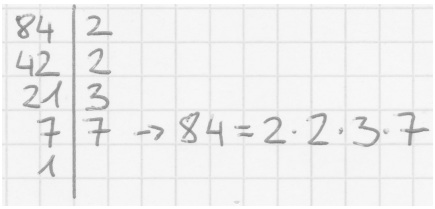
Was ist ein Algorithmus?

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.

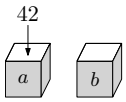


*„Ein **Algorithmus** ist [...] eine wohldefinierte Rechenvorschrift, die eine Größe oder eine Menge von Größen als Eingabe verwendet und eine Größe oder eine Menge von Größen als Ausgabe erzeugt. Somit ist ein Algorithmus eine Folge von Rechenschritten, die die Eingabe in die Ausgabe umwandeln.“*

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

Variablen & Wertzuweisungen

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 als Wert 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 jeweils nach Durchführung der Wertzuweisung haben.

Variablen & Wertzuweisungen

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

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: $T \leftarrow 8$
2: $K \leftarrow 5$
3: **if** $T \leq 10$ **then**
4: $\quad K \leftarrow 10$
5: $\quad T \leftarrow T + 4$
6: **end if**
7: $T \leftarrow 20$

In Zeile 3 wird überprüft, ob der Wert von T höchstens 10 ist.
Falls diese Bedingung erfüllt ist, werden die eingerückten Befehle bis zum **end if** durchgeführt.
Falls sie *nicht* erfüllt ist, werden die eingerückten Befehle bis zum **end if** *nicht* durchgeführt.

Zeile	1	2	3	4	5	6	7
T	8	8	**8**	8	12	12	20
K	–	5	5	10	10	10	10

if-Abfrage MmF

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

1: $x \leftarrow -5$
2: $y \leftarrow 3$
3: **if** $x \cdot y < 0$ **then**
4: $\quad x \leftarrow x \cdot y$
5: $\quad y \leftarrow x$
6: **end if**
7: **if** $x \neq y$ **then**
8: $\quad x \leftarrow 42$
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

if – else if – else MmF

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

Ein Programm soll die erreichten Punkte P in die entsprechende Ziffernote N (1, 2, 3, 4, 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** $P \geq 30$ **then**
2: $\quad N \leftarrow 1$
3: **end if**
4: **if** $26 \leq P < 30$ **then**
5: $\quad N \leftarrow 2$
6: **end if**
7: **if** $21 \leq P < 26$ **then**
8: $\quad N \leftarrow 3$
9: **end if**
10: **if** $16 \leq P < 21$ **then**
11: $\quad N \leftarrow 4$
12: **end if**
13: **if** $P < 16$ **then**
14: $\quad N \leftarrow 5$
15: **end if**

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

1: **if** $P \geq 30$ **then**
2: $\quad N \leftarrow 1$
3: **else if** $P \geq 26$ **then**
4: $\quad N \leftarrow 2$
5: **else if** $P \geq 21$ **then**
6: $\quad N \leftarrow 3$
7: **else if** $P \geq 16$ **then**
8: $\quad N \leftarrow 4$
9: **else**
10: $\quad N \leftarrow 5$
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.

if – else if – else MmF

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** $a \cdot b > 0$ **then**
2: $\quad s \leftarrow 1$
3: **else if** $a \cdot b < 0$ **then**
4: $\quad s \leftarrow -1$
5: **else**
6: $\quad s \leftarrow 0$
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 hier 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

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: $\quad a \leftarrow a + b$
5: $\quad b \leftarrow a + b$
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				2				3			

for - Schleife

Welchen Wert hat die Variable a schließlich?

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

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

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

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

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

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

for - Schleife

Trage jene Zahl in das Kästchen ein, damit die Variable a schließlich den Wert 42 hat.

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

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

for - Schleife

Das folgende Programm verwendet in einer **for**-Schleife den Wert der Zählvariable i.
Welchen Wert hat die Variable s nach Durchführung der **for**-Schleife?

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

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

repeat until - Schleife

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.

```
1: a ← 1
2: repeat
3:     a ← 2 · 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

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

a)

```
1: a ← 13
2: repeat
3:     a ← a − 2
4: until a ≤ 4
```

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

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

b)

```
1: a ← 5
2: repeat
3:     a ← 3 · a
4: until a > 2306
```

$5 \cdot 3^x > 2306 \implies x > \underbrace{\log_3\left(\frac{2306}{5}\right)}_{=5{,}58\ldots}$

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

Endlosschleife

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

```
1: a ← 1
2: repeat
3:     a ← a + 2
4: until a = 10
```

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

while - Schleife

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

```
1: a ← 2
2: while a ≤ 16 do
3:     a ← a · 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

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

a)

```
1: a ← 25
2: while a > 1 do
3:     a ← a − 3
4: end while
```

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

8 vollständige Durchführungen, $a = 1$

b)

```
1: a ← 7
2: while a² ≤ 42 do
3:     a ← a + 1
4: end while
```

0 vollständige Durchführungen, $a = 7$