



**INFORMATIK-BIBER SCHWEIZ  
CASTOR INFORMATIQUE SUISSE  
CASTORO INFORMATICO SVIZZERA**

# Aufgaben und Lösungen 2016 Schuljahre 11/12/13

<http://www.informatik-biber.ch/>

**Herausgeber:**

Christian Datzko, Hanspeter Erni

010100110101011001001001  
010000010010110101010011  
010100110100100101000101  
001011010101001101010011  
010010010100100100100001

# SV!A

[www.svia-ssie-ssii.ch](http://www.svia-ssie-ssii.ch)  
schweizerischerverein für informatik und  
erziehung // société suisse de l'inform  
atique dans l'enseignement // società sviz  
zera per l'informatica nell'insegnamento





# Mitarbeit Informatik-Biber 2016

Andrea Adamoli, Christian Datzko, Susanne Datzko, Olivier Ens, Hanspeter Erni, Martin Guggisberg, Corinne Huck, Carla Monaco, Nicole Müller, Gabriel Parriaux, Jean-Philippe Pellet, Julien Ragot, Beat Trachsler.

Herzlichen Dank an:

Juraj Hromkovič, Giovanni Serafini, Urs Hauser, Tobias Kohn, Ivana Kosírová, Serena Pedrocchi, Björn Steffen: ETHZ

Valentina Dagienė: Bebras.org

Hans-Werner Hein, Wolfgang Pohl, Peter Rossmann: Bundesweite Informatikwettbewerbe (BWINF), Deutschland

Anna Morpurgo, Violetta Lonati, Mattia Monga: Italien

Gerald Futschek: Oesterreichische Computer Gesellschaft, Österreich

Zsuzsa Pluhár: ELTE Informatikai Kar, Ungarn

Eljakim Schrijvers, Daphne Blokhuis, Marissa Engels: Eljakim Information Technology bv, Niederlande

Roman Hartmann: hartmannGestaltung (Flyer Informatik-Biber Schweiz)

Christoph Frei: Chragokyberneticks (Logo Informatik-Biber Schweiz)

Pamela Aeschlimann, Andreas Hieber, Aram Loosmann, Daniel Vuille, Peter Zurflüh: Lernetz.ch (Webseite)

Andrea Leu, Maggie Winter, Brigitte Maurer: Senarclens Leu + Partner

Die deutschsprachige Fassung der Aufgaben wurde ähnlich auch in Deutschland und Österreich verwendet.

Die französischsprachige Übersetzung wurde von Nicole Müller und die italienischsprachige Übersetzung von Andrea Adamoli erstellt.



**INFORMATIK-BIBER SCHWEIZ**  
**CASTOR INFORMATIQUE SUISSE**  
**CASTORO INFORMATICO SVIZZERA**

Der Informatik-Biber 2016 wurde vom Schweizerischen Verein für Informatik in der Ausbildung SVIA durchgeführt und von der Hasler Stiftung unterstützt.

## HASLERSTIFTUNG

Hinweis: Alle Links wurden am 1. November 2016 geprüft. Dieses Aufgabenheft wurde am 13. November 2016 mit dem Textsatzsystem  $\text{\LaTeX}$  erstellt.



Die Aufgaben sind lizenziert unter einer Creative Commons Namensnennung – Nicht-kommerziell – Weitergabe unter gleichen Bedingungen 4.0 International Lizenz. Die Autoren sind auf S. 43 genannt.



## Vorwort

Der Wettbewerb „Informatik-Biber“, der in verschiedenen europäischen Ländern schon seit mehreren Jahren bestens etabliert ist, will das Interesse von Kindern und Jugendlichen an der Informatik wecken. Der Wettbewerb wird in der Schweiz in Deutsch, Französisch und Italienisch vom Schweizerischen Verein für Informatik in der Ausbildung SVIA durchgeführt und von der Hasler Stiftung im Rahmen des Förderprogramms FIT in IT unterstützt.

Der Informatik-Biber ist der Schweizer Partner der Wettbewerbs-Initiative „Bebras International Contest on Informatics and Computer Fluency“ (<http://www.bebas.org/>), die in Litauen ins Leben gerufen wurde.

Der Wettbewerb wurde 2010 zum ersten Mal in der Schweiz durchgeführt. 2012 wurde zum ersten Mal der Kleine Biber (Stufen 3 und 4) angeboten.

Der „Informatik-Biber“ regt Schülerinnen und Schüler an, sich aktiv mit Themen der Informatik auseinander zu setzen. Er will Berührungängste mit dem Schulfach Informatik abbauen und das Interesse an Fragenstellungen dieses Fachs wecken. Der Wettbewerb setzt keine Anwenderkenntnisse im Umgang mit dem Computer voraus – ausser dem „Surfen“ auf dem Internet, denn der Wettbewerb findet online am Computer statt. Für die Fragen ist strukturiertes und logisches Denken, aber auch Phantasie notwendig. Die Aufgaben sind bewusst für eine weiterführende Beschäftigung mit Informatik über den Wettbewerb hinaus angelegt.

Der Informatik-Biber 2016 wurde in fünf Altersgruppen durchgeführt:

- Stufen 3 und 4 (Kleiner Biber)
- Stufen 5 und 6
- Stufen 7 und 8
- Stufen 9 und 10
- Stufen 11 bis 13

Die Stufen 3 und 4 hatten 9 Aufgaben zu lösen, jeweils drei davon aus den drei Schwierigkeitsstufen leicht, mittel und schwer.

Jede der anderen Altersgruppen hatte 15 Aufgaben zu lösen, jeweils fünf davon aus den drei Schwierigkeitsstufen leicht, mittel und schwer.

Für jede richtige Antwort wurden Punkte gutgeschrieben, für jede falsche Antwort wurden Punkte abgezogen. Wurde die Frage nicht beantwortet, blieb das Punktekonto unverändert. Je nach Schwierigkeitsgrad wurden unterschiedlich viele Punkte gutgeschrieben beziehungsweise abgezogen:

	leicht	mittel	schwer
richtige Antwort	6 Punkte	9 Punkte	12 Punkte
falsche Antwort	−2 Punkte	−3 Punkte	−4 Punkte

Das international angewandte System zur Punkteverteilung soll dem erfolgreichen Erraten der richtigen Lösung durch die Teilnehmenden entgegenwirken.

Jede Teilnehmerin und jeder Teilnehmer hatte zu Beginn 45 Punkte (Kleiner Biber 27) auf dem Punktekonto.

Damit waren maximal 180 (Kleiner Biber: 108) Punkte zu erreichen, das minimale Ergebnis betrug 0 Punkte.

Bei vielen Aufgaben wurden die Antwortalternativen am Bildschirm in zufälliger Reihenfolge angezeigt. Manche Aufgaben wurden in mehreren Altersgruppen gestellt.



---

## Für weitere Informationen:

SVIA-SSIE-SSII Schweizerischer Verein für Informatik in der Ausbildung

Informatik-Biber

Hanspeter Erni

biber@informatik-biber.ch

<http://www.informatik-biber.ch/>

 <https://www.facebook.com/informatikbiberch>



# Inhaltsverzeichnis

Mitarbeit Informatik-Biber 2016	i
Vorwort	ii
1. KIX-Code	1
2. Medianfilter	3
3. Hände schütteln	5
4. Nim	9
5. Sortierte Trikot-Nummern	11
6. Cassy, die Schildkröte	13
7. Palettenlift	17
8. Kugelspiel	21
9. Zwei Möglichkeiten	25
10. Tramkreuzung	27
11. Codierung von Flaggen	29
12. Schuhbündel	33
13. Rekursive Malerei	35
14. Formenspiel	37
15. Egoistische Eichhörnchen	41
A. Aufgabenautoren	43
B. Sponsoring: Wettbewerb 2016	44
C. Weiterführende Angebote	47



# 1. KIX-Code

In den Niederlanden sind die Postleitzahlen vierstellig und enthalten Buchstaben und Ziffern. Es gibt sogar einen eigenen Strichcode für die Postleitzahlen, den KIX-Code. In jedem Zeichen des KIX-Codes gibt es einen oberen Teil, zwei lange und zwei kurze Balken, und einen unteren Teil, ebenfalls zwei lange und zwei kurze Balken. In der Mitte überdecken sich die kurzen Balken. In der Tabelle sind die KIX-Code-Zeichen für 0, 7, G und Y zusammengesetzt:

	0	1	2	3	4	5
	6	7	8	9	A	B
	C	D	E	F	G	H
	I	J	K	L	M	N
	O	P	Q	R	S	T
	U	V	W	X	Y	Z

Der KIX-Code der Postleitzahl G7Y0 ist also:

Zu welcher Postleitzahl gehört dieser KIX-Code:



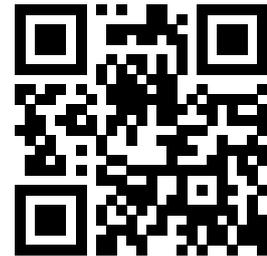
## Lösung

Die richtige Antwort lautet BC16:

	0	1	2	3	4	5
	6	7	8	9	A	B
	C	D	E	F	G	H
	I	J	K	L	M	N
	O	P	Q	R	S	T
	U	V	W	X	Y	Z

## Dies ist Informatik!

In den Niederlanden wird der KIX-Code tatsächlich bei der Post verwendet. Maschinenlesbare Codes ermöglichen es, Briefe und Pakete automatisch zu sortieren. Derartige Codes kommen auch sonst häufig zum Einsatz, zum Beispiel Barcodes ("Bar" ist das englische Wort für "Balken"), die man von Scanner-Kassen im Supermarkt kennt. QR-Code wurden in der Autoindustrie erfunden, um Bauteile zu kennzeichnen. Mittlerweile sind sie überall in der Werbung zu finden und es gibt Smartphone-Apps zum Scannen. Wofür wohl der QR-Code in diesem Text steht?



## Webseiten und Stichwörter

KIX-Code, Strichcode, QR-Code

- <https://nl.wikipedia.org/wiki/KIX-code>
- <https://de.wikipedia.org/wiki/Strichcode>
- <https://de.wikipedia.org/wiki/RM4SCC>



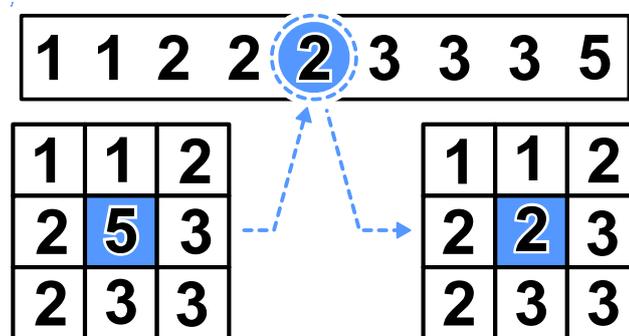
## 2. Medianfilter

Ein Bild wird als Tabelle mit Helligkeitswerten für jedes Pixel zwischen 1 und 5 gespeichert. Der Wert 1 steht für Schwarz, der Wert 5 steht für Weiss und die Werte von 2 bis 4 stehen für die heller werdenden Grautöne dazwischen.

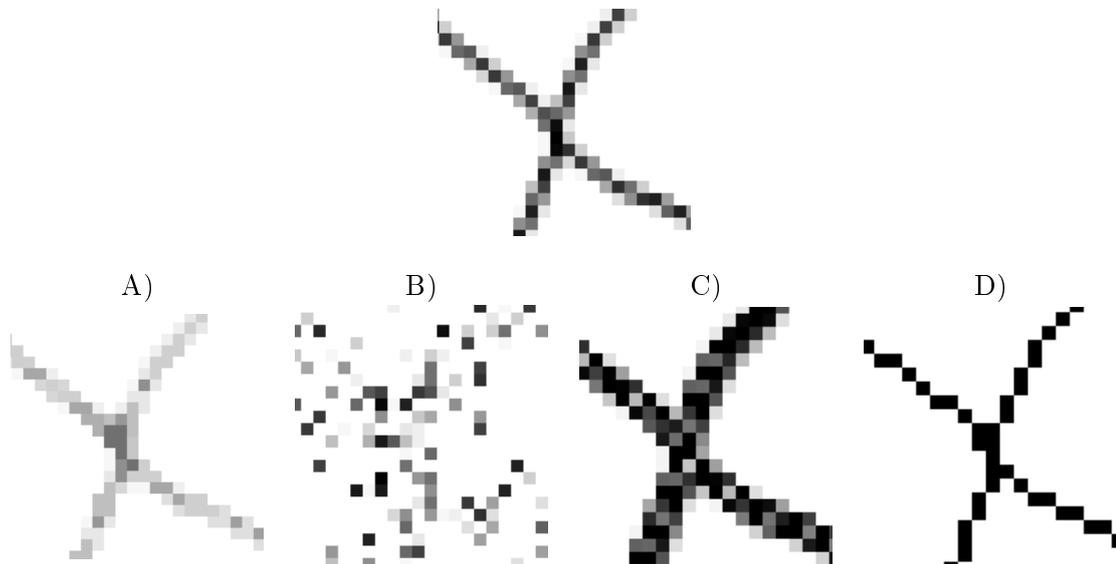
Ein sogenannter „Medianfilter“ verändert den Grauton von jedem Pixel des Bildes so, dass...

- ...der Wert des Pixels und die Werte seiner acht Nachbarn in einer Reihe aufgeschrieben und dabei sortiert werden, ...
- ...und das Pixel den fünften Wert, also den mittleren Wert der Reihe, als neuen Grauton erhält.

Hier erhält das mittlere Pixel als neuen Wert eine 2:



Wie wird dieses Bild aussehen, wenn der Medianfilter es verändert hat?





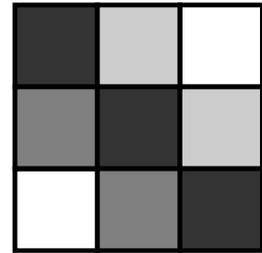
## Lösung

Die Antwort A) ist richtig:

Bei diesem Medianfilter werden quadratische Bildausschnitte mit neun Pixeln verarbeitet. Das schwarze Pixel in der Mitte bekommt einen neuen Wert. Wenn man das Beispiel rechts betrachtet, fällt auf, dass von den neun Pixeln nur drei schwarz sind. Der Median der sortierten Grautonfolge ist folglich heller als das ungefilterte Original. Aus Schwarz wird ein Grauton.

In dem Bild in der Aufgabe sind in allen quadratischen 9-Pixel-Ausschnitten des Bildes die schwarzen Pixel in der Minderheit. Deshalb enthält das gefilterte Bild kein einziges schwarzes Pixel.

Das ist nur bei Bild A) der Fall.



## Dies ist Informatik!

Beim Bearbeiten von Fotos möchte man schnell mal bestimmte Effekte erzielen. Häufig wünscht man sich mehr Schärfe oder lebendigere Farben. Manchmal möchte man auch künstlerische Effekte erzielen, um einem Bild eine besondere Note zu geben. Diese Effekte kann man mit Hilfe von Bildfiltern erzeugen.

Ein solcher Filter ist der Median-Filter. Er wird beispielsweise dafür verwendet, um einzelne Pixelfehler, die beispielsweise aufgrund eines Defekts des Bildsensors entstanden sind, auszugleichen. Der Effekt ist, dass das Bild ein geglättet wirkt und kein einzelnes Pixel mehr hervorsteht. Gewisse Formen von Rauschen können so verringert werden.

## Webseiten und Stichwörter

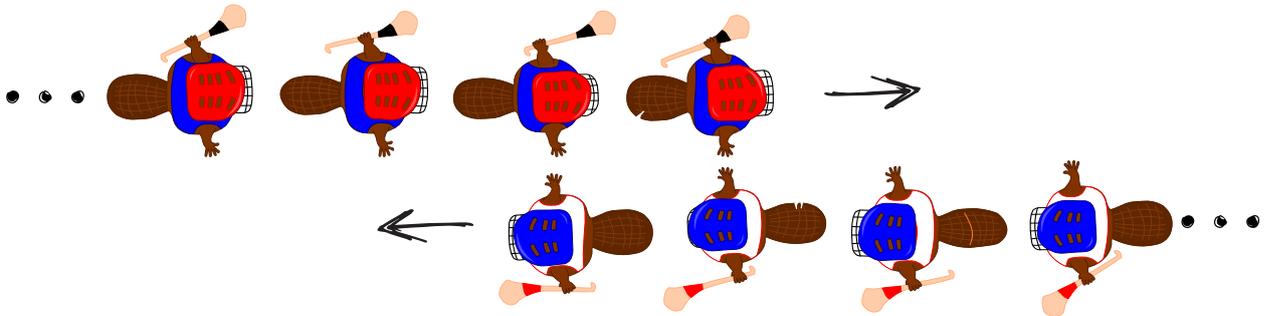
Bildverarbeitung, Medianfilter, Graustufen

- <https://de.wikipedia.org/wiki/Rangordnungsfilter>



### 3. Hände schütteln

Biber spielen gerne das irische Spiel Hurling. Am Schluss einer Partie Hurling stellen sich beide Mannschaften hintereinander in einer Reihe auf. Dann laufen die Spieler aneinander vorbei, schütteln sich nach und nach die Hände und sagen „Danke für das Spiel!“



Das Händeschütteln läuft im einzelnen so ab: Zuerst schütteln sich die beiden ersten Spieler die Hände. Dann schütteln die ersten Spieler den zweiten Spielern der jeweils anderen Mannschaft die Hände (siehe Bild). Dies geht so weiter, bis auch die beiden letzten Spieler sich die Hände geschüttelt haben.

Beim Hurling gibt es 15 Spieler pro Mannschaft. Dass zwei Spieler sich die Hände schütteln und zum jeweils nächsten Spieler gehen, dauert 1 Sekunde.

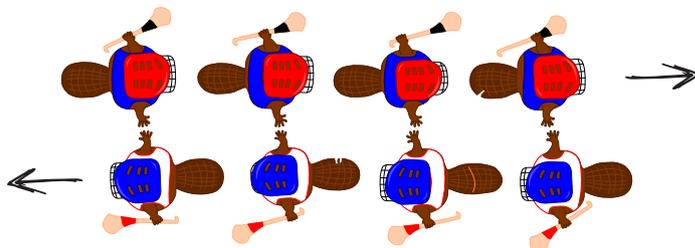
*Wie viele Sekunden dauert das Händeschütteln der beiden Mannschaften insgesamt?*



## Lösung

29 ist die richtige Antwort.

Das Händeschütteln geht so: Der jeweils erste Spieler jeder Mannschaft schüttelt jedem Spieler der anderen Mannschaft die Hand. Bei vier Spielern wären das vier Sekunden und die Situation wäre die folgende:



Gleichzeitig schüttelt der jeweils letzte Spieler jeder Mannschaft zuerst einem Spieler der anderen Mannschaft die Hand. Er muss aber noch allen anderen Spielern die Hand schütteln, was bei vier Spielern weitere drei Sekunden wären.

Allgemein dauert es bei  $n$  Spielern also zuerst  $n$  Sekunden und danach noch einmal  $n - 1$  Sekunden, so dass insgesamt  $n + n - 1 = 2n - 1$  Sekunden sind. Bei 15 Spielern sind es also  $15 + 15 - 1 = 29$  Sekunden, die das Händeschütteln dauert.

## Dies ist Informatik!

Für Hurling-Teams mit 15 Spielern konnten wir genau berechnen, wie lange das Händeschütteln braucht. 29 Sekunden sind für die Zuschauer gut auszuhalten. Doch wie sieht diese „Laufzeit“ bei Eishockey-Teams mit insgesamt 22 Spielern aus? Ist der Algorithmus des Hurling-Händeschütteln dann immer noch brauchbar, oder würde es zu lange dauern? Es wäre gut, eine allgemeine Einschätzung der Laufzeit eines Algorithmus zu haben, ohne dass wir immer alles einzeln berechnen müssen.

Die Informatik befasst sich intensiv mit allgemeinen Einschätzungen von Algorithmen-Laufzeit. Solche Laufzeitanalysen liefern einen mathematischen Ausdruck, der eine Variable  $n$  für die Grösse der Eingabe enthält. Für das Hurling-Händeschütteln erhalten wir einen solchen Ausdruck, wenn wir im zweiten Satz der Answererklärung „Anzahl der Spieler einer Mannschaft“ durch  $n$  ersetzen:  $2n - 1$ . Damit lässt sich auch für andere Spielerzahlen die Händeschüttel-Laufzeit exakt berechnen: für 22 Spieler 43 Sekunden, für 40 Spieler 79 Sekunden usw.

Hinter dem Laufzeitausdruck  $2n - 1$  steckt eine lineare Funktion. Damit gehört der Händeschüttel-Algorithmus zur Klasse der Algorithmen mit linearer Laufzeit, die man auch als  $O(n)$  bezeichnet. Wie wäre es aber, wenn man anders die Hände schütteln würde? Beispielsweise dass jeder jedem einzeln die Hand gibt? Dann gehörte er zur Klasse  $O(n^2)$ , die Hurling-Teams würden dann  $15^2 = 225$  Sekunden lang Hände schütteln, also fast vier Minuten. Hätte der Algorithmus gar exponentielle Laufzeit, wäre also in der Klasse  $O(2^n)$ , würde ca.  $2^{15} = 32768$  Sekunden lang Hände geschüttelt, also über 9 Stunden. Da wären die Zuschauer längst im Bett. Es lohnt sich also, gut darüber nachzudenken, ob man nicht wie in diesem Fall einige Dinge parallel erledigen kann, um Zeit zu sparen.

## Webseiten und Stichwörter

Laufzeitkomplexität, Laufzeitanalyse



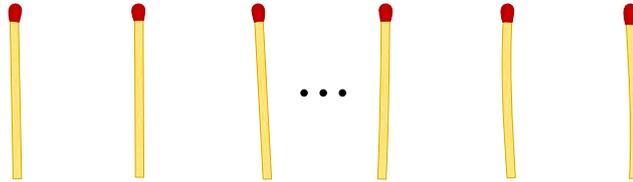
- [https://de.wikipedia.org/wiki/Laufzeit\\_\(Informatik\)](https://de.wikipedia.org/wiki/Laufzeit_(Informatik))





## 4. Nim

Beat und sein Freund spielen das Nim-Spiel. 13 Hölzchen liegen auf dem Tisch. Die beiden Spieler nehmen abwechselnd 1, 2 oder 3 Hölzchen weg. Wer das letzte Hölzchen nimmt, hat gewonnen.



Hinweis: Wenn noch vier Hölzchen auf dem Tisch liegen, kann Beat nicht mehr gewinnen. Diese Situation möchte er vermeiden.

*Beat fängt an. Wie viele Hölzchen muss er wegnehmen, um das Spiel zu gewinnen?*

- A) 1
- B) 2
- C) 3
- D) Das spielt keine Rolle.



## Lösung

1 ist richtig. Dann bleiben 12 Hölzchen übrig. Der Freund nimmt 1, 2 oder 3 weg und Beat nimmt so viele, dass 8 übrig bleiben. Wieder nimmt der Freund 1, 2 oder 3 weg. Beat nimmt so viele, dass 4 übrig bleiben und der Freund nicht mehr gewinnen kann.

Wenn Beat 2 oder 3 Hölzchen nimmt, kann der Freund so reagieren, dass ein Vielfaches von 4 übrig bleibt. Dann kann Beat nicht mehr gewinnen.

## Dies ist Informatik!

In der Spieltheorie werden Spiele wie das Nim-Spiel oder das berühmte Gefangenens-Dilemma als Modelle verwendet, um strategische Probleme in der Wirklichkeit zu analysieren und Lösungswege zu finden. In einer Marktwirtschaft dienen die Erkenntnisse zum Beispiel dazu, Preise optimal zu gestalten. Eine Preissenkung kann zwar den Absatz erhöhen, aber sie verringert gleichzeitig den Gewinn pro verkauftem Produkt. Umgekehrt vergrößert eine Preissteigerung den Gewinn pro verkauftem Produkt, aber sie kann auch dazu führen, dass der Absatz und damit der Gesamtgewinn sinkt. Mit Modellen aus der Spieltheorie können mögliche Reaktionen der Käufer auf Preisänderungen vorhergesagt werden. Wie bedeutend diese Modelle für die Wirtschaft sind, sieht man daran, dass bereits mehrere Nobelpreise für spieltheoretische Arbeiten verliehen wurden.

## Webseiten und Stichwörter

Nim-Spiel, Spieltheorie, Entscheidungsbaum

- <http://www.mathematische-basteleien.de/nimspiel.html>
- <http://scienceblogs.de/zoonpolitikon/2008/04/22/spieltheorie-einfach-erklart-i-einleitung-und-gefangenendilemma/>



## 5. Sortierte Trikot-Nummern

Die folgenden Bilder zeigen zwei Teams mit je 15 Spielern. Beide Teams haben mit Nummern bedruckte Trikots. Die Spieler der ersten Mannschaft sind nach ihrer Nummer sortiert. Die Spieler der zweiten Mannschaft sind nicht sortiert.

Team 1:



Team 2:



Wie kann man am schnellsten feststellen, welche Nummern sowohl im Team 1 als auch im Team 2 verwendet werden?

- A) Man durchläuft die Nummern von Team 1 (1, 4, 5, ...). Bei jeder Nummer stellt man fest, ob diese in Team 2 vorkommt.
- B) Man durchläuft die Nummern von Team 2 (8, 28, 12, ...). Bei jeder Nummer stellt man fest, ob diese in Team 1 vorkommt.
- C) Es ist egal bei welchem Team man beginnt. Beides geht gleich schnell.
- D) Zuerst muss man feststellen, wie viele Nummern *nicht* in beiden Teams vorkommen. Ziehe ich diese Zahl von 15 ab, dann erhalte ich die gesuchte Zahl.



## Lösung

Die richtige Antwort ist B). In einer sortierten Folge findet man Zahlen schneller als in einer nicht sortierten Folge. Ein Computer würde beispielsweise nur  $\log_2(n)$  anstelle von  $n$  Suchschritten benötigen. Somit benötigt Antwort B) im dümmsten Fall  $n \cdot \log_2(n)$  Schritte.

Für die Antwort A) würde man zum Durchsuchen der sortierten Folge nicht schnell suchen, sondern müsste alle Zahlen einzeln durchgehen. Somit benötigt Antwort A) im dümmsten Fall  $n^2$  Schritte, was mehr als  $n \cdot \log_2(n)$  Schritte ist.

Damit ist auch widerlegt, dass die Antwort C) richtig wäre, denn  $n^2$  ist nicht gleich  $n \cdot \log_2(n)$ .

Die Antwort D) gibt in der Regel nicht die richtige Antwort, daher kann sie also schon gar nicht der schnellste Weg zur richtigen Antwort sein.

## Dies ist Informatik!

Eine sortierte Folge kannst du schneller durchsuchen als eine unsortierte Folge. Wie suchst du z.B. die Nummer 9 auf den Trikots von Team 1? Du schaust zuerst auf den mittleren Spieler. Er hat die Nummer 17. Die gesuchte Nummer 9 ist kleiner. Also suchst du links weiter und guckst auf die Mitte der linken Hälfte der Mannschaft. Da steht Nummer 7. Also suchst du rechts von 7 weiter und hast die Nummer 9 rasch gefunden. Der Trick ist, dass du das Suchgebiet bei jedem Suchschritt in zwei Teile teilst und damit immer kleiner machst. Dieses Verfahren nennt man in der Informatik binäre Suche. Das kommt vom lateinischen Wort „bis“, was übersetzt „zwei Mal“ bedeutet. Weil sortierte Sammlungen von Daten durch binäre Suche schnell durchsucht werden können, ist das Sortieren in der Informatik eine wichtige Sache. Wenn  $n$  Zahlen durchsucht werden, sind das dann nur noch ungefähr  $\log_2(n)$  Suchschritte.

## Webseiten und Stichwörter

Sortierung, binäre Suche

- [https://de.wikipedia.org/wiki/Binäre\\_Suche](https://de.wikipedia.org/wiki/Binäre_Suche)



## 6. Cassy, die Schildkröte

Die Schildkröte Cassy lebt in Gitterland, auf einem Acker von fünf mal fünf Gitterzellen. Sie isst für ihr Leben gern frische Salatpflanzen. Jeden Morgen wachsen neue Salatpflanzen. Cassy weiss nicht, an welcher Stelle sie sind, aber sie will alle essen. Cassy startet jeden Morgen in der Mitte des Ackers und folgt den Anweisungen im Anweisungsblock.

*Sorge dafür, dass Cassy über jede Gitterzelle des Ackers läuft. Wähle links Anweisungen aus. Du kannst sie mehrfach verwenden.*

- $R$  ist eine Zählvariable. Wenn der Anweisungsblock das erste Mal ausgeführt wird, hat  $R$  den Wert 1, bei der zweiten Ausführung 2, usw.
- Cassy darf den Acker verlassen, aber nicht das Gitterland.
- Mit „Testen“ kannst Du Dein Programm testen.

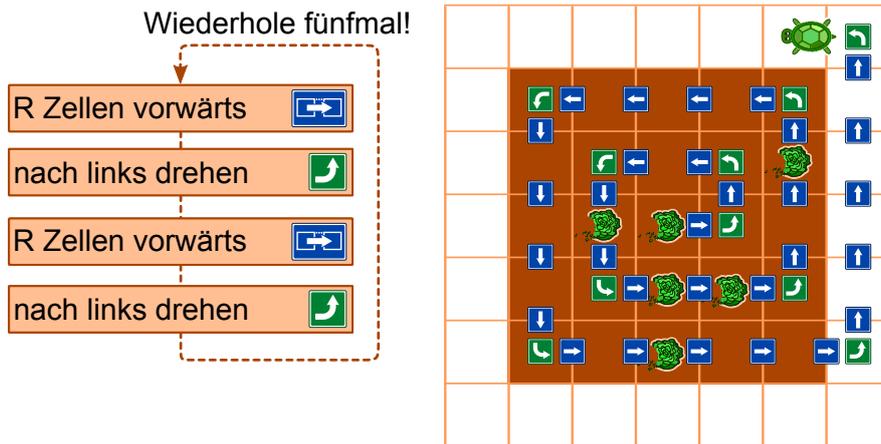
The screenshot shows the Cassy programming environment. On the left, a yellow box labeled 'Anweisungen' contains three instruction blocks: 'nach links drehen' (left turn), 'nach rechts drehen' (right turn), and 'R Zellen vorwärts' (move forward R cells). In the center, a green box labeled 'Anweisungsblock' contains four empty slots for instructions. A dashed red arrow labeled 'Wiederhole fünfmal!' (Repeat five times!) points to the top of the block, indicating a loop. Below the block is a red 'Testen' button. On the right, a 5x5 grid represents the field, with a green turtle icon in the center cell.



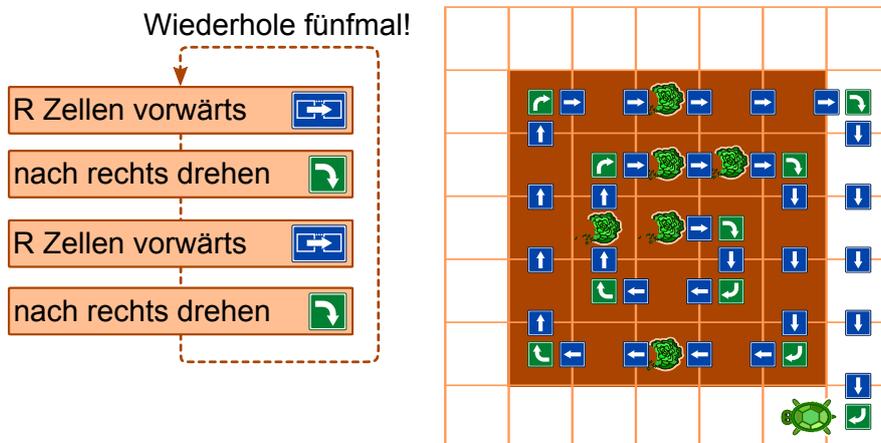
## Lösung

Es werden maximal vier Anweisungen fünfmal wiederholt. Dies bedeutet, dass der Lösungsweg einer Spirale folgen muss. Es gibt vier verschiedene Befehlsfolgen, bei welchen Cassy den gesamten Acker spiralförmig ablaufen kann:

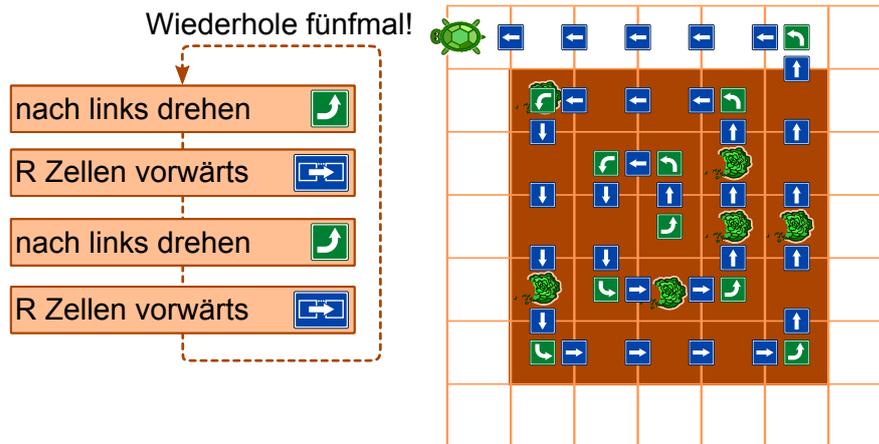
- $R$  Zellen vorwärts, nach links drehen,  $R$  Zellen vorwärts, nach links drehen



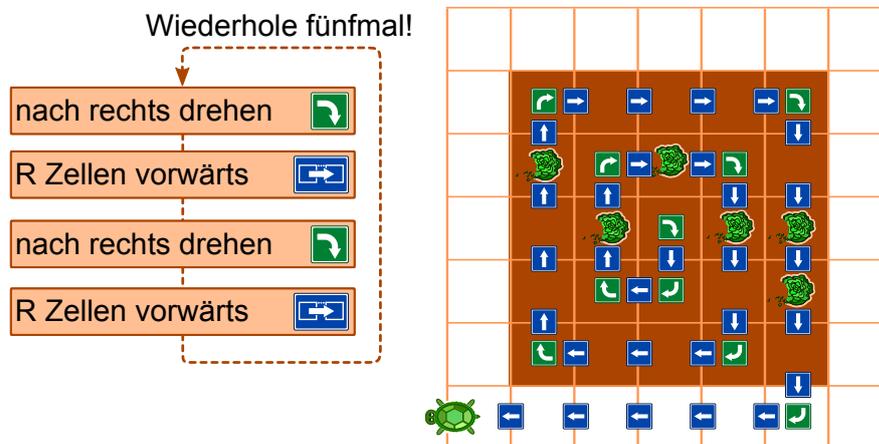
- $R$  Zellen vorwärts, nach rechts drehen,  $R$  Zellen vorwärts, nach rechts drehen



- nach links drehen,  $R$  Zellen vorwärts, nach links drehen,  $R$  Zellen vorwärts



- nach rechts drehen,  $R$  Zellen vorwärts, nach rechts drehen,  $R$  Zellen vorwärts



## Dies ist Informatik!

Für diese Aufgabe muss ein Programm geschrieben werden. Es besteht aus einer Sequenz (Abfolge) von vier einzelnen Anweisungen, die dann fünfmal wiederholt werden. Diese Wiederholung nennt man eine Schleife. In diesem Fall ist es eine sogenannte Zählschleife, weil  $R$  von 1 bis 5 hoch zählt. Ein Computer führt diese Anweisungen der Reihe nach aus. Alle nützlichen Programmiersprachen unterstützen Schleifen – und weitere Befehle die den Programmverlauf steuern können, wie z.B. Verzweigungen und die Möglichkeit Unterprogramme aufzurufen.

Falls das Programm korrekt ist, macht der Computer genau das, was du wolltest. Falls das Programm nicht korrekt ist, führt der Computer die Anweisungen zwar aus, aber die Schildkröte bewegt nicht so, wie du es wolltest. Ein Computer ist in der Regel nicht in der Lage zu erkennen, ob ein erstelltes Programm korrekt ist.

## Webseiten und Stichwörter

Turtle-Graphik

- <http://www.turtlegrafik.ch/>
- <http://primalogo.ch/>

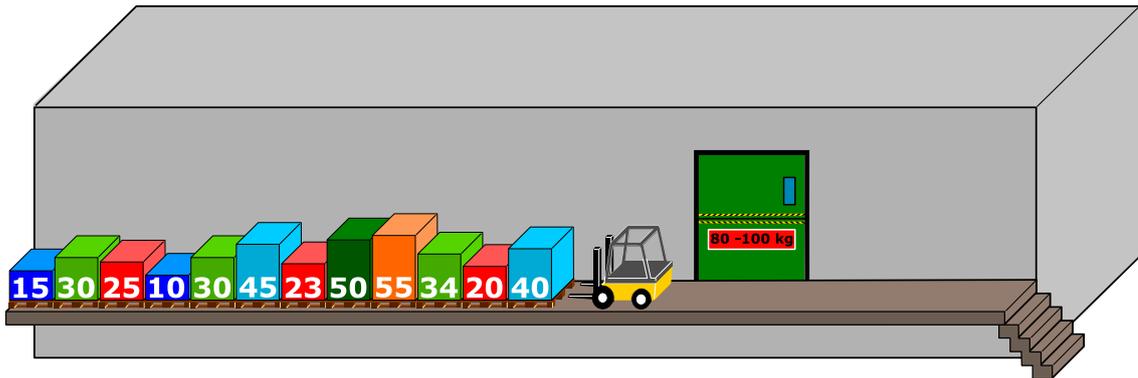




## 7. Palettenlift

Auf einer engen Laderampe stehen Paletten nebeneinander. Die Paletten wiegen (von links nach rechts): 15 kg, 30 kg, 25 kg, 10 kg, 30 kg, 45 kg, 23 kg, 50 kg, 55 kg, 34 kg, 20 kg, 40 kg. Die Rampe ist so eng, dass die Paletten nicht aneinander vorbei gehoben werden können.

Die Paletten werden in einen Lift geladen, der sie zum Lager bringt. Der Lift bringt die Paletten weg, sobald er mit mindestens 80 kg Ware geladen ist. Es darf jedoch nicht mehr als 100 kg wiegen. Er kommt dann wieder leer zurück.



Beim Laden der Paletten in den Lift wird immer diejenige Palette genommen, die sich am nächsten zum Lift befindet. Wenn das Gesamtgewicht des Liftes beim Laden von der letzten Palette 100 kg übersteigt, wird die Palette auf die andere Seite der Rampe gebracht. Ansonsten bleibt die Palette im Lift.

Wenn alle Paletten von links zum Lager gebracht worden sind, werden die auf der anderen Seite der Rampe in derselben Art und Weise in den Lift geladen.

*Welche der folgenden Aussagen ist korrekt?*

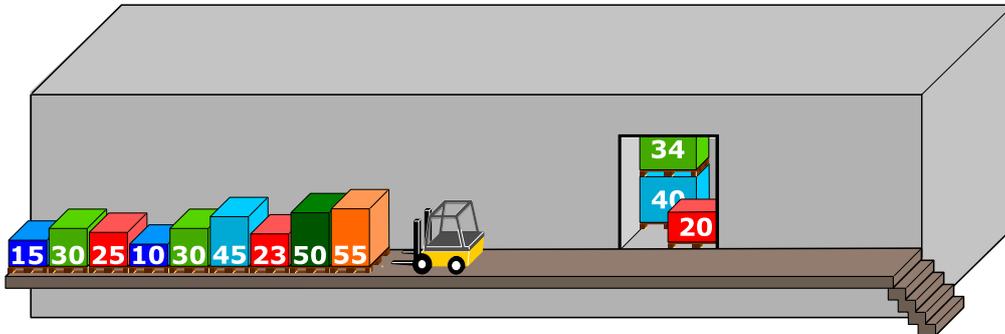
- A) Bei der zweiten Fahrt des Lifts wiegt seine Ladung 98 kg.
- B) Die Paletten am anderen Ende der Rampe werden nicht verwendet.
- C) Bei einer Fahrt des Lifts wiegt seine Ladung 100 kg.
- D) Der Lift macht insgesamt fünf Fahrten.
- E) Es ist nicht möglich, die Paletten nach der oben genannten Prozedur zum Lager zu bringen.



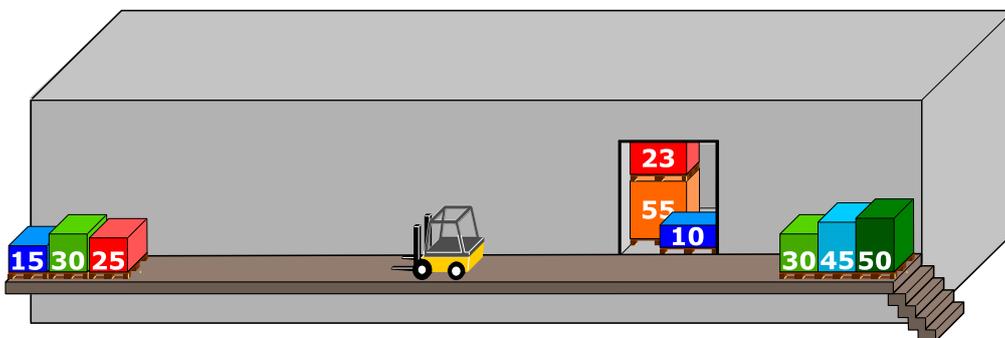
## Lösung

Die korrekte Antwort ist C):

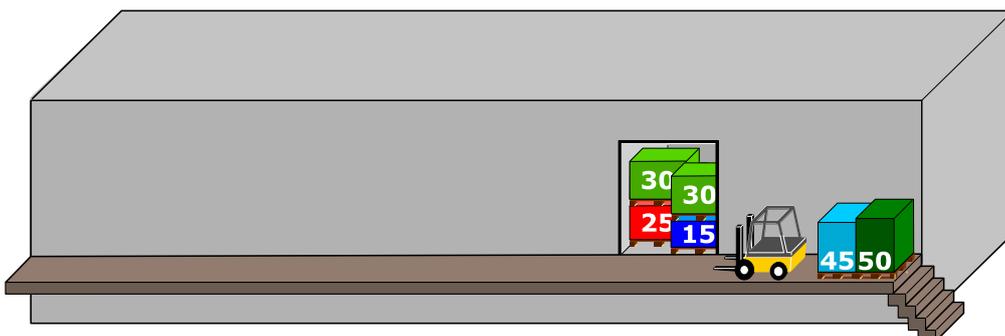
- Die erste Ladung besteht aus den ersten drei Paletten. Diese wiegen  $40\text{ kg} + 20\text{ kg} + 34\text{ kg} = 94\text{ kg}$ .



- Die zweite Ladung besteht zunächst aus einer einzigen Palette (55 kg). Wenn die nächste Palette (50 kg) ebenfalls im Lift geladen werden würde, würde die Ladung die 100 kg übersteigen, deswegen wird diese Palette zum anderen Ende der Rampe gebracht. Die nächste Palette (23 kg) wird in den Lift gestellt; jedoch ist die Ladung nun zu leicht mit  $55\text{ kg} + 23\text{ kg} = 78\text{ kg}$ . Die nächste Palette der Reihe (45 kg) würde wieder das zugelassene Gewicht im Lift übersteigen. Aus diesem Grund wird es ebenfalls auf der anderen Seite der Rampe gebracht. Dasselbe geschieht mit der nächsten Palette (30 kg). Letztendlich kann die nächste Palette (10 kg) in den Lift gestellt werden. Der Lift wird zum Laden gefahren mit einer Ladung von  $55\text{ kg} + 23\text{ kg} + 10\text{ kg} = 88\text{ kg}$ .

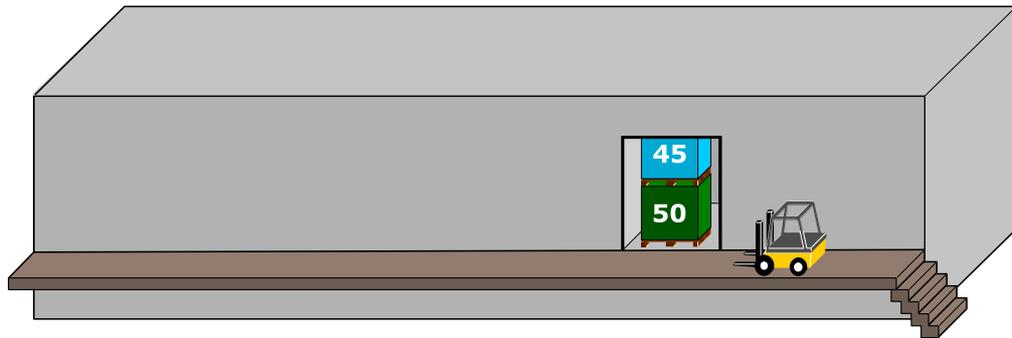


- Die letzten drei Paletten der Reihe ( $25\text{ kg} + 30\text{ kg} + 15\text{ kg} = 70\text{ kg}$ ) werden ebenfalls in den Lift geladen. Der dritten Ladung wird noch die erste (30 kg) der Paletten in der gegenüberliegenden Reihe hinzugefügt, so dass die Ladung im Lift genau 100 kg wiegt.





- Zu diesem Zeitpunkt sind nur noch zwei Paletten auf der anderen Seite der Rampe übrig: 45 kg + 50 kg = 95 kg. Diese werden als vierte Ladung im Lift zum Laden gefahren.



Ausgehend von dieser Erklärung sind die Antworten A), B), D) und E) nicht korrekt.

## Dies ist Informatik!

Weil die Rampe so eng ist, kann immer nur die erste Palette der Reihe bearbeitet werden. Das ist wie bei einem Stapel: Nur das oberste Ding auf dem Stapel kann genommen werden, und neue Dinge können nur oben auf den Stapel gelegt werden. In der Informatik heisst diese Datenstruktur auch konsequenterweise „Stapel“. In diesem Fall kommen drei Stapel vor: die linke Reihe von Paletten (mit dem obersten Element ganz rechts), die (anfangs leere) rechte Reihe von Paletten (mit dem obersten Element ganz links) und der Lift.

Stapel kommen auch sonst häufig vor: wenn ein Kind einen Turm aus Bauklötzen baut, kann es in der Regel nur den obersten Bauklotz entfernen oder oben einen drauflegen. Eine Glace mit mehreren Kugeln verhält sich meist ebenfalls wie ein Stapel: die Kugel, die zuerst auf dem Cornet platziert wird, wird zuletzt gegessen. Beim Surfen im Web können die besuchten Seiten im Tab ebenfalls als Stapel betrachtet werden: wenn auf „Zurück“ geklickt wird, erscheint die zuletzt besuchte Seite als erste.

Das Prinzip dahinter nennt man auch LIFO: „last in – first out“ (im Gegensatz zu einer Schlange, die nach dem FIFO-Prinzip funktioniert: „first in – first out“). Dieses Prinzip wird beispielsweise bei einer Warteschlange am Schalter oder beim Arzt verwendet (wobei Ärzte zum Glück Notfälle vorziehen können).

## Webseiten und Stichwörter

Datenstruktur, Stapel, Algorithmus, LIFO-Prinzip

- <https://de.wikipedia.org/wiki/Stapelspeicher>
- [https://de.wikipedia.org/wiki/Last\\_In\\_\\T1\\textendash\\_First\\_Out](https://de.wikipedia.org/wiki/Last_In_\\T1\\textendash_First_Out)





## 8. Kugelspiel

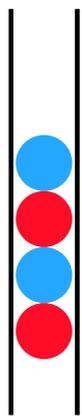
Emil spielt ein neues Spiel auf seinem Computer. Das Spiel beginnt mit einem Stapel aus mindestens drei farbigen Kugeln (rot oder blau), die sich in einer Röhre befinden.

Nach einem Klick auf eine Taste fallen die beiden jeweils unteren Kugeln aus der Röhre. Ausserdem fallen von oben neue Kugeln auf den Stapel. Abhängig von der Farbe der bisher untersten Kugel können zwei Dinge geschehen:

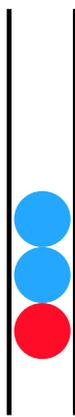
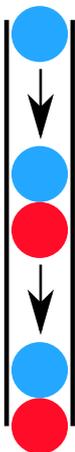
Falls die bisher unterste Kugel rot war, fällt eine blaue Kugel auf den Stapel:

Falls die bisher unterste Kugel blau war, fallen drei Kugeln mit den Farben rot, blau und rot auf den Stapel:

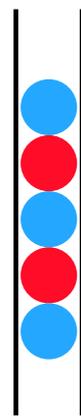
vorher



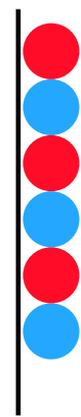
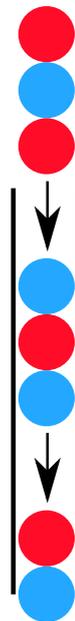
nachher



vorher



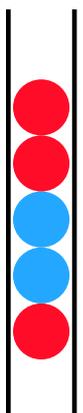
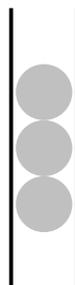
nachher



Solange mindestens drei Kugeln in der Röhre sind, drückt Emil immer wieder auf die Taste. Das Spiel endet, wenn sich weniger als drei Kugeln in der Röhre befinden.

Falls Emil mit einem Stapel wie in der Figur rechts beginnt, bleiben nach fünfmaligem Drücken nur zwei blaue Kugeln übrig, und das Spiel ist beendet.

*Ziehe von rechts passend Farben auf die drei Stapelplätze, so dass Du einen Startstapel hast, bei dem das Spiel niemals enden wird.*

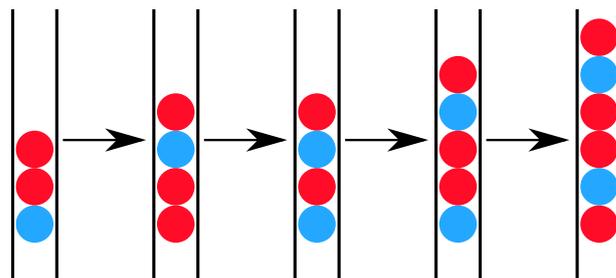
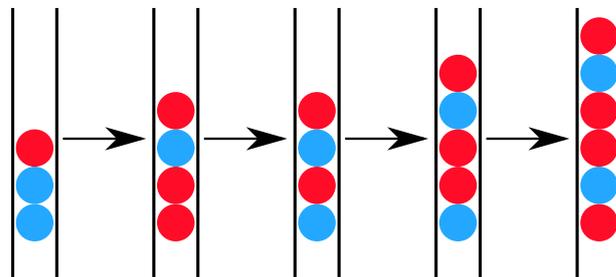
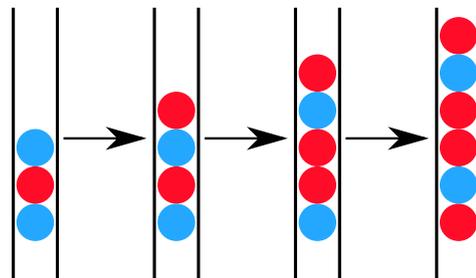
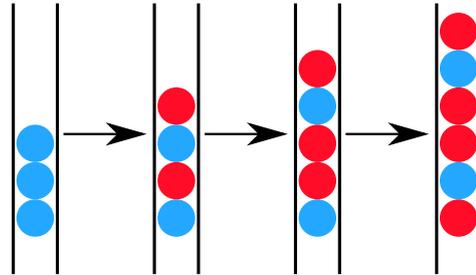




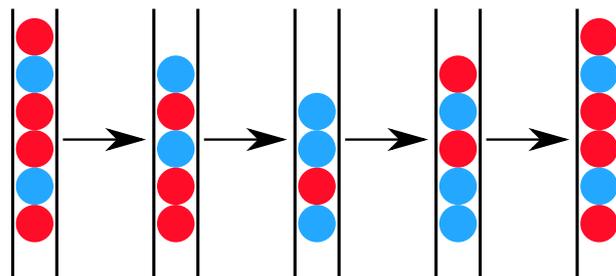
## Lösung

Wenn die unterste Kugel eines Dreierstapels rot ist, endet das Spiel schon nach dem ersten Drücken: Dann sind nur noch zwei Kugeln in der Röhre.

Hingegen ist jeder Dreierstapel, dessen unterste Kugel blau ist, ein Endlos-Stapel. Mit höchstens viermaligem Drücken auf die Taste wird aus jedem der vier möglichen Dreierstapel mit einer blauen Kugel als unterste der Sechserstapel rot-blau-rot-rot-blau-rot:



Danach verläuft das Spiel immer in einem Viererzyklus:





## Dies ist Informatik!

Das vorgestellte Spiel wurde als Beispiel von Emil Leon Post verwendet, um zu zeigen, dass es bei einer Verarbeitung von Zeichenketten (Strings) zu nicht abbrechenden Prozessen kommen kann. Emil Leon Post (1897-1954) war ein Polnischer Mathematiker und Logiker, welcher viele wissenschaftliche Beiträge zur theoretischen Informatik auf dem Gebiet der Aussagenlogik veröffentlichte.

Ein Ersetzungssystem wie in diesem Beispiel lässt sich mit Hilfe einer formalen Grammatik beschreiben. Dabei werden neben dem Eingabealphabet Regeln definiert, nach denen ersetzt wird. In unserem Fall wären beispielsweise die beiden vier Regeln spannend (wobei  $X$  für eine beliebige Kombination von blauen oder roten Kugeln steht):

$$Xbb \rightarrow rbrX$$

$$Xrb \rightarrow rbrX$$

$$Xbr \rightarrow bX$$

$$Xrr \rightarrow bX$$

## Webseiten und Stichwörter

Rechenmodell, Formale Sprache, Produktionsregeln, Zeichenketten, verarbeitendes System

- [http://esolangs.org/wiki/Post\\_canonical\\_system](http://esolangs.org/wiki/Post_canonical_system)
- [https://en.wikipedia.org/wiki/Tag\\_system](https://en.wikipedia.org/wiki/Tag_system)
- [https://en.wikipedia.org/wiki/Post\\_canonical\\_system](https://en.wikipedia.org/wiki/Post_canonical_system)



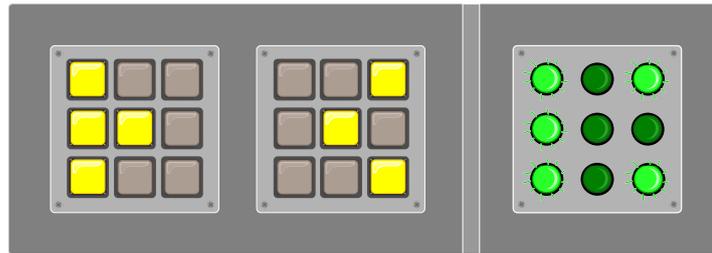


## 9. Zwei Möglichkeiten

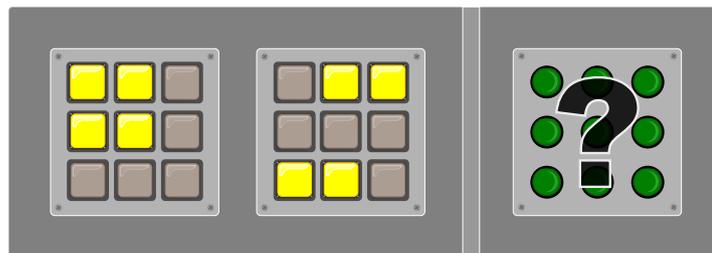
Erich hat ein altes elektronisches Geräte gefunden. Auf der linken Seite hat es zwei Felder mit je 9 Tasten, die man drücken kann. Auf der rechten Seite hat es ein Feld mit 9 Lampen. Je nachdem wie die Tasten gedrückt werden, gehen die Lampen an oder aus.

Erich beobachtet nun, dass die Position einer Lampe, die an- oder ausgeht, dieselbe ist, wie eine entsprechende Tastenkombination in den zwei Feldern.

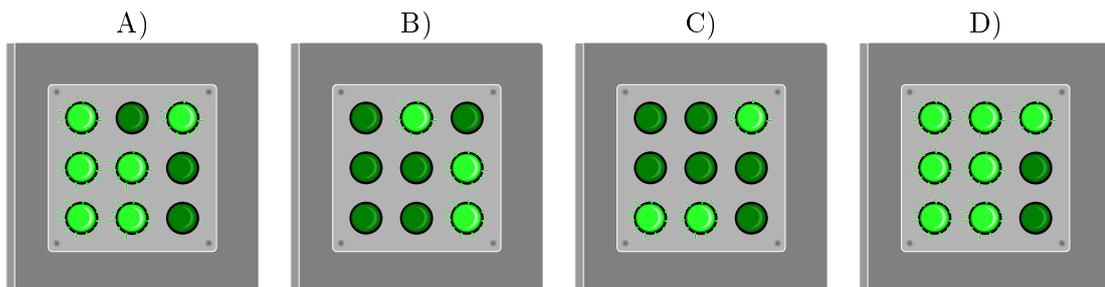
Momentan leuchten die Lampen wie folgt:



Erich ändert nun die Kombinationen, so dass sie so aussehen:



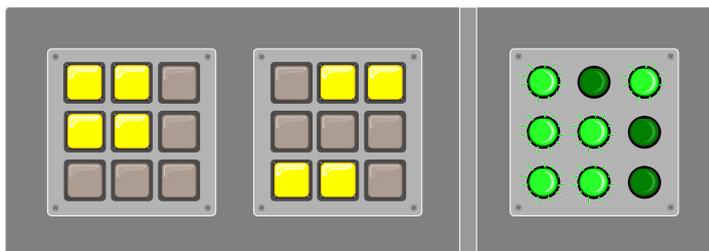
Wie leuchten die Lampen rechts?





## Lösung

Die Regel ist: Wenn auf den beiden Eingabefeldern die gleiche Taste genau einmal eingeschaltet ist, dann leuchtet die entsprechende Lampe im Ausgabefeld, ansonsten leuchtet sie nicht.



## Dies ist Informatik!

In allen Bereichen der Informatik wird gerne zweiwertige Logik benutzt, um über die Wahrheit und Falschheit von Aussagen innerhalb einer Anwendungssituation zu entscheiden. Das geschieht immer unter der Voraussetzung, dass es dafür genau zwei Möglichkeiten (zwei Werte) gibt. Eine dritte Möglichkeit ist ausgeschlossen („tertium non datur“). Eine logische Funktion liefert also entweder den Wahrheitswert „falsch“ oder den Wahrheitswert „wahr“.

In dieser Aufgabe benutzen wir die Funktion XOR („ausschliessendes Oder“, englisch: „exclusive or“), weil sie besonders häufig vorkommt. XOR funktioniert so: Wenn von zwei Aussagen genau eine „wahr“ ist und die andere „falsch“, dann liefert XOR den Wahrheitswert „wahr“. In den anderen Fällen liefert XOR den Wahrheitswert „falsch“. Im Beispiel zu dieser Aufgabe wird die logische Funktion XOR neunmal (Tasten) mit zwei Aussagen (Felder) vorgeführt.

## Webseiten und Stichwörter

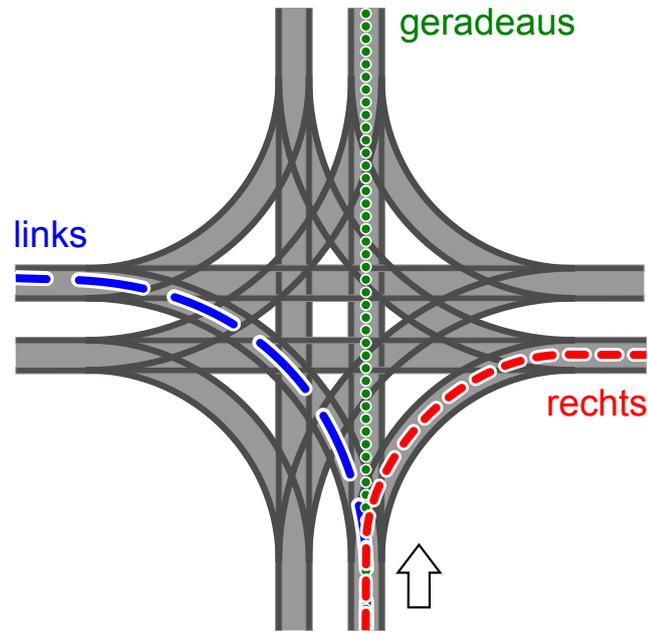
Logik, Zweiwertigkeit, Funktion XOR

- <https://de.wikipedia.org/wiki/Kontravalenz>
- <https://de.wikipedia.org/wiki/XOR-Gatter>



## 10. Tramkreuzung

Auf den Strassen von Sankt Petersburg fahren Trams. Dort gibt es eine Tramkreuzung, an der alle Wege möglich sind: Die Trams können aus jeder Richtung kommen und können jeweils gradeaus fahren, nach links oder nach rechts abbiegen.



Die Richtung, in die die Trams fahren, ist durch die Stellung der Weichen bestimmt. Die Stellungen der Weichen für Trams werden durch eine Kombination aus den Wörtern GERADEAUS, LINKS und RECHTS beschrieben. Beispiel: Die Kombination LINKS-GERADEAUS-LINKS-RECHTS bedeutet, dass die Weichen so gestellt sind, dass ein Tram nach links abbiegt, im Uhrzeigersinn das nächste Tram gradeaus fährt, das wieder im Uhrzeigersinn nächste Tram links fährt und das vierte Tram rechts fährt.

Nun kann es vorkommen, dass aus allen Richtungen Trams gleichzeitig an der Kreuzung ankommen. Welche Weichenstellungen können zu einem Zusammenstoß führen?

- A) RECHTS-RECHTS-RECHTS-RECHTS
- B) RECHTS-RECHTS-LINKS-LINKS
- C) LINKS-RECHTS-LINKS-RECHTS
- D) RECHTS-LINKS- RECHTS-LINKS



## Lösung

Die richtige Antwort ist B). Das erste Tram (das nach rechts abbiegt) wird mit dem dritten Tram (das nach links abbiegt) zusammenstossen. Ebenso wird das zweite Tram (das nach rechts abbiegt) mit dem vierten Tram (das nach links abbiegt) zusammenstossen.

Bei allen anderen Lösungen fahren die Trambahnen kollisionsfrei aneinander vorbei: bei A) sind alle in der äusseren Kurve, bei C) und D) fahren immer zwei Trams nebeneinander vorbei durch dieselben Strassen. Bei C) und D) ist es noch wichtig zu sehen, dass die Kurven im Inneren sich nicht stören. Das nennt man „tangenciales“ oder auch „amerikanisches“ Linksabbiegen.

## Dies ist Informatik!

Die Tramschienen – vor allem im Bereich von Kreuzungen – sind ein Beispiel für Ressourcen, die von mehreren Akteuren gemeinsam genutzt werden. Gleichzeitig darf aber an einer Stelle nur ein Tram sein ... sonst gibt es eine Kollision. Die Koordination und Optimierung der gemeinsamen Nutzung von Ressourcen durch parallel laufende Prozesse ist ein wichtiges Gebiet der Informatik. Wie bei der Trambahn muss man auch in der Digitaltechnik potenzielle Risiken beachten und Regeln für den sicheren Zugriff auf Datenbestände oder gemeinsam genutzte Geräte finden. Sonst droht der Verlust von Daten oder die Zerstörung von Hardware.

Im Fall der Tramkreuzung ist es eine gute Idee, nur sichere Weichenstellungen zuzulassen. Das kann man dort mit Hilfe von Signalen lösen. In der Informatik gibt es ebenfalls „Signal“-Lösungen für solche Probleme: sogenannte „Semaphoren“ (engl. für “Signal”) zeigen an, dass bestimmte Ressourcen belegt sind.

## Webseiten und Stichwörter

Tramkreuzung, Prozesssynchronisation, Zugriff auf beschränkte Ressourcen

- <https://de.wikipedia.org/wiki/Prozesssynchronisation>
- [https://de.wikipedia.org/wiki/Abbiegen\\_\(Stra%C3%9Fenverkehr\)#Tangenciales\\_und\\_nichttangenciales\\_Abbiegen](https://de.wikipedia.org/wiki/Abbiegen_(Stra%C3%9Fenverkehr)#Tangenciales_und_nichttangenciales_Abbiegen)
- <http://www.swisseduc.ch/informatik/infotraffic/logictraffic/>



## 11. Codierung von Flaggen

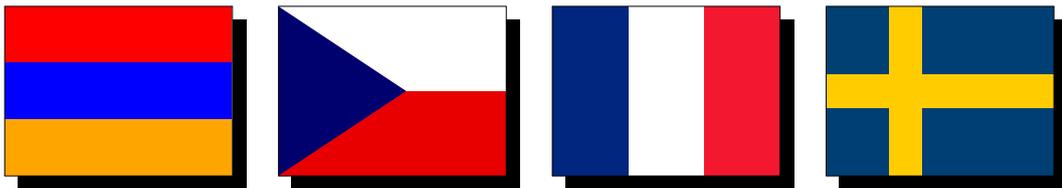
Computerbilder bestehen aus Zeilen mit Bildpunkten (Pixel). Wenn Computerbilder als Dateien gespeichert werden, wird im einfachsten Fall die Farbe jedes Pixels einzeln beschrieben. Mit dem (erfundenen) Dateiformat GIW werden Computerbilder komprimiert, also mit geringerer Dateigrösse gespeichert. Das funktioniert so:

- Jede Pixelzeile wird einzeln beschrieben.
- Jede Farbe wird durch ein Kürzel aus drei Buchstaben beschrieben.
- Eine Folge gleichfarbiger Pixel wird durch ein Klammerpaar beschrieben, das ein Farbkürzel und die Anzahl der gleichfarbigen Pixel enthält.

Eine Pixelzeile zum Beispiel, die durch die beiden Klammerpaare (grü,20)(wei,13) beschrieben wird, enthält zuerst 20 grüne und danach 13 weisse Pixel.

Unten siehst du vier Computerbilder von Flaggen. Die Bilder bestehen alle aus gleich vielen Pixelzeilen mit jeweils gleich vielen Pixeln. Sie wurden als Dateien im GIW-Format gespeichert.

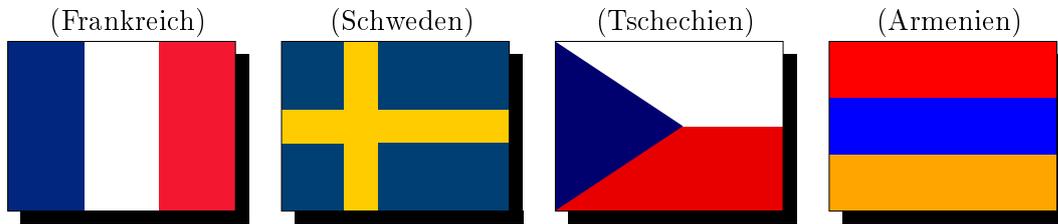
*Ordne die Bilder nach der Grösse ihrer GIW-Datei!*





## Lösung

Die richtige Reihenfolge ist:



Falls eine ganze Zeile aus einer gleichen Farbe besteht, ist nur ein Klammerpaar zur Beschreibung nötig. Für jeden Farbwechsel in der Zeile kommt ein weiteres Klammerpaar hinzu. Beispielsweise hat die Flagge von Armenien in jeder Zeile die selbe Farbe, damit kann jede Zeile durch genau ein Klammerpaar kodiert werden.

Auf der anderen Seite wechselt bei der Flagge von Tschechien die Farbe innerhalb jeder Zeile genau einmal, von blau nach weiss oder von blau nach rot. Eine komprimierte Beschreibung dieser Flagge benötigt jeweils zwei Klammerpaare pro Zeile. Die französische Flagge wechselt in jeder Zeile zweimal die Farbe und benötigt folglich drei Klammerpaare pro Zeile für die komprimierte Beschreibung einer Zeile.

Die schwedische Flagge hat einen waagerechten gelben Streifen und benötigt für diesen Bereich ein Klammerpaar pro Zeile. In allen anderen Zeilen wechselt die Farbe zweimal und benötigt jeweils drei Klammerpaare. Daraus ergibt sich, dass die schwedische Fahne mehr Klammerpaare als die armenische, jedoch weniger als die französische braucht.

Wir müssen nun die schwedische Flagge mit der tschechischen Flagge vergleichen. Hätte die schwedische Flagge gleich viele Zeilen mit einer Farbe, wie mit drei Farben, dann bräuchte es im Mittel genau 2 Klammerpaare für die Beschreibung. Der mittlere Streifen der schwedischen Flagge ist jedoch schmaler als der restliche Teil der Flagge, so dass durchschnittlich mehr als zwei Klammerpaare für die Beschreibung benötigt werden. Daraus ergibt sich, dass die schwedische Flagge insgesamt mehr Klammerpaare als die tschechische Flagge benötigt.

## Dies ist Informatik!

Die Datenkomprimierung ist ein wichtiger Teil der Informatik. Dank einer Reduktion von Speicherplatz lassen sich digitale Elemente oder Objekte mit weniger Zeitaufwand durch ein Netzwerk transportieren. Datenkomprimierungsalgorithmen können den Transferaufwand innerhalb eines Netzwerks signifikant verringern. Würde beispielsweise die Musik eines Webradios ohne Komprimierung versendet, so würde die zehnfache Datenmenge anfallen als bei heutzutage typischer Komprimierung. Aus diesem Grund wird intensiv an neuen Datenkomprimierungsalgorithmen geforscht, welche Fotos, Musik und Videos effizienter speichern können.

Das in dieser Aufgabe vorgestellte Komprimierungsverfahren gehört zum Typ der Lauflängenkodierung. Weitere Informationen kann das folgende YouTube Video liefern: [https://www.youtube.com/watch?v=ydpNscvym\\_E](https://www.youtube.com/watch?v=ydpNscvym_E).

## Webseiten und Stichwörter

Kodierung, Komprimierung, Komprimierungsalgorithmus, Bitmap-Grafik

- [https://www.youtube.com/watch?v=ydpNscvym\\_E](https://www.youtube.com/watch?v=ydpNscvym_E)



- <https://de.wikipedia.org/wiki/Lauflängenkodierung>





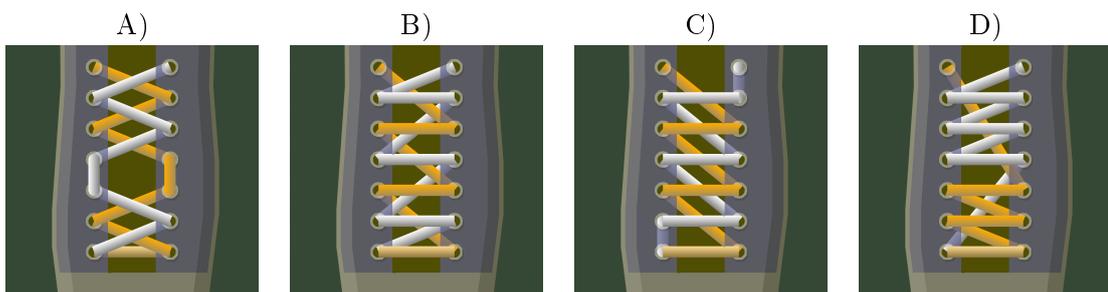
## 12. Schubbandel

Biber mögen schicke Schubbandel. Sie programmieren eine Maschine, die die Schubbandel für sie binden kann. Diese beherrscht folgende Befehle:

Befehl	Bedeutung
{ ... }	Der Inhalt der Klammern wird so oft wie möglich wiederholt.
$n$ { ... }	Der Inhalt der Klammern wird $n$ mal wiederholt.
orange:	Die folgenden Befehle gelten für die orangen Schubbandel.
weiss:	Die folgenden Befehle gelten für die weissen Schubbandel.
vorne:	Der Schubbandel geht vor die Öse.
hinten:	Der Schubbandel geht hinter die Öse.
hoch:	Der Schubbandel wird nach oben bewegt (zur nächsten Öse).
runter:	Der Schubbandel wird nach unten bewegt (zur nächsten Öse).
wechsel:	Der Schubbandel wird von links nach rechts, respektive umgekehrt bewegt; der Schubbandel wird entsprechend nachgeführt.

Welche Schnürung erzeugt das folgende Programm:

```
orange: vorne
weiss: vorne
2{
    orange: hoch wechsel vorne
    weiss: hoch wechsel vorne
}
orange: hoch hinten
weiss: hoch hinten
{
    orange: hoch wechsel vorne
    weiss: hoch wechsel vorne
}
```





## Lösung

Die richtige Lösung ist A). Bei den ersten zwei Ösen funktioniert das Programm so wie das Beispiel. Danach gehen die Schuhbündel eine Öse direkt nach oben. Danach wechseln die Schuhbündel die Seiten und es geht weiter wie anfangs.

## Dies ist Informatik!

Die Maschine zum Binden von Schuhbündeln wird mit Hilfe einer einfachen Programmiersprache programmiert. Diese enthält typische Elemente von Programmiersprachen:

- Hintereinander Ausführen von Befehlen (*Sequenz*)
- Wiederholen von Blöcken von Befehlen (*Schleife*), sowohl als Zählschleife als auch als Schleife mit Abbruchbedingung

Was häufig in Programmiersprachen vorkommt, aber nicht hier, ist das das Ausführen von Blöcken von Befehlen unter bestimmten Bedingungen (*Selektion*) sowie das separate Speichern von häufig verwendeten Blöcken von Befehlen (*Modularisierung*).

Die Maschine zum Binden von Schuhbündeln kann übrigens auch im Simulator programmiert werden: <http://www.klbg.com/lacing/>.

## Webseiten und Stichwörter

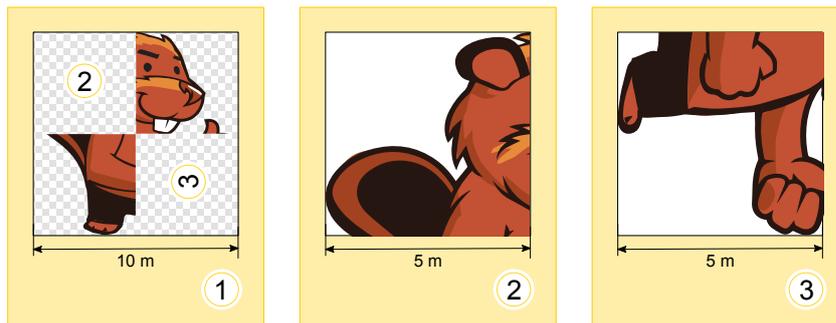
Schleife, Befehl, Simulator, Schuhbündel

- <https://de.wikipedia.org/wiki/Interpreter>
- <https://de.wikipedia.org/wiki/Programmiersprache>
- <http://www.klbg.com/lacing/>
- <http://www.fieggen.com/shoelace/lacingmethods.htm>

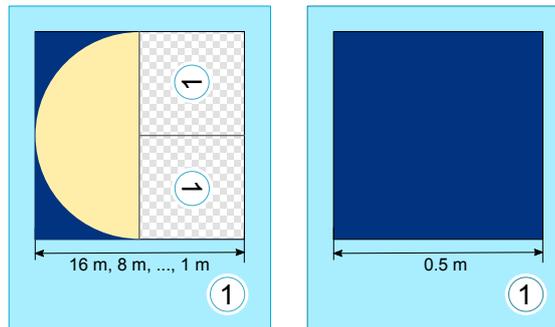


# 13. Rekursive Malerei

Tina und Ben helfen bei der Vorbereitung einer Sonderausstellung im Informatik-Museum. Auf den Boden eines Ausstellungsraums sollen sie ein 16 Meter  $\times$  16 Meter grosses Bild malen. Vom Künstler bekommen sie einen Satz Malkarten mit Anweisungen in dessen berühmter Malkartensprache, mit Hinweisen zu den Bildelementen, Massen und Drehungen. Auf manchen Malkarten sind nummerierte Felder, die auf andere Malkarten verweisen. Hier ein Beispiel aus einem früheren Malkartenprojekt. Wenn man diese drei Malkarten richtig ausführt, entsteht das Bild eines Bibers. Beachte dabei die unterschiedlichen Kantenlängen der Karten.

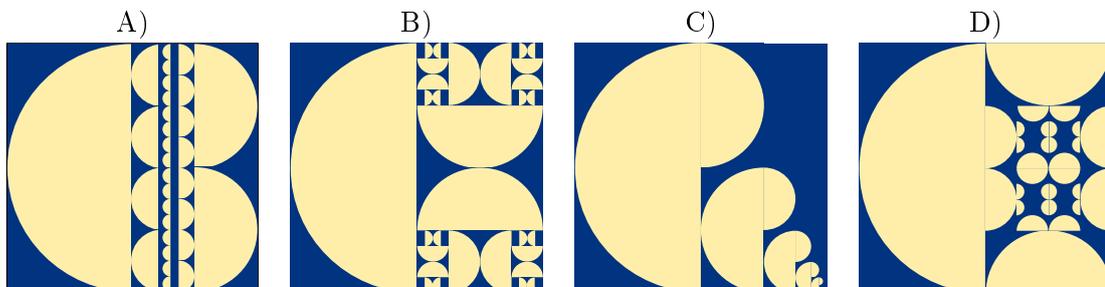


Für die Sonderausstellung bekommen Tina und Ben nun die folgenden zwei Malkarten:



Ben runzelt die Stirn. „Wie soll das gehen? Die linke Karte verweist auf sich selbst, und ausserdem haben beide Karten die selbe Nummer!“ Tina lacht: „Wir kriegen das hin! Zuerst verwenden wir nur die linke Karte. Die rechte Karte wird uns später anweisen, wann wir mit dem Malen aufhören sollen.“

*Wie wird der Boden des Ausstellungsraums aussehen?*





## Lösung

Die linke Malkarte weist an, dass die linke Hälfte des Bodens mit einer Halbkreisfläche gefüllt werden soll, deren runde Seite nach links zeigt. Für die rechte Hälfte soll die selbe Malkarte zwei Mal verwendet werden, nur eben kleiner. Die Orientierungen der Bildelemente muss den Orientierungen der Einsen entsprechen.

Bei dem oberen Bildelement ist die Eins um  $90^\circ$  nach links gedreht. Deshalb muss das Bildelement ebenfalls nach links gedreht sein und die Rundung der Halbkreisfläche liegt unten.

Bei dem unteren Bildelement ist die Eins um  $90^\circ$  nach rechts gedreht. Deshalb muss das Bildelement ebenfalls nach rechts gedreht sein, und die Rundung dieser Halbkreisfläche liegt oben.

Das ist allein bei Antwort B) der Fall. Daher ist B) die einzige richtige Antwort.

## Dies ist Informatik!

In der Informatik werden Anweisungen, die sich selbst aufrufen, als „rekursiv“ bezeichnet. Der Begriff kommt von lateinisch „recurrere“ (deutsch für „zurücklaufen“, hier im Sinne von „zu sich selber zurückkehren“). Rekursion ist ein mächtiges Konzept. Manche komplexen Aufgaben kann man kurz und überschaubar mit einer rekursiven Anweisung lösen. Manche rekursiv formulierten Programme hingegen sind nur sehr schwer zu verstehen.

Eine rekursive Anweisung enthält neben dem Rekursionsaufruf auch immer eine Bedingung, die festlegt, wann die Rekursion abgebrochen werden soll. Sonst arbeitet die Rekursion unendlich lange (oder bis irgend eine Ressource wie der Arbeitsspeicher erschöpft ist).

## Webseiten und Stichwörter

Programmieren, Rekursion, Abbruchbedingung

- <https://de.wikipedia.org/wiki/Rekursion>



# 14. Formenspiel

Alicia spielt ein Spiel mit geometrischen Formen. Dabei ersetzt sie alle Formen nach bestimmten Regeln, die sie für jedes Spiel neu festlegt. Alicia fängt immer mit einer einzigen Form an. Eine mögliche Regelmenge könnte sein:

$$\square \rightarrow \triangle \triangle \text{ und } \triangle \rightarrow \square \triangle \square$$

Wenn Alicia mit einem Quadrat startet und die beiden Regeln oben befolgt, wären die ersten drei Schritte:



In einer anderen Spielrunde hat Alicia diese Formenreihe produziert:



Welche Ersetzungsregeln hat sie sich für diese Spielrunde überlegt?

- |   |   |   |   |
|---|---|---|---|
| A)  | B)  | C)  | D)  |
| $\square \rightarrow \triangle \square \square$ | $\square \rightarrow \square \circ \circ$ | $\triangle \rightarrow \triangle \triangle$     | $\triangle \rightarrow \square \circ$               |
| $\triangle \rightarrow \circ$                   | $\triangle \rightarrow \triangle \square$ | $\square \rightarrow \circ \circ$               | $\square \rightarrow \triangle \triangle \triangle$ |
| $\circ \rightarrow \triangle \triangle$         | $\circ \rightarrow \square \triangle$     | $\circ \rightarrow \triangle \square \triangle$ | $\circ \rightarrow \square \triangle$               |



## Lösung

Die richtige Antwort ist B). Wenn Alicia mit einem Dreieck anfängt, finden die folgenden Ersetzungen statt:



Die anderen Antwortmöglichkeiten kann man durch die folgenden Überlegungen ausschliessen: Wenn Alicia bei der Antwort A) mit einem Dreieck oder einem Kreis startet, kann sie nie Quadrate erzeugen. Wenn sie mit einem Quadrat startet, erhält sie:



Da alle Ersetzungen immer länger werden, kann diese Ersetzung nicht mehr zum gewünschten Ergebnis führen.

Wenn sie bei der Antwort C) mit einem Dreieck startet, wird sie nie Quadrate oder Kreise erzeugen können. Wenn sie mit einem Quadrat startet, erhält sie:



Die beiden Dreiecke am Anfang jedoch können nie zu einem Dreieck und einem Quadrat werden, was aber gefordert ist. Wenn sie mit einem Kreis startet, erhält sie:



Auch hier hat sie wieder das Problem der beiden Dreiecke am Anfang, das sie nicht lösen kann. Bei der Antwort D) ist es nicht möglich, zwei Kreise nebeneinander zu erzeugen. Es ist noch nicht einmal möglich, mehr als einen Kreis zu erzeugen, da jeder erzeugte Kreis im nächsten Schritt wieder in ein Quadrat und ein Dreieck umgewandelt würde. Das ist aber notwendig für die gewünschte Kombination.

## Dies ist Informatik!

Die Regeln des Spiels stellen ein sogenanntes Wortersetzungssystem dar. Konkret beschreibt es eine kontextfreie Grammatik, bei der jeweils ein Wort (in diesem Fall ein geometrisches Symbol) durch ein oder mehrere Wörter (geometrische Symbole) ersetzt werden.

Eine kontextfreie Grammatik besteht aus:

- einer Menge von Wörtern (in diesem Fall die Symbole , und , üblicherweise wird noch zwischen Wörtern, die später ersetzt werden und Wörtern, die später nicht mehr ersetzt werden dürfen, unterschieden)
- einem Startsymbol (das ist in diesem Fall frei aus der Menge der Wörter wählbar)
- eine Menge von Produktionen (in unserem Falle eines der Regelsysteme)

Im Gegensatz zum Spiel von Alicia werden in kontextfreien Grammatiken aber nicht immer alle möglichen Produktionen gleichzeitig angewendet, sondern man kann auswählen, welche Produktionen man wann verwendet.

Kontextfreie Grammatiken werden an verschiedensten Stellen verwendet, von formalen Sprachen wie Programmiersprachen über Beschreibungen für natürliche Sprache bis hin zum Beschreiben von Wachstumsvorgängen in Pflanzen.



Um diese Aufgabe zu lösen, muss man neben dem Verständnis der Regeln auch die Analyse von Symbolketten vornehmen können. Dieses Konzept heisst *parsing* in der Informatik und wird dort unter anderem verwendet, um ein Computerprogramm, das in einer Programmiersprache geschrieben ist, so zu übersetzen, dass der Computer es ausführen kann.

## Webseiten und Stichwörter

Kontextfreie Grammatik, Parsing

- [https://de.wikipedia.org/wiki/Kontextfreie\\_Grammatik](https://de.wikipedia.org/wiki/Kontextfreie_Grammatik)
- <https://de.wikipedia.org/wiki/Lindenmayer-System>
- <https://de.wikipedia.org/wiki/Parser>





## 15. Egoistische Eichhörnchen

Eichhörnchen leben in Baumhöhlen. Ein Baum hat fünf übereinanderliegende Höhlen. Auf diesem Baum wohnen sechzehn Eichhörnchen. Dies bedeutet, dass die Eichhörnchen zusammen in diesen fünf Höhlen leben.

Jeden Tag kontrolliert jedes Eichhörnchen, in welcher Höhle sich die wenigsten Eichhörnchen befinden. Dies bedeutet, es zählt, wie viele Nachbarn in die Höhlen oberhalb oder unterhalb von ihnen wohnen. Für die nächste Übernachtung wird insgeheim von jedem Eichhörnchen diejenige benachbarte Höhle ausgewählt, welche die wenigsten Übernachtungen hatte. Wenn die



Höhlen untereinander dieselben Übernachtungszahlen aufweisen, bevorzugen die Eichhörnchen die eigene Höhle vor der Höhle oberhalb, und sie bevorzugen die Höhle oberhalb vor der Höhle unterhalb. Wenn sich beispielsweise heute 5, 0, 0, 4 und 7 Eichhörnchen in den Höhlen von oben bis unten befinden, wird am nächsten Tag die Situation wie folgt aussehen: Alle 5 Eichhörnchen die in der obersten Höhle übernachtet haben, werden in der Höhle gleich unterhalb ziehen (denn 0 Nachbarn sind besser als 4). Die 7 Eichhörnchen der untersten Höhle werden nach oben ziehen (4 Nachbarn sind besser als 6), und die 4 Eichhörnchen der Höhle neben der unteren Höhle werden nach oben ziehen (0 Nachbarn sind besser als 3).

*Wenn sich heute anfangs 6, 3, 3, 0 und 4 Eichhörnchen in den Höhlen von oben bis unten befinden, in wie vielen Tagen werden alle Eichhörnchen am Ende in derselben Höhle sein?*

- A) In zwei Tagen.
- B) In drei Tagen.
- C) In vier Tagen.
- D) Sie werden nie alle in der derselben Höhle sein.



## Lösung

Die richtige Lösung ist: In drei Tagen.

Heute:	6, 3, 3, 0, 4
Nach 1 Tag:	0, 9, 0, 7, 0
Nach 2 Tagen:	9, 0, 7, 0, 0
Nach 3 Tagen:	0, 16, 0, 0, 0

## Dies ist Informatik!

Dieses Problem ist ein Beispiel für Schwarmintelligenz. Die Idee solcher Algorithmen ist, dass man komplexe Probleme mit sehr einfachen Geräten lösen kann, wenn man ganz viele von diesen Geräten hat. Beispielsweise verhalten sich Ameisen nach simplen Regeln und unabhängig voneinander. Wenn es jedoch viele Ameisen hat, sind diese in der Lage, sehr anspruchsvolle Dinge zu tun, wie beispielsweise einen Ameisenhaufen zu bauen, in einem Diagramm den optimalen Weg zu suchen, eine Lösung für das Verkäuferproblem zu finden oder sogar Blätter zu zerschneiden.

In dieser Aufgabe gibt es ebenfalls eine „grosse“ Menge an Geräten (in diesem Fall durch Eichhörnchen repräsentiert), die sich nach simplen Regeln verhalten. In diesem Fall ist ihr kollektives Verhalten allerdings sehr weit davon entfernt, „intelligent“ zu sein. Sie möchten so viel Platz wie möglich haben, enden jedoch am Schluss alle in derselben Höhle. Daraus lässt sich folgern, dass das Ameisenverhalten sich nicht eins zu eins in einem Ameisen-Algorithmus übertragen lässt. Manchmal ist es eben besser zu kooperieren, als sich egoistisch zu verhalten.

## Webseiten und Stichwörter

Schwarmintelligenz, Ameisen-Algorithmen

- [https://en.wikipedia.org/wiki/Swarm\\_intelligence](https://en.wikipedia.org/wiki/Swarm_intelligence)
- [https://de.wikipedia.org/wiki/Kollektive\\_Intelligenz](https://de.wikipedia.org/wiki/Kollektive_Intelligenz)
- <https://de.wikipedia.org/wiki/Ameisenalgorithmus>



## A. Aufgabenautoren

 Nursultan Akhmetov  
 Wilfried Baumann  
 Andrea Brabcová  
 Eugenio Bravo  
 Nicolas Brunner  
 Valentina Dagienė  
 Christian Datzko  
 Susanne Datzko  
 Olivier Ens  
 Hanspeter Erni  
 Gerald Futschek  
 Haris Gavranovic  
 Martin Guggisberg

 Hans-Werner Hein  
 Filiz Kalelioğlu  
 Akiko Kikui  
 Hiroki Manabe  
 Khairul A. Mohamad Zaki  
 Hamed Mohebbi  
 Anna Morpurgo  
 Tom Naughton  
 Erkulan Nurtazanov  
 Henry Ong  
 Serena Pedrocchi  
 Wolfgang Pohl  
 Ilya Posov

 Sergei Pozdniakov  
 J. P. Pretti  
 Lorenzo Repetto  
 Kirsten Schlüter  
 Eljakim Schrijvers  
 Seiichi Tani  
 Willem van der Vegt  
 Jiří Vaníček  
 Troy Vasiga  
 Lina Vinikienė  
 Michael Weigend



## B. Sponsoring: Wettbewerb 2016

### HASLERSTIFTUNG

<http://www.haslerstiftung.ch/>

Stiftungszweck der Hasler Stiftung ist die Förderung der Informations- und Kommunikationstechnologie (IKT) zum Wohl und Nutzen des Denk- und Arbeitsplatzes Schweiz. Die Stiftung will aktiv dazu beitragen, dass die Schweiz in Wissenschaft und Technologie auch in Zukunft eine führende Stellung innehat.



<http://www.roborobo.ch/>

Die RoboRobo Produkte fördern logisches Denken, Vorstellungsvermögen, Fähigkeiten Abläufe und Kombinationen auszudenken und diese systematisch aufzuzeichnen.

Diese Produkte gehören in innovative Schulen und fortschrittliche Familien. Kinder und Jugendliche können in einer Lektion geniale Roboter bauen und programmieren. Die Erwachsenen werden durch die Erfolgserlebnisse der „Erbauer“ miteinbezogen.

RoboRobo ist genial und ermöglicht ein gemeinsames Lern-Erlebnis!



<http://www.digitec.ch/>

digitec ist der Online-Marktführer der Schweiz. Egal, ob Fernseher, Smartphones oder Grafikkarten – bei digitec findest du alles rund um IT, Unterhaltungselektronik und Telekommunikation. Überzeuge dich selbst von der grossen Auswahl und stöbere in über 100'000 Produkten zu den besten Preisen.



<http://www.baerli-biber.ch/>

Schon in der vierten Generation stellt die Familie Bischofberger ihre Appenzeller Köstlichkeiten her. Und die Devise der Bischofbergers ist dabei stets dieselbe geblieben: «Hausgemacht schmeckt's am besten». Es werden nur hochwertige Rohstoffe verwendet: reiner Bienenhonig und Mandeln allererster Güte. Darum ist der Informatik-Biber ein „echtes Biberli“.



<http://www.verkehrshaus.ch/>



Kanton Zürich  
Volkswirtschaftsdirektion  
Amt für Wirtschaft und Arbeit

Standortförderung beim Amt für Wirtschaft und Arbeit  
Kanton Zürich



### i-factory (Verkehrshaus Luzern)

Die i-factory bietet ein anschauliches und interaktives Erproben von vier Grundtechniken der Informatik und ermöglicht damit einen Erstkontakt mit Informatik als Kulturtechnik. Im optischen Zentrum der i-factory stehen Anwendungsbeispiele zur Informatik aus dem Alltag und insbesondere aus der Verkehrswelt in Form von authentischen Bildern, Filmbeiträgen und Computer-Animationen. Diese Beispiele schlagen die Brücke zwischen der spielerischen Auseinandersetzung in der i-factory und der realen Welt.

<http://www.ubs.com/>

Wealth Management IT and UBS Switzerland IT



<http://www.bbv.ch/>

bbv Software Services AG ist ein Schweizer Software- und Beratungsunternehmen. Wir stehen für Top-Qualität im Software Engineering und für viel Erfahrung in der Umsetzung. Wir haben uns zum Ziel gesetzt, unsere Expertise in die bedeutendsten Visionen, Projekte und Herausforderungen unserer Kunden einzubringen. Wir sind dabei als Experte oder ganzes Entwicklungsteam im Einsatz und entwickeln individuelle Softwarelösungen.

Im Bereich der Informatik-Nachwuchsförderung engagiert sich die bbv Software Services AG sowohl über Sponsoring als auch über die Ausbildung von Lehrlingen. Wir bieten Schnupperlehrtage an und bilden Informatiklehrlinge in der Richtung Applikationsentwicklung aus. Mehr dazu erfahren Sie auf unserer Website in der Rubrik Nachwuchsförderung.



<http://www.presentex.ch/>

Beratung ist keine Nebensache

Wir interessieren uns, warum, wann und wie die Werbeartikel eingesetzt werden sollen – vor allem aber, wer angesprochen werden soll.



<https://www.hslu.ch/de-ch/informatik/agenda/veranstaltungen/fuer-schulen/itgirls/>

HLSU, Lucerne University of Applied Sciences and Arts Engineering & Architecture



<http://www.phlu.ch/>

Pädagogische Hochschule Luzern



**ABZ**

AUSBILDUNGS- UND BERATUNGSZENTRUM  
FÜR INFORMATIKUNTERRICHT

<http://www.abz.inf.ethz.ch/>

Ausbildungs- und Beratungszentrum für Informatikunter-  
richt der ETH Zürich.



## C. Weiterführende Angebote

### Das Lehrmittel zum Informatik-Biber

#### Module

Verkehr – Optimieren

Musik – Komprimieren

Geheime Botschaften – Verschlüsseln

Internet – Routing

Apps

Auszeichnungssprachen

<http://informatik-biber.ch/einleitung/>

Das Lehrmittel zum Biber-Wettbewerb ist ein vom SVIA, dem schweizerischen Verein für Informatik in der Ausbildung, initiiertes Projekt und hat die Förderung der Informatik in der Sekundarstufe I zum Ziel.

Das Lehrmittel bringt Jugendlichen auf niederschwellige Weise Konzepte der Informatik näher und zeigt dadurch auf, dass die Informatikbranche vielseitige und spannende Berufsperspektiven bietet.

Lehrpersonen der **Sekundarstufe I** und weiteren interessierten Lehrkräften steht das Lehrmittel als Ressource zur Vor- und Nachbereitung des Wettbewerbs kostenlos zur Verfügung.

Die sechs Unterrichtseinheiten des Lehrmittels wurden seit Juni 2012 von der LerNetz AG in Zusammenarbeit mit dem Fachdidaktiker und Dozenten Dr. Martin Guggisberg der PH FHNW entwickelt. Das Angebot wurde zweisprachig (Deutsch und Französisch) entwickelt.



I learn it: <http://ilearnit.ch/>

In thematischen Modulen können Kinder und Jugendliche auf dieser Website einen Aspekt der Informatik auf deutsch und französisch selbständig entdecken und damit experimentieren. Derzeit sind sechs Module verfügbar.



Der Informatik-Biber neu auf Facebook:

<https://www.facebook.com/informatikbiberch>

010100110101011001001001  
010000010010110101010011  
010100110100100101000101  
001011010101001101010011  
010010010100100100100001

**SV!A**

[www.svia-ssie-ssii.ch](http://www.svia-ssie-ssii.ch)  
schweizerischer vereinfürinformatikind  
erausbildung//sociétésuissedel'inform  
atiquedansl'enseignement//societàsviz  
zeraperl'informaticanell'insegnamento

Werden Sie SVIA Mitglied – <http://svia-ssie-ssii.ch/svia/mitgliedschaft> und unterstützen Sie damit den Informatik-Biber.

Ordentliches Mitglied des SVIA kann werden, wer an einer schweizerischen Primarschule, Sekundarschule, Mittelschule, Berufsschule, Hochschule oder in der übrigen beruflichen Aus- und Weiterbildung unterrichtet.

Als Kollektivmitglieder können Schulen, Vereine oder andere Organisationen aufgenommen werden.