

INFORMATIK-BIBER

Aufgaben und Lösungen 2014

Herausgeber:
Wolfgang Pohl, BWINF
Hans-Werner Hein, BWINF
Agnieszka Wolny, BWINF
Christian Sebaly, BWINF

Aufgabenausschuss Informatik-Biber 2014

Christiane Beyer, Rhein-Sieg-Gymnasium Sankt Augustin

Hans-Werner Hein, BWINF Bonn

Ulrich Kiesmüller, Simon-Marius-Gymnasium Gunzenhausen

Wolfgang Pohl, BWINF Bonn

Kirsten Schlüter, St.-Emmeram-Realschule Aschheim

Nicole Schulte, Petrus-Legge-Gymnasium Brakel

Michael Weigend, Holzkamp-Gesamtschule Witten

Die deutschsprachigen Fassungen der Aufgaben wurden auch in Österreich und der Schweiz verwendet. An ihrer Erstellung haben mitgewirkt:

Ivo Blöchliger, Kantonsschule Wohlen

Christian Datzko, Wirtschaftsgymnasium und Wirtschaftsmittelschule Basel

Jürgen Frühwirth, Technische Universität Wien

Gerald Futschek, Technische Universität Wien

Peter Garscha, Technische Universität Wien

Bernd Kurzmann, BG / BRG / WISKU 11, Wien

Roman Ledinsky, Technische Universität Wien

Barbara Müllner, Bundesrealgymnasium Bad Vöslau

Jacqueline Peter, Schweizerischer Verein für Informatik in der Ausbildung (SVIA)

Der Informatik-Biber ist ein Projekt der Bundesweiten Informatikwettbewerbe (BWINF) und das Einstiegsformat zum Bundeswettbewerb Informatik.

BWINF ist eine Initiative der Gesellschaft für Informatik (GI), des Fraunhofer-Verbunds IuK-Technologie und des Max-Planck-Instituts für Informatik.

BWINF wird vom Bundesministerium für Bildung und Forschung (BMBF) gefördert. Die Bundesweiten Informatikwettbewerbe gehören zu den von den Kultusministerien empfohlenen Schülerwettbewerben.

Einleitung

Der Informatik-Biber ist ein Online-Test mit Aufgaben zur Informatik. Er erfordert Köpfchen, aber keine Vorkenntnisse.

Der Informatik-Biber will das allgemeine Interesse für das Fach Informatik wecken und gleichzeitig die Motivation für eine Teilnahme am Bundeswettbewerb Informatik stärken.

Schülerinnen und Schüler, die mehr wollen, sind herzlich eingeladen, sich anschließend am Bundeswettbewerb Informatik zu versuchen (siehe Seite 59).

Der Informatik-Biber findet jährlich im November statt. An der 8. Austragung im Jahr 2014 beteiligten sich 1232 Schulen mit 217.506 Schülerinnen und Schülern.



Der deutsche Informatik-Biber ist Partner der internationalen Initiative Bebras (siehe Seite 4). 2013 nahmen 680.000 Schülerinnen und Schüler aus 26 Ländern daran teil.

Auf den folgenden Seiten finden sich die 39 Aufgaben des Informatik-Biber 2014. Im oberen hellblauen Feld sind Schwierigkeitsgrade und Altersstufen vermerkt. Die grau unterlegten Felder am Seitenende enthalten Erläuterungen zu den Lösungen und Lösungswegen sowie eine kurze Darstellung des Aufgabenthemas hinsichtlich seiner Relevanz in der Informatik. Die Interaktivität der Aufgaben hat sich weiter diversifiziert. Es kommen vor:

- 17 mal „Klicke auf einen von n Antworttexten“,
- 7 mal „Klicke auf eines von n Bildern“,
- 6 mal „Klicke auf mehrere von n Objekten“,
- 5 mal „Schreibe eine Zahl/einen Buchstaben in ein Feld“,
- 4 mal „Schiebe mehrere von n Objekten auf eine andere Position“.

Es wurden von den Teilnehmenden neben den herkömmlichen Bürocomputern und Laptops vermehrt auch Tablets eingesetzt.

Der Informatik-Biber 2014 wurde in vier Altersgruppen durchgeführt: Stufen 5 bis 6, 7 bis 8, 9 bis 10 und 11 bis 13. In jeder Altersgruppe waren innerhalb von 40 Minuten 18 Aufgaben zu lösen, jeweils sechs davon in den drei Schwierigkeitsstufen leicht, mittel und schwer.

Die Veranstalter bedanken sich bei allen Lehrkräften, die mit einem weit über die Pflichten hinausgehenden Engagement es ihren Klassen möglich gemacht haben, den Informatik-Biber zu erleben. Wir laden die Schülerinnen und Schüler ein, auch im November 2015 wieder teilzunehmen.

Bebras: International Contest on Informatics and Computer Fluency



Der niederländische Biber

Der deutsche Informatik-Biber ist Partner der internationalen Initiative Bebras. 2004 fand in Litauen der erste Bebras Contest statt. 2006 traten Estland, Niederlande und Polen der Initiative bei, und auch Deutschland veranstaltete im Jahr der Informatik als „El:Spiel blitz!“ einen ersten Biber-Testlauf. Seitdem kamen viele Bebras-Länder hinzu. Zum Drucktermin sind es weltweit 30 und weitere Länderteilnahmen sind in Planung. Insgesamt hatte der Bebras Contest 2013 über 680.000 Teilnehmerinnen und Teilnehmer.

Die Bebras-Community erarbeitet jedes Jahr auf einem Workshop anhand von Vorschlägen der Länder eine größere Auswahl möglicher Aufgaben. Die Aufgaben des Informatik-Biber 2014 stammen aus 16 Ländern: Australien, Deutschland, Finnland, Frankreich, Japan, Kanada, Litauen, Österreich, Russland, Schweiz, Slowakei, Slowenien, Spanien, Tschechien, Ukraine und Ungarn.



Der ukrainische Biber



Die australischen Biber

Deutschland nutzt zusammen mit einer Vielzahl anderer Länder zur Durchführung des Bebras Contest ein gemeinsames Online-System. Dieses „International Bebras Challenge System“ wird von der niederländischen Firma Eljakim IT betrieben und fortentwickelt.

Informationen über die Aktivitäten aller Bebras-Länder finden sich unter:

bebras.org

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Am Rand entlang

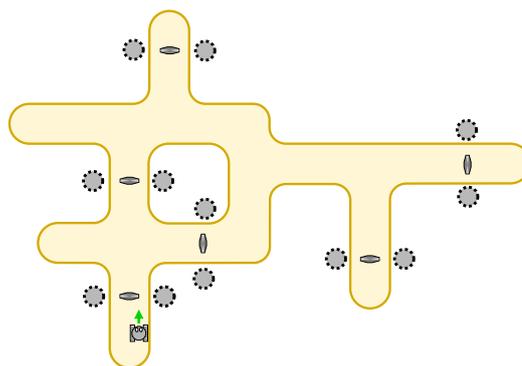
Ein Roboter fährt immer am Rand seiner Fahrbahn entlang. Der Roboter kann die folgenden Anweisungen bekommen und ausführen:

Anweisung	Ausführung
START-GO	Starte den Motor und fahre in der Startrichtung los.
GO	Fahre weiter am Rand entlang.
CROSS-GO	Wechsle zum anderen Rand der Fahrbahn und fahre in der gleichen Richtung weiter.
STOP	Bleib stehen.

Wenn der Roboter steht, muss er zuerst die Anweisung **START-GO** bekommen. Auf der Fahrbahn sind Steuermarken. Immer wenn der Roboter über eine Steuermarke fährt, führt er die nächste Anweisung aus. Das Bild zeigt die Fahrbahn des Roboters mit den Steuermarken. Unten siehst du den Roboter und seine Startrichtung.

Der Roboter steht.
Nun bekommt er diese Anweisungen:

START-GO
CROSS-GO
GO
GO
GO
STOP



An welcher Stelle bleibt der Roboter stehen?
Die grauen Kreise im Bild zeigen die möglichen Stellen.

So ist es richtig:

Der Roboter bleibt rechts am oberen Fahrbahnrand stehen. Das Bild zeigt den Verlauf der Fahrt.

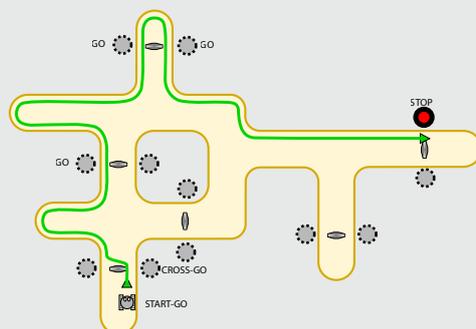
Das ist Informatik!

Fahrroboter (also automatische Fahrzeuge ohne Fahrer) findet man zum Beispiel auf Flughäfen und in Fabriken.

Diese Maschinen werden durch Programme gesteuert. Im einfachsten Fall ist ein Programm eine simple Folge von Anweisungen – wie in dieser Biber-Aufgabe.

Bei echten Fahrrobotern können die Programme aber deutlich komplizierter sein.

In der Informatik arbeiten viele Leute an Programmen für Roboter: Fahrroboter, Konstruktionsroboter, Medizinroboter, Fußballroboter, Flugroboter und so weiter. Das Verhalten von Robotern wirkt sich oft auch auf deren Umwelt und damit auch auf Menschen aus. Programme für Roboter müssen deshalb besonders verlässlich sein.





Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Baumstammbilder

Wenn die Biber Baumstämme zerteilen, legen sie die Teile auf eine besonders kunstvolle Art aus. Am Anfang liegt nur ein einfacher großer Baumstamm da. Der wird dann auf eine bestimmte Art durch kürzere Baumstämme ersetzt. Diese kürzeren Baumstämme werden wiederum auf dieselbe Art durch noch kürzere Baumstämme ersetzt.

Hier sieht man drei verschiedene Beispiele:

Anfang			
Erstes Ersetzen			
Zweites Ersetzen			

Wenn es nach dem zweiten Ersetzen so aussieht:



Wie sah es nach dem ersten Ersetzen aus?

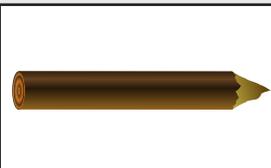
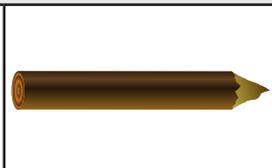
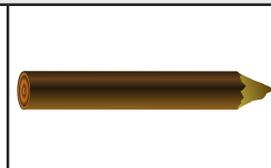
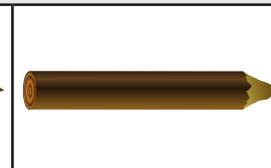
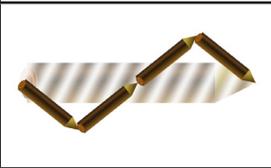
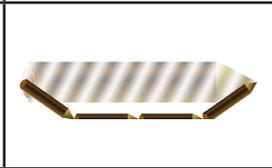
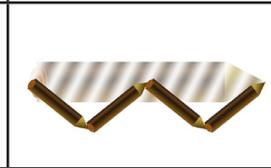
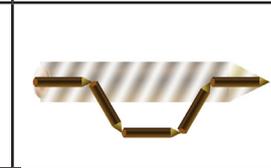
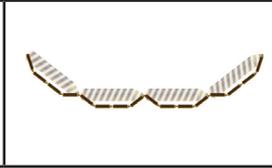
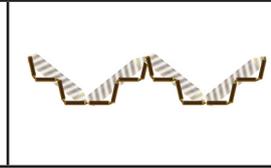
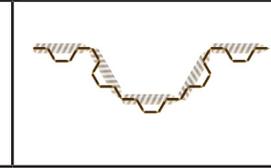
- A)
- B)
- C)
- D)

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Antwort A ist richtig:

Bei den anderen Antworten würde es nach dem ersten Ersetzen jeweils anders aussehen:

Antwort A	Antwort B	Antwort C	Antwort D
			
			
			

Das ist Informatik!

Die Ergebnisse von Ersetzungsfolgen dieser Art werden Fraktale genannt. Die besondere Eigenschaft von Fraktalen ist es, dass sie sich auf den verschiedenen Größenstufen selbstähnlich sind.

Theoretisch entsteht ein Fraktal erst durch unendlich viele Ersetzungen. In der Praxis der Informatik kann man so lange nicht warten und bricht daher die Ersetzungsfolge ab, wenn ein genügend brauchbares Zwischenergebnis erreicht ist.

Das zweite Beispiel in dieser Biber-Aufgabe ist ein besonders bekanntes Fraktal; es heißt nach seinem Erfinder Koch-Kurve.

<http://de.wikipedia.org/wiki/Koch-Kurve>



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

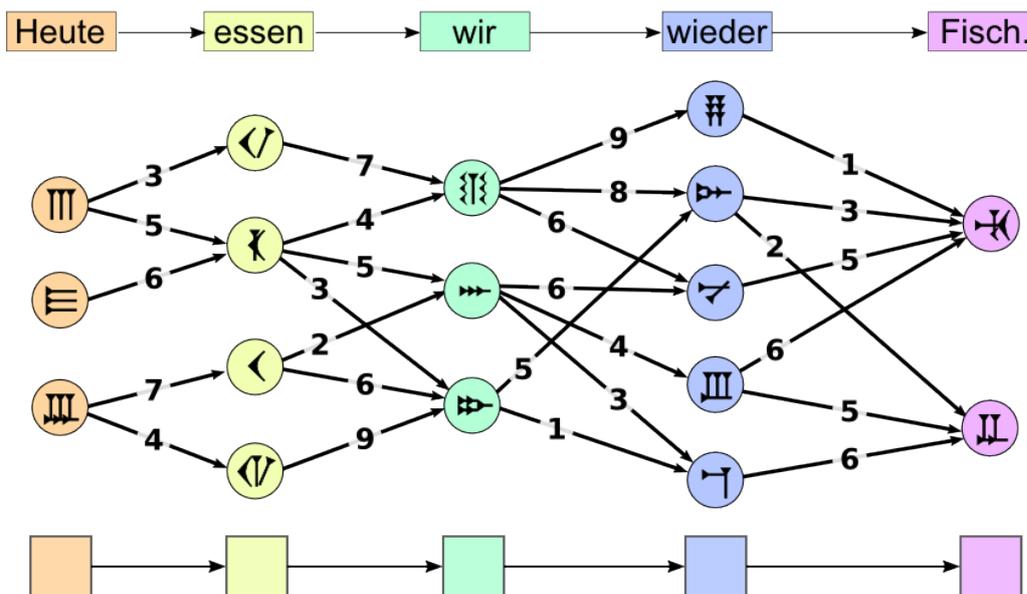
Beste Übersetzung

Betty erforscht, wie deutsche Sätze in eine uralte Zeichensprache übersetzt werden. Jedes Wort wird in ein Zeichen übersetzt. Dabei gibt es meist mehrere Möglichkeiten. Deshalb berücksichtigt Betty, welche Zeichen in der Übersetzung aufeinander folgen.

Betty bereitet die Übersetzung eines Satzes gründlich vor: Zunächst malt sie unter jedem Wort auf, in welche Zeichen sie das Wort übersetzen kann. Außerdem verbindet sie Zeichenpaare, die in der Übersetzung aufeinander folgen können, mit Pfeilen und gibt mit „Passzahlen“ an, wie gut die beiden Zeichen zueinander passen.

Die beste Übersetzung ist nun die Folge von Zeichen, bei der die Summe der Passzahlen aller aufeinander folgenden Zeichenpaare möglichst groß ist.

Für die Übersetzung des Satzes „Heute essen wir wieder Fisch.“ hat Betty alles vorbereitet:

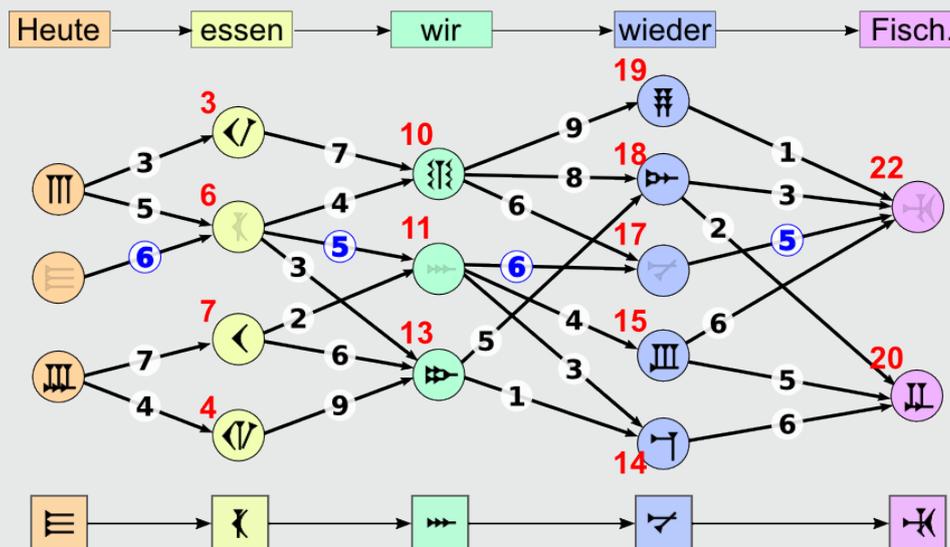


Bestimme die beste Übersetzung!

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



So ist es richtig!



Für diese Übersetzung ist die Summe der zugehörigen Passzahlen (blau gefärbt):
 $6 + 5 + 6 + 5 = 22$.

Dies ist die größtmögliche Summe.

Wie kann man die beste Übersetzung schnell herausfinden, ohne alle Möglichkeiten auszuprobieren? Eine gute Methode ist, von links nach rechts für jedes Zeichen zu bestimmen, wie die größte Passzahlsumme bis zu diesem Zeichen lautet. Dazu muss man nur die vorher bestimmten besten Wege der Vorgängerzeichen berücksichtigen – und nicht alle Wege. Das Bild zeigt für jedes Zeichen in Rot die so ermittelten Werte. Ganz rechts sieht man, dass 22 wirklich die insgesamt größte Summe ist.

Wenn man sich gemerkt hat, von wo aus man jeweils zur besten Summe gekommen ist, kann man vom besten Gesamtergebnis aus (also 22) rückwärts (entlang der blauen Zahlen) die beste Übersetzung bestimmen.

Die hier verwendete Methode, eine optimale Möglichkeit von links nach rechts schrittweise zu konstruieren, heißt „Dynamische Programmierung“.

Das ist Informatik!

Das Internet bietet die technischen Möglichkeiten dafür, dass wir uns leicht mit anderen Menschen auf der ganzen Welt austauschen können. Doch die meisten Menschen sprechen eine andere Sprache. Da sind Informatiksysteme hilfreich, die Texte oder sogar gesprochene Sprache automatisch übersetzen können.

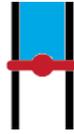
Moderne Übersetzungssysteme berücksichtigen – wie Betty –, welche Wörter in der Zielsprache der Übersetzung häufig gemeinsam vorkommen. Solche Wortmuster finden die Systeme, indem sie möglichst viele Texte statistisch untersuchen. Die Grammatik der Sprachen wird hingegen nicht so genau beachtet. Das würde die Menge der zu verarbeitenden Informationen zu sehr in die Höhe treiben und die Übersetzung verlangsamen. Deshalb fallen automatische Übersetzungen häufig etwas merkwürdig aus.



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Bewässerung

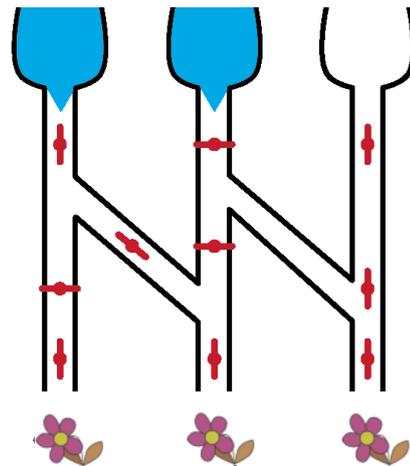
Wenn das Ventil zu ist,
fließt kein Wasser.



Wenn das Ventil offen ist,
fließt Wasser durch.



Welche der drei durstigen Blumen bekommen
Wasser bei dieser Stellung der Ventile?



Blumen, die Wasser bekommen, sollen so  aussehen.

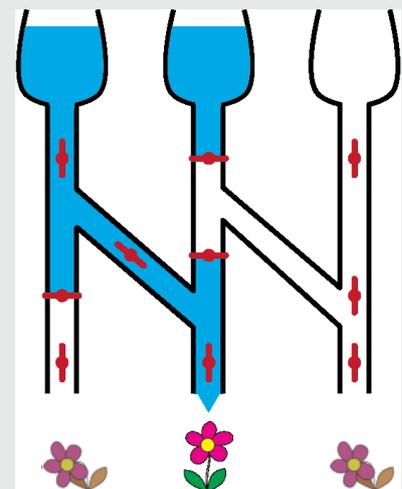
Blumen, die kein Wasser bekommen, sollen so  aussehen.

So ist es richtig:

Nur die mittlere Blume bekommt Wasser
bei dieser Stellung der Ventile.

Das ist Informatik!

Für die Informatik ist unser Bewässerungssystem eine Schaltung. Die Ventile sind die Schalter – mit den zwei Stellungen „auf“ und „zu“. Entsprechend den Eingabetrichtern und den Schalterstellungen bewegen sich die Informationen „Wasser fließt“ und „Wasser fließt nicht“ durch die Schaltung – bis hin zu den Blumen. Elektronik-Geräte enthalten elektronische Schaltungen, durch die Elektrizität fließt. Bei Schaltungen mit Glasfasern fließen die Informationen als Laserlicht. Es gibt robotische Geräte, die in Umgebungen arbeiten müssen, in denen elektronische Schaltungen schnell kaputt gehen: Starke Magnetfelder, hohe Feuchtigkeit, extreme Temperaturen. Solche Robotik kann schon mal robuste Schaltungen enthalten, in denen Hydrauliköl oder Pressluft fließen.



<<http://de.wikipedia.org/wiki/Elektronik>>
<<http://de.wikipedia.org/wiki/Fluidik>>

<<http://de.wikipedia.org/wiki/Photonik>>
<<http://de.wikipedia.org/wiki/Hydraulik>>

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Biber-Ausweis

Jeder Biber hat einen Ausweis mit einer Ausweisnummer.
Um Lesefehlern vorzubeugen, trägt jeder Ausweis noch einen Prüfbuchstaben.

Der Prüfbuchstabe wird so ermittelt:

1. Zähle die Ziffern der Ausweisnummer zusammen.
2. Suche das Ergebnis in der Tabelle.
3. In der gleichen Zeile steht rechts der passende Prüfbuchstabe.

Ergebnis	Prüfbuchstabe
0 7 14 21 28	T
1 8 15 22 29	R
2 9 16 23 30	W
3 10 17 24 31	A
4 11 18 25 32	G
5 12 19 26 33	M
6 13 20 27 34	Y

Biber-Ausweis



Erik Musterbiber
Wasserburg am See

Ausweisnummer
4517

Prüfbuchstabe

Schreibe den passenden Prüfbuchstaben in den Biber-Ausweis!

Der Prüfbuchstabe „A“ ist richtig:

4+5+1+7 ergibt 17.

Das Ergebnis 17 steht in der Tabelle als dritte Zahl der vierten Zeile.
In der vierten Zeile steht rechts dazu der Prüfbuchstabe „A“.

Das ist Informatik!

Die Informatik hat viele Methoden und Geräte entwickelt, um Zeichengruppen einzulesen, welche in alltäglichen Situationen die „Identität“ eines Objekts oder einer Person behaupten. Die Prüfung von Identität kann in vielen Bereichen wichtig sein: der Wert einer Banknote oder eines Gutscheins, die Gültigkeit einer Konzertkarte oder eines Flugtickets, Kennzeichen von Autos und anderen Fahrzeugen. All dies – und vieles mehr – muss sicher erkannt werden können. Beim maschinellen Einlesen von Zeichengruppen kommt es aber manchmal zu einem Lesefehler. Wenn der Fehler nicht sofort bemerkt wird, kann das später äußerst ärgerlich ausgehen – für den Geprüften, für den Prüfer, oder für beide. Eine sehr verbreitete Methode, Lesefehler erkennbar zu machen, ist es, mit einem Algorithmus zu einer Identitäts-Zeichengruppe ein oder mehrere Prüfzeichen zu berechnen und hinzuzufügen. Bei einem Lesefehler passen dann gelesene Zeichengruppe und Prüfzeichen meist nicht zusammen.

Übrigens: Autokennzeichen enthalten keine Prüfzeichen.



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Biber in der Grube

Die Biber gehen häufig in Gruppen durch den dunklen Wald. Im Wald sind die Pfade sehr schmal. Deshalb gehen sie dort immer in einer Reihe, ohne sich zu überholen.

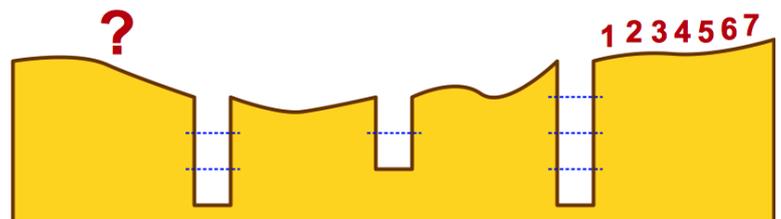
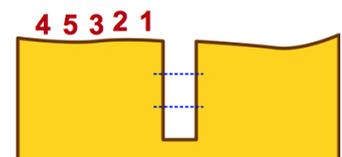
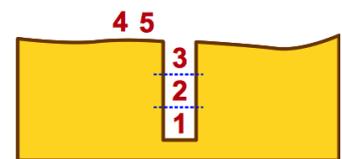
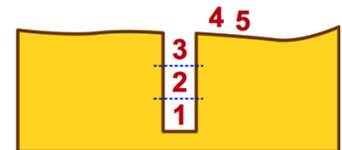
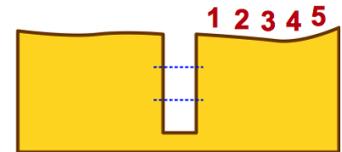
Auf den Pfaden im Wald gibt es viele Gruben. Die Biber überwinden eine Grube so: Zuerst springen so viele Biber in die Grube, wie dort hinein passen. Danach gehen die anderen aus der Gruppe über die volle Grube. Schließlich klettern die Biber in der Grube der Reihe nach aus der Grube hinaus. Dann kann die Gruppe weitergehen.

Die Bilder zeigen, wie 5 Biber eine Grube überwinden. In diese Grube passen 3 Biber hinein.

Eine Gruppe von 7 Bibern geht durch den dunklen Wald. Die Biber müssen drei Gruben überwinden. In die erste Grube passen 4 Biber, in die zweite passen 2 Biber, und in die dritte passen 3 Biber.

In welcher Reihenfolge gehen die Biber nach der dritten Grube weiter?

- A) 4 7 5 6 1 2 3
- B) 2 1 6 5 3 4 7
- C) 6 5 7 4 3 2 1
- D) 5 7 6 1 4 3 2



Antwort B ist richtig:

Zu Beginn ist die Reihenfolge: 1 2 3 4 5 6 7
 Nach der ersten Grube (in die 4 Biber passen): 5 6 7 4 3 2 1
 Nach der zweiten Grube (in die 2 Biber passen): 7 4 3 2 1 6 5
 Nach der dritten Grube (in die 3 Biber passen): 2 1 6 5 3 4 7

Das ist Informatik!

Die strukturierte Organisation von Daten ist in der Informatik sehr wichtig, damit man effizient auf gespeicherte Daten wieder zugreifen kann. Es gibt viele verschiedene Arten, Daten zu speichern, und man muss für die jeweilige Anwendungssituation die passende Art auswählen. Eine häufig verwendete Struktur ist die in diesem Beispiel gezeigte – der Stapel (engl. „stack“).

Legt man etwas am Stapel ab, so verdeckt dies die zuvor abgelegte Information, man kann also immer nur die zuletzt abgelegte Information wieder abrufen. Eine solche Struktur wird auch LIFO-Struktur genannt, nach der englischen Bezeichnung für „last in, first out“.

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Brezeln

Zwei Biber arbeiten in einer Bäckerei. Susanna ist die Bäckerin. Sie holt immer drei Brezeln gleichzeitig aus dem Ofen und hängt sie von rechts an die Stange: zuerst eine A-Brezel, dann eine B-Brezel und zum Schluss eine O-Brezel. Peter ist der Verkäufer. Er verkauft immer die Brezel, die ganz rechts an der Stange hängt. Susanna backt schneller als Peter die Brezeln verkaufen kann.



Wie viele Brezeln hat Peter mindestens verkauft, wenn die Stange wie im Bild behängt ist?

- A) 5 Brezeln
- B) 7 Brezeln
- C) 9 Brezeln
- D) 11 Brezeln

Antwort C ist richtig:

Susanna muss mindestens sechsmal 3 Brezeln (das sind 18 Stück) an die Stange hängen, damit 6 A-Brezeln übrigbleiben können. Insgesamt 9 Brezeln sind noch da, also hat Peter mindestens 9 Stück verkauft: 4 B-Brezeln und 5 O-Brezeln.

Wie viele komplette ABO-Brezel-Tripel Peter darüber hinaus noch verkauft hat, bleibt unbekannt.

Das ist Informatik!

Die Stange verbildlicht einen Kellerspeicher (stack). Das ist in der Informatik ein Speicherkonzept, bei dem eine neue Information nur auf die „zuoberst“ liegende Information abgelegt werden kann (push) und immer nur die „zuoberst“ liegende Information entnommen werden kann (pop).

An der Stange können neue Brezeln nur zuvorderst aufgehängt und weggenommen werden. Hier ist das „zuoberst“ des Kellerspeicher-Konzepts als „zuvorderst“ implementiert.

Die Zugriffsweise auf Kellerspeicher wird LIFO („last in, first out“) genannt.



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

De-Anonymisierung

Krankenakten enthalten sensible persönliche Daten, die nicht öffentlich bekannt sein sollten. Zu Forschungszwecken hat ein Krankenhaus aktuelle Daten deshalb anonymisiert veröffentlicht; die erste Tabelle zeigt einen Auszug aus dieser Liste.

Geb.Datum	Geschlecht	PLZ	Krankheit
01.01.1974	männlich	29400	Diabetes
01.01.1976	männlich	18250	Lungenkrebs
01.01.1976	weiblich	29400	Brustkrebs
01.01.1976	weiblich	29400	Fehlgeburt
01.01.1984	weiblich	18250	Herzanfall
01.01.1985	weiblich	16300	Brustkrebs
01.01.1987	weiblich	25340	Hautkrebs
01.01.1988	männlich	18250	Diabetes
01.01.1988	weiblich	18250	Grippe

Gleichzeitig hat – wegen anstehender Wahlen – die Gemeinde mit PLZ 18250 eine Liste der Wahlberechtigten veröffentlicht; die zweite Tabelle zeigt die Daten **ALLER** Wahlberechtigten, die an einem 1. Januar geboren wurden.

Geb.Datum	Geschlecht	Name
01.01.1958	weiblich	Melanie Meyer
01.01.1976	männlich	Georg Schmidt
01.01.1976	männlich	Robert Schlumpf
01.01.1984	weiblich	Kathrin Frei
01.01.1984	weiblich	Eva Müller
01.01.1988	weiblich	Agnes Bachmann
01.01.1988	männlich	Roman Schröder
01.01.1988	weiblich	Isabelle Beyer
01.01.1989	männlich	Martin Klaus

Anhand der beiden Tabellen kannst du eine Person mit Namen identifizieren (de-anonymisieren), die mit absoluter Sicherheit eine Krankheit hat.

Wie lautet der Name dieser Person?

- A) Georg Schmidt
- B) Eva Müller
- C) Roman Schröder
- D) Isabelle Beyer

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Antwort C ist richtig:

Die Patienten der Zeilen 1, 3, 4, 6 und 7 können es nicht sein, da sie nicht in der Gemeinde mit der Postleitzahl 18250 wohnen. Der Patient der Zeile 2 ist im Jahr 1976 geboren, männlich und wohnt in der Gemeinde mit der Postleitzahl 18250. Es gibt aber zwei Einwohner, die diesen Daten entsprechen: Georg Schmidt und Robert Schlumpf.

Die Patientin der Zeile 5 ist im Jahr 1984 geboren, weiblich und wohnt in der Gemeinde mit der Postleitzahl 18250. Es gibt aber zwei Einwohnerinnen, die diesen Daten entsprechen: Kathrin Frei und Eva Müller.

Die Patientin in der Zeile 9 ist im Jahr 1988 geboren, weiblich und wohnt in der Gemeinde mit der Postleitzahl 18250. Es gibt aber zwei Einwohnerinnen, die diesen Daten entsprechen: Agnes Bachmann und Isabelle Beyer.

Der Patient in der Zeile 8 jedoch, geboren im Jahr 1988, männlich und in der Gemeinde mit der Postleitzahl 18250 wohnend, lässt sich eindeutig als Roman Schröder identifizieren.

Das ist Informatik!

Wenn Daten über einen Menschen gespeichert sind, dann kann man diesen Menschen anhand seiner Daten eindeutig wiedererkennen. Oder nicht: Dann bleibt der Mensch anonym. Ob jemand anonym bleibt, hängt von der Bedeutung der gespeicherten Daten und von der Anzahl der potenziell in Frage kommenden Menschen ab. Ist zum Beispiel über jemand nur der Name „Müller“ und das Gewicht „ca. 85 Kilo“ bekannt, dann bliebe er in einer Millionenstadt sicher anonym – da gibt es viele Müllers dieser Gewichtsklasse – in einem Mietshaus mit nur vier Parteien wohl eher nicht.

Legt man für verschiedene Zwecke unabhängig voneinander zustande gekommene Datensammlungen zusammen und alle Betroffenen waren hier wie da anonym, dann kommt diese Anonymität in Gefahr. Die Daten pro Mensch werden mehr und selektiver, die Anzahl der Selektierten sinkt.

Das Gegenmittel wäre die ausnahmslose Verpflichtung zur Zweckgebundenheit von Datensammlungen. Das Zusammenlegen mit anderen Daten wäre dann ein gegenüber dem ursprünglichen Sammelzweck neuer Zweck und also grundsätzlich verboten.



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Drawbot

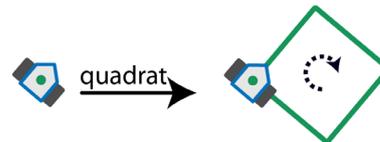
Der Roboter Drawbot kann fahren und dabei zeichnen!

Man kann Drawbot die folgenden Befehle eingeben: **quadrat**, **dreieck**, **vor**, **drehen**

Die Wirkung der Befehle ist so:

quadrat:

Drawbot zeichnet ein Quadrat.
An den Ecken dreht er sich nach rechts.



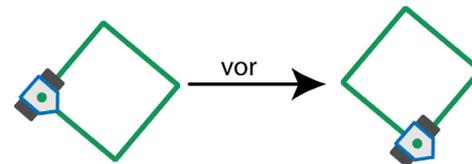
dreieck:

Drawbot zeichnet ein Dreieck.
An den Ecken dreht er sich nach rechts.



vor:

Drawbot fährt auf einer vorher gezeichneten Linie bis zur nächsten Ecke.



drehen:

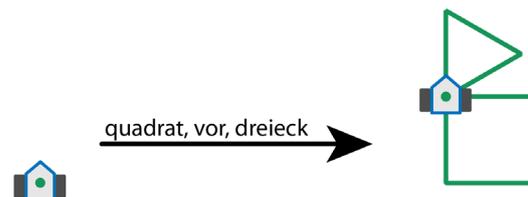
Drawbot dreht sich nach rechts bis zur nächsten gezeichneten Linie.



Man kann Drawbot auch eine Folge von Befehlen eingeben.

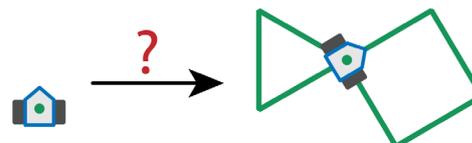
Ein Beispiel: **quadrat, vor, dreieck**

Die Wirkung dieser Befehlsfolge ist rechts zu sehen:



Welche Befehlsfolge hat diese Wirkung?

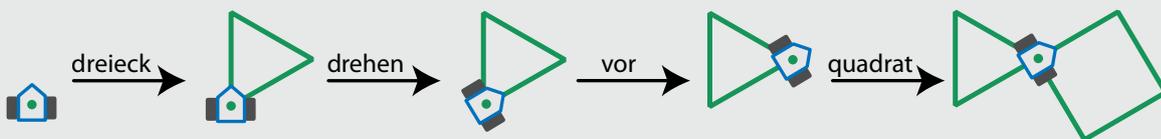
- A) quadrat, drehen, vor, dreieck
- B) dreieck, drehen, vor, quadrat
- C) dreieck, drehen, quadrat
- D) quadrat, vor, quadrat, drehen, dreieck



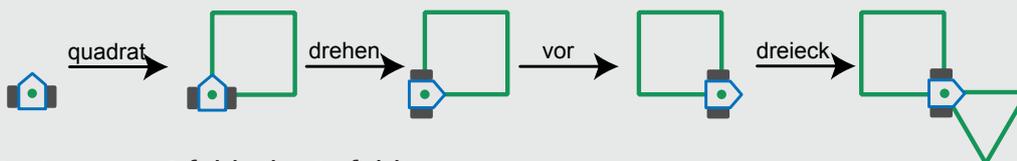
Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



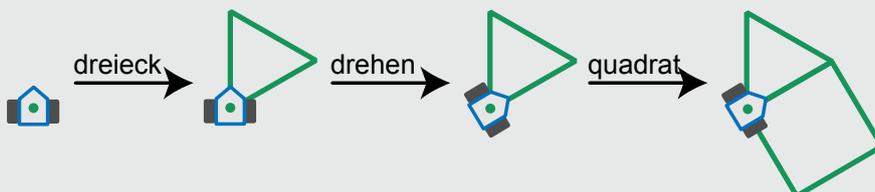
Antwort B ist richtig:



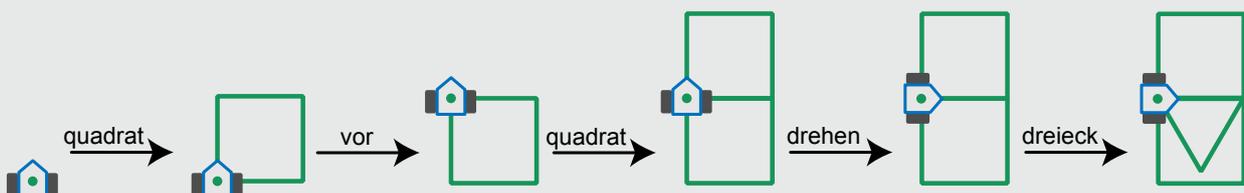
In Antwort A sind die Befehle **dreieck** und **quadrat** vertauscht:



In Antwort C fehlt der Befehl **vor**:



Antwort D ist offensichtlich falsch; die Wirkung dieser Befehlsfolge enthält zwei Quadrate.



Das ist Informatik!

Die einfachsten Bausteine von Programmen für Roboter (und auch für Computer) sind Befehle und Folgen von Befehlen. Weil echte Roboter meistens nicht zeichnen, sondern Autos zusammenbauen oder bei medizinischen Therapien helfen, kennen sie viel mehr und deutlich kompliziertere Befehle als der Drawbot. Und ihre Befehle haben eine starke Wirkung. Da ist es wichtig, dass ihre Programmierer sehr genau arbeiten.

Mit den einfachen Zeichenbefehlen von Drawbot kann man aber gut Programmieren lernen. Solche Befehle wurden zuerst von dem amerikanischen Informatiker Seymour Papert in der Sprache Logo eingeführt. In Logo zeichnet eine kleine Schildkröte, die „Turtle“. Turtle-Grafik gibt es heute in vielen Programmiersprachen, zum Beispiel Python.



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Eiscreme-Stapel

Beim Eissalon LIFO werden die gewünschten Eiscreme-Kugeln auf eine Eistüte gestapelt. Und zwar genau in der Reihenfolge, wie es der Kunde sagt.

Was muss der Kunde sagen, wenn er eine Eistüte haben will, wie hier gezeigt?

Ich hätte gerne eine Eistüte mit ...

- A) ... Schokolade, Pfefferminze und Heidelbeere!
- B) ... Schokolade, Heidelbeere und Pfefferminze!
- C) ... Heidelbeere, Pfefferminze und Schokolade!
- D) ... Heidelbeere, Schokolade und Pfefferminze!



Antwort C ist richtig:

„Ich hätte gerne eine Eistüte mit Heidelbeere, Pfefferminze und Schokolade!“

Was zuerst genannt wird, landet im Stapel zuunterst.

Was zuletzt genannt wird, landet im Stapel zuoberst.

Bei Antwort A ist die Reihenfolge genau verkehrt herum.

Bei den Antworten B und D ist die Pfefferminze nicht in der Mitte.

Das ist Informatik!

Reihenfolge ist wichtig. Werden die Eissorten in anderer Reihenfolge genannt, ergibt das eine andere Eistüte. In der Informatik lernt man, wie nützlich es ist, wenn etwas geordnet ist. Und dass man verstehen muss, welche Ordnungen in welchen Situationen gelten.

Ohne zu verstehen, wie der Eissalon handelt, kann man nicht gezielt eine bestimmte Eistüte bestellen.

Ohne eine Situation zu verstehen, kann man kein dazu passendes Programm entwickeln. Die in dieser Biber-Aufgabe benutzte Ordnung heißt LIFO („last in, first out“).

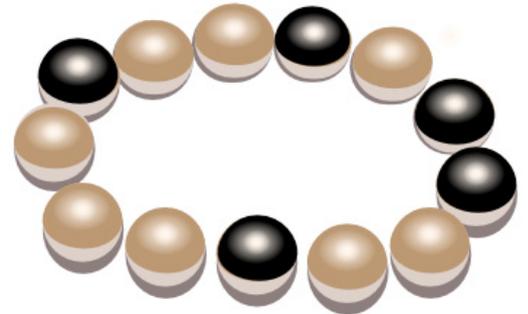


Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Falsche Armbänder

Beim letzten Wasserfest trug die Biberprinzessin dieses magische Armband aus hellen und dunklen Perlen. Danach hat sie es geöffnet und in ein Kästchen gelegt.

Nun braucht sie ihr magisches Armband wieder und schaut in das Kästchen. Oje: Jemand hat drei falsche Armbänder dazu gelegt.



Welches der vier Armbänder ist ihr magisches Armband?

- A)
- B)
- C)
- D)

Antwort B ist richtig:

Das magische Armband hat 13 Perlen, davon 5 dunkle. Zwei der dunklen Perlen liegen als Paar nebeneinander. Armband A ist falsch, es hat kein Paar dunkler Perlen. Armband C ist falsch, es hat nur 12 Perlen. Armband D ist falsch, es hat 6 dunkle Perlen.

Das ist Informatik!

Die Armbänder sind Beispiele für strukturierte Anordnungen von Objekten. Die Informatik spricht von Mustern (patterns). Muster kann man anhand zweckdienlicher Struktur-Eigenschaften miteinander vergleichen – auf Ähnlichkeit oder sogar Gleichheit (pattern matching). In dieser Biber-Aufgabe sind die Struktur-Eigenschaften die Anzahl heller Perlen, die Anzahl dunkler Perlen und die Paarungen. Die Informatik spricht von Mustererkennung (pattern recognition), wenn sie Muster als Bestandteile größerer Strukturen sucht. Das kann ein bestimmtes Wort in einem Text sein. Das kann ein Gesicht in den Aufnahmen einer Überwachungskamera sein. Und vieles mehr.

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Falsche Kachel

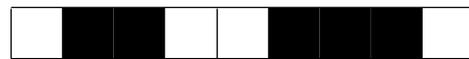
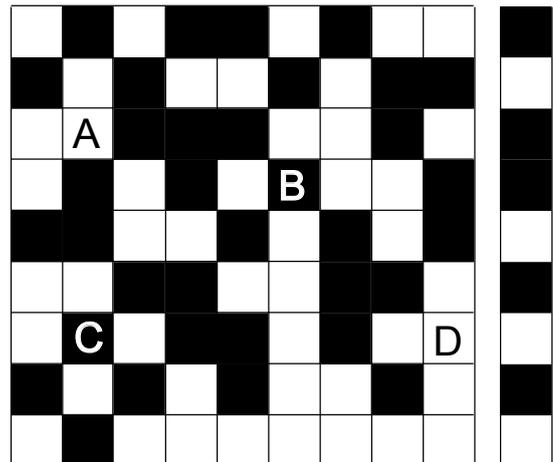
Der Platz vor dem Computer-Clubhaus soll einen neuen Belag aus 9 mal 9 schwarzen und weißen Kacheln bekommen. Ein Designer entwirft den Plan. Er fügt dem Plan rechts und unten je einen Streifen von Kontrollfeldern hinzu.

Wenn die Anzahl der schwarzen Kacheln in einer Zeile gerade ist, dann ist das Kontrollfeld rechts daneben schwarz. Sonst ist es weiß.

Wenn die Anzahl der schwarzen Kacheln in einer Spalte gerade ist, dann ist das Kontrollfeld darunter schwarz. Sonst ist es weiß.

Leider hat sich ein Fehler eingeschlichen. Die Kontrollfelder sind in Ordnung, aber eine Kachel ist falsch. Welche?

- A) Die Kachel A; sie muss schwarz sein.
- B) Die Kachel B; sie muss weiß sein.
- C) Die Kachel C; sie muss weiß sein.
- D) Die Kachel D; sie muss schwarz sein.



Antwort C ist richtig:

Die Kachel C muss weiß sein. Die siebte Zeile von oben hat eine gerade Anzahl (4) schwarze Kacheln, aber ein weißes Kontrollfeld. Also muss irgendeine Kachel in der siebten Zeile falsch sein. Die Kontrollfelder der anderen Zeilen sind richtig. Die zweite Spalte von links hat eine ungerade Anzahl (5) schwarze Kacheln, aber ein schwarzes Kontrollfeld. Also muss irgendeine Kachel in der zweiten Spalte falsch sein. Die Kontrollfelder der anderen Spalten sind richtig. Die Kachel C ist in der siebten Zeile und in der zweiten Spalte. Also muss sie die falsche Kachel sein.

Das ist Informatik!

Diese Biber-Aufgabe ist ein einfaches Beispiel für einen fehlertoleranten Code. Die Kacheln repräsentieren mit ihren zwei Zuständen die Bits des Codes. Zusammen mit den Kontrollfeldern muss die Anzahl schwarzer Bits in jeder Zeile und jeder Spalte ungerade sein. Es wird auch davon ausgegangen, dass nicht viele Bits gleichzeitig falsch sind. Die Fehlertoleranz jedes Codes hat ihre Grenzen. Die Prüfung der Zeilen oder der Spalten allein ergibt hier nur den Hinweis, dass ein Bitfehler existiert. Die Prüfung der Zeilen und der Spalten gemeinsam erlaubt die Lokalisierung und Reparatur des Bitfehlers. Die Informatik kennt zum Speichern und Übertragen von Informationen viele Codes mit unterschiedlicher Fehlertoleranz. Manche Anwendungen, wie das Kaufen im WWW, brauchen eine höhere Datensicherheit als andere, wie zum Beispiel das Anschauen lustiger Katzenvideos.



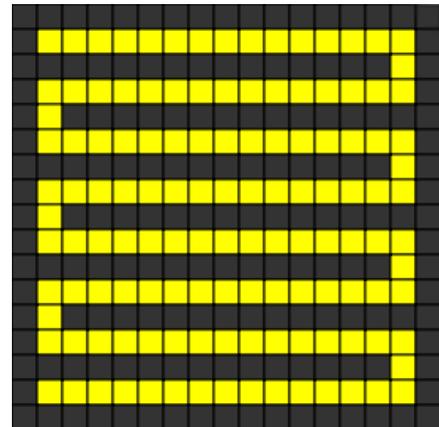
Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Fang das Monster

Im Keller der Biberburg lebt ein Monster.
Es versteckt sich irgendwo in den gelben Zellen.
Wo graue Wände sind, kann das Monster nicht sein.

Du willst das Monster fangen.
Klicke auf eine gelbe Zelle. Die gelben Zellen werden weniger.
Klicke wieder auf eine gelbe Zelle. Und so weiter.

Wenn nur noch eine gelbe Zelle übrig ist,
hast Du das Monster darin gefangen.

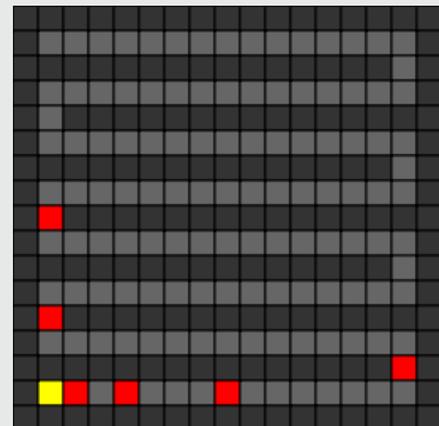


So ist es richtig:

Die 127 gelben Zellen bilden eine lange Reihe.
Die beste Fang-Strategie ist es, immer auf die mittlere gelbe Zelle zu klicken. Nach dem ersten Klick bleiben dann 63 Zellen übrig. Nach dem zweiten Klick bleiben 31, dann 15, 7 und 3 Zellen übrig. Nach dem sechsten Klick bleibt nur eine gelbe Zelle übrig. Mit weniger als 6 Klicks kann man das Monster nicht fangen.

Das ist Informatik!

Bei jedem Trennschritt den Suchbereich zu halbieren, nennt die Informatik „Binäres Suchen“. Binäres Suchen ist möglich, wenn man genau weiß, wo im Suchraum die Mitte ist. Programme können das leicht berechnen. Binäres Suchen ist möglich, wenn die Objekte im Suchraum passend geordnet sind, zum Beispiel als Liste oder als balancierter Binärbaum. Bei dieser Biber-Aufgabe stellt die Nachbarschaftsbeziehung der Kellerzellen die Ordnung her. Menschen „halbieren“ in zwei ungleiche Teile, wenn sie zum Beispiel im Lexikon einen Begriff oder in einem Buch eine bestimmte Seitennummer suchen – falls diese Bücher noch aus Papier sind. Hier ist es eine menschliche Strategie, bei jedem Trennschritt möglichst nahe an die gesuchte Stelle zu kommen. Man weiß ja ungefähr, wo das „D“ im Lexikon oder die Seite 550 in einem 600-seitigen Schmöker positioniert ist.



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Flussaufwärts

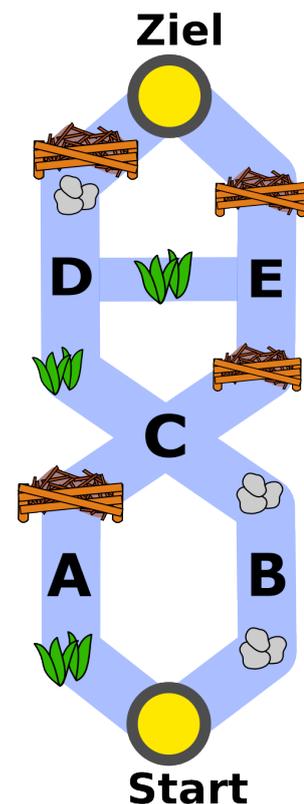
Um zu seinem Ziel zu kommen, muss der Biber einen passenden Weg durch das Flusssystem nehmen. Auf seinem Weg muss er Hindernisse überwinden. Dabei verbraucht der Biber folgende Mengen an Energie:

Hindernisse	Benötigte Energie
	2 Zweige
	3 Zweige
	5 Zweige

Um genug Energie zu haben, isst der Biber vor dem Start 15 Zweige. Im abgebildeten Flusssystem siehst du die Hindernisse. A, B, C, D und E sind die Zwischenstationen auf den möglichen Wegen.

Welchen der folgenden Wege wird der Biber nehmen? Beachte, dass er vor dem Start nur 15 Zweige gegessen hat.

- A) Start → C → E → Ziel
- B) Start → A → C → E → D → Ziel
- C) Start → B → C → D → E → Ziel
- D) Start → B → C → D → Ziel



Antwort C ist richtig:

Die verschiedenen Wege brauchen folgende Energiemengen:

Start → A → C → E → Ziel : $2+5+5+5 = 17$

Start → A → C → E → D → Ziel : $2+5+5+2+3+5 = 22$

Start → B → C → D → E → Ziel : $3+3+2+2+5 = 15$;

das ist der einzige Weg, der nicht mehr Energie benötigt, als der Biber hat.

Start → B → C → D → Ziel: $3+3+2+3+5 = 16$

Das ist Informatik!

Das Flusssystem ist ein Netzwerk mit „Start“, „Ziel“ und den Zwischenstationen „A“ bis „E“ als Knoten. Der Energieverbrauch für das Überwinden von Hindernissen kann als Abstand zwischen zwei verbundenen Knoten angesehen werden. So braucht der Biber nur nach dem kürzesten Weg vom Start zum Ziel zu suchen. Für das Kürzeste-Wege-Problem ist der Algorithmus von Dijkstra der berühmteste. Bekannt ist auch der Algorithmus von Floyd und Warshall, der von allen Knoten zu allen anderen Knoten die Länge des kürzesten Weges bestimmt. Du hast die Anwendung dieser Algorithmen in einem Navigationssystem vielleicht schon erlebt.

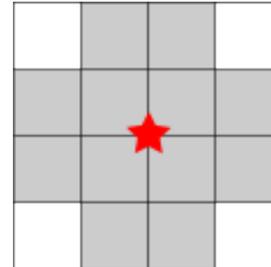


Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Funknetz im Dorf

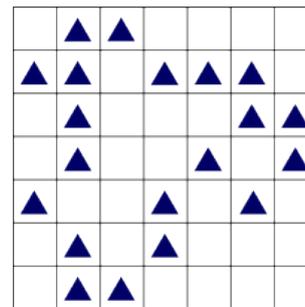
In einem Dorf wird ein Funknetz mit mehreren Funkmasten eingerichtet. Es soll den Einwohnern Zugang zum Internet bieten.

Jeder Funkmast hat ein begrenztes Sende- und Empfangsgebiet. Das ist im Bild zu sehen: Nur auf den zwölf umliegenden Grundstücken (grau) erhält ein Haus Verbindung zum Funkmast in der Mitte (roter Stern).



Ein Funkmast kann immer nur auf dem Schnittpunkt zweier Grundstücksgrenzen aufgestellt werden. Die Sende- und Empfangsgebiete von Funkmasten dürfen sich überlappen.

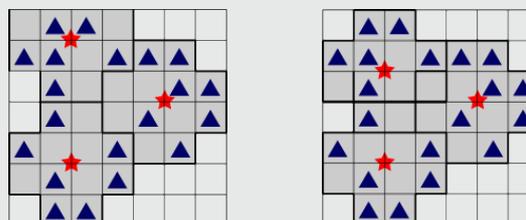
Das Bild zeigt die Karte des Dorfes. Jedes Dreieck ▲ kennzeichnet ein Haus.



Wie viele Funkmasten müssen mindestens aufgestellt werden, damit alle Häuser eine Verbindung zum Funknetz erhalten?

3 ist richtig:

Mit nur zwei Funkmasten kann man nicht alle Häuser mit dem Funknetz verbinden. Zwei verschiedene Aufstellungen von je drei Funkmasten würden allen Häusern eine Verbindung zum Funknetz ermöglichen:



Das ist Informatik!

Die Informatik kennt Algorithmen, große Flächen durch kleine Flächen unterschiedlicher Formen kostengünstig und mehr oder weniger genau abzudecken. Beispiele sind das Zuschneiden von Stoffen in der Bekleidungsindustrie oder das Ausstanzen von Blechteilen im Maschinenbau. Auch die Planung von Funknetzen für Mobilfunk, für digitales terrestrisches Radio und Fernsehen oder für WLAN ist von diesem Problemtyp. Algorithmen, die für alle Abdeckungsprobleme immer die besten Lösungen finden, sind aber häufig unpraktikabel. Bei größeren Problemen würde die Rechenzeit viel zu lange dauern. Die Informatik entwickelt für größere Abdeckungsprobleme Algorithmen, die zwar nicht immer die besten, aber immerhin verlässlich gute Lösungen finden – und insbesondere sehr schnell sind.

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Fußstapfen

Fußstapfen-Bäume! Die werden nach einem bestimmten Schema gestapft. Dies ist das Stapfprogramm für einen **1-Baum**:

Gehe 1 Schritt vor, dabei machst du 1 Fußstapfen.
Gehe wieder zurück.



1-Baum

Kennt man das Stapfprogramm für einen 1-Baum, sieht das Stapfprogramm für einen **2-Baum** so aus:

Gehe 2 Schritte vor, dabei machst du 2 Fußstapfen.
Wende dich nach rechts und stapfe einen 1-Baum.
Wende dich nach links und stapfe einen 1-Baum.
Gehe in deiner Spur wieder zurück.



2-Baum

Das Stapfprogramm für einen **3-Baum** ist nun auch schnell erklärt, denn ein 3-Baum enthält 2-Bäume:

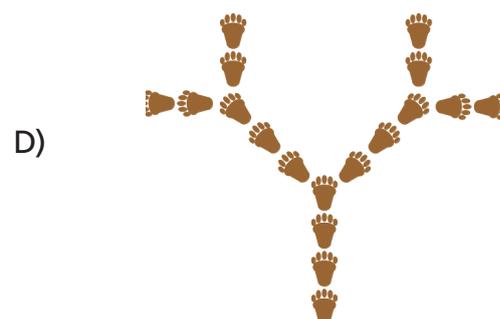
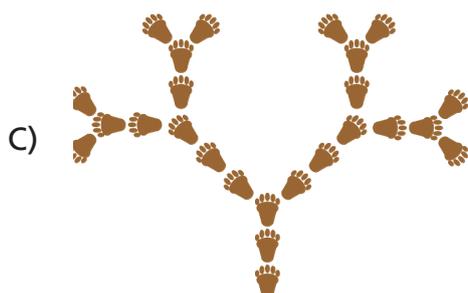
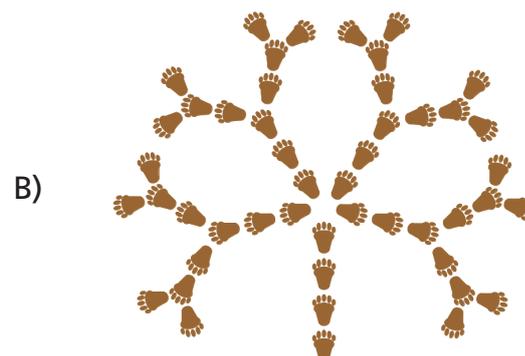
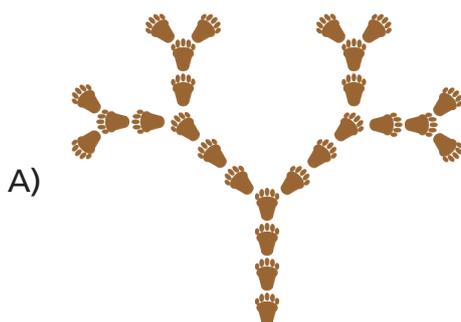
Gehe 3 Schritte vor, dabei machst du 3 Fußstapfen.
Wende dich nach rechts und stapfe einen 2-Baum.
Wende dich nach links und stapfe einen 2-Baum.
Gehe in deiner Spur wieder zurück.



3-Baum

Das Stapfprogramm für einen **4-Baum** folgt dem gleichen Schema.

Welcher Baum ist nach diesem Schema ein 4-Baum?





Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Antwort A ist richtig:

Wenn man das Stapfprogramm für einen 2-Baum mit dem für einen 3-Baum vergleicht, erkennt man das Schema und kann das Stapfprogramm für einen 4-Baum aufschreiben: Gehe 4 Schritte vor, dabei machst du 4 Fußstapfen. Wende dich nach rechts und stapfe einen 3-Baum. Wende dich nach links und stapfe einen 3-Baum.

Gehe in deiner Spur wieder zurück.

Nur die Figur in Antwort A entsteht durch dieses Programm. Sie ist deshalb ein 4-Baum, der aus 4 Fußstapfen und zwei 3-Bäumen besteht.

Die Figur in Antwort B besteht aus 4 Fußstapfen und fälschlich vier 3-Bäumen.

Die Figur in Antwort C enthält zwar zwei 3-Bäume, beginnt aber mit nur 3 Fußstapfen.

Die Figur in Antwort D beginnt mit 4 Fußstapfen, aber die beiden Unterbäume sind keine 3-Bäume.

Das ist Informatik!

Das Schema funktioniert für alle möglichen Zahlen. N-Bäume stapfen heißt n Schritte vorgehen, dabei n Fußstapfen machen, zwei (n-1)-Bäume stapfen und wieder zurückgehen. Ein (n-1)-Baum wiederum besteht aus n-1 Fußstapfen und zwei (n-2)-Bäumen, und so weiter, bis man beim 1-Baum ankommt, für den es ein spezielles Programm gibt.

In der Informatik spricht man von Rekursion, wenn eine Aufgabe erledigt wird, indem man einfachere Versionen der gleichen Aufgabe erledigt, so lange bis die einfachste(n) Version(en) der Aufgabe auf spezielle Art und Weise erledigt wird. In vielen Fällen kann mit Rekursion elegant beschrieben werden, wie eine Aufgabe zu erledigen ist.

Aber Achtung: Um einen n-Baum zu stapfen muss man 2(n-1)-Bäume stapfen, also 4(n-2)-Bäume stapfen, also 8(n-3)-Bäume stapfen, ... , also 2(n-1) 1-Bäume stapfen. Für eine große Zahl n kann das sehr lange dauern. Rekursion kann also elegant, aber auch mühsam sein.

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Geburtstagstorte

Bei ihrem letzten Geburtstag wollte Beatrix eine Torte backen. Das Rezept enthielt 8 Gewürze. Doch nach dem Backen war die Torte giftgrün. Den Gästen grauste beim Anblick der Torte. Weil die Torte trotzdem sehr lecker war, möchte Beatrix sie wieder backen. Nur giftgrün soll die Torte nicht mehr werden. Beatrix vermutet, dass nur ein einziges Gewürz die Torte giftgrün gemacht hat. Sie will systematisch herausfinden, welches Gewürz das schuldige ist. Dazu überlegt sie sich, mehrere Probe-Torten zu backen und dabei die Gewürze zu variieren.



Wie viele Probe-Torten muss Beatrix mindestens backen, damit sie mit Sicherheit das schuldige Gewürz bestimmen kann?

3 ist richtig:

Die 8 Gewürze teilt Beatrix so auf die 3 Probe-Torten auf:

Probe-Torte	–	1	2	1,2	3	1,3	2,3	1,2,3
Gewürz	0	1	2	3	4	5	6	7

Das Gewürz 0 befindet sich also in keiner Probe-Torte, Das Gewürz 1 nur in Probe-Torte 1, das Gewürz 2 nur in Probe-Torte 2, das Gewürz 3 in den Probe-Torten 1 und 2 usw. Damit hat Beatrix jedes Gewürz genau einer Kombination von Probe-Torten zugeordnet. Spätestens nach der dritten Probe-Torte weiß Beatrix, welche Probe-Torten giftgrün geworden sind. Dann kann sie das schuldige Gewürz bestimmen. Mit weniger als drei Probe-Torten kommt Beatrix nicht aus. Bei zwei Probe-Torten gibt es nur vier Kombinationen (– ; 1 ; 2 ; 1,2), denen sie die 8 Gewürze nicht eindeutig zuordnen kann.

Das ist Informatik!

In den fertigen Torten ist Information gespeichert. Beatrix interessiert sich dabei nur für zwei Werte: „Die Farbe ist giftgrün.“ oder „Die Farbe ist nicht giftgrün.“ Für Beatrix' Zwecke ist jede Probe-Torte deshalb wie ein Bit im Computer. In einem Bit kann genau einer von zwei Werten gespeichert werden: Ein Bit ist entweder „wahr“ oder „falsch“. In der Informatik werden diese beiden Werte häufig durch die Ziffern 1 und 0 dargestellt. Hat man mehrere Bits zur Verfügung, kann man damit Zahlen darstellen – mit Hilfe des Binärsystems. Bei drei Bits sieht das ganz ähnlich aus wie oben:

Bits	000	001	010	011	100	101	110	111
Zahl	0	1	2	3	4	5	6	7

Mit mehr Bits kann man größere Zahlen darstellen. Mit Bits kann man aber auch ganz andere Dinge darstellen, zum Beispiel Buchstaben. Dazu muss man die Dinge bestimmten Zahlen zuordnen, die man wiederum mit Bits darstellt. Mit Bits kann man beinahe alles darstellen, was man sich vorstellen kann. Aber eines ganz sicher nicht: die Unendlichkeit. Es gibt nämlich in jedem Informatiksystem, ganz egal wie groß es ist, immer nur endlich viele Bits.

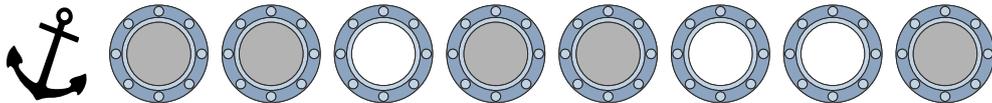


Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

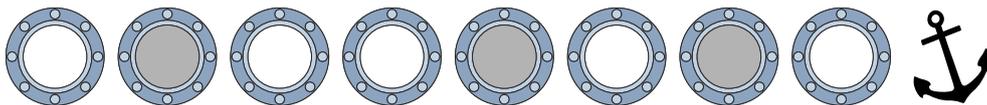
Getöntes Glas

Kapitän Schwarz lässt an seiner Yacht die Gläser der Bullaugen erneuern.
Jedes neue Glas ist entweder ganz klar oder getönt.
Der Glasermeister bekommt folgenden Auftrag:

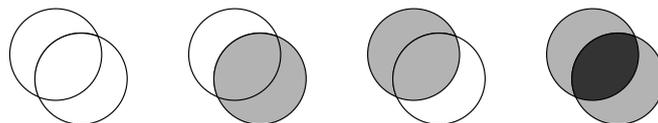
Bullaugen auf der linken Seite



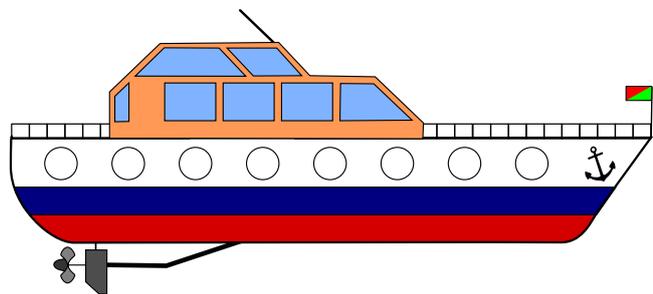
Bullaugen auf der rechten Seite



Da sich immer zwei Bullaugen genau gegenüber liegen, kann man von jeder Seite durch die Yacht hindurch sehen. Je nach Tönung der Gläser ist die Durchsicht ganz klar, getönt oder stark getönt.



Klicke auf die Bullaugen.
Ändere die Durchsichten so,
dass sie dem Auftrag des
Glasermeisters entsprechen.
Tipp: Achte auf die Position der Anker.



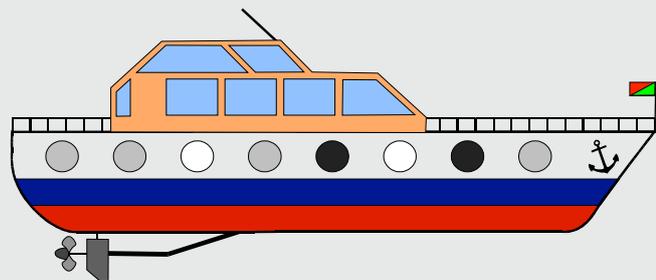
Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



So ist es richtig:

Zuerst müssen wir klären, welche der Bullaugen sich gegenüber liegen. Dafür nutzen wir den Anker als Anhaltspunkt.

Wenn wir jetzt durch die Bullaugen-Paare blicken, ergeben sich folgende Durchsichten:



Das ist Informatik!

Die Darstellung von Information ist ein wichtiger Aspekt in der Informatik.

In dieser Biber-Aufgabe wird eine Addition (genauer eigentlich eine Vektoraddition – es kann ja nie zu einem Übertrag kommen) durch Überdeckungen der Tönungen von Fenstergläsern motiviert.

Dabei steht ein klares Fensterglas für 0, ein schwach getöntes für 1 und ein stark getöntes für 2.

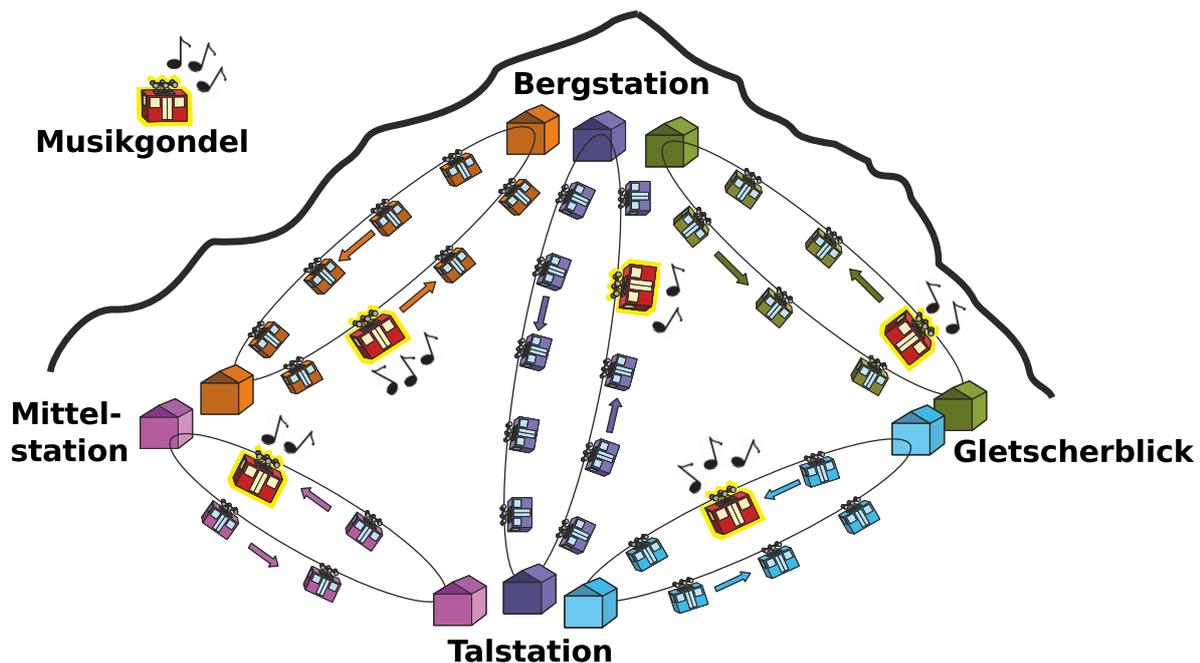
Um die Addition einfach durchführen zu können, muss man allerdings vorab die zu addierenden Elemente (Fenstergläser) zusammenführen. Dazu ist es wichtig zu erkennen, dass hier eine Fensterreihe in spiegelverkehrter Form vorliegt.



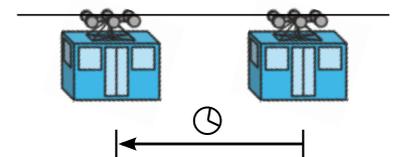
Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Gipfelstürmer

Tom möchte zur Bergstation fahren. Er kann von der Talstation aus mit verschiedenen Gondelbahnen dorthin gelangen. Aber er möchte nur in Gondeln fahren, in denen Musik gespielt wird. Das Bild zeigt die aktuellen Positionen der Gondeln zu dem Zeitpunkt, als Tom von der Talstation wegfahren möchte.



Alle Gondelbahnen bewegen sich gegen den Uhrzeigersinn. Der Fahrzeitabstand von einer Gondel zur vorigen Gondel ist die Zeit, welche die Gondel braucht, um von ihrer aktuellen Position zur aktuellen Position der vorigen Gondel zu fahren (siehe Bild).

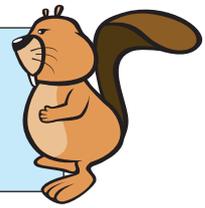


Der Fahrzeitabstand ist bei allen Gondelbahnen gleich. Er ändert sich auch nicht, wenn Gondeln durch Stationen fahren. Die Gondeln aller Gondelbahnen fahren immer zu gleichen Zeitpunkten durch die Stationen. Wenn Tom durch eine Station fährt, kann er flink und ohne Zeitverlust in die Gondel einer anderen Gondelbahn umsteigen. Manchmal muss er aber auch auf die nächste Musikgondel warten.

Auf welchem Weg kommt Tom am schnellsten zur Bergstation?

- A) Talstation → Mittelstation → Bergstation
- B) Talstation → Mittelstation → Talstation → Bergstation
- C) Talstation → Bergstation
- D) Talstation → Gletscherblick → Bergstation

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Antwort D ist richtig:

GT sei die Musikgondel der Strecke Gletscherblick-Talstation. GB sei die Musikgondel der Strecke Gletscherblick-Bergstation. Tom braucht 8 Fahrzeitabstände (F).

Der Ablauf ist so:

F-1: GT erreicht Talstation. Tom steigt ein. GB fährt Richtung Bergstation.

F-2: GT verlässt Talstation. GB erreicht Bergstation.

F-3: GT fährt Richtung Gletscherblick. GB verlässt Bergstation.

F-4: GT erreicht Gletscherblick. Tom steigt aus. GB fährt Richtung Gletscherblick.

F-5: GB erreicht Gletscherblick. Tom steigt ein.

F-6: GB verlässt Gletscherblick.

F-7: GB fährt Richtung Bergstation.

F-8: GB erreicht Bergstation. Tom steigt aus.

Bei Antwort A braucht Tom 10 Fahrzeitabstände.

Bei den Antworten B und C braucht Tom 11 Fahrzeitabstände.

Das ist Informatik!

Beim Einstieg in die Informatik scheint es zunächst so zu sein: Ein Computer führt die Befehle seines Programms schön nacheinander aus (sequentiell).

Für viele Systeme, die von der Informatik modelliert werden, wäre ein Nacheinander aber keine gute Idee. Zum Beispiel bei einer Gondelbahn.

Die dürfte immer nur ein Stückchen fahren, müsste dann warten, bis die anderen Gondelbahnen nacheinander ihr Stückchen gefahren sind, würde dann wieder ein Stückchen fahren, usw.

Die Informatik würde das System gleichzeitig fahrender Gondelbahnen besser als „nebenläufige Prozesse“ modellieren.



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Gruppenarbeit

Für eine Gruppenarbeit bildeten die Schüler einer Klasse vier Gruppen. Alle Gruppen teilten ihre Arbeit in einzelne Aufgaben auf. Drei Gruppen konnten alle ihre Aufgaben erledigen, aber eine Gruppe wurde nicht fertig. Was ist passiert?

Die schlauesten Schüler, Ada und Charles, haben die vier Gruppen analysiert. Sie fanden heraus, dass die meisten Gruppenmitglieder auf andere warten mussten, bevor sie mit ihrer eigenen Aufgabe beginnen konnten. Ada und Charles haben für jede Gruppe eine Skizze gezeichnet, die sich auf das Wesentliche konzentriert.

Ein Kreis stellt eine Person dar. Ein Pfeil von Person 1 nach Person 2 bedeutet, dass Person 1 ihre Aufgabe erledigt haben musste, bevor Person 2 mit ihrer Aufgabe beginnen konnte.

Welches Bild entspricht der Gruppe, die nicht fertig wurde?



Antwort D ist richtig:

Die Skizzen stellen Abhängigkeitsgraphen für die Aufgaben der vier Gruppen dar. Die Gruppenmitglieder werden blockiert, wenn ein Zyklus (ein Rundgang) vorhanden ist. Dann kann keiner im Zyklus mit der Aufgaben beginnen, weil jeder auf den Vorgänger wartet. Nur der Graph D enthält einen solchen Zyklus.

Das ist Informatik!

Die meisten Computersysteme erledigen verschiedene Aufgaben scheinbar nebeneinander her. Ein Laptop kann zugleich Musik abspielen, E-Mails abrufen und die Festplatte auf Viren prüfen. Auf einem Smartphone kann man ein Spiel spielen und erhält trotzdem zwischendurch seine SMS und Telefonanrufe. Manche solcher „Prozesse“ können aber voneinander abhängen. Wenn ein Dokument geöffnet werden soll, muss der Editor-Prozess warten, bis der Betriebssystem-Prozess die entsprechenden Daten von der Festplatte kopiert hat. Was passiert aber, wenn zugleich ein anderer Prozess auf die Festplatte zugreifen will? Warten beide Prozesse höflich, bis der andere seine Daten hat?

Die Informatik hat sehr intensiv erforscht, wie solche Deadlocks (Verklemmungen) vermieden werden können. Wer programmiert, muss sicherstellen, dass niemals zwei oder mehr seiner Prozesse aufeinander warten müssen.

<<http://de.wikipedia.org/wiki/Philosophenproblem>>

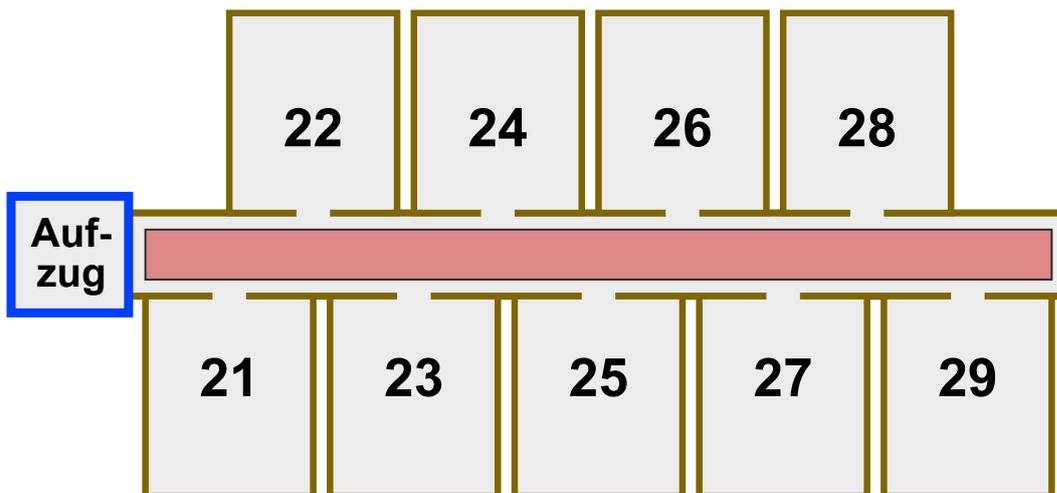
Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Hotel Comfort

Im Hotel Comfort sind die Zimmernummern zweistellig:

- Die erste Ziffer gibt das Stockwerk an, in welchem das Zimmer liegt.
- Die zweite Ziffer gibt an, wie weit das Zimmer vom Aufzug entfernt ist. Die Zimmer sind also in jedem Stockwerk so angeordnet wie hier für den zweiten Stock gezeigt:



Die Gäste im Hotel Comfort sollen sich nur wenig anstrengen. Je näher ein Zimmer am Aufzug ist, desto komfortabler liegt es. Sind zwei Zimmer in verschiedenen Stockwerken gleich weit vom Aufzug entfernt, liegt das Zimmer im niedrigeren Stockwerk komfortabler. Zimmer 32 liegt also komfortabler als Zimmer 15, und Zimmer 22 liegt komfortabler als Zimmer 32.

Im Hotel Comfort gilt die Vorschrift: Ein neuer Gast bekommt stets dasjenige freie Zimmer, das am komfortabelsten liegt.

Folgende zehn Zimmer sind derzeit frei: **12, 25, 11, 43, 22, 15, 18, 31, 44, 52**

Nun kommen nach und nach zehn neue Gäste.

In welcher Reihenfolge müssen die freien Zimmer vergeben werden?

- 18, 15, 12, 11, 25, 22, 31, 44, 43, 52
- 52, 43, 44, 31, 22, 25, 11, 12, 15, 18
- 11, 31, 12, 22, 52, 43, 44, 15, 25, 18
- 11, 12, 15, 18, 22, 25, 31, 43, 44, 52



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Antwort C ist richtig:

Die Vorschrift des Hotels zur Zimmervergabe bedeutet, dass die Zimmernummern zuerst nach der zweiten Ziffer und dann nach der ersten Ziffer sortiert werden müssen. Um die Zimmer in der richtigen Reihenfolge zu vergeben, kann man die einzelnen Zimmernummern also von rechts nach links lesen und dann wie üblich nach kleinerem Wert sortieren. Beispiel: aus 32 wird 23, aus 15 wird 51, und weil $23 < 51$, muss Zimmer 32 vor Zimmer 15 vergeben werden. Liest man die Nummern auf diese Weise, so liefert nur Antwort C eine aufsteigende Zahlenfolge (11, 13, 21, ..., 52, 81).

Antwort A ist falsch: Die Reihenfolge der ersten beiden Nummern (18, 15) gehorcht nicht der Vergabevorschrift ($81 > 51$). Antwort B ist falsch: Die Reihenfolge der dritten und vierten Nummer (44, 31) gehorcht nicht der Vergabevorschrift ($44 > 13$). Antwort D ist falsch: Die Reihenfolge der vierten und fünften Nummer (18, 22) gehorcht nicht der Vergabevorschrift ($81 > 22$). Hier wurde erst nach Stockwerken und dann nach Abstand vom Aufzug sortiert.

Das ist Informatik!

Die Sortierung der im Hotel Comfort zu vergebenden Zimmernummern ist speziell. Aber sie hat eine besondere Eigenschaft: Wenn normal sortierte Zimmernummern – zum Beispiel 11, 12, 18, 22, 25 – für das Hotel Comfort umsortiert werden, bleiben die Zimmernummern eines Stockwerks untereinander weiterhin sortiert: 11, 12, 22, 25, 18. Der Grund: Die Reihenfolge zweier Nummern wird nur geändert, wenn es nach dem neuen vorrangigen Sortierkriterium (also der zweiten Ziffer, die die Entfernung vom Aufzug angibt) nötig ist.

Sortierverfahren, die beim Umsortieren die Reihenfolge nach der bisherigen Sortierpriorität beibehalten, heißen in der Informatik „stabile Sortierverfahren“. Die sind in der Praxis recht nützlich, zum Beispiel in einem E-Mail-Programm: Dort können die E-Mails etwa nach Datum, Absender oder Betreff sortiert werden. Sind die E-Mails erst nach Datum sortiert, und ordnet man sie dann nach Betreff an, so bleiben die E-Mails mit demselben Betreff untereinander immer noch nach Datum sortiert. Das scheint selbstverständlich, funktioniert aber nur mit einem stabilen Sortierverfahren.

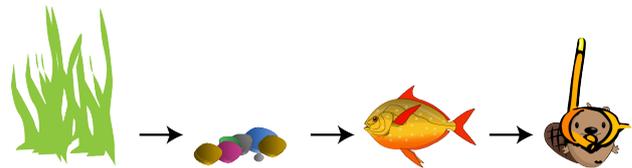
Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



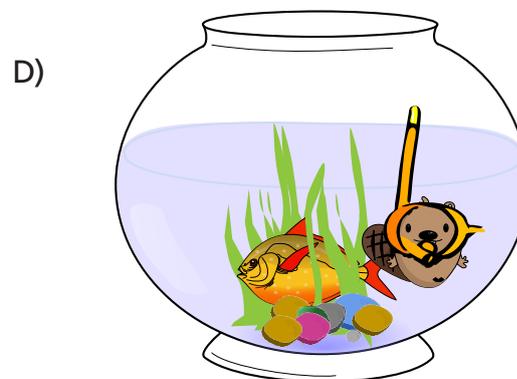
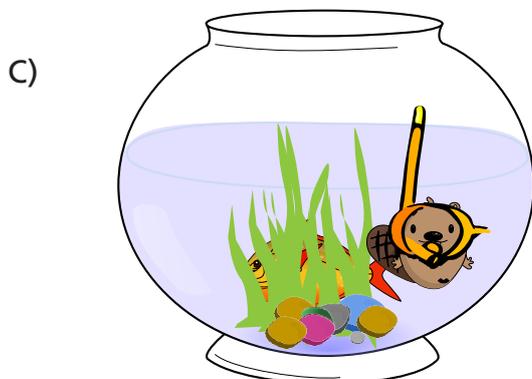
Klebebildchen

Johnny hat ein Fischglas gemacht.
Das verziert er noch mit Klebebildchen.

Zuerst klebt er das Gras, dann die Steine,
dann den Fisch und dann den Tauch-Biber.



Wie sieht das Bild danach aus?



Antwort A ist richtig:

Die Bildchen kleben in der richtigen Reihenfolge übereinander.

B ist falsch, weil der Tauch-Biber nicht ganz vorn ist, der Fisch ist ganz vorn.

C ist falsch, weil das Gras nicht ganz hinten ist, der Fisch ist ganz hinten.

D ist falsch, weil der Fisch nicht vor dem Gras schwimmt, er schwimmt durch das Gras.

Das ist Informatik!

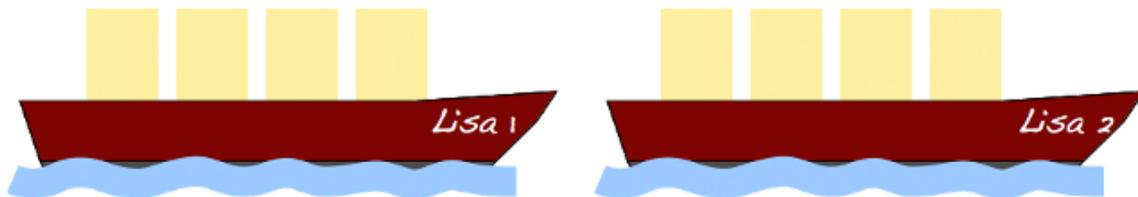
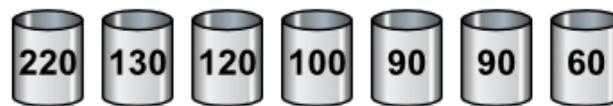
Bei vielen Informationsarten spielen Reihenfolgen eine wichtige Rolle. Hier zum Beispiel bei einem Bild mit vier Bildebenen, die auf eine bestimmte Weise übereinander liegen. Eine andere Reihenfolge der Ebenen würde ein anderes Bild ergeben, obwohl die Bildebenen gleich bleiben.



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Lisas laden

Falke und Folke, den beiden Fischern, gehören die Boote „Lisa 1“ und „Lisa 2“ – die beiden Lisas. Jedes der Boote kann mit höchstens 300 Kilogramm beladen werden. Falke und Folke sollen mit den beiden Lisas einige Fässer mit verschiedenen Sorten Fisch transportieren. Die Fischer werden nach dem transportierten Gewicht bezahlt.



Belade die beiden Lisas mit so viel Kilogramm Fisch wie möglich!

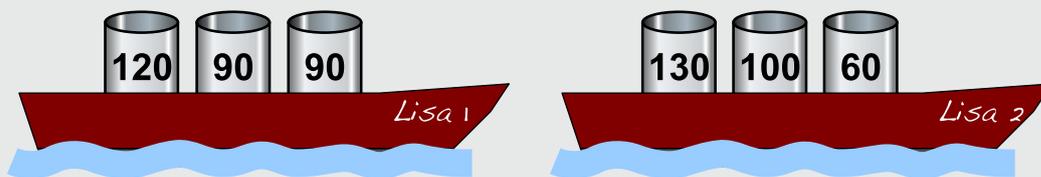
Über den Booten siehst du die Fässer, die zur Verfügung stehen. Jedes Fass ist mit seinem Gewicht (in Kilogramm) beschriftet. Ziehe nun einzelne Fässer jeweils auf eines der Boote. Du kannst jedes Boot mit höchstens 300 Kilogramm beladen.

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



So ist es richtig:

Insgesamt können die Boote mit 590 Kilogramm Fisch beladen werden:
 $120+90+90=300$ Kilogramm auf dem einen Boot, $130+100+60=290$ Kilogramm auf dem anderen.



Achtung, nicht gierig werden! Wenn man die schwersten Fässer zuerst nimmt, um die beiden Lisas zu beladen, kann man die Boote höchstens mit $220+60=280$ Kilogramm bzw. $130+120=250$ Kilogramm beladen. Das macht nur 530 Kilogramm insgesamt. Mit mehr als 590 Kilogramm können die Lisas nicht beladen werden. Denn dazu müssten beide Boote mit 300 Kilogramm beladen werden. Es gibt aber nur eine Möglichkeit, Fässer zu einem Gesamtgewicht von 300 Kilogramm zu kombinieren, nämlich $120+90+90=300$ Kilogramm.

Das ist Informatik!

Viele Menschen sind fasziniert davon, Dinge zu optimieren – häufig übrigens, um Kosten zu sparen und ihren Profit zu maximieren. Bei nicht ganz einfachen Problemen werden meist Computerprogramme zum Optimieren verwendet: um kürzeste Routen, optimale Beladungen, ideale Stundenpläne usw. zu finden.

Manche Optimierungsprobleme lassen sich mit einem „gierigen“ (engl.: greedy) Algorithmus lösen. Dabei wird jeder Schritt zur Lösung (hier: die Auswahl eines Fasses) so gewählt, dass er so viel Profit (hier: so viel Gewicht) wie möglich bringt – das ist gierig.

Das Schöne an der Informatik: In den meisten Fällen hilft Gier nicht weiter, und komplexere Algorithmen werden benötigt, um optimale Lösungen zu finden. Für einige Probleme lässt sich sogar nachweisen, dass diejenigen Algorithmen, die garantiert optimale Lösungen finden, selbst von Computern nur mit unvermeidbar hohem Aufwand ausgeführt werden können. Für viele solcher schwierigen Optimierungsprobleme hat die Informatik effiziente Algorithmen entwickelt, die zwar keine optimalen Lösungen, aber nachweisbar sehr gute, fast optimale Lösungen finden.

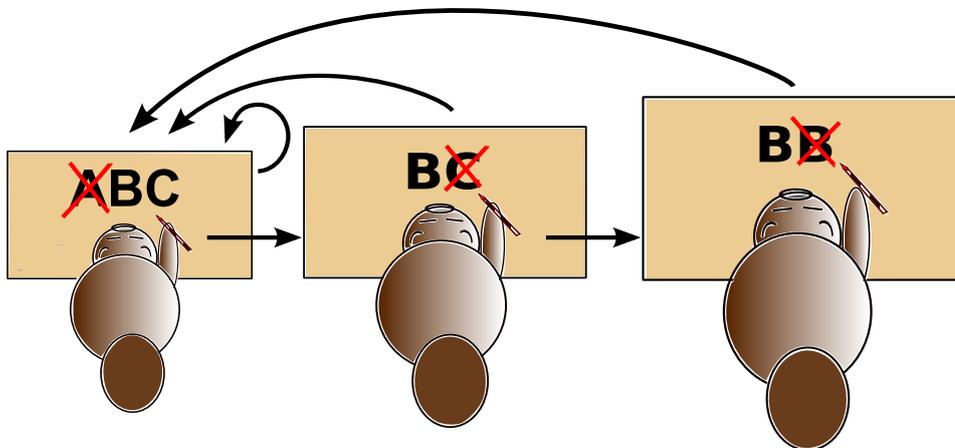


Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Nachricht aus Bibirien

Im weit entfernten Bibirien werden die – ohnehin nur aus den Buchstaben A, B und C bestehenden – Zeitungsmeldungen gelegentlich vom Nachrichtenamt „korrigiert“. Die drei Korrektoren des Amtes lesen eine Nachricht jeweils von links nach rechts und suchen dabei nach bestimmten Buchstabenmustern:

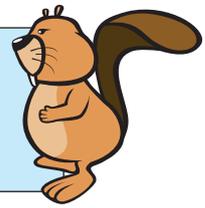
1. Der **Unterkorrektor** sucht nach der Folge **ABC**. Wenn er die Folge findet, ersetzt er sie durch **BC** und fängt mit der geänderten Nachricht wieder von vorne an. Findet er die Folge nicht, gibt er die Nachricht an den Hauptkorrektor weiter.
2. Der **Hauptkorrektor** sucht nach der Folge **BC**. Wenn er die Folge findet, ersetzt er sie durch **B** und gibt die geänderte Nachricht an den Unterkorrektor zurück. Findet er die Folge nicht, gibt er die Nachricht an den Chefkorrektor weiter.
3. Der **Chefkorrektor** sucht nach der Folge **BB**. Wenn er die Folge findet, ersetzt er sie durch **B** und gibt die geänderte Nachricht an den Unterkorrektor zurück. Findet er die Folge nicht, ist die Korrektur zu Ende.



Drei der folgenden Nachrichten bestehen am Ende der Korrektur nur noch aus einem Buchstaben B. Nur eine **NICHT** – welche?

- A) AAABCB
- B) ABCABC
- C) ABABCB
- D) ABCCCC

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Antwort C ist richtig:

Die einzelnen Nachrichten werden so „korrigiert“:

A)	AAABCB	AABCB	ABCB	BCB	BB	B
B)	ABCABC	BCABC	BCBC	BBC	BB	B
C)	ABABCB	ABBCB	ABBB	ABB	AB	
D)	ABCCCC	BCCCC	BCCC	BCC	BC	B

Das ist Informatik!

Was kann berechnet werden? Das hat man sich insbesondere in der ersten Hälfte des 20. Jahrhunderts in der Wissenschaft gefragt.

Man hat nach Möglichkeiten gesucht, das Wesen des „Berechnens“ bzw. den Begriff des „Berechnungsverfahrens“ (des Algorithmus) formal zu modellieren.

Am bekanntesten ist die Turing-Maschine. Sie ist trotz ihres Namens keine Maschine – wurde also nie gebaut — sondern ein Gedankenmodell des Briten Alan Turing.

Nicht ganz so bekannt sind Textersetzungssysteme, wie sie etwa der Russe Andrei Markow beschrieben hat. Ein solches Textersetzungssystem wird von den biberischen Korrektoren in dieser Biber-Aufgabe angewendet.

Die Informatik ist darüber froh, dass alle bisher gefundenen Modelle, weitere sind etwa der Lambda-Kalkül und die μ -rekursiven Funktionen, sich als gleichwertig und gleichmächtig erwiesen haben.



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Nur neun Tasten

Daniel schreibt auf seinem alten Handy Nachrichten.

Für jeden Buchstaben muss er die passende Taste einmal, zweimal, dreimal oder viermal tippen.

Danach kommt eine kurze Pause.

Für das Zeichen „C“ tippt er zum Beispiel dreimal die Taste mit der Ziffer 2, denn C ist der dritte Buchstabe auf dieser Taste. Für das Wort GUT tippt er insgesamt vier Mal: einmal die 4, zweimal die 8, einmal die 8.

Daniel tippt sechs Mal, um den Namen einer Freundin zu schreiben.

Welches ist der Name der Freundin?

- A) Miriam
- B) Emma
- C) Iris
- D) Ina



Antwort D ist richtig:

„Miriam“ hat zwar sechs Buchstaben, erfordert aber zwölf Mal tippen: einmal die 6, dreimal die 4, dreimal die 7, dreimal die 4, einmal die 2 und einmal die 6. „Emma“ erfordert fünf Mal tippen: zweimal die 3, einmal die 6, einmal die 6 und einmal die 2. „Iris“ erfordert sogar dreizehn Mal tippen: dreimal die 4, dreimal die 7, dreimal die 4 und viermal die 7. „Ina“ erfordert sechs Mal tippen: dreimal die 4, zweimal die 6 und einmal die 2.

Das ist Informatik!

Auf nur neun Tasten können 26 Buchstaben und 4 Satzzeichen eindeutig eingegeben werden. Das Prinzip ist es, Zeichen auf gleichen Tasten dadurch zu unterscheiden, wie oft diese Taste gedrückt wird.

Ältere Telefonapparate, Handys, Fernbedienungen und Drucker verlangen noch solche Eingaben. Aktuelle Smartphones und Tablets können dagegen mit ihren sensitiven Oberflächen beliebige zur Situation passende Tasten anbieten.

Welche Eingaben wir in zehn Jahren vorwiegend benutzen werden, ist schwer vorherzusagen. Vielleicht eine Mischung von Gestikulieren, Sprechen und Hindenken. Die Informatik experimentiert jedenfalls schon intensiv mit allem, was an Techniken gerade reif wird.

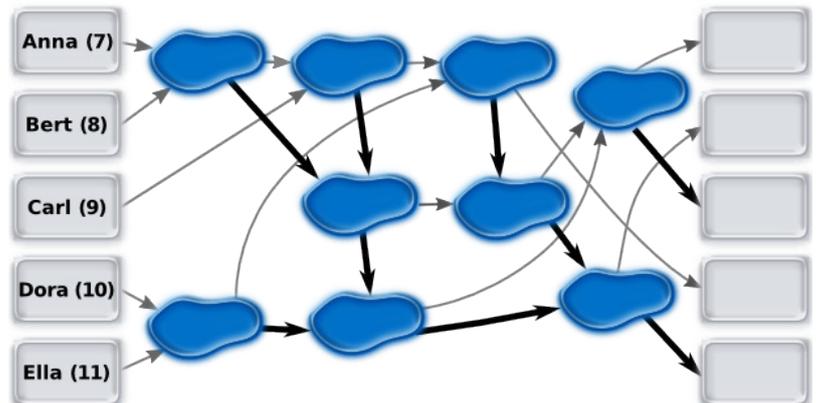
Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Pfützenspringen

Anna (7 Jahre), Bert (8 Jahre), Carl (9 Jahre), Dora (10 Jahre) und Ella (11 Jahre) spielen ein Spiel, bei dem sie von Pfütze zu Pfütze springen. Dazu haben sie Pfeile auf den Boden gemalt.

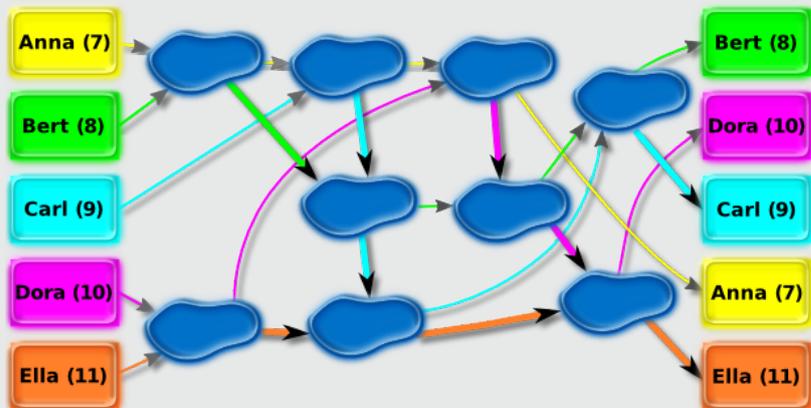
Am Anfang stehen die Kinder auf den Feldern links und springen entlang der Pfeile jeweils zu einer Pfütze. Ein Kind, das zuerst in einer Pfütze ist, wartet, bis ein zweites dazu kommt. Das ältere Kind springt dann weiter entlang des dicken Pfeiles, das jüngere entlang des dünnen Pfeiles.



Ziehe jeden Namen auf das Feld rechts, auf dem das Kind am Ende ankommt.

So ist es richtig:

Das folgende Bild zeigt die Wege der fünf Kinder.

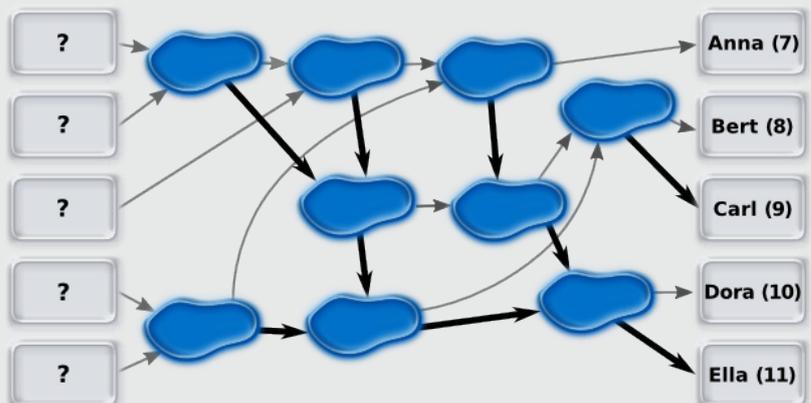


Das ist Informatik!

Die Pfützen und Pfeile bilden ein Netzwerk. Die Pfützen funktionieren wie Vergleichselemente.

Wären die Vergleichselemente richtig verbunden, könnte das Netzwerk die fünf Kinder nach ihrem Alter sortieren. Es wäre ein Sortiernetz.

Das Netz in dieser Biber-Aufgabe ist kein Sortiernetz. Aber diese Grafik zeigt ein Sortiernetz – mit den richtigen Verbindungen:





Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Richtige Rechtecke?

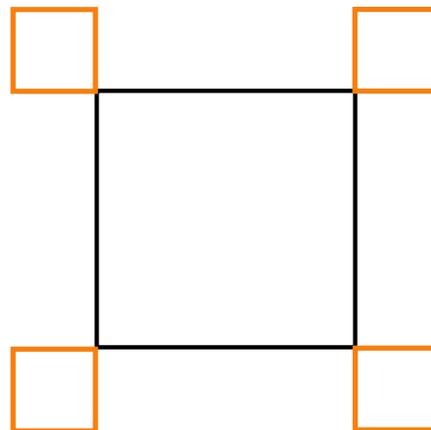
Ein Roboter ist darauf spezialisiert, Rechtecke zu zeichnen. Er kann die folgenden einfachen Anweisungen ausführen:

Orange	zeichne eine orange Linie der Länge 1
Black	zeichne eine schwarze Linie der Länge 1
Turn	drehe dich um 90° im Uhrzeigersinn

Außerdem kann der Roboter die folgenden Anweisungen ausführen, die aus anderen Anweisungen A und B zusammengesetzt werden; A und B selbst können einfache oder zusammengesetzte Anweisungen sein:

A, B	führe A aus und führe danach B aus
n×(B)	führe B n-mal aus

Der Roboter soll die folgende Zeichnung erstellen.
Welche Anweisung ergibt **NICHT** die gewünschte Zeichnung?



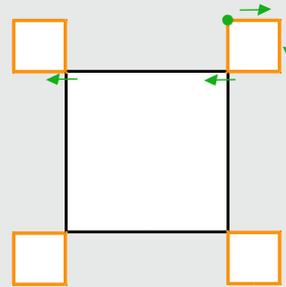
- A) 4×(2×(Orange, Turn), Orange, 3×(Black), Orange, Turn)
- B) 4×(3×Black, 3×(Orange, Turn), Orange)
- C) 4×(2×(Orange, Turn), 3×(Black), 2×(Orange, Turn))
- D) 4×(Black, 3×(Orange, Turn), Orange, 2×(Black))

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

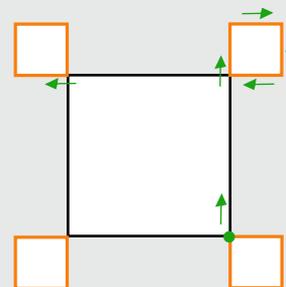


Antwort C ist richtig:

Die Anweisung A ergibt die gewünschte Zeichnung:



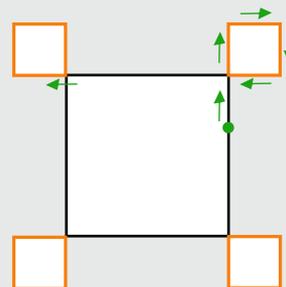
Die Anweisung B ergibt die gewünschte Zeichnung:



Die Anweisung C ergibt **NICHT** die gewünschte Zeichnung:



Die Anweisung D ergibt die gewünschte Zeichnung:



Das ist Informatik!

Selbst wenn man eine sehr einfache Programmiersprache wie die des Rechteck-Roboters verwendet, kann man leicht Fehler machen. Fehler in Programmen heißen in der Informatik „Bugs“, und „Debugging“ bezeichnet die Suche nach Fehlern in einem Programm. Im Jahr 2014 wurde der „Heartbleed Bug“ bekannt. Es handelte sich um einen Fehler in einer weit verbreiteten Programmbibliothek für verschlüsselten Datenaustausch. Dieser Fehler machte Angriffe auf viele Internet-Dienste möglich, zum Beispiel zum Stehlen von Zugangsdaten (Benutzernamen und Passwörtern). Bugs können also weitreichende Auswirkungen haben. Debugging gehört – neben dem systematischen Vermeiden von Fehlern – zu den ganz besonders wichtigen Arbeiten in der Informatik.

<<http://de.wikipedia.org/wiki/Programmfehler>>

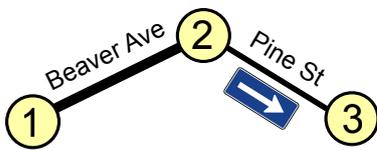
<<http://xkcd.com/1354/>>



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

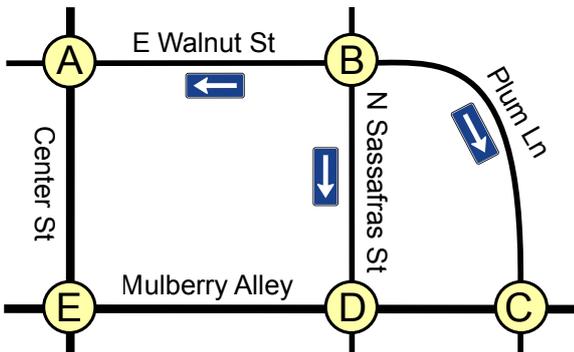
Stadtverkehr

In Beaver Springs wurde neulich die Pine Street zur Einbahnstraße gemacht. Nun muss Jack, der einzige Taxifahrer in der Gegend, sich neu merken, wie er von einem Ort zum anderen kommt. Für die drei Knotenpunkte 1, 2 und 3 legt Jack die folgende Tabelle an. Er setzt Häkchen in einige Felder der Tabelle, um sich zu merken, in welche Richtungen er welche Straßen befahren darf.



	1	2	3
1		✓	
2	✓		✓
3			

Auch im Nachbarort Beavertown wurden einige Straßen zu Einbahnstraßen gemacht.



Jack braucht also auch für Beavertown eine Tabelle mit Häkchen.

	A	B	C	D	E
A					
B					
C					
D					
E					

Hier siehst du die leere Tabelle für Beavertown. Setze die Häkchen in die richtigen Felder!

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



So ist es richtig:

Ein Häkchen im Tabellenfeld in Zeile X und einer Spalte Y (kurz: im Feld (X,Y)) bedeutet, dass Jack vom Knotenpunkt X zum Knotenpunkt Y fahren kann.

Für eine in beiden Richtungen befahrbare Straße wie Mulberry Alley zwischen den Knotenpunkten D und E müssen also zwei Häkchen gesetzt werden: eines ins Feld (D,E) und eines ins Feld (E,D). Verläuft zwischen zwei Knotenpunkten eine Einbahnstraße, wie etwa die Plum Lane von B nach C, darf nur ein Häkchen gesetzt werden, in diesem Fall ins Feld (B,C).

Das ist Informatik!

Die ausgefüllte Tabelle sagt genau, von welchem Knotenpunkt man direkt zu welchem anderen fahren darf. Gleichzeitig sagt sie aber nichts über besondere Eigenschaften der Verbindung zwischen zwei Knotenpunkten.

Ein Taxifahrer würde über eine Verbindung vermutlich auch gerne wissen, wie schnell gefahren werden kann, wie wahrscheinlich ein Stau ist, ob der Straßenbelag gut in Schuss ist usw. Aber um zu entscheiden, ob man überhaupt von A nach B gelangt, evtl. auch über mehrere Verbindungen, genügt die durch die Häkchen in der Tabelle gegebene Information. Informatiksysteme (und Menschen wohl auch) verarbeiten meist nur die Informationen, die für die gewünschte Funktion nötig sind. Sie verwenden ein abstraktes Modell der Wirklichkeit.

	A	B	C	D	E
A					✓
B	✓		✓	✓	
C				✓	
D			✓		✓
E	✓			✓	

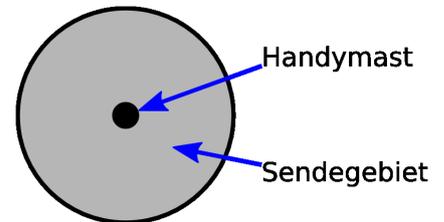


Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Sturmsicheres Netzwerk

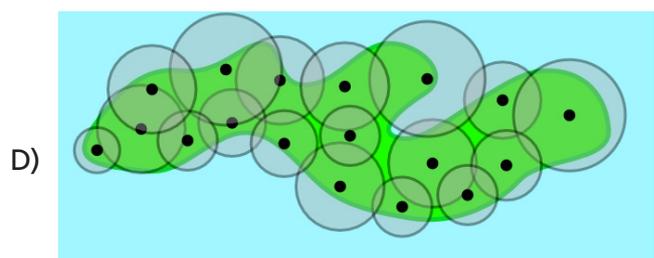
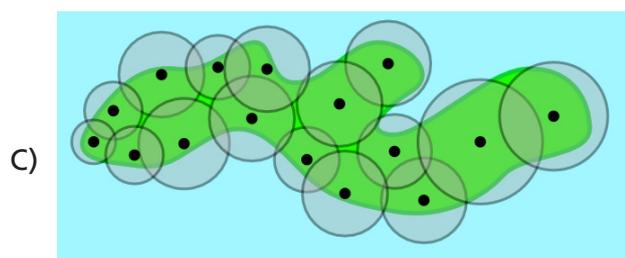
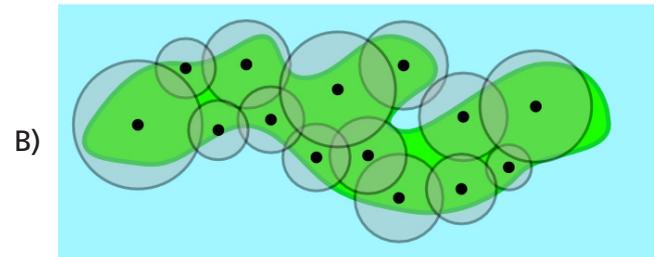
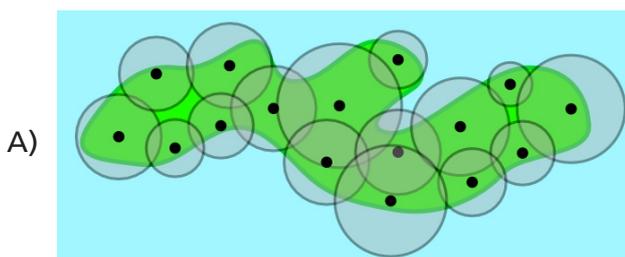
Auf einer sturmumrauten Insel sollen Handymasten aufgestellt werden. Jeder Handymast deckt ein kreisförmiges Sendegebiet ab.

Wenn sich ihre Sendegebiete überlappen, dann sind zwei Handymasten direkt über Funk verbunden. Ein Handymast kann mit einem anderen auch indirekt verbunden sein; über eine Kette direkt miteinander verbundener Handymasten.



Wegen des ständigen Sturmes sollen die Handymasten so aufgestellt werden, dass der Ausfall eines einzelnen Handymastes verkraftet werden kann. Fällt ein Handymast aus, dann sollen alle anderen trotzdem noch immer verbunden sein.

Wie müssen die Handymasten aufgestellt werden?

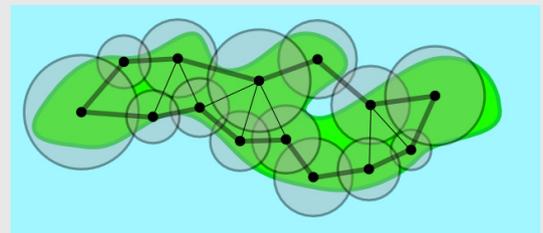


Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



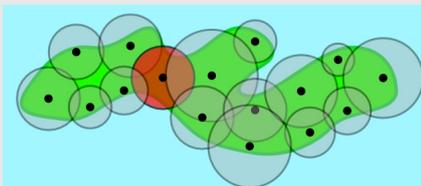
Antwort B ist richtig:

Zeichnet man die direkten Verbindungen der Handymasten in den Plan B, dann entsteht unter anderem auch eine ringförmige Verbindung aller Handymasten (dicker Linienzug).

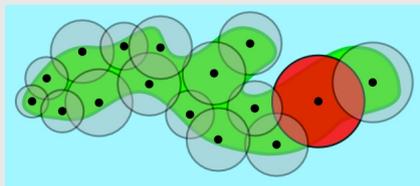


Entfernt man einen beliebigen Handymast, dann bleiben alle übrigen trotzdem noch immer miteinander verbunden.

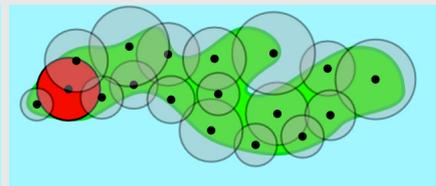
Bei den anderen Plänen gibt es einen solchen Verbindungsring nicht, sondern einen kritischen Handymast (rotes Sendegebiet). Wenn der ausfällt, verbleiben zwei nicht miteinander verbundene Gruppen von Handymasten.



Antwort A



Antwort C



Antwort D

Das ist Informatik!

Bei technischen Systemen können Komponenten ausfallen. Das mobile Telefonieren ist so ein technisches System, die Funkmasten sind Komponenten davon, aber noch vieles andere mehr. Fällt eine Komponente aus, kann das unterschiedlich schwere Folgen haben. Das ganze System könnte schlagartig ausfallen. Oder nur bestimmte Funktionen des Systems könnten ausfallen, während alles andere weiter funktioniert. Oder alles funktioniert normal weiter.

Der Grad der Ausfallsicherheit hängt davon ab, wie stark die Komponenten des Systems aufeinander angewiesen sind und für welche besonders wichtigen Komponenten Ersatz-Komponenten eingebaut sind, die bei Ausfällen einspringen.

Die Informatik ist vor allem zuständig für die Software-Komponenten in technischen Systemen. Wenn eine Software-Komponente einen Programmierfehler enthält oder mit unzulässigen Eingaben „gefüttert“ wird, kann sie falsche Ausgaben produzieren oder ganz „abstürzen“.

Da nutzt es nichts, dem technischen System einen Ersatz-Computer mit derselben fehlerhaften Software hinzuzufügen. Wenn ein technisches System besonders ausfallsicher sein soll, dann muss die Ersatz-Software von einem anderen Team ein zweites Mal geplant und in einer anderen Sprache programmiert werden (diversitäre Redundanz).



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Suanpan

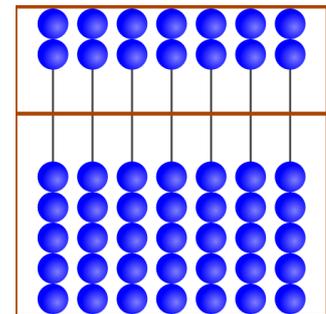
Der „Suanpan“ ist ein traditionelles chinesisches Rechenbrett. Mit seinen Kugeln kann man Zahlen einstellen. Dazu stellt man an den Stangen die einzelnen Ziffern der gewünschten Zahl ein.

Im oberen Feld hat jede Kugel den Wert „5“. Im unteren Feld hat jede Kugel den Wert „1“.

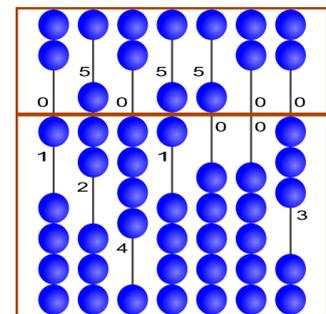
Sind an einer Stange alle Kugeln von der Mittellinie weggeschoben, dann ist die eingestellte Ziffer die „0“.

Will man eine andere Ziffer einstellen, dann schiebt man die notwendigen Kugeln zur Mittellinie.

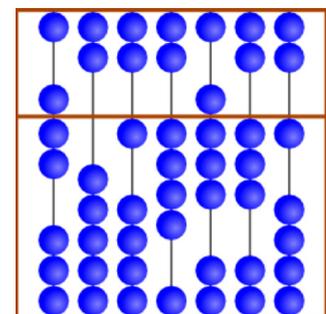
Im Beispiel sind an den Stangen die Ziffern 1, 7, 4, 6, 5, 0 und 3 eingestellt. Insgesamt ist also die Zahl 1746503 eingestellt.



0 0 0 0 0 0 0



1 7 4 6 5 0 3



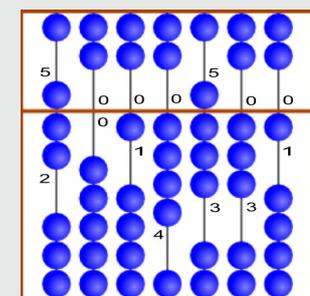
Welche Zahl ist hier eingestellt?

Das ist Informatik!

Seit tausenden von Jahren benutzen die Menschen Hilfsmittel, um sich große Zahlen zu merken und damit zu rechnen. In dieser Biber-Aufgabe wird der Suanpan vorgestellt, eine chinesische Variante des bekannten Abakus. Suanpans sind schon lange Zeit in Gebrauch, und für viele Menschen sind sie bis heute noch ein regelmäßiges Hilfsmittel. Der Suanpan wurde zusammen mit seiner Rechenmethode Zhusuan im Jahr 2013 in die „Repräsentative Liste des immateriellen Kulturerbes der Menschheit“ der UNESCO aufgenommen.

<http://www.unesco.de/ike_neuaufnahmen_2013.html>

Die richtige Zahl ist:



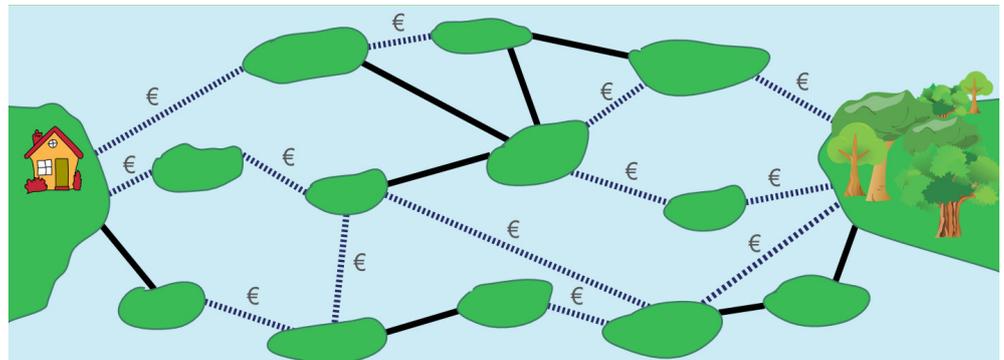
7 0 1 4 8 3 1

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Teure Brücken

Die Inseln im See sind über öffentliche und private Brücken verbunden. Über eine private Brücke (gestrichelte Linie) zu gehen kostet eine Gebühr. Über eine öffentliche Brücke (durchgezogene Linie) zu gehen kostet nichts.



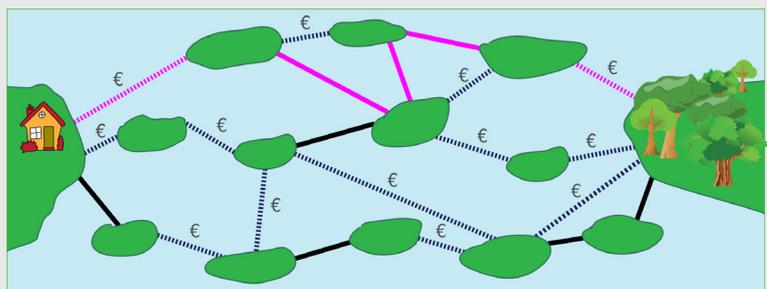
Sandy möchte von ihrem Haus zum Wald gehen. Sandy sucht einen Weg mit möglichst wenigen Brücken. Aber sie ist knapp bei Kasse und kann sich nur Wege mit höchstens zwei privaten Brücken leisten.

Finde unter den Wegen mit höchstens zwei privaten Brücken den mit den insgesamt wenigsten Brücken.

Wie viele Brücken hat dieser Weg?

5 ist richtig:

Es gibt keinen Weg von Sandys Haus zum Wald mit weniger als vier Brücken. Alle Wege mit vier Brücken enthalten drei oder mehr private Brücken; diese Wege kann Sandy sich nicht leisten. Das Bild zeigt einen Weg mit fünf Brücken, von denen zwei privat sind. Das ist der kürzeste Weg, den Sandy sich leisten kann.



Das ist Informatik!

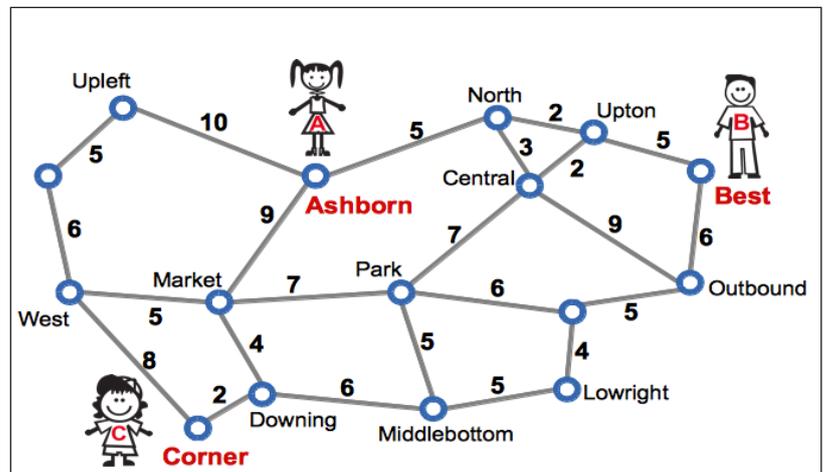
Brücken zwischen Inseln, Straßen zwischen Orten, Netzwerkverbindungen zwischen Computern, Leiterbahnen zwischen den Lötstellen einer Platine: Es gibt viele, verschiedene Lebensbereiche, in denen Objekte untereinander auf irgendeine Art und Weise verbunden sind. Um Systeme zu entwickeln, die in solchen Bereichen nützlich sind, greift die Informatik gerne auf ein Modell aus der Mathematik zurück: den Graph. Als Ursprung der Graphentheorie gilt die Arbeit des Schweizer Universalgenies Leonhard Euler über das „Königsberger Brückenproblem“. Euler zeigte im Jahr 1736, dass es einen Rundweg über die damals existierenden Brücken der Stadt Königsberg (das heutige Kaliningrad) nicht geben kann.



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Treffpunkt

Anne, Bernie und Clara wohnen in einer Stadt mit einem gut ausgebauten U-Bahn-Netz. Der Netzplan (siehe Bild) zeigt Haltestellen und die Strecken zwischen den Haltestellen. Für jede Strecke ist angegeben, wie viele Minuten reine Fahrzeit man für die Strecke benötigt.



Anne wohnt bei der Haltestelle Ashborn, Bernie bei Best und Clara bei Corner. Sie wollen sich an irgendeiner Haltestelle treffen, aber jeder möchte höchstens 15 Minuten reine Fahrzeit benötigen.

Welche Haltestellen kommen als Treffpunkt in Frage?

So ist es richtig:

Park und Ashborn kommen als Treffpunkt in Frage. Zu diesen Haltestellen benötigen alle drei bei folgender Streckennutzung höchstens 15 Minuten reine Fahrzeit:

Park: Ashborn-North-Central-Park: 15 Min.;

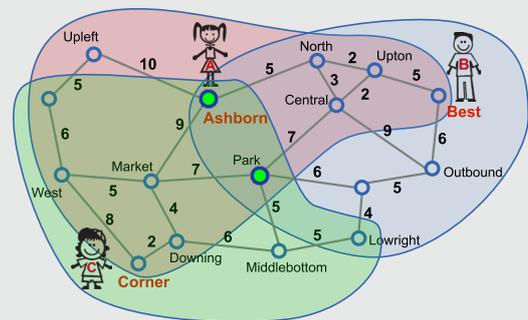
Best-Upton-Central-Park: 14 Min.;

Corner-Downing-Market-Park oder Corner-Downing-Middlebottom-Park: 13 Min. (Für den anscheinend direkteren Weg Ashborn-Market-Park würde Anne mehr Zeit benötigen als gewünscht, nämlich 16 Minuten.);

Ashborn: Ashborn-Ashborn: 0 Min. (Anna muss hier also gar nicht fahren);

Best-Upton-North-Ashborn: 12 Min.;

Corner-Downing-Market-Ashborn: 15 Min. Im Bild zeigen die farbigen Bereiche, welche Haltestellen Anne, Bernie bzw. Clara mit höchstens 15 Minuten reiner Fahrzeit erreichen können. Nur Ashborn und Park liegen im Schnitt der drei Bereiche. Es gibt also keine anderen möglichen Treffpunkte.



Das ist Informatik!

Beziehungen zwischen Elementen nur einer Menge werden häufig zu „Graphen“ erweitert: Die Elemente heißen dann „Knoten“, und „Kanten“ sind Paare von Knoten, die miteinander in Beziehung stehen. In manchen Graphen haben die Kanten eine Richtung: Knoten a steht in Beziehung mit Knoten b, aber nicht umgekehrt. Kanten können auch bestimmte Werte als „Gewichte“ zugewiesen werden. Ein Verkehrsnetz kann sehr gut durch einen Graph modelliert werden, und zwar mit Kantengewichten, die der reinen Fahrzeit entsprechen. Erfreulicherweise hat die Informatik viele leistungsfähige Algorithmen für Graphen entwickelt; zum Beispiel welche, die kürzeste Wege zwischen Knoten finden.

„Kürzeste-Wege-Algorithmen“ wie der von Edsger W. Dijkstra bilden die Grundlage für Systeme zur Routenplanung, zum Beispiel Navigationssysteme für Autos.

<<http://de.wikipedia.org/wiki/Dijkstra-Algorithmus>>

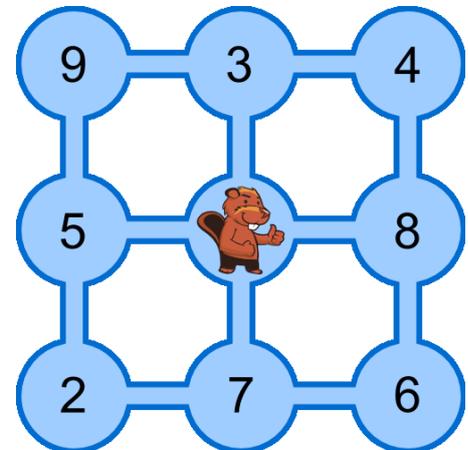
Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Viele Freunde

Im Bild siehst du neun Teiche. Sie sind durch Kanäle verbunden. Tobi Biber lebt im mittleren Teich, seine Freunde leben in den anderen Teichen. Die Zahlen zeigen an, wie viele Freunde in jedem Teich leben.

Tobi möchte seine Freunde besuchen. Er startet zuhause, schwimmt jeden Tag durch einen Kanal in einen anderen Teich, besucht dort seine Freunde und bleibt über Nacht. Am nächsten Tag schwimmt er weiter.

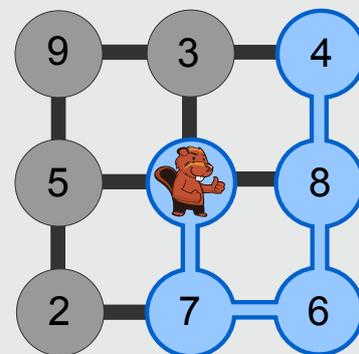


Wie viele verschiedene Freunde kann Tobi innerhalb von vier Tagen höchstens besuchen? Es ist ihm egal, in welchem Teich er nach den vier Tagen ankommt.

- A) 21 Freunde B) 24 Freunde C) 25 Freunde D) 30 Freunde

Antwort C ist richtig:

Tobi kann innerhalb von vier Tagen 25 Freunde besuchen. Der Weg (7-6-8-4), den er dazu nehmen muss, ist im Bild blau dargestellt. Die Antworten A und B ergeben sich aus Schwimmrouten, auf denen Tobi zwar einzelne Teiche mit vielen Freunden, aber nicht insgesamt möglichst viele Freunde besucht. Antwort D ergibt sich, wenn die vier größten Freunde-Zahlen summiert werden, obwohl Tobi die zugehörigen Teiche nicht innerhalb von vier Tagen erreichen kann. Auch alle anderen Kombinationen von Teichen, in denen insgesamt mehr als 25 Freunde leben, kann Tobi nicht innerhalb von vier Tagen erreichen.



Das ist Informatik!

Aus Sicht der Informatik bilden die Teiche und Kanäle einen Graph. Die Teiche sind die Knoten im Graph, die Freunde-Zahlen die Werte der Knoten, die Kanäle sind die Kanten im Graph.

Gesucht wird im Graph ein zusammenhängender Weg, der die Bedingungen erfüllt:

1. Tobis Teich ist der Startknoten. Sein Knotenwert ist 0.
2. Der Weg führt über höchstens 4 Kanten. Jeder Knoten kommt nur einmal vor.
3. Die Summe der Knotenwerte, die auf dem Weg liegen, ist so groß wie möglich.

In dem kleinen Graphen dieser Biber-Aufgabe lässt sich der gesuchte Weg mit Durchprobieren rasch finden. Für große Graphen mit vielen Kanten gebraucht die Informatik Methoden, eine gute Lösung zu finden, ohne die gigantisch vielen möglichen Wege alle durchprobieren zu müssen.

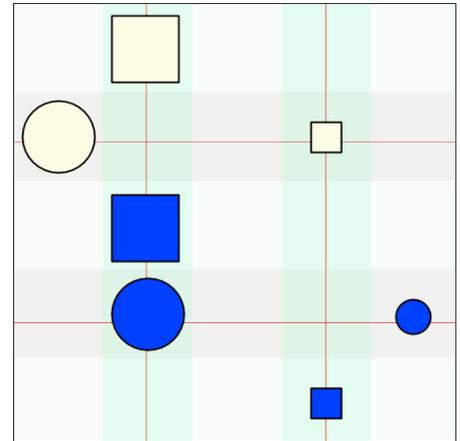


Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Wahr oder falsch

Alice und Tom spielen „Wahr oder falsch“ an der Magnettafel im Klassenzimmer. Alice bringt sieben verschiedene Magnete an der Tafel an. Dann stellt sie Behauptungen über Form, Farbe, Größe und Position der Magnete auf.

Eine Behauptung ist wahr, die anderen Behauptungen sind falsch. Tom muss herausfinden, welche Behauptung wahr ist.



Welche dieser Behauptungen ist wahr?

- A) Es gibt zwei Magnete X und Y, so dass X dunkelblau ist und Y hellgelb ist und X sich oberhalb von Y befindet.
- B) Für je zwei Magnete X und Y gilt: Falls X ein Quadrat ist und Y ein Kreis ist, dann befindet sich X oberhalb von Y.
- C) Für je zwei Magnete X und Y gilt: Falls X klein ist und Y groß ist, dann befindet sich X rechts von Y.
- D) Für je zwei Magnete X und Y gilt: Falls X hellgelb ist und Y dunkelblau ist, dann befindet sich X unterhalb von Y.

Antwort C ist richtig:

Denn alle kleinen Magnete befinden sich rechts von allen großen Magneten.

A ist falsch: Es gibt keinen dunkelblauen Magnet, der sich oberhalb eines hellgelben

Magneten befindet. B ist falsch: Es befinden sich nicht alle quadratischen Magnete oberhalb

von kreisförmigen Magneten. D ist falsch: Nicht alle hellgelben Magnete befinden sich unterhalb von dunkelblauen Magneten.

Das ist Informatik!

Bei dieser Biber-Aufgabe geht es darum, für die Behauptungen festzustellen, ob sie wahr oder falsch sind. Die Eigenschaften einzelner Magnete können mit den Prädikaten „quadratisch(X)“, „kreisförmig(X)“, „groß(X)“, „klein(X)“, „dunkelblau(X)“ und „hellgelb(X)“ beschrieben werden. Die Beziehungen zwischen zwei Magneten können mit den Prädikaten „oberhalb-von(X,Y)“, „unterhalb-von(X,Y)“ und „rechts-von(X,Y)“ beschrieben werden. In der formalen Sprache der Prädikatenlogik sehen die Behauptungen dann so aus:

A) exist X, Y: dunkelblau(X) and hellgelb(Y) and oberhalb-von(X, Y)

B) forall X, Y: (quadratisch(X) and kreisförmig(Y)) implies oberhalb-von(X, Y)

C) forall X, Y: (klein(X) and groß(Y)) implies rechts-von(X,Y)

D) forall X, Y: (hellgelb(X) and dunkelblau(Y)) implies unterhalb-von(X,Y)

Die Informatik kennt Programmiersprachen, mit denen man direkt in Prädikatenlogik programmiert. Die Programmiersprache Prolog ist ein Beispiel für so eine logikorientierte Programmiersprache.

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Welches Foto?

Johnny hat 8 Fotos gemacht. Eines davon will er gerne Bella geben. Er will herausfinden, welches Foto sie haben möchte.

Dazu stellt er ihr einige Fragen:

„Möchtest du ein Foto mit einem Sonnenschirm“? – „Ja.“

„Möchtest du ein Foto, auf dem ich eine Mütze oder einen Hut trage?“ – „Nein.“

„Möchtest du ein Foto, auf dem das Meer zu sehen ist?“ – „Ja.“

Welches Foto möchte Bella haben?



A)



B)



C)



D)



E)



F)



G)



H)

Antwort H ist richtig:

Die Fotos B, E, G und H passen zu Bellas Antwort auf Johnnys erste Frage.

Die Fotos C, D, G und H passen zu Bellas Antwort auf die zweite Frage.

Die Fotos A, B, D und H passen zur Antwort auf die dritte Frage.

Nur Foto H passt zu allen Antworten.

Das ist Informatik!

Um Daten zu speichern und zu verarbeiten, verwenden aktuelle Computer Bits. Bits können immer nur einen von zwei verschiedenen Werten annehmen: „Ja“ oder „Nein“ (bzw. „an“ oder „aus“, „wahr“ oder „falsch“, „1“ oder „0“). In dieser Biber-Aufgabe kann Bellas Fotowunsch mit drei Bits dargestellt werden; eins für jede Frage, die Johnny stellt. Bellas Antworten bedeuten, dass der erste Bit-Wert „Ja“ ist UND der zweite Bit-Wert „Nein“ (bzw. „NICHT Ja“) ist UND der dritte Bit-Wert „Ja“ ist. Die Informatik weiß, dass die logischen Operationen UND und NICHT genügen, um Bit-Werte auf jede beliebige Art und Weise zu anderen Bit-Werten zu verarbeiten. Alles was Algorithmen leisten, könnten sie alleine mit diesen zwei einfachen logischen Operationen erreichen – zum Beispiel die Auswahl eines Daten-Elements mit bestimmten Eigenschaften (hier: einem Foto) aus einer Daten-Sammlung (hier: Johnnys acht Fotos).



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

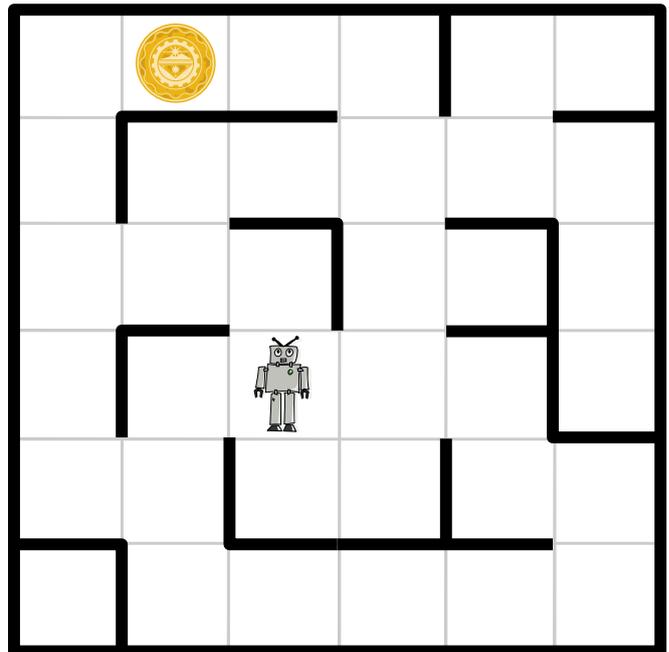
Weltraumlabyrinth

Raumfahrer sind auf einem verlassenen Planeten gelandet. Auf ihren Tele-Brillen sehen sie rätselhafte Bilder. Sie folgen den Signalen und machen als Quelle einen Roboter aus. Er steht in einem Labyrinth, das die Raumfahrer von ihrer erhöhten Position gut überblicken und sendet offensichtlich Nahaufnahmen seiner Umgebung.

Das Labyrinth ist in Quadrate eingeteilt. In einem davon befindet sich der Roboter. In einem anderen Quadrat befindet sich ein unbekanntes Objekt. Die Raumfahrer würden den Roboter gerne zum Objekt steuern, um Nahaufnahmen davon zu sehen.

Plötzlich flimmern vier kryptische Textzeilen mit insgesamt vier verschiedenen Wörtern über die Tele-Brillen. Auch der Roboter und das Objekt sind zu erkennen. Nach einigem Grübeln vermuten die Raumfahrer: Die vier Wörter sind Befehle, die den Roboter jeweils in ein benachbartes Quadrat steuern;

für jede der vier möglichen Richtungen gibt es einen eigenen Befehl. Außerdem sind die Raumfahrer sicher, dass eine der Textzeilen eine Befehlsfolge ist, die den Roboter zum Objekt steuert.



Welche der vier Textzeilen steuert den Roboter zum unbekanntem Objekt?

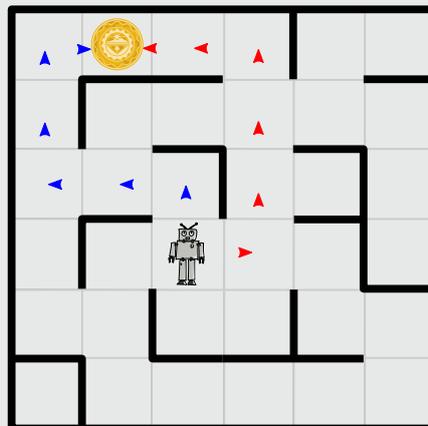
- A) Ha' poS poS Ha' Ha' nIH
- B) Ha' Ha' poS Ha'
- C) Ha' poS poS Ha' nIH Ha'
- D) Ha' poS nIH vl'ogh Ha' poS

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Antwort A ist richtig:

Keine der durch die Textzeilen gegebenen Befehlsfolgen enthält mehr als sechs Befehle. Mit jedem Befehl kann der Roboter einen Schritt in ein benachbartes Quadrat machen. Das Bild zeigt die beiden Wege, die den Roboter in sechs Schritten zum Objekt führen:



Die Befehlsfolge muss also den Roboter entweder so steuern (rote Pfeile): rechts, vor, vor, vor, links, links. Dazu passt keine der vier Textzeilen. Oder die Befehlsfolge muss den Roboter so steuern (blaue Pfeile): vor, links, links, vor, vor, rechts. Dazu passt nur Textzeile A mit Ha' = vor, poS = links und nIH = rechts.

Das ist Informatik!

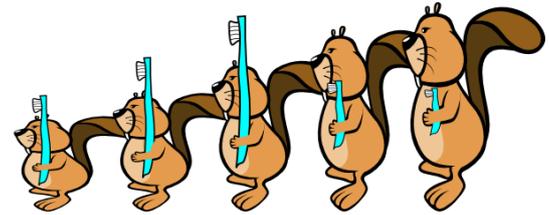
Kryptoanalyse ist die Wissenschaft des Lesens verschlüsselter Botschaften. Seit der Antike versuchen Kryptoanalytiker geheime Nachrichten zu entschlüsseln. Dabei wird auch das Wissen über die mögliche Bedeutung der verschlüsselten Botschaften verwendet. Als im Zweiten Weltkrieg versucht wurde, die von der Enigma-Maschine verschlüsselten Botschaften zu entschlüsseln, suchte man gezielt nach deutschen Städtenamen und nach Wörtern, die in Wetterberichten vorkommen. Denn wichtige Nachrichten begannen oft mit einer Wettervorhersage. Bei dieser Biber-Aufgabe konntest du dich als Kryptoanalytikerin oder Kryptoanalytiker betätigen. Die Entschlüsselung ist übrigens wesentlich einfacher, wenn man klingonisch spricht. ;-)



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Zahnbürsten

„Nicht so schnell!“ sagt Mutter Biber.
„Eve und Chad, tauscht sofort die Zahnbürsten!
Ann und Chad,
danach tauscht ihr die Zahnbürsten!“



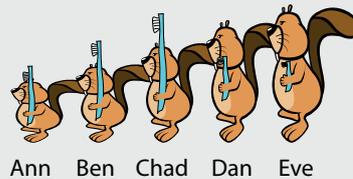
Ann Ben Chad Dan Eve

Welche zwei Biber müssen noch ihre Zahnbürsten tauschen, so dass jeder die richtige Bürste hat?

- A) Ben und Chad
- B) Ben und Dan
- C) Ann und Eve
- D) Niemand

Antwort B ist richtig:

Anfangszustand:



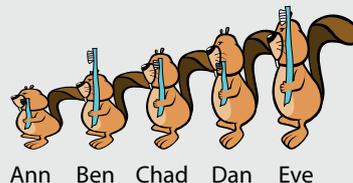
Ann Ben Chad Dan Eve

„Eve und Chad, tauscht eure Zahnbürsten!“



Ann Ben Chad Dan Eve

„Ann und Chad, ihr beiden auch!“



Ann Ben Chad Dan Eve

Nun müssen nur noch Ben und Dan ihre Bürsten tauschen.

Das ist Informatik!

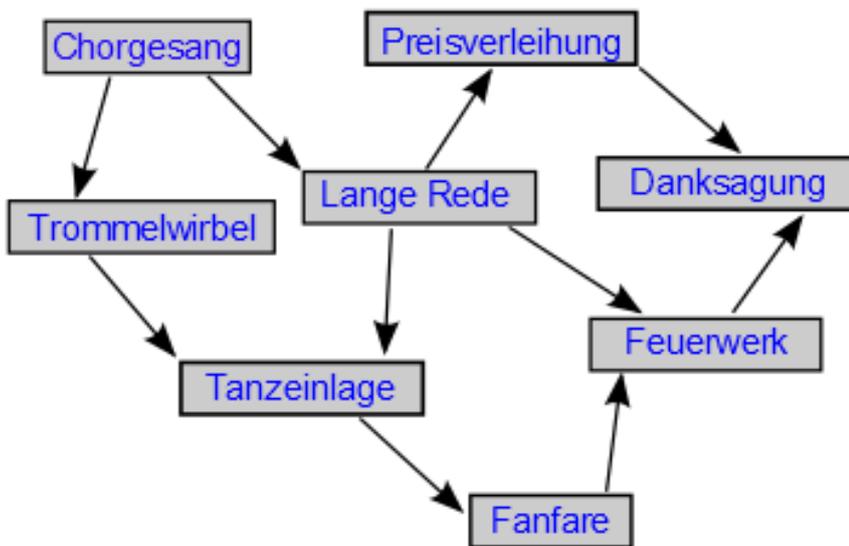
Programmierer sind oft wie Mütter, die für Ordnung sorgen. Doch an Stelle von Zahnbürsten bewegen sie Zahlen in Speicherzellen des Computers. Das Vertauschen von Daten ist eine Grundoperation der Programmierung. Häufig muss eine Gruppe von Zahlenwerten nach der Größe sortiert werden. Die Zahlen sind in aufeinander folgenden Zellen gespeichert. Das Computerprogramm muss dafür sorgen, dass die kleinste Zahl in die erste Zelle kommt, die zweitkleinste in die zweite und so weiter, bis schließlich die größte Zahl in der letzten Zelle landet. Dieses Sortieren kann man bewerkstelligen, indem man mehrfach die Inhalte von Speicherzellen vertauscht.

Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer



Zeremonie

Eine feierliche Zeremonie besteht aus einzelnen Aktionen, die unbedingt in der richtigen Reihenfolge durchzuführen sind. Links siehst du die Bestandteile einer Zeremonie. Ein Pfeil von einer ersten Aktion zu einer zweiten bedeutet, dass die erste vor der zweiten stattfinden muss. Zum Beispiel muss der Chorgesang vor dem Trommelwirbel und der langen Rede stattfinden.



Plane eine Zeremonie!

Ziehe die einzelnen Aktionen in der Box rechts in eine richtige Reihenfolge. Die erste Aktion der Zeremonie soll oben sein, darunter kommt die nächste, usw.



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

So ist es richtig:

Eine Zeremonie kann nach der folgenden Vorschrift geplant werden:

Solange es eine Aktion gibt, zu der Pfeile nur von bereits durchgeführten Aktionen zeigen, führe sie durch.

Nach dieser Vorschrift ergibt sich der Chorgesang als einzig mögliche erste Aktion. Dann folgt wahlweise der Trommelwirbel oder die lange Rede. So geht es weiter, bis als letzte Aktion die Danksagung stattfindet.

Dies ist eine Lösung:	Dies ist eine andere Lösung:
Chorgesang	Chorgesang
Trommelwirbel	Lange Rede
Lange Rede	Trommelwirbel
Tanzeinlage	Preisverleihung
Fanfare	Tanzeinlage
Preisverleihung	Fanfare
Feuerwerk	Feuerwerk
Danksagung	Danksagung

Das ist Informatik!

Selbst im Alltag gibt es Vorrangbeziehungen zwischen Handlungen: Beim Anziehen sind die Socken vor den Schuhen und die Unterhose vor der Hose an der Reihe. Die Hose sollte wiederum vor den Schuhen angezogen werden. Ob aber die Socken oder die Unterhose zuerst dran sind, wäre egal.

Wenn beim Anziehen alles klappt, war aus Sicht der Informatik die Reihenfolge der Handlungen „topologisch sortiert“.

In der Informatik sind topologische Sortierungen wichtig: Zum Beispiel müssen Programmteile, deren Ergebnisse von anderen Programmteilen benötigt werden, vor diesen ausgeführt werden.

Ist eine topologische Sortierung vorhanden, dann ist es sicher, dass es zwischen den Handlungen keine wechselseitigen Vorrangbeziehungen gibt. Solche „Zyklen“ sind unerwünscht, weil sie zu einer Blockade des Handlungsablaufs führen.

<http://de.wikipedia.org/wiki/Topologische_Sortierung>

Bundeswettbewerb Informatik

Talente zu entdecken und zu fördern ist Ziel dieses Leistungswettstreits, an dem jährlich über 1.000 junge Menschen unter 21 Jahren teilnehmen. Allen Teilnehmern stehen weitergehende Fördermaßnahmen offen, die Sieger werden ohne Aufnahmeverfahren in die Studienstiftung des deutschen Volkes aufgenommen.

Lange Tradition, hohe Qualität

Der Bundeswettbewerb Informatik (BwInf) wurde 1980 von der Gesellschaft für Informatik e.V. (GI) auf Initiative von Prof. Dr. Volker Claus ins Leben gerufen. Ziel des Wettbewerbs ist, das Interesse an Informatik zu wecken und zu intensiver Beschäftigung mit ihren Inhalten und Methoden sowie den Perspektiven ihrer Anwendung anzuregen. Er gehört zu den bundesweiten Schülerwettbewerben, die von den Kultusministerien der Länder empfohlen werden. Der Bundeswettbewerb Informatik ist der traditionsreichste unter den Bundesweiten Informatikwettbewerben (BWINF), zu denen auch der Informatik-Biber gehört. Gefördert wird er, wie BWINF insgesamt, vom Bundesministerium für Bildung und Forschung und steht unter der Schirmherrschaft des Bundespräsidenten. Träger von BWINF sind die Gesellschaft für Informatik, der Fraunhofer-Verbund IuK-Technologie und das Max-Planck-Institut für Informatik.

Start und Ziel im September

Der Wettbewerb beginnt jedes Jahr im September, dauert etwa ein Jahr und besteht aus drei Runden. Die Aufgaben der ersten und zweiten Runde werden zu Hause selbstständig bearbeitet, einzeln oder in einer Gruppe. An der zweiten Runde dürfen jene teilnehmen, die wenigstens drei Aufgaben weitgehend richtig gelöst haben. In der zweiten Runde ist dann eigenständiges Arbeiten gefordert. Die ca. dreißig bundesweit Besten werden zur dritten Runde, einem Kolloquium, eingeladen.

Internationale Informatik-Olympiade

Die Jüngeren unter den Finalisten können sich in mehreren Trainingsrunden und Vorbereitungswettbewerben im europäischen Ausland für das vierköpfige deutsche Team qualifizieren, das im Folgejahr an der Internationalen Informatik-Olympiade (IOI) teilnimmt.

Lebenslange Vernetzung

Die bereits mehr als 30 Jahrgänge von BwInf-Teilnehmenden bilden ein langfristig wachsendes Netzwerk, vor allem im BWINF Alumni und Freunde e. V. und innerhalb der Studienstiftung des deutschen Volkes. Erste Anknüpfungspunkte bieten auch BwInf – Informatik erleben bei Facebook, Einstieg Informatik (unterstützt vom Fakultätentag Informatik) und die Website des Bundeswettbewerb Informatik.



Träger:



Gesellschaft
für Informatik



GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung