelle:

```

1: if a · b > 0 then
2:   s ← 1
3: else if a · b < 0 then
4:   s ← -1
5: else
6:   s ← 0
7: end if
    
```

a	3	5	-4	2	0	-3
b	4	-2	-1	0	0	1
s	1	-1	1	0	0	-1

2) Wie kannst du den Wert von  $s$  unmittelbar erkennen, ohne  $a \cdot b$  zu berechnen?

- $s = 1 \iff a$  und  $b$  haben das gleiche Vorzeichen.
- $s = -1 \iff a$  und  $b$  haben verschiedene Vorzeichen.
- $s = 0 \iff a = 0$  und/oder  $b = 0$ .

for - Schleife



MATHEMATIK  
macht  
FREU(N)DE

- 1:  $a \leftarrow 1$
- 2:  $b \leftarrow 1$
- 3:  $a \leftarrow a + b$
- 4:  $b \leftarrow a + b$
- 5:  $a \leftarrow a + b$
- 6:  $b \leftarrow a + b$
- 7:  $a \leftarrow a + b$
- 8:  $b \leftarrow a + b$

Trage in die Tabelle jene Werte ein, die die Variablen jeweils nach Durchführung der Zeile haben.

Zeile	1	2	3	4	5	6	7	8
$a$	1	1	2	2	5	5	13	13
$b$	-	1	1	3	3	8	8	21

Bei diesem Programm werden die Befehle  $a \leftarrow a + b$ ,  $b \leftarrow a + b$  dreimal hintereinander ausgeführt.

Mit einer **for**-Schleife können wir das gleiche Programm kürzer anschreiben:

- 1:  $a \leftarrow 1$
- 2:  $b \leftarrow 1$
- 3: **for**  $i \leftarrow 1$  **to** 3 **do**
- 4:      $a \leftarrow a + b$
- 5:      $b \leftarrow a + b$
- 6: **end for**

In Zeile 3 wird die Zählvariable  $i$  definiert.

$i$  erhält den Wert 1. Dann werden die Befehle bis zum **end for** werden ausgeführt.

$i$  erhält den Wert 2. Dann werden die Befehle bis zum **end for** werden ausgeführt.

$i$  erhält den Wert 3. Dann werden die Befehle bis zum **end for** werden ausgeführt.

Zeile	1	2	3	4	5	6	3	4	5	6	3	4	5	6
$a$	1	1	1	2	2	2	2	5	5	5	5	13	13	13
$b$	-	1	1	1	3	3	3	3	8	8	8	8	21	21
$i$	-	-	1	1	1	1	2	2	2	2	3	3	3	3

for - Schleife



MATHEMATIK  
macht  
FREU(N)DE

Welchen Wert hat die Variable  $a$  schließlich?

- a) 1:  $a \leftarrow 4$   
 2: **for**  $i \leftarrow 1$  **to** 5 **do**  
 3:      $a \leftarrow a + 2$   
 4: **end for**

$$a = 4 + 5 \cdot 2 = 14$$

- b) 1:  $a \leftarrow 25$   
 2: **for**  $i \leftarrow 3$  **to** 9 **do**  
 3:      $a \leftarrow a - 3$   
 4: **end for**

$$a = 25 - 7 \cdot 3 = 4$$

- c) 1:  $a \leftarrow 3$   
 2: **for**  $i \leftarrow 1$  **to** 10 **do**  
 3:      $a \leftarrow a \cdot 2$   
 4: **end for**

$$a = 3 \cdot 2^{10} = 3072$$

for - Schleife



MATHEMATIK  
macht  
FREU(N)DE

Fülle jene Zahl in die Lücke, damit die Variable  $a$  schließlich den Wert 42 hat.

- a) 1:  $a \leftarrow 63$   
 2: **for**  $i \leftarrow 1$  **to** 7 **do**  
 3:      $a \leftarrow a - 3$   
 4: **end for**

$$x - 7 \cdot 3 = 42 \implies x = 63$$

- b) 1:  $a \leftarrow 6$   
 2: **for**  $i \leftarrow 1$  **to** 9 **do**  
 3:      $a \leftarrow a + 4$   
 4: **end for**

$$6 + x \cdot 4 = 42 \implies x = 9$$

- c) 1:  $a \leftarrow 72$   
 2: **for**  $i \leftarrow 4$  **to** 13 **do**  
 3:      $a \leftarrow a - 3$   
 4: **end for**

$$72 - 10 \cdot x = 42 \implies x = 3$$

for - Schleife



MATHEMATIK  
macht  
FREU(N)DE

Das folgende Programm verwendet in einer **for**-Schleife den Wert der Zählvariable  $i$ .

- 1:  $s \leftarrow 0$
- 2: **for**  $i \leftarrow 1$  **to** 9 **do**
- 3:      $s \leftarrow s + i$
- 4: **end for**

Wir dürfen  $i$  in der Schleife aber *nicht* neue Werte zuweisen.

Welchen Wert hat die Variable  $s$  nach Durchführung der **for**-Schleife?

$$0 + 1 + 2 + 3 + \dots + 9 = \frac{9 \cdot 10}{2} = 45$$

repeat until - Schleife



MATHEMATIK  
macht  
FREU(N)DE

Bei einer **for**-Schleife legt man *zu Beginn* fest, wie oft die Schleife ausgeführt werden soll.  
Bei einer **repeat until**-Schleife legt man stattdessen *am Ende* eine Abbruchbedingung fest.  
Die Schleife wird *so lange wiederholt, bis* die Abbruchbedingung erfüllt ist.

repeat until

- 1:  $a \leftarrow 1$
- 2: **repeat**
- 3:      $a \leftarrow 2 \cdot a$
- 4: **until**  $a > 100$

Nach Durchlauf	1	2	3	4	5	6	7
$a$	2	4	8	16	32	64	128

Nach 7 Durchläufen ist die Abbruchbedingung  $a > 100$  erstmals erfüllt.  
Die **repeat until**-Schleife bricht also ab. Die Variable  $a$  hat danach den Wert 128.

repeat until - Schleife



MATHEMATIK  
macht  
FREU(N)DE

Wie oft wird die **repeat until**-Schleife durchgeführt? Welchen Wert hat die Variable  $a$  schließlich?

- |   |   |
|---|---|
| <ol style="list-style-type: none"> <li>a) 1: <math>a \leftarrow 13</math></li> <li>2: <b>repeat</b></li> <li>3:     <math>a \leftarrow a - 2</math></li> <li>4: <b>until</b> <math>a \leq 4</math></li> </ol> | <ol style="list-style-type: none"> <li>b) 1: <math>a \leftarrow 5</math></li> <li>2: <b>repeat</b></li> <li>3:     <math>a \leftarrow 3 \cdot a</math></li> <li>4: <b>until</b> <math>a &gt; 2306</math></li> </ol> |
|---|---|

$$13 - 2 \cdot x \leq 4 \implies x \geq 4,5$$

5 Durchführungen,  $a = 13 - 2 \cdot 5 = 3$

$$5 \cdot 3^x > 2306 \implies x > \log_3 \left( \frac{2306}{5} \right)$$

$= 5,58\dots$

6 Durchführungen,  $a = 5 \cdot 3^6 = 3645$

Endlosschleife



MATHEMATIK  
macht  
FREU(N)DE

Was passiert bei der folgenden **repeat until**-Schleife?

- 1:  $a \leftarrow 1$
- 2: **repeat**
- 3:      $a \leftarrow a + 2$
- 4: **until**  $a = 10$

Die Abbruchbedingung wird *nie* erfüllt.  
( $1 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 11 \rightarrow 13 \rightarrow \dots$ )  
Die Schleife wird also *nie* abgebrochen. („Endlosschleife“)

while - Schleife



MATHEMATIK  
macht  
FREU(N)DE

Bei einer **while**-Schleife legt man *zu Beginn* der Schleife eine Bedingung fest.  
*Solange* diese Bedingung erfüllt ist, wird die Schleife durchgeführt.

while

- 1:  $a \leftarrow 2$
- 2: **while**  $a \leq 16$  **do**
- 3:      $a \leftarrow a \cdot a$
- 4: **end while**

Vor Durchlauf	1	2	3	4
$a$	2	4	16	256

Vor dem 4. Durchlauf ist die Bedingung  $a \leq 16$  erstmals *nicht* erfüllt.  
Die **while**-Schleife bricht also ab. Die Variable  $a$  hat danach den Wert 256.

while - Schleife



MATHEMATIK  
macht  
FREU(N)DE

Wie oft wird die **while**-Schleife vollständig durchgeführt? Welchen Wert hat die Variable  $a$  schließlich?

- |  |  |
|--|--|
| <ol style="list-style-type: none"> <li>a) 1: <math>a \leftarrow 25</math></li> <li>2: <b>while</b> <math>a &gt; 1</math> <b>do</b></li> <li>3:     <math>a \leftarrow a - 3</math></li> <li>4: <b>end while</b></li> </ol> | <ol style="list-style-type: none"> <li>b) 1: <math>a \leftarrow 7</math></li> <li>2: <b>while</b> <math>a^2 \leq 42</math> <b>do</b></li> <li>3:     <math>a \leftarrow a + 1</math></li> <li>4: <b>end while</b></li> </ol> |
|--|--|

$$25 - 3 \cdot x > 1 \implies x < 8$$

8 vollständige Durchführungen,  $a = 1$

0 vollständige Durchführungen,  $a = 7$