



10 Jahre + 2006 - 2016

BIBER DER INFORMATIK 2016

AUFGABEN und LÖSUNGEN



www.ocg.at/biber-der-informatik



Herausgeber

Gerald Futschek, Technische Universität Wien

Die Biberaufgaben 2016 wurden von Deutschland, Österreich und der Schweiz gemeinsam erarbeitet und waren letztendlich nahezu identisch. Der Aufgabenteil dieser Broschüre wurde vom deutschen Bundeswettbewerb Informatik (BWINF) unter Federführung von Wolfgang Pohl und Hans-Werner Hein sorgfältig erstellt und für unser österreichisches Biberheft übernommen.

Das österreichische Biber Team 2016

Wilfried Baumann, Österreichische Computer Gesellschaft

Jürgen Frühwirth, GWIKU-18, Wien

Gerald Futschek, Technische Universität Wien

Peter Garscha, Technische Universität Wien

Bernd Kurzmann, BG / BRG / WISKU 11, Wien

Roman Ledinsky, Technische Universität Wien

Peter Micheuz, Alpe Adria Universität Klagenfurt

Katharina Resch-Schobel, Österreichische Computer Gesellschaft

Der Biber der Informatik

ist ein Projekt der Österreichischen Computer Gesellschaft OCG in Zusammenarbeit mit dem Institut für Softwaretechnik und Interaktive Systeme der Technischen Universität Wien. Der Biber der Informatik wird vom Bundesministerium für Bildung (BMB) empfohlen.

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

Andrea Adamoli, SVIA

Christian Datzko, Wirtschaftsgymnasium und Wirtschaftsmittelschule Basel

Susanne Datzko, freischaffende Graphikerin

Olivier Ens, Student Pädagogische Hochschule Luzern

Hanspeter Erni, Pädagogische Hochschule Luzern

Martin Guggisberg, Fachhochschule Nordwestschweiz

Hans-Werner Hein, BWINF Bonn

Juraj Hromkovic, ETH Zürich

Ulrich Kiesmüller, Simon-Marius-Gymnasium Gunzenhausen

Wolfgang Pohl, BWINF Bonn

Kirsten Schlüter, St.-Emmeram-Realschule Aschheim

Michael Weigend, Holzkamp-Gesamtschule Witten

Copyright 2016 - OCG, Lizenz:



Einleitung

Der Biber der Informatik, international unter der Bezeichnung „Bebras-Challenge“ bekannt, ist ein Online-Wettbewerb mit Aufgaben zur Informatik. Er erfordert Informatisches Denken, aber keine Vorkenntnisse. Der Biber der Informatik will das allgemeine Interesse für das Fach Informatik wecken.

Schülerinnen und Schüler, die mehr im Bereich Informatik tun wollen, sind herzlich eingeladen, sich auch am Computer Creative Wettbewerb oder bei der Informatik Olympiade zu versuchen (siehe Seite 5).

Der Biber der Informatik findet jährlich im November statt. 2016 bereits zum 10. Mal! In der Biberwoche vom 7. bis 11. November 2016 beteiligten sich 229 Schulen mit insgesamt 21291 Schülerinnen und Schülern. Die Möglichkeit, auch in Zweierteams zu arbeiten, wurde gerne genutzt.

Die Online-Teilnahme am Biber der Informatik 2016 war mit Desktops, Laptops und Tablets möglich. Etwa die Hälfte der Antworteingaben waren multiple-choice. Verschiedene andere Interaktionsformen machten die Bearbeitung abwechslungsreich. Der Umgang mit dem Wettbewerbssystem konnte vor der Austragung geübt werden.

Der Biber der Informatik 2016 wurde in fünf Altersgruppen durchgeführt. In den Schulstufen 3 bis 4 (Volksschule) waren innerhalb von 30 Minuten 9 Aufgaben zu lösen, jeweils drei in den Schwierigkeitsstufen leicht, mittel und schwer. In den Schulstufen 5 bis 6, 7 bis 8, 9 bis 10 und 11 bis 13 waren innerhalb von 40 Minuten 15 Aufgaben zu lösen, jeweils fünf in den Schwierigkeitsstufen leicht, mittel und schwer.

Die 41 Aufgaben des Biber der Informatik 2016 sind auf Seite 6 gelistet, nach ungefähr steigender Schwierigkeit und mit einer informatischen Klassifikation ihres Aufgabenthemas. Ab Seite 7 folgen die Aufgaben nach ihrem Titel alphabetisch sortiert. Im Seitenkopf sind die zugeordneten Altersgruppen und Schwierigkeitsgrade vermerkt. Eine kleine Flagge gibt an, aus welchem Bebras-Land die Idee zu dieser Aufgabe stammt.

Der Kasten am Aufgabenende enthält Erläuterungen zu den Lösungen und Lösungswegen sowie eine kurze Darstellung des Aufgabenthemas hinsichtlich seiner Relevanz in der Informatik. Die Veranstalter bedanken sich bei allen Lehrkräften, die mit großem Engagement ihren Klassen und Kursen ermöglicht haben, den Biber der Informatik zu erleben. Wir laden die Schülerinnen und Schüler ein, auch 2017 wieder mitzumachen. Der genaue Termin und vieles mehr wird auf der Website biber.ocg.at und per E-Mail an die Koordinatorinnen und Koordinatoren bekannt gegeben.



Gerald Futschek

Bebras: International Challenge on Informatics and Computational Thinking



Der polnische Biber

Die Bebras-Community erarbeitet jedes Jahr auf einem internationalen Workshop anhand von Vorschlägen der Länder eine größere Auswahl möglicher Aufgabenideen. Die 41 Aufgabenideen des Biber der Informatik 2016 stammen aus 18 Ländern: Australien, Deutschland, Frankreich, Großbritannien, Irland, Island, Italien, Japan, Kanada, Litauen, Niederlande, Österreich, Russland, Schweiz, Slowakei, Tschechien, Ungarn und USA.



Der taiwanesische Biber



Der ägyptische Biber

Der österreichische Biber der Informatik ist Partner der internationalen Initiative Bebras (Biber auf Litauisch). 2004 fand in Litauen die erste Bebras Challenge statt. 2006 traten Estland, die Niederlande und Polen der Initiative bei, und auch Deutschland veranstaltete einen ersten Biber-Testlauf. 2007 war Österreichs erster Biber der Informatik. Seitdem kamen viele Bebras-Länder hinzu. Zum Drucktermin sind es weltweit 43 aus allen 5 Kontinenten, und weitere Länderteilnahmen sind in Planung.

Insgesamt hatte die Bebras Challenge 2016 international weit über eine Million Teilnehmerinnen und Teilnehmer.

Österreich nutzt gemeinsam mit einer Vielzahl anderer Länder zur Durchführung des Biber der Informatik ein gemeinsames Online-System. Dieses „International Bebras Challenge System“ wird von der niederländischen Firma Eljakim IT betrieben und weiterentwickelt. Informationen über die Aktivitäten aller Bebras-Länder finden sich auf der internationalen Website bebras.org.



Wettbewerbe der OCG für Schülerinnen und Schüler

Computer Creative Wettbewerb (CCW)

Lange Tradition, hohe Qualität

Dieser Wettbewerb hat eine offene Themenstellung. Alle Projekte mit Schwerpunkt Informatik können eingereicht werden. Einzel-, Gruppen- und Klassenprojekte. Von der Volksschule bis zur Matura in verschiedenen Kategorien. Kreativität in der Themenstellung und in der Ausführung ist gefragt, ebenso informatisches Geschick.

Die Österreichische Computer Gesellschaft (OCG) hat diesen Bewerb bereits 1980 auf Initiative von Univ. Prof. Dr. Helmut Schauer ins Leben gerufen. Ziel des Wettbewerbs war ursprünglich wie auch heute noch das Interesse an Informatik zu wecken und zu intensiver Beschäftigung mit ihren Inhalten und Methoden sowie den Perspektiven ihrer Anwendung anzuregen. Der Einsendeschluss ist üblicherweise nach den Osterferien.

Weitere Informationen: www.ocg.at/ccw

Informatik Olympiade (IOI)

Diejenigen unter den Schülerinnen und Schülern, die sich beim Computer-Programmieren fit fühlen und gerne in einer Gruppe von Gleichgesinnten auch anspruchsvolle Programmieraufgaben lösen wollen, können sich bei der Informatikolympiade anmelden. Dabei wird allerdings verlangt, dass vorgegebene Programmieraufgaben gelöst und hochgeladen werden. Siehe www.ocg.at/ioi. Nach 2 Qualifikationsworkshops fahren die 4 Bestplatzierten der Österreichischen Informatik-Olympiade zur Internationalen Olympiade um sich mit den Besten der anderen Länder zu messen.

Alles über die OCG Wettbewerbe finden Sie unter www.ocg.at/de/wettbewerbe.

Aufgabenliste

Dies sind die 41 Aufgaben des Biber der Informatik 2016, gelistet nach ungefähr steigender Schwierigkeit und mit einer informatischen Klassifikation ihres Aufgabenthemas. Die farbigen Kästchen zeigen den Schwierigkeitsgrad (grün=leicht, orange=mittel, rot=schwer) für die verschiedenen Altersstufen:

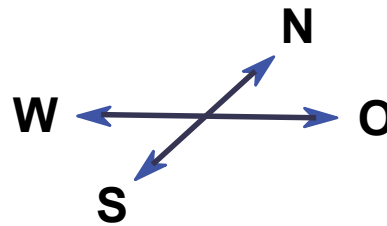
Titel	Thema	3-4	5-6	7-8	9-10	11-13	Seite
3D-Labyrinth	Programmieren, Anweisungsfolge, Programmstrukturen		orange	grün			7
Blumen und Sonnen	Codierung, Datenkompression, Präfixcode		rot	grün			12
Laut oder Leise	Mensch-Computer-Interaktion, Benutzung, grafische Symbolik	orange					46
Wer war's?	Datenbank, Attribut, Selektion	grün					63
Wer gewinnt?	Spam, Datenschutz, Internet-Risiken		grün				62
Puzzlesteine	Algorithmus, Optimierung, Greedy-Methode	orange	grün				52
Cassy, die Schildkröte	Programmieren, Korrektheit, Anweisungsblock, Schleife			rot		grün	19
Flaggen am Strand	Codierung, Bedeutung, Unicode	orange	grün				24
Regel-Regal	Regelmenge, Strukturen, Handlungen		grün				53
Zwei Möglichkeiten	Logik, Zweiwertigkeit, Funktion XOR				rot	orange	66
Bewässerung	Schaltung, Informationsfluss, physikalisches Medium	grün					9
Kugelspiel	Informatik-Theorie, Ersetzungssystem, Wortproblem				rot	orange	44
Partygäste	Planen, Bedingungen, gerichteter Graph				grün		51
Schuhbänder	Programmieren, Anwendungssprache, Befehlstypen					rot	56
Geheime Nachrichten	Kryptologie, Datenschutz		rot		grün		31
Karten und Kegel	Programmieren, Variable, Zuweisung				grün		36
Formenspiel	Informatik-Theorie, Formale Sprachen, Grammatik					rot	28
Eiscreme-Stapel	Datenstrukturen, Reihenfolge, Stapelspeicher	grün					23
Geburtstagskerzen	Binärsystem, Binäre Informationsdarstellung		orange	grün			30
Biberchat	Informationelle Selbstbestimmung, Benutzerprofile, Big Data			orange			10
Arten	Wissensrepräsentation, Objektorientierung, Vererbung		rot	orange	grün		8
Zugleich	Parallelität, Supercomputer, Computer-Cluster		orange	grün			64
Spielernummern	Algorithmus, Suchen, binäre Suche				orange	grün	58
Brücken-Bach	Algorithmus, Graphen, Brücken		rot	orange			16
Blumenfarben	Algorithmus, Logikspiel, Teststrategie	rot	orange	grün			13
Vegetarische Schnitzeljagd	Datenstrukturen, einfach verkettete Liste, Listenelement	rot	orange				61
Trambahn (трамвай)	Programmieren, parallele Prozesse, gemeinsame Ressourcen				rot		60
Bonbonnieren	Objektorientierung, Modellieren, Attribute			rot	orange		14
Rekursive Malerei	Programmieren, Rekursion, Abbruchbedingung					rot	54
IP-Adressen	Netzwerke, Internet-Protokoll, IP-Adressierung				rot	orange	34
Medianfilter	Algorithmus, Bildverarbeitung, Graustufen, Pixel			rot		orange	47
Egoistische Eichhörnchen	Algorithmus, Schwarmintelligenz					rot	21
Mittagspause	Planen, Constraints, Optimierung		rot	orange			48
Kix	Grafische Codierung, Kix-Code, QR-Code, Bar-Code			rot		grün	42
Durch den Tunnel	Planen, autonomes Handeln				orange		20
Flaggenbilder	Algorithmus, Bildverarbeitung, Datenkompression					rot	26
Kistenlift	Datenstruktur, Stapel, PUSH und POP				rot	orange	40
Brückenplan	Datenstrukturen, Höhe eines Baums, balancierter Baum			orange	grün		18
Hände schütteln	Algorithmus, Laufzeit, Parallelität			rot	orange	grün	32
Käsegänge	Programmieren, Ereignis, Mensch-Computer-Schnittstelle (GUI)	rot	grün				38
Nim	Strategie, Vorhersage, Spielregeln				orange	grün	50



3D-Labyrinth

Ein 3D-Labyrinth hat vier Ebenen mit jeweils vier Feldern. Eine Kugel liegt auf der obersten Ebene. Auf der untersten Ebene ist das Ziel: das gelbe Feld.

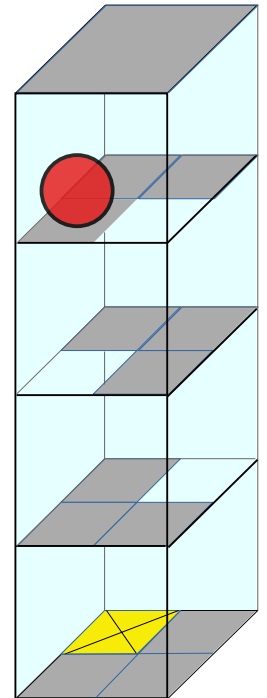
Du kannst die Kugel mit den Richtungsbefehlen N, O, S und W steuern. Auf einem weißen Feld fällt die Kugel eine Ebene nach unten. Das Labyrinth ist geschlossen; du kannst die Kugel also nicht nach außen steuern.



Steuere die Kugel ins Ziel!

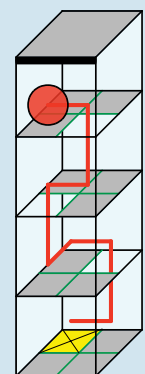
Tippe dazu unten eine Folge von Richtungsbefehlen ein. Tippe die Befehle direkt nacheinander, z.B.: NOS, um die Kugel nach N, O und dann nach S zu bewegen.

Klicke auf „Ausprobieren“, um zu sehen, wie deine Befehlsfolge die Kugel steuert.



So ist es richtig:

Mit Befehlsfolgen wie „OWNOW“ (siehe die rote Linie im Bild) oder „OWONW“ steuerst du die Kugel ins Ziel. Es gibt noch viele andere Möglichkeiten, die Kugel ins Ziel zu steuern, denn Umwege sind nicht verboten.



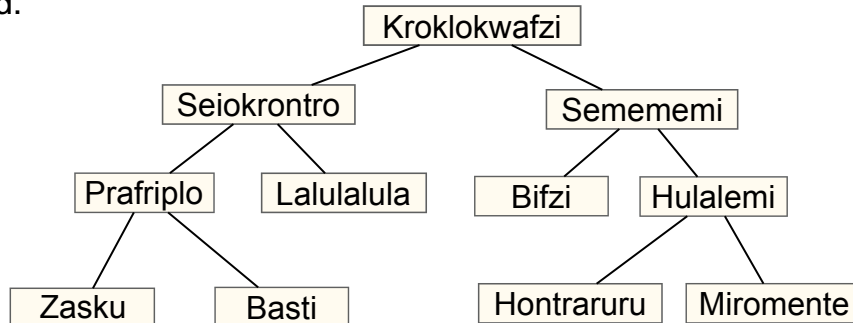
Das ist Informatik!

Die Befehlsfolge, mit der die Kugel durch das Labyrinth gesteuert wird, ist ein kurzes Computerprogramm. Die zugehörige Programmiersprache kennt nur vier Befehle, nämlich N, O, S und W; in der Informatik spricht man auch von Anweisungen. Ein Programm in dieser Sprache ist eine Folge von Anweisungen, die nacheinander (in der Informatik sagt man auch sequenziell) ausgeführt werden. Auch bei den meisten „richtigen“ Programmiersprachen ist die Abfolge bzw. Sequenz von Anweisungen die grundlegendste Möglichkeit, einzelne Anweisungen zu einem Programm zu strukturieren. Weitere grundlegende Strukturierungsmöglichkeiten sind die Wiederholung, die Bedingte Anweisung sowie das Wiederverwenden von häufigen Programmteilen durch Unterprogramme. Und das genügt auch schon; auf diese scheinbar einfachen Strukturen lassen sich selbst die komplexesten Computerprogramme zurückführen.



Arten

Das Bild beschreibt die Beziehungen zwischen Tierarten auf dem Planet Morgenstern. Eine Linie zwischen zwei Tierarten bedeutet, dass alle Tiere der unteren Art auch Tiere der oberen Art sind.



Beispielsweise sind alle Hulalemi auch Semememi. Manche Seiokrontro sind hingegen keine Basti.

Nur eine der folgenden Behauptungen ist richtig. Welche?

- A) Alle Basti sind auch Seiokrontro.
- B) Manche Hontraruru sind keine Semememi.
- C) Alle Zasku sind auch Bifzi.
- D) Alle Prafriplo sind auch Basti.

Antwort A ist richtig:

Alle Tiere der Art Basti sind auch Tiere der Art Prafriplo. Da alle Prafriplos Tiere der Art Seiokrontro sind, sind auch alle Bastis Tiere der Art Seiokrontro.

Antwort B ist nicht richtig: Hontrarurus sind Tiere der Art Hulalemi, welche wiederum Tiere der Art Semememi sind. Deshalb sind alle Hontrarurus Tiere der Art Semememi.

Antwort C ist nicht richtig: So wie manche Seiokrontro keine Basti sind, sind manche Kroklokwapzi keine Semememi – und zwar genau diejenigen Kroklokwapzi, die Seiokrontro sind. Da alle Bafzi auch Seiokrontro sind, können Basti keine Semememi sein, und damit auch keine Bifzi, denn die sind alle Semememi.

Antwort D ist nicht richtig: Alle Basti sind auch Prafriplo, aber auch alle Zasku sind Prafriplo. Es gibt also Prafriplo, die keine Basti sind.

Das ist Informatik!

Computerprogramme verarbeiten Informationen über Objekte der Welt: Menschen, Maschinen, Bankkonten, Waren im Online-Versandhaus oder im Supermarkt und vieles mehr. Für die verschiedenen Arten von Objekten (in der Informatik spricht man von Klassen statt Arten) beschreiben die Programm-Entwickler die mit allen Objekten der Klasse verbundenen Eigenschaften und Funktionen. Objekte einer Klasse können gleichzeitig auch Objekte einer Oberklasse sein: ein Joghurt ist auch ein Milchprodukt, genauso wie Butter. Die Eigenschaften der Oberklasse werden an die Unterklassen vererbt: Das Warenverwaltungsprogramm des Supermarkts weiß für Milchprodukte, dass sie gekühlt werden müssen, und damit weiß es das auch für Joghurts und Butter. Da auch andere Produkte gekühlt werden müssen (z.B. Wurstwaren), sollte diese Eigenschaft für die Klasse der Kühlprodukte beschrieben sein, die dann wieder Oberklasse der Milchprodukte ist, aber auch der Wurstwaren. Deswegen sind Wurstwaren noch lange keine Milchprodukte, und nicht alle Kühlprodukte sind Joghurts. Das ist ganz ähnlich wie bei den Tierarten auf dem Planeten Morgenstern. Dem Computer ist es übrigens egal, wie die im Programm beschriebenen Klassen bezeichnet werden. Im Programm können die Kühlwaren auch Seiokrontro heißen, die Milchprodukte Prafriplo, die Joghurts Basti und die Butter Zasku; alles funktioniert unverändert. Nur die Menschen, die so ein Morgenstern-Programm lesen sollen, haben ihre Schwierigkeiten – aber vielleicht auch ihren Spaß.



3-4: leicht

5-6: –

7-8: –

9-10: –

11-13: –



Bewässerung

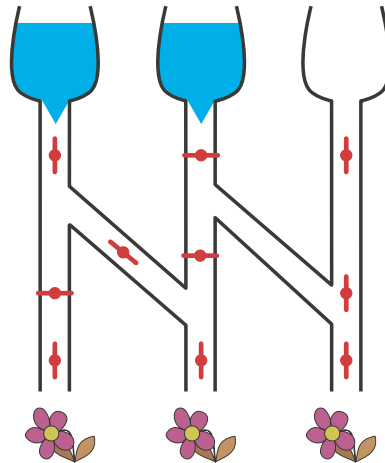
Wenn das Ventil zu ist,
fließt kein Wasser.



Wenn das Ventil offen ist,
fließt Wasser durch.



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



Klicke auf die Blumen!

Blumen, die Wasser bekommen, sollen so  aussehen.

Blumen, die kein Wasser bekommen, sollen so  aussehen.

So ist es richtig:

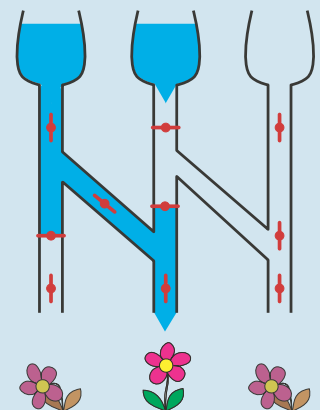
Nur die mittlere Blume bekommt Wasser bei dieser Stellung der Ventile.

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.



de.wikipedia.org/wiki/Elektronik

de.wikipedia.org/wiki/Photonik

de.wikipedia.org/wiki/Fluidik

de.wikipedia.org/wiki/Hydraulik



Biberchat

Der Biberchat kann kostenlos verwendet werden und wird dann durch Werbung finanziert. Der Reiseveranstalter „Sunshine Travel“ verwendet dabei Werbebilder für unterschiedliche Zielgruppen.

Sendet ein Biber eine Nachricht, wird darin nach bestimmten Wörtern gesucht. Die Wörter werden mit Hilfe dieser Regeln bewertet:

- Die Anrede „Liebe“ wird gerne von älteren Bibern verwendet und bringt zwei Punkte.
- Die Anreden „Hi“, „Hey“ und „Arriba“ sind unter jüngeren Bibern populär und bringen jeweils zwei Minuspunkte.
- Die Abkürzungen „bf“, „gr8“, „np“ und „thx“ bringen je einen Minuspunkt.
- Jedes Wort mit zehn oder mehr Buchstaben bringt einen Punkt.

Sendet ein Biber eine Nachricht, wird er anhand der Gesamtpunktzahl seiner Nachricht einer Zielgruppe zugeordnet. Dann wird ein passendes Bild angezeigt:

Gesamtpunktzahl	Zielgruppe	passendes Bild
> 0 (größer Null)	Senior	Strandbild
< 0 (kleiner Null)	Jugendlich	Surfbild
= 0 (gleich Null)	keine Zuordnung	Eiffelturm

Welche Bilder werden zu den folgenden Nachrichten angezeigt?

**A)**

Liebe Freunde, der Sommer kommt, und ich suche nach einer netten Unterkunft nahe am Rhein. Danke für eure Vorschläge, Richie.

B)

Arriba! Jemand hier?

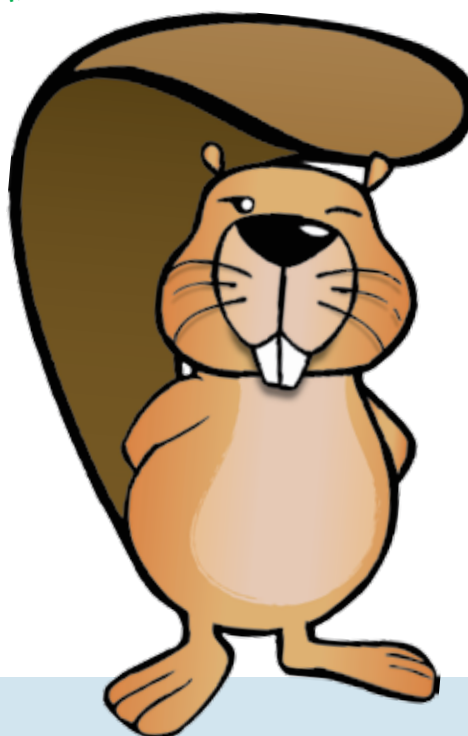
C)

@Mia: <3 <3 <3

D)

das passt. gr8. thx

Du liest jetzt schon die vierte Biberaufgabe
in diesem Heft. Zielgruppe: Informatik-interessiert.
Du solltest Programmieren lernen:
cscircles.cemc.uwaterloo.ca/de

**So ist es richtig:**

Nachricht A: Sunshine Travel zeigt das Strandbild an. Die Anrede „Liebe“ bringt zwei Punkte. Die zehn Buchstaben langen Wörter „Unterkunft“ und „Vorschläge“ bringen je einen Punkt. Der Biber wird mit vier Punkten der Zielgruppe „Senior“ zugeordnet.

Nachricht B: Sunshine Travel zeigt das Surfbild an. Die Anrede „Arriba“ bringt zwei Minuspunkte. Der Biber wird der Zielgruppe „Jugendlich“ zugeordnet.

Nachricht C: Sunshine Travel zeigt das Bild des Eiffelturms an. Die Nachricht bringt keine Punkte, weil keine der Regeln zutrifft. Der Biber kann keiner Zielgruppe zugeordnet werden.

Nachricht D: Sunshine Travel zeigt das Surfbild an. Die Abkürzungen „gr8“ und „thx“ bringen je einen Minuspunkt. Der Biber wird mit zwei Minuspunkten der Zielgruppe „Jugendlich“ zugeordnet.

Das ist Informatik!

Schon länger kennt die Informatik leistungsfähige Algorithmen um Texte zu analysieren. Man kann Stichwörter herausfiltern wie in dieser Biberaufgabe. Man kann wiederkehrende Wortgruppen entdecken, kann Schreibstile von Personen unterscheiden und vieles mehr. Inzwischen sind die Speicher- und Verarbeitungskapazitäten der dafür installierten Computer so groß geworden, dass alle Texte, die von Milliarden Benutzern laufend in die sozialen Netze eingetippt werden, noch in der gleichen Minute irgendwo, irgendwie analysiert werden – weltweit.

Das ist für die Provider (Betreiber) von sozialen Netzen eine sprudelnde Informationsquelle, um sehr detaillierte und aussagekräftige Profile über ihre Benutzer aufzustellen. Diese heranwachsenden Profile erlauben es den Providern unter anderem, über das Internet weitere Serviceleistungen anzubieten und dabei die Benutzer noch genauer kennenzulernen — und noch mehr Geld damit zu verdienen. Die Benutzer müssen sich dagegen zunehmend Sorgen machen. Was ist wirklich über sie gespeichert? Können sie mit ihrem vollständigen Profil zu anderen Providern wechseln? Enthält ihr Profil Unwahrheiten, die zu einem finanziellen oder beruflichen Schaden führen könnten? Können ihre Profile unbemerkt an andere Organisationen verkauft werden?

Gut wäre es, wenn die Provider jedem Benutzer auf Anforderung sein Profil zeigen müssten und staatliche Organisationen es sicher stellen, dass er dabei nicht belogen wird — und dass er löschen kann, was er will. Für die digitale Gesellschaft und eine sie mitgestaltende Informatik ist es noch ein schwerer Weg bis zur vollständigen Implementierung des bestehenden Rechts auf „Informationelle Selbstbestimmung“.



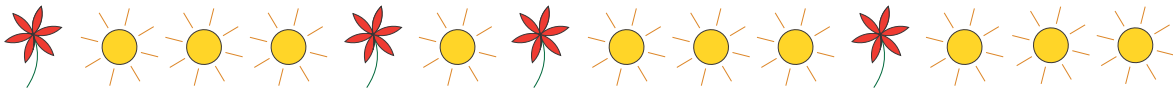
Blumen und Sonnen

Barbara hat zwei Stempel bekommen. Einer druckt eine Blume, der andere eine Sonne. Sie überlegt, wie sie nur mit Blumen und Sonnen ihren Namen stempeln kann.

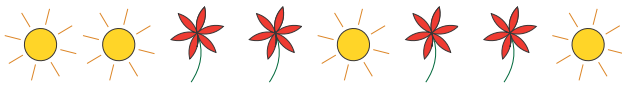
Für verschiedene Buchstaben bestimmt sie verschiedene Folgen von Blumen und Sonnen:

Buchstabe	B	A	R	E	Y
Folge		 	  	   	   

Ihren eigenen Namen „Barbara“ muss sie dann so stempeln:



Nun stempelt Barbara den Namen eines ihrer Freunde:



Welchen Namen hat sie gestempelt?

A) Abby B) Arya C) Barry D) Ray

Antwort A ist richtig:

Barbara stempelt die Namen ihrer Freunde so:

Abby        

Arya           

Barry             

Ray         

Das ist Informatik!

Wenn Daten wie etwa die Buchstaben des Alphabets durch andere Daten (wie hier die Folgen von Blumen und Sonnen) beschrieben werden, spricht man in der Informatik von einer Codierung bzw. einem Code. Daten, die mit Hilfe von Computern gespeichert und verarbeitet werden sollen, werden binär codiert, also durch Folgen von 0- und 1-Werten: Textzeichen wie Buchstaben und Ziffern, Bilder, Musikstücke, Videos ...

Ein solcher binärer Code ist einfach zu organisieren, wenn jedes Datenelement, also z.B. jedes Textzeichen, durch gleich viele Bits codiert werden kann. Lange Zeit war es üblich, ein Textzeichen durch 8 Bits zu kodieren – eine Folge von 8 Bits nennt man auch ein Byte. Neuere Codes wie UTF-8 setzen teilweise mehrere Bytes ein, um Zeichen aus Sprachen der ganzen Welt zu kodieren.

Hat man sehr viele Daten, ist es sinnvoll, sie platzsparend zu codieren. Textzeichen kommen in langen Texten unterschiedlich häufig vor; in deutschen Texten z.B. sind „E“ oder „N“ viel häufiger als „X“.

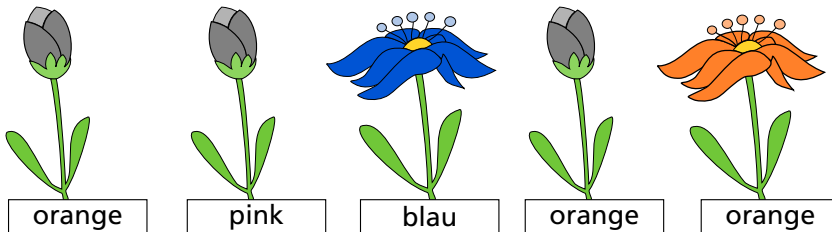
Wenn man die häufigen Zeichen mit weniger Bits codiert als die seltenen, spricht man von einem Code mit variabler Länge. Damit man bei der Decodierung die Zeichen erkennen kann, darf die Codierung eines Zeichens niemals mit dem Anfang der Codierung eines anderen Zeichens übereinstimmen (überlege einmal, warum). Ein Code, der diese Regel einhält, heißt Präfixcode; ein bekanntes Beispiel ist der Morsecode.

Blumenfarben

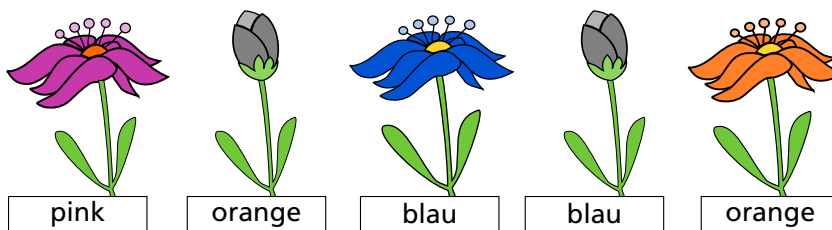
Jana spielt ein Computerspiel. Sie soll die Farben von fünf Blumen raten. Mögliche Farben sind blau, orange oder pink. Die Blumen sind geschlossen.

Jana rät zuerst die Farben so: orange, pink, blau, orange und orange.

Sie hat zweimal richtig geraten. Die richtigen Blumen gehen auf:



Für die falschen Blumen rät sie andere Farben. Nun ist noch eine Blume richtig:



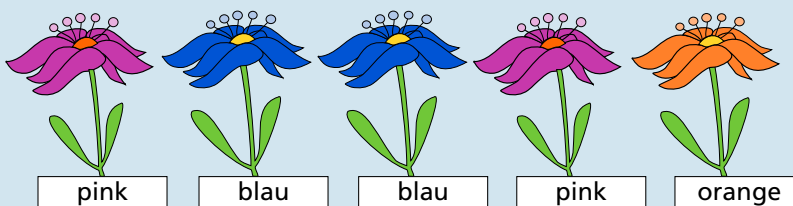
Rate die Farben der beiden anderen Blumen.

So ist es richtig:

Für die Blumen links, in der Mitte und rechts stehen die Farben bereits fest: pink, blau und orange.

Bei der zweiten Blume von links hat Jana pink und orange bereits gewählt, aber beide Male falsch geraten. Von den drei möglichen Farben ist nur noch Blau übrig. Die Blume ist also blau.

Bei der zweiten Blume von rechts hat Jana orange und blau bereits gewählt, aber beide Male falsch geraten. Von den drei möglichen Farben ist nur noch Pink übrig. Die Blume ist also pink.



Das ist Informatik!

Aus Ereignissen darf man logische Schlüsse ziehen. Aus einem Ereignis „Jana hat pink geraten, doch die Blume öffnet sich nicht“ darf man schließen, dass die Blume nicht pink ist. Logisch! Die Blumen können jede nur pink, orange oder blau sein. Nach Ereignissen, aus denen sich schließen lässt, dass eine Blume nicht pink und nicht orange ist, darf man schließen, dass diese Blume blau ist. Auch logisch!

Manche Menschen denken nicht gerne logisch. Es strengt an. Aber viele Menschen spielen zum Beispiel gerne Fußball, obwohl das auch anstrengt. Wenn man etwas gut kann, macht es Spaß. Wenn man es sehr gut kann, lässt sich mit logischem Denken sogar Geld verdienen, zum Beispiel als Schach-Profi. In der Informatik hat man Spaß am logischen Denken. Man programmiert es gern in Spiele hinein, wie bei dieser Biberaufgabe. Auch ist logisches Denken beim Arbeiten an Problemen der realen Welt nützlich – und beim Programmieren selbst. Faszinierend!



Bonbonnieren

Eine Bonbonniere ist ein Glasgefäß für Süßigkeiten. Carl und Judy haben je drei davon. Jede Bonbonniere kann mehrere der folgenden zehn Eigenschaften haben:

Die Bonbonniere ist offen (1) oder ist geschlossen (2).

Die Bonbonniere enthält rot-weiß gestreifte Bonbons (3) oder enthält keine (4).

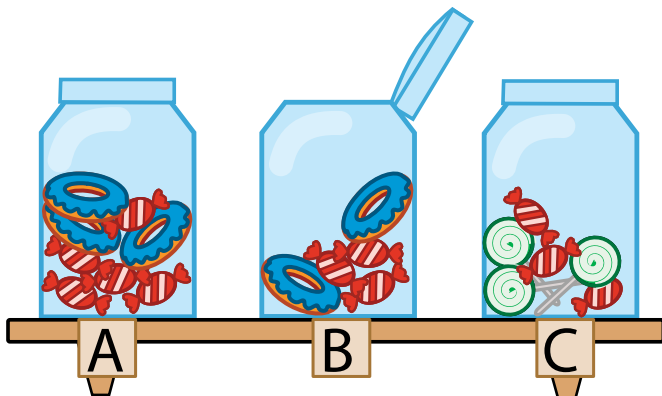
Die Bonbonniere enthält blaue Zuckerringe (5) oder enthält keine (6).

Die Bonbonniere enthält Lutscher mit grünen Spiralen (7) oder enthält keine (8).

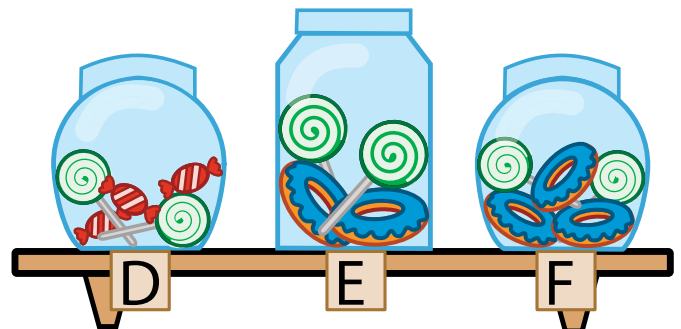
Die Bonbonniere ist rund (9) oder ist eckig (10).

Carls Bonbonniere A hat zum Beispiel die Eigenschaften 2, 3, 5, 8 und 10.

Schau genau hin! Carls Bonbonnieren haben einige Eigenschaften gemeinsam. Auch Judys Bonbonnieren haben einige Eigenschaften gemeinsam.



Carls Bonbonnieren

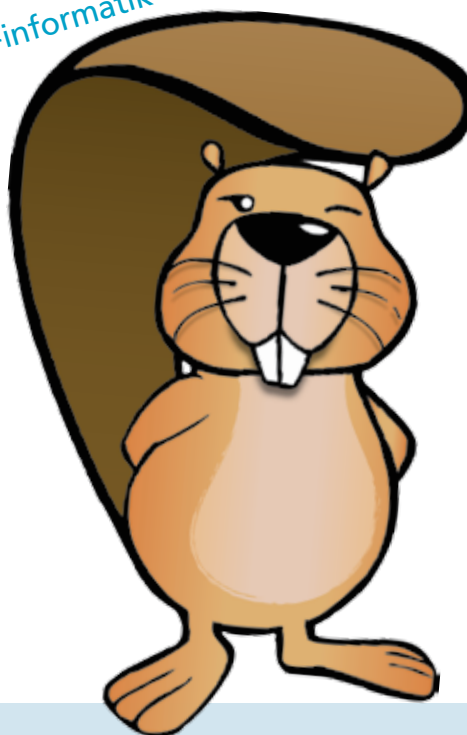


Judys Bonbonnieren

Eine der Bonbonnieren hat sogar sowohl die gemeinsamen Eigenschaften von Carls Bonbonnieren als auch die gemeinsamen Eigenschaften von Judys Bonbonnieren.

Klicke auf diese Bonbonniere!

Aufgabenhefte vergangener Biber-Wettbewerbe kannst
du dir hier herunterladen:
www.ocg.at/de/ueber-den-biber-der-informatik

**Antwort C ist richtig:**

Carls Bonbonnieren A, B und C haben zwei gemeinsame Eigenschaften:

- + Die Bonbonnieren sind eckig (10)
- + Die Bonbonnieren enthalten rot-weiß gestreifte Bonbons (3)

Judys Bonbonnieren D, E und F haben zwei andere gemeinsame Eigenschaften:

- + Die Bonbonnieren enthalten Lutscher mit grünen Spiralen (7)
- + Die Bonbonnieren sind geschlossen (2)

Nur die Bonbonniere C hat die Eigenschaften 2, 3, 7 und 10. Sie ist geschlossen, enthält rot-weiß gestreifte Bonbons, enthält Lutscher mit grünen Spiralen und ist eckig. Dass sie auch die Eigenschaft 6 hat, spielt keine Rolle.

Das ist Informatik!

Die Informatik fasst bei ihren Datenmodellen gern Objekte aufgrund von Eigenschaften in Gruppen zusammen. In dieser Biberaufgabe haben wir es mit zehn Eigenschaften und zwei Dreiergruppen zu tun. Wir suchen nach Objekten, welche die gemeinsamen Eigenschaften beider Gruppen haben. Bei relationalen Datenbanken nennt man das auch „den Durchschnitt zweier Mengen bilden“. Bei vielen Datenbanken kann man gesuchte Objekte anhand gewünschter Eigenschaften aus großen unübersichtlichen Objektmengen herausfiltern („eine Teilmenge bilden“).

Zum Beispiel kann man so in der Datenbank eines Online-Shops gezielt nach Smartphones mit einer bestimmten Displaygröße und anderen dort abgespeicherten Eigenschaften suchen. Beim Aufbau einer Datenbank ist sehr darauf zu achten, welche Eigenschaften im Datenmodell vorkommen. Werden wichtige Eigenschaften vergessen, ist alles spätere Suchen weniger treffsicher. Werden überflüssige Eigenschaften modelliert, macht das die spätere Datenpflege teurer, ohne dass sich der Nutzen der Datenbank erhöht.



Brücken-Bach

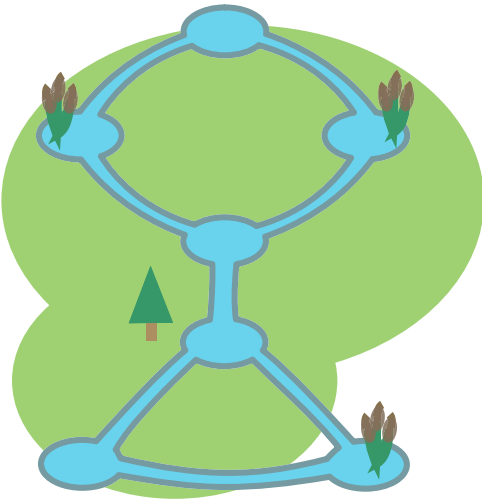
Im Biberrevier leben die Biber in Teichen. Jeder Teich hat mindestens einen Bach, durch den die Biber zum nächsten Teich schwimmen können. Die Biber schaffen aber auch Schwimmstrecken durch mehrere Bäche. So können sie zu jedem Teich schwimmen, egal wo sie starten.

Gerne blockieren die Biber einen Bach mit einem Damm. Aber sie müssen aufpassen: Wenn die Biber den falschen Bach blockieren, werden manche Teiche vom Rest des Biberreviers abgetrennt. Die Biber können dann nicht mehr zu jedem Teich schwimmen. Solch einen Bach nennen die Biber „Brücken-Bach“. Ein Brücken-Bach ist die einzige Verbindung zwischen zwei Teilen des Reviers.

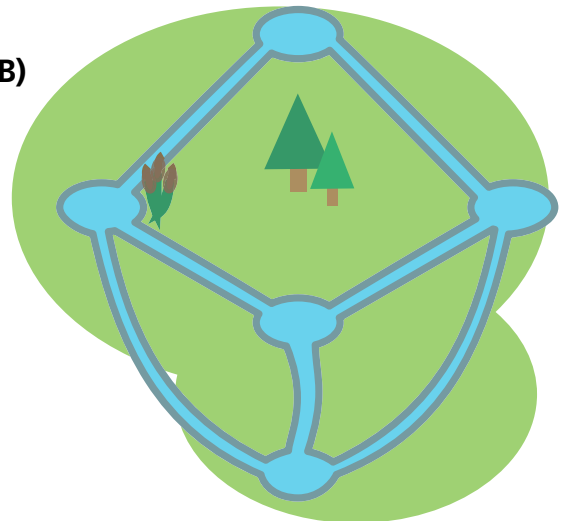
Auf den Bildern sind einige Biberreviere zu sehen.

Nur ein Biberrevier hat keinen Brücken-Bach. Welches?

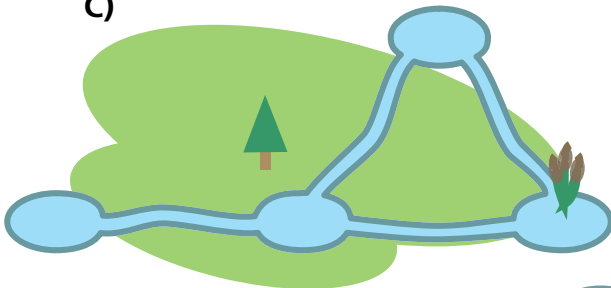
A)



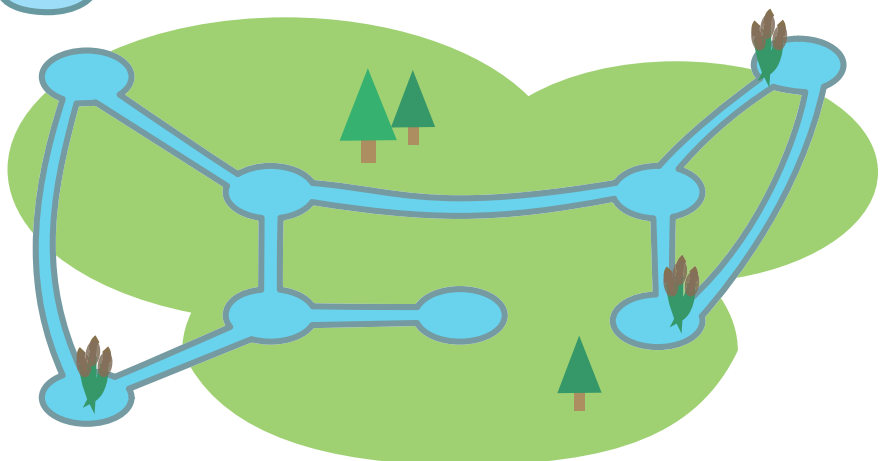
B)



C)



D)





3-4: –

5-6: schwer

7-8: mittel

9-10: –

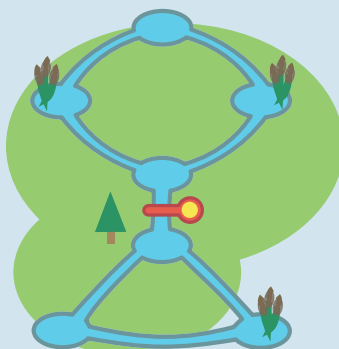
11-13: –

**Antwort B ist richtig:**

Egal, welcher einzelne Bach durch einen Damm blockiert wird, können die Biber in diesem Revier noch von jedem Teich zu jedem anderen schwimmen.

In Revier A ist der Fluss in der Mitte ein Brücken-Bach.

Ist er blockiert, können die Biber nicht mehr von unten nach oben schwimmen (und umgekehrt).

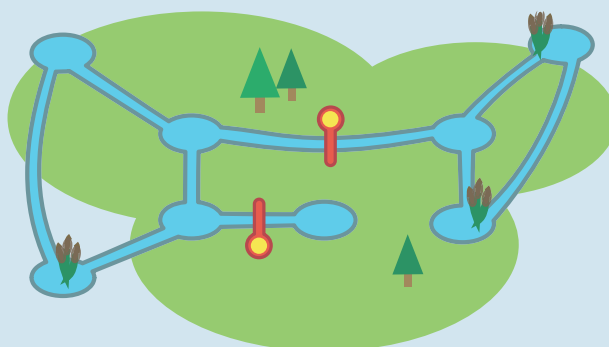


In Revier C ist der Bach links ein Brücken-Bach.

Ist er blockiert, können die Biber nicht mehr zum linken Teich schwimmen.



In Revier D sind die beiden Bäche in der Mitte Brücken-Bäche.

**Das ist Informatik!**

Das Biberrevier ist ein Netzwerk aus Bächen und Teichen. Das kann man mit dem Internet vergleichen. Dort sind Computer, Mobiltelefone, Fernseher etc. die Teiche. Leitungen oder Funkverbindungen sind die Bäche. Das Internet sollte ursprünglich Universitätsstandorte in den USA verbinden. Schon für die Gründer des Internets war es ein Thema, Brücken-Verbindungen zu vermeiden. Denn wenn eine Brücken-Verbindung ausfällt, sind Teile des Netzwerks abgetrennt.

Die Informatik bedient sich der Graphentheorie, um Netzwerke zu betrachten. Graphen werden als Systeme aus Knotenpunkten (Teiche) und Verbindungskanten (Bäche) definiert. Mit Graphen kann man viele Arten von Netzwerken modellieren, z. B. ein Verkehrsnetz oder ein Kommunikationsnetz. In vielen Netzwerken ist es wichtig, dass jeder Knoten von jedem anderen Knoten aus erreichbar ist. Das Graphenmodell eines Netzwerks mit dieser Eigenschaft heißt „zusammenhängend“. Kanten, die in einem (Netzwerk-)Graphen vorhanden sein müssen, damit der Graph zusammenhängend bleibt, heißen „Brücken“. Die Brücken-Bäche im Biber-Teich-Bach-Netz sind genau solche Graphen-Brücken. Die Informatik kennt viele Algorithmen, um Netzwerkprobleme zu analysieren. Eine ganze Reihe davon hat der amerikanische Informatiker Robert Tarjan erfunden. Von ihm stammt auch ein effizienter Algorithmus zur Bestimmung der Brücken in einem Graphen.

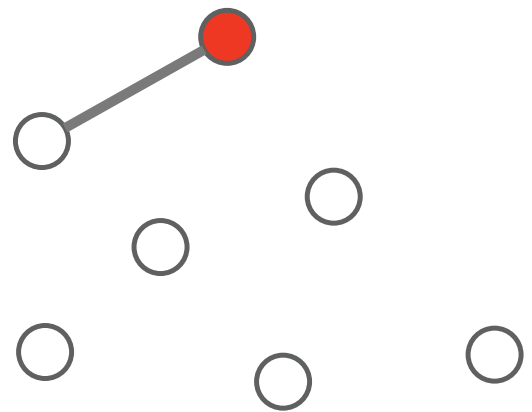


Brückenplan

Der Biber-Opä ist ein wenig wasserscheu geworden. Er möchte von seiner Burg aus zu allen anderen Burgen der Biber-Familie über Brücken gehen können. Die Biber meinen es gut mit Opä und wollen beim Brückenbauen diese beiden Bedingungen beachten:

- Opä soll von seiner Burg aus höchstens über zwei Brücken gehen müssen.
- Von jeder Burg aus sollen höchstens zwei Brücken weiterführen.

Die Biber beginnen mit einem Brückenplan. Sie zeichnen alle Burgen als Kreise. Opas Burg ist ein roter Kreis. Eine erste Brücke von Opas Burg aus zeichnen sie ein. Aber dann wissen sie nicht mehr weiter.

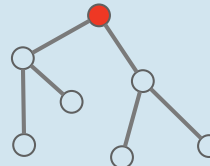


Vervollständige den Plan so, dass er beide Bedingungen erfüllt.

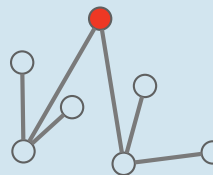
Dazu gibt es mehrere richtige Möglichkeiten. Auf jeden Fall werden fünf weitere Brücken benötigt.

So ist es richtig:

Dieser Plan erfüllt beide Bedingungen, nämlich:
Vom roten Kreis aus ist jeder andere Kreis über höchstens zwei Linien zu erreichen.
Von jedem Kreis führen höchstens zwei Linien weiter.



Aber auch andere Pläne erfüllen beide Bedingungen, zum Beispiel dieser: Vielleicht brauchen die Biber noch weitere Bedingungen, um den allerbesten Plan zu erstellen.



Das ist Informatik!

Die Burgen und Brücken bilden zusammen eine Struktur, in der von einem Knoten (nämlich Opas Burg) alle anderen Knoten (die anderen Burgen) auf genau einem Weg über Kanten (die Brücken) und andere Knoten erreicht werden können. Solch eine Struktur nennt die Informatik einen Baum. Der besondere Knoten (also Opas Burg) wird als Wurzel bzw. Wurzelknoten bezeichnet. Die Länge des weitesten Weges von der Wurzel eines Baums zu irgendeinem seiner Knoten, gemessen in der Anzahl der auf dem Weg passierteten Kanten, nennt man die Höhe des Baums.

Bäume können für viele Anwendungen nützlich sein; sieh nach bei der Aufgabe „Mittagessen“ aus dem Biber der Informatik 2015. Besonders verbreitet ist ihre Anwendung als Suchbaum, in dem Werte einer möglicherweise sehr großen Menge (z.B. alle Wörter eines Wörterbuchs) an Knoten gespeichert werden. Um einen Wert zu suchen, geht man den Weg von der Wurzel zu seinem Knoten entlang. Die Suchen in einem Suchbaum sind also im Durchschnitt sehr viel schneller, wenn die Höhe möglichst klein ist. Ein Baum, dessen Höhe begrenzt ist, nennt man auch balanciert. Der Biber-Brückenplan ist ein balancierter Baum, da seine Höhe durch die erste Bedingung ideal begrenzt wird.



3-4: –

5-6: –

7-8: schwer

9-10: –

11-13: leicht



Cassy, die Schildkröte

Die Schildkröte Cassy lebt in Gitterland, auf einem Acker von fünf mal fünf Gitterzellen. Sie isst für ihr Leben gern frische Salatpflanzen. Jeden Morgen wachsen neue Salatpflanzen. Cassy weiß nicht, in welchen Gitterzellen sie sind, aber sie will alle essen! Cassy startet jeden Tag in der Mitte des Ackers und will den ganzen Acker ablaufen.

Sage Cassy, wie sie sich bewegen soll!

Neben dem Gitterland ist ein Anweisungsblock mit vier Feldern. Fülle jedes Feld mit einer Bewegungs-Anweisung für Cassy, so dass sie insgesamt den ganzen Acker ablauft:

- Wähle durch Klicken die passenden Anweisungen aus.
- Eine Anweisung kann mehr als einmal verwendet werden.
- R ist ein Wiederholungszähler.
- Wenn der Anweisungsblock das erste Mal ausgeführt wird, enthält R den Wert $R = 1$; bei der zweiten Ausführung wird $R = 2$; usw.
- Cassy darf den Acker verlassen, aber nicht das Gitterland.
- Klicke auf den Knopf „Testen“, um Cassy zu starten.

Mache es fünf Mal

Testen

nach links drehen

nach rechts drehen

R Zellen vorwärts

So ist es richtig:

Es gibt vier verschiedene Möglichkeiten, die Felder des Anweisungsblocks so zu füllen, dass Cassy den gesamten Acker ablauft. Bei allen vier Möglichkeiten hat Cassys Weg die Form einer Spirale:

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

Das ist Informatik!

Bei dieser Biberaufgabe solltest du ein Programm für Cassy entwerfen. Cassy führt die Anweisungen der Reihe nach aus (und zwar fünf Mal). Falls das Programm korrekt ist, macht Cassy genau das, was du willst. Aber auch wenn das Programm nicht korrekt ist, führt die Schildkröte die Anweisungen aus, nur bewegt sie sich dann nicht so wie beabsichtigt. Genau so ist das auch im Allgemeinen mit Computern und Computer-Programmen. Ein Computer ist nicht in der Lage zu erkennen, ob ein Programm korrekt ist. In dieser Aufgabe wird ein Anweisungsblock erstellt, der aus vier Anweisungen besteht. Alle Anweisungen in diesem Block werden genau fünfmal ausgeführt. Eine wiederholte Ausführung von Anweisungen wird in der Informatik als Schleife bezeichnet. Beachte, dass die Schleife in dieser Aufgabe einen Schleifenzähler besitzt, der sich mit jeder Ausführung des Blocks verändert. In allen nützlichen Programmiersprachen können Schleifen programmiert werden – und weitere Möglichkeiten zur Steuerung des Programmverlaufs, wie bedingte Verzweigungen oder Aufrufe von Unterprogrammen.



Durch den Tunnel

Anna und Benno machen mit ihren Eltern eine Wanderung. Auf ihrer Strecke liegt ein Tunnel. Aus Erfahrung wissen sie, dass jeder von ihnen unterschiedlich viel Zeit für die Tunnelpassage benötigt: Anna benötigt 10 Minuten, Benno 5 Minuten, die Mutter 20 Minuten und der Vater 25 Minuten. Den dunklen und engen Tunnel kann man nur alleine oder zu zweit passieren. Sie müssen also mehrere Passagen machen. Zu zweit benötigt man so viel Zeit wie die langsamere der beiden Personen. Im Tunnel muss man auf jeden Fall eine Lampe benutzen.

Als sie an den Eingang des Tunnels kommen, stellen sie fest: Der Akku ihrer einzigen Lampe reicht nur noch für 60 Minuten. Können sie innerhalb dieser 60 Minuten alle durch den Tunnel kommen? Anna hat einen Plan: „Ja, können wir, und zwar mit fünf Passagen!“

Wie könnte Annas Plan aussehen?

Ziehe die Namen in die Felder der Tabelle, um Annas Plan zu beschreiben.

Passage	Person 1	Person 2
1: Hin →		
2: Zurück ←		
3: Hin →		
4: Zurück ←		
5: Hin →		

Anna

Benno

Mutter

Vater

So ist es richtig:

Dass alle innerhalb von 60 Minuten am Ausgang sind, ist nur möglich, wenn Mutter und Vater (die beiden langsamsten) nur einmal den Tunnel passieren. Das wiederum ist nur möglich, wenn vorher Anna und Benno den Tunnel passieren und einer von ihnen zurück geht. Der eine kann dann die Lampe Vater und Mutter geben, und der andere bringt sie durch den Tunnel zurück, damit dann beide rechtzeitig zusammen zum Ausgang gehen können. Dabei ist es egal, ob in Passage 2 Anna zurückgeht oder Benno: In Passage 4 ist der jeweils andere dran, so dass diese beiden Passagen in beiden Fällen zusammen 15 Minuten dauern.

Das ist Informatik!

Anna hat die Lösung des Tunnelproblems mit einem Plan beschrieben. Ihr Plan erfüllt die Bedingung, dass bei jeder Passage die Lampe dabei ist. Die Ausführung des Plans kommt mit den verfügbaren Mitteln aus: Der 60-Minuten Akku-Reserve der Lampe. Die Informatik entwickelt immer komplexere Systeme, die in der Lage sind, eigenes Handeln selbständig bzw. autonom zu planen, dabei die vorgegebenen Bedingungen zu erfüllen und mit den verfügbaren Mitteln auszukommen; so wie Anna. Die Konsequenzen autonomen Planens sind weitreichender als in unserer gut überschaubaren Biberaufgabe. Zum Beispiel bei selbstfahrenden Autos: Die müssen einen sinnvollen Weg zum Ziel planen, dabei alle Verkehrsregeln einhalten und berücksichtigen, dass sie während der Ausführung immer genug Energie haben. Souveräne und sichere Informatik-Planung wird auch ethisch und juristisch vertretbar sein müssen, wenn das Handeln autonomer Systeme Auswirkungen für Menschen, andere Lebewesen und die Umwelt hat. Ein selbstfahrendes Auto muss stets die zentrale Bedingung erfüllen, dass durch sein Handeln kein Mensch zu Schaden kommt – wenn irgend möglich. Solche Überlegungen hat Isaac Asimov bereits im Jahr 1942 angestellt, als gerade die ersten Computer gebaut wurden. Er hat sich Gesetze für Roboter überlegt. Sein erstes Robotergesetz lautet: „Ein Roboter darf keinen Menschen verletzen oder durch Untätigkeit zu Schaden kommen lassen.“ Und ein selbstfahrendes Auto ist ein Roboter.



3-4: –

5-6: –

7-8: –

9-10: –

11-13: schwer



Egoistische Eichhörnchen

In einem Baum leben 16 egoistische Eichhörnchen. Nachts bewohnt jedes Eichhörnchen eine von fünf Baumhöhlen.

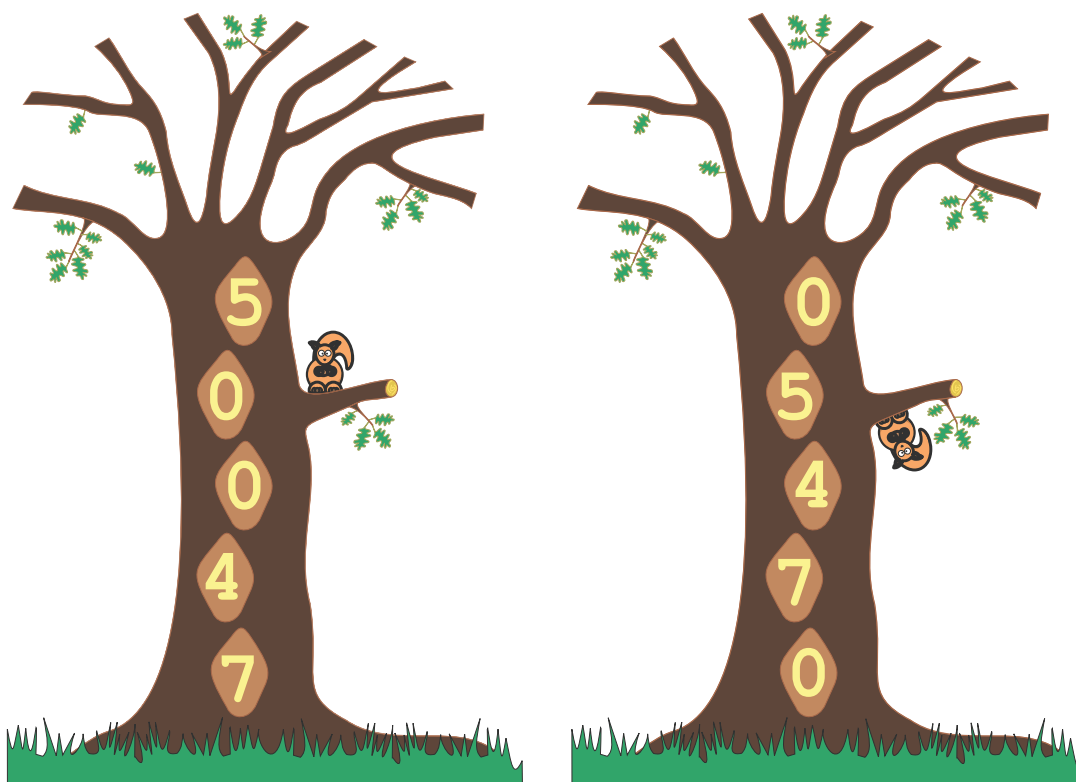
Die Eichhörnchen, die zusammen eine Höhle bewohnen, müssen sich das dort vorhandene Futter teilen. Jedes einzelne Eichhörnchen möchte deshalb für die nächste Nacht eine „bessere“ Höhle wählen. Es prüft dazu drei Zahlen, nämlich: für die aktuelle Höhle die Zahl seiner Mitbewohner und für die Höhlen unterhalb und oberhalb jeweils die Zahl der Bewohner. Für die nächste Nacht wählt es dann die Höhle mit dem niedrigsten Wert. Sind zwei Werte gleich, wählt das Eichhörnchen die aktuelle Höhle lieber als die Höhle oberhalb und die Höhle oberhalb lieber als die Höhle unterhalb.

Ein Beispiel:

In einer Nacht bewohnen 5, 0, 0, 4 bzw. 7 Eichhörnchen die Höhlen (von oben nach unten). Für die nächste Nacht wählen alle 5 Eichhörnchen aus der obersten Höhle die Höhle unterhalb (denn 0 Mitbewohner sind besser als 4). Die 4 Eichhörnchen der zweiten Höhle von unten wählen die Höhle oberhalb (0 Nachbarn sind besser als 3). Auch die 7 Eichhörnchen der untersten Höhle wählen die Höhle oberhalb (4 Nachbarn sind besser als 6).

In einer Nacht bewohnen 6, 3, 3, 0 und 4 Eichhörnchen die Höhlen (von oben nach unten). Nach wie vielen Tagen werden alle Eichhörnchen in derselben Höhle sein?

- A) Nach zwei Tagen.
- B) Nach drei Tagen.
- C) Nach vier Tagen.
- D) Niemals werden alle Eichhörnchen in der derselben Höhle sein.





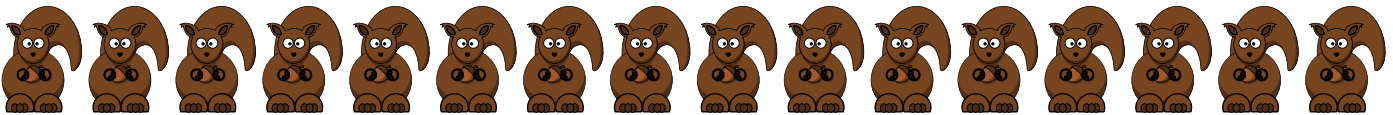
3-4: –

5-6: –

7-8: –

9-10: –

11-13: schwer

**Antwort B ist richtig:**

Die Zahlen der egoistischen Eichhörnchen in den Höhlen entwickeln sich so:

Aktuelle Nacht:	6, 3, 3, 0, 4
Nach einem Tag:	0, 9, 0, 7, 0
Nach zwei Tagen:	9, 0, 7, 0, 0
Nach drei Tagen:	0, 16, 0, 0, 0

Das ist Informatik!

Kommen viele Lebewesen zu einem Schwarm zusammen, kann der Schwarm komplexe Dinge schaffen, auch wenn jedes einzelne Lebewesen sich nach einfachen Regeln und unabhängig von den anderen verhält. Ein gutes Beispiel für eine solche Schwarmintelligenz oder auch kollektive Intelligenz bieten Ameisen: Jede einzelne Ameise hat sehr einfache Verhaltensmuster, die unabhängig von anderen Ameisen sind. In einem Ameisenvolk können viele Ameisen gemeinsam dennoch komplexe Aufgaben erledigen wie Blätter zu zerschneiden, einen Ameisenhaufen zu bauen oder in einem Wegenetz den optimalen Weg zu finden.

Auch einige Gebiete der Informatik experimentieren mit Schwarmintelligenz. Die Hoffnung ist, dass viele einfache Agenten (z.B. einfache Roboter, aber auch Softwarekomponenten) durch gemeinsames Verhalten und Kommunikation untereinander komplexe Probleme lösen können, wie etwa die optimale Verteilung der Arbeitslast in einem Server-Cluster oder das Problem des Handlungsreisenden.

Diese Biberaufgabe zeigt, dass kollektives Verhalten nicht immer intelligent sein muss. Die Eichhörnchen verhalten sich zwar gleichermaßen nach einfachen Regeln, landen jedoch am Schluss in derselben Höhle. Der für einen einzelnen Agenten gewählte Algorithmus muss also sorgfältig bedacht werden, wenn Schwarmintelligenz entstehen soll. Und Kommunikation und Kooperation sind meist besser als Egoismus. Diese Erkenntnis ist für uns Menschen nicht neu.

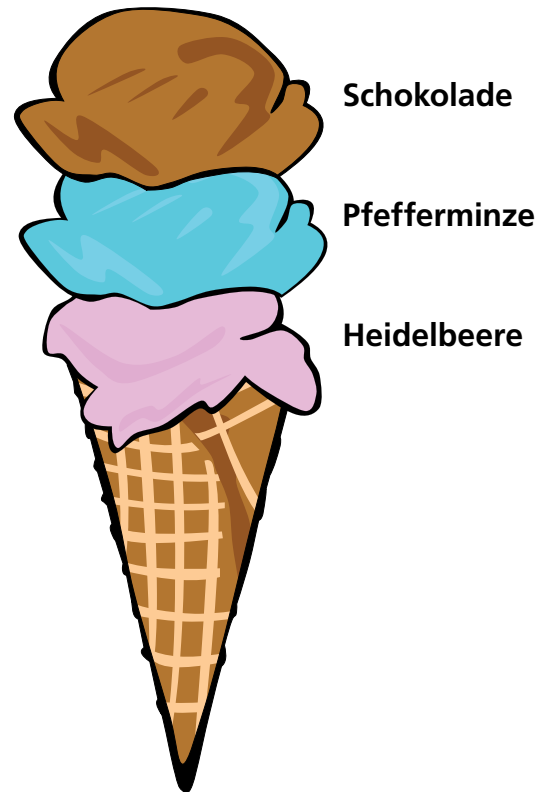
Eiscreme-Stapel

Benni kauft sich beim Eissalon LIFO ein Eis. Die gewünschten Eiscreme-Kugeln werden auf eine Eistüte gestapelt – und zwar genau in der Reihenfolge, die man sagt.

Was muss Benni sagen, wenn er eine Eistüte haben will, die genauso aussieht wie diese hier?

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 auch in der Informatik kein dazu passendes Programm entwickeln.

Die in dieser Biberaufgabe benutzte Ordnung heißt so wie der Eissalon: „last in, first out“ (LIFO).

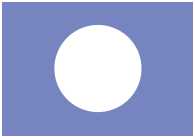
Frei übersetzt heißt das: „Das Letzte, das rein kommt, kommt als Erstes wieder raus“.



Flaggen am Strand

Albert ist Rettungsschwimmer und am Strand auf Posten. Seine Kinder Beatrix und Richard spielen weiter weg. Das Meer rauscht, der Wind übertönt alles Rufen und die Mobiltelefone finden kein Netz. Um doch mit den Kindern kommunizieren zu können, verwendet Albert eine Stange mit drei Positionen für farbige Flaggen.

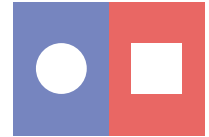
Die obere Flagge kann folgendes bedeuten:



Die Nachricht ist für Richard.



Die Nachricht ist für Beatrix.



Die Nachricht ist für Richard und Beatrix.

Die mittlere Flagge kann folgendes bedeuten:



Es gibt etwas zu essen.



Es gibt etwas zu trinken.

Die untere Flagge kann folgendes bedeuten:



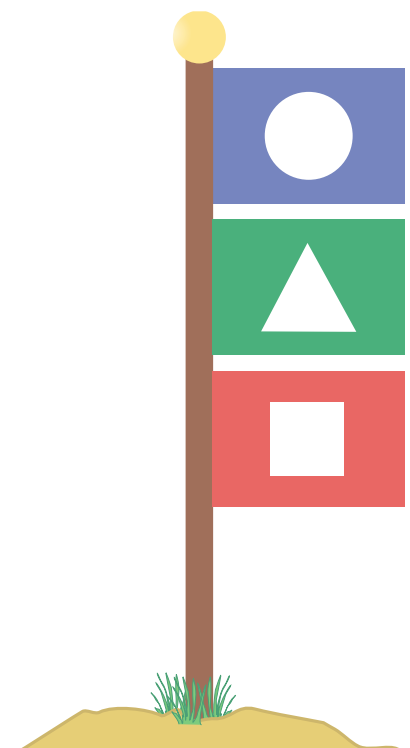
Beeile dich!



Du kannst dir Zeit lassen!

Was bedeutet es, wenn diese Flaggen so an der Stange hängen?

- A) Beatrix, es gibt etwas zu essen, beeile dich!
- B) Richard, es gibt etwas zu essen, beeile dich!
- C) Richard und Beatrix, es gibt etwas zu trinken, ihr könnt euch Zeit lassen!
- D) Beatrix, es gibt etwas zu trinken, du kannst dir Zeit lassen!





3-4: mittel

5-6: leicht

7-8: –

9-10: –

11-13: –

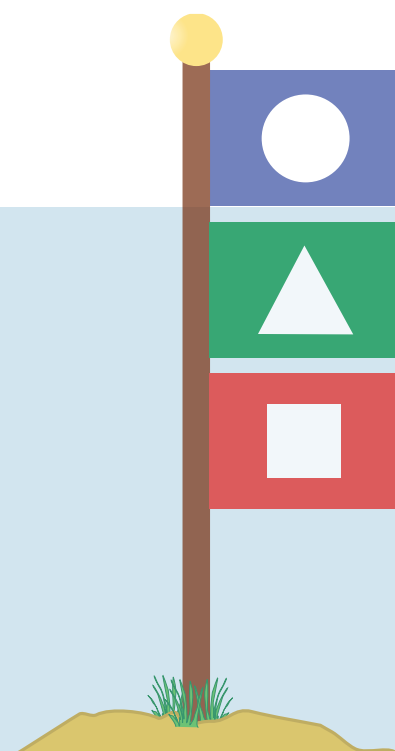
**Antwort B ist richtig:**

Die blaue Flagge an der oberen Position bedeutet, dass die Nachricht für Richard ist. Die grüne Flagge an der mittleren Position bedeutet, dass es für ihn etwas zu essen gibt. Die rote Flagge an der unteren Position bedeutet, dass er sich beeilen soll.

Das ist Informatik!

In dieser Biberaufgabe werden mehrteilige Nachrichten mit Flaggen codiert. Dabei sind die Farben der Flaggen, die Zeichen auf den Flaggen und die Positionen der Flaggen an der Stange von Bedeutung. Die Nachrichten werden hier über einen optischen Kanal gesendet, das ist die freie Sicht am Strand. Die Nachrichten werden nur in einer Richtung gesendet – die Kinder haben keine Flaggen. Sie müssen zum Vater hinlaufen und, wenn sie nahe genug sind, einen akustischen Kanal benutzen, also rufen.

In der Informatik wird über viele Kanäle Text übermittelt. Dafür kennt die Welt einen gemeinsamen Code, den Unicode. Der ist weltweit genormt, damit alle Sender und Empfänger die Bedeutung von Nachrichten verstehen können. Unicode umfasst nicht nur die Schriftzeichen fast aller menschlichen Sprachen, sondern auch ältere Binärcodes wie ASCII, und Symbolmengen aus vielen Kulturen. Zum Beispiel ist das Bildsymbol mit dem Unicode „U+1F60A“ ein Emoji und bedeutet „Lächelndes Gesicht mit lächelnden Augen“. Natürlich gibt es auch geheime Codes für private Mitteilungen. Man stelle sich vor, alle am Strand wüssten um die Bedeutung der grünen Essensflagge und kämen herbei gerannt, wenn Albert sie eigentlich nur für seine Kinder hissen wollte. Die Bedeutung seiner Flaggen sollte wohl besser ein Familiengeheimnis bleiben!





Flaggenbilder

Computerbilder bestehen aus Zeilen von Bildpunkten, genannt Pixel. Wenn Computerbilder als Dateien gespeichert werden, wird im einfachsten Fall die Farbe jedes Pixels einzeln beschrieben.

Mit dem (erfundenen) Dateiformat GIW werden Computerbilder komprimiert, also mit geringerer Dateigröße gespeichert. Das funktioniert so:

Jede Pixelzeile wird einzeln beschrieben.

Jede Farbe wird durch ein Kürzel aus drei Buchstaben beschrieben.

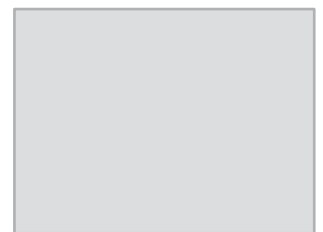
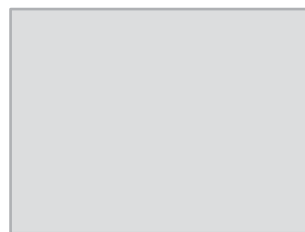
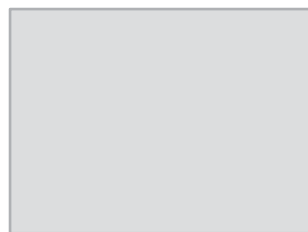
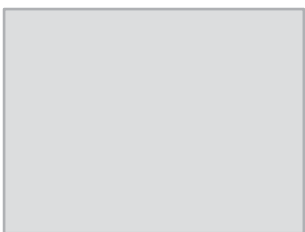
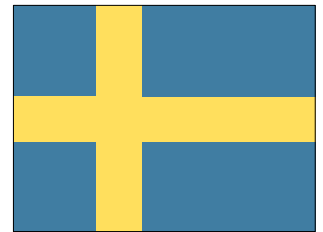
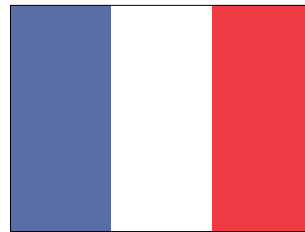
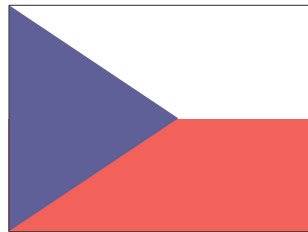
Eine Folge gleichfarbiger Pixel wird durch ein Klammerpaar beschrieben, das ein Farbkürzel und die Anzahl der gleichfarbigen Pixel enthält.

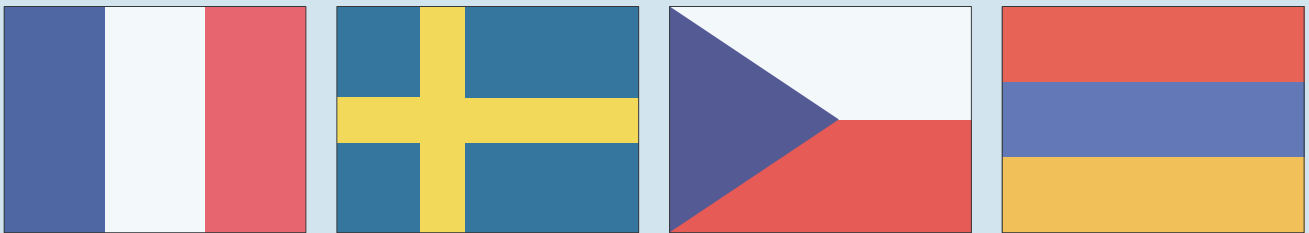
Ein Beispiel: Eine Pixelzeile, die durch die beiden Klammerpaare (grü,20) (wei,13) beschrieben wird, enthält zuerst 20 grüne und danach 13 weiße Pixel.

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

Ordne die Bilder nach der Größe ihrer GIW-Datei!

Ziehe die Bilder so in die freien Felder, dass links das Bild mit der größten GIW-Datei liegt.



**So ist es richtig:**

Eine Zeile, deren Pixel alle die gleiche Farbe haben, kann mit einem einzigen Klammerpaar beschrieben werden. Bei der Flagge von Armenien ist das in allen Zeilen so. Die GIW-Datei enthält also ein Klammerpaar pro Zeile.

In Bildern, bei denen die Farben in einer Pixelzeile wechseln, wird nach jedem Farbwechsel ein neues Klammerpaar benötigt. Bei der Flagge von Tschechien gibt es in jeder Zeile genau einen Farbwechsel, von Blau nach Weiß bzw. Blau nach Rot. Die GIW-Datei enthält also zwei Klammerpaare pro Zeile.

Bei der schwedischen Flagge gibt es in den meisten Zeilen zwei Farbwechsel. Die GIW-Datei enthält also für diese Zeilen je drei Klammerpaare. In der Mitte sind einige Zeilen, die keinen Farbwechsel enthalten; für jede dieser Zeilen enthält die Datei nur ein Klammerpaar. Es gibt aber viel mehr Zeilen mit zwei Farbwechseln als Zeilen ohne Farbwechsel. Deshalb enthält die GIW-Datei im Mittel mehr als zwei, aber weniger als drei Klammerpaare pro Zeile.

Bei der französischen Flagge gibt es in jeder Zeile zwei Farbwechsel. Die GIW-Datei enthält also drei Klammerpaare pro Zeile.

Das ist Informatik!

Datenkompression ist ein wichtiger Bereich der Informatik. Je geringer die Datenmenge ist, mit der digitale Daten (Bilder, Audiodaten, Videos, ...) beschrieben werden können, desto geringer ist der zur Speicherung der Daten benötigte Platz und desto schneller lassen sich die Daten durch ein Netzwerk transportieren. Kompressionsverfahren können den Transferaufwand innerhalb eines Netzwerks also deutlich verringern. Würde beispielsweise die Musik eines Webradios ohne Kompression versendet, so würde die zehnfache Datenmenge anfallen als bei heutzutage typischer Kompression.

Da die Menge der Daten, die durch Netzwerke transportiert werden sollen, immer weiter steigt (zum Beispiel Sprachnachrichten in Messenger-Anwendungen), forscht die Informatik weiter im Bereich der Datenkompression. Das in dieser Aufgabe vorgestellte Kompressionsverfahren arbeitet nach dem Prinzip der Lauflängenkodierung. Weitere Informationen kann das folgende Video liefern:

www.youtube.com/watch?v=ypdNscvym_E



Formenspiel

Alice spielt mit geometrischen Formen. In einer Reihe von Formen ersetzt sie nach und nach einzelne Formen durch eine Reihe anderer Formen.

Für jede Spielrunde überlegt sich Alice einige Ersetzungsregeln. Bei jeder Ersetzung wendet sie die Regeln so oft wie möglich an. Sie beginnt jede Spielrunde mit einer einzigen Form.

In einer ersten Spielrunde beginnt Alice mit einem Quadrat und hat sich diese Ersetzungsregeln überlegt:



Mit drei Ersetzungen produziert sie die Formenreihe rechts:

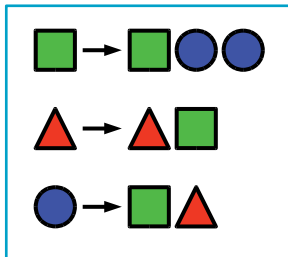


In einer anderen Spielrunde produziert Alice diese Formenreihe:

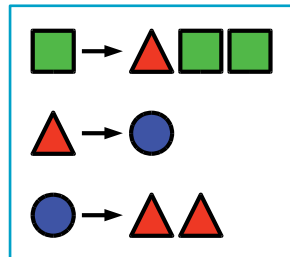


Welche Ersetzungsregeln hat sie sich für diese Spielrunde überlegt?

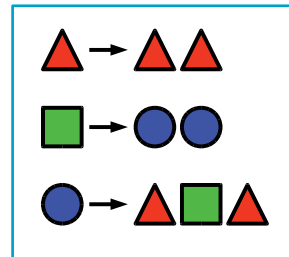
A)



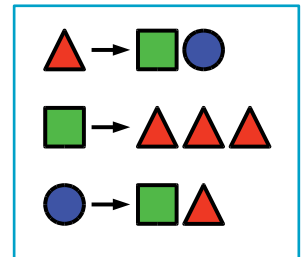
B)



C)



D)



**Antwort A ist richtig:**

Alice fängt mit einem Dreieck an und produziert die Formenreihe mit drei Ersetzungen:



Die anderen Antwortmöglichkeiten kann man durch die folgenden Überlegungen ausschließen:

Antwort B: Wenn Alice mit einem Dreieck oder einem Kreis beginnt, kann sie keine Quadrate produzieren. Wenn sie mit einem Quadrat beginnt, muss sie so ersetzen:

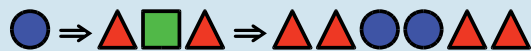


Da die Formenreihe mit jeder Ersetzung länger wird, kann sie die gewünschte Formenreihe mit diesen Regeln nicht produzieren.

Antwort C: Wenn Alice mit einem Dreieck beginnt, kann sie weder Quadrate noch Kreise produzieren. Wenn sie mit einem Quadrat beginnt, ersetzt sie so:



Nun hat sie keine Möglichkeit mehr, die beiden Dreiecke am Anfang durch ein Dreieck und ein Quadrat zu ersetzen, die am Anfang der gewünschten Formenreihe stehen. Auch wenn sie mit einem Kreis beginnt, stehen nach zwei Ersetzungen zwei Dreiecke am Anfang der produzierten Formenreihe:



Antwort D: Alice hat keine Möglichkeit, zwei Kreise nebeneinander zu produzieren. Es ist noch nicht einmal möglich, mehr als einen Kreis nebeneinander zu produzieren, da jeder Kreis bei der nächsten Ersetzung wieder in ein Quadrat und ein Rechteck umgewandelt würde. Die gewünschte Formenreihe enthält aber zwei Kreise nebeneinander.

Das ist Informatik!

Die Symbole und Regeln des Formenspiels bilden zusammen ein System zur Veränderung von Symbolketten. In der theoretischen Informatik werden solche Ersetzungssysteme unter bestimmten Bedingungen auch als Grammatiken bezeichnet. Die Symbolfolgen, die, wie hier im Formenspiel, durch Anwendung der Regeln erzeugt werden können, sind dann die Wörter einer Sprache. In der theoretischen Informatik ist eine Sprache also eine Menge von Wörtern, und eine Grammatik erzeugt eine solche formale Sprache. Für jede Grammatik lässt sich das Wortproblem als Frage formulieren: Gehört ein gegebenes Wort zu der von der Grammatik erzeugten Sprache? Wie schwer das Wortproblem zu lösen ist, hängt von der Form der Regeln der Grammatik ab. Aus der Form der Regeln ergeben sich Klassen von Grammatiken mit unterschiedlichen Eigenschaften. Zum Beispiel:

Reguläre Grammatiken erzeugen Sprachen, die auch durch reguläre Ausdrücke beschrieben werden können und von endlichen Automaten erkannt werden; das Wortproblem ist also leicht zu lösen.

Reguläre Grammatiken bzw. Ausdrücke sind sehr nützlich z.B. bei der Beschreibung von Suchmustern.

Kontextfreie Grammatiken erzeugen Sprachen, für die das Wortproblem effizient gelöst werden kann.

Programmiersprachen werden deshalb in der Regel durch kontextfreie Grammatiken beschrieben.

Ein gültiges, vom Compiler akzeptiertes Programm ist aus formaler Sicht ein Wort der Sprache.

Unbeschränkte Grammatiken erzeugen Sprachen, für die das Wortproblem nur zum Teil gelöst werden kann. Gehört ein Wort zu der von einer unbeschränkten Grammatik erzeugten Sprache, kann man eine Turingmaschine angeben, die dies erkennt. Die gleiche Turingmaschine würde aber nicht anhalten für ein Wort, das nicht zu der Sprache gehört.



Geburtstagskerzen

Heute ist Bodos elfter Geburtstag. Bodos Mutter findet aber nur noch fünf Kerzen.
Zum Glück weiß sie, wie sie die Zahl elf mit fünf Kerzen darstellen kann.
Sie steckt sie alle nebeneinander auf den Kuchen:

Die Kerze ganz rechts ist 1 wert.
Jede andere Kerze ist das Doppelte der Kerze rechts daneben wert.
Die Werte aller brennenden Kerze werden addiert.

Zum Beispiel:

1



2



4

 $2 + 1 = 3$  $16 + 4 + 1 = 21$ 

Welche Kerzen brennen an Bodos elftem Geburtstag?

A)



B)



C)



D)



E)



Antwort A ist richtig:

Die Kerzen mit den Werten 8, 2 und 1 brennen: $0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 11$.

Falsch sind:

B, denn die Kerzen mit den Werten 8, 4 und 2 brennen: $0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 14$.

C, denn nur die Kerze mit Wert 16 brennt: $1 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 = 16$.

D, denn die Kerzen mit den Werten 16, 8 und 2 brennen: $1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 26$.

E, denn alle Kerzen brennen: $1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 31$.

Das ist Informatik!

Mit „Kerze an“ oder „Kerze aus“ wird ausgedrückt, ob ein Zahlenwert addiert wird oder nicht. Durch die Position der Kerze wird die Höhe des Zahlenwerts festgelegt. Das Gleiche kann man auch mit Einsen (für Kerze an) und Nullen (für Kerze aus) beschreiben. Bodos Geburtstagskerzen lassen sich also mit 01011 beschreiben. Das ist die Darstellung der Zahl 11 im Binärsystem. Jede beliebige Zahl kann im Binärsystem dargestellt werden. Das Binärsystem (auch Dualsystem genannt) wird intern von fast allen modernen Computern genutzt. Das hat praktische Gründe: Logische Schaltkreise, die in Computern eingebaut sind, sind für das Binärsystem einfacher zu realisieren als z. B. für das Dezimalsystem.



3-4: –

5-6: schwer

7-8: –

9-10: leicht

11-13: –



Geheime Nachrichten

Boris und Berta senden sich geheime Nachrichten, die niemand außer ihnen verstehen soll. Boris möchte Berta diesen Text mitteilen:

TREFFENMITBILLYUM6

Boris schreibt die Zeichen des Texts nacheinander in die Felder einer Tabelle mit vier Spalten und fünf Zeilen. Dabei füllt er die Tabelle Zeile für Zeile von oben nach unten, in jeder Zeile von links nach rechts. In leere Felder am Ende der Tabelle schreibt Boris ein #.

Das Bild zeigt das Ergebnis.

Nun nimmt Boris die Zeichen aus der Tabelle aber diesmal Spalte für Spalte von links nach rechts, aus jeder Spalte von oben nach unten.

Er sendet also diese geheime Nachricht:

TFILMRETL6ENBY#FMIU#

Berta verwendet für ihre Antwort dieselbe Methode.

Sie sendet folgende geheime Nachricht an Boris:

OHDRIKWETNIEDS#CROE#

Wie lautet Bertas Antwort?

- A) OITDRHKNSODWIEREEC
- B) OKICHWERDEDORTSEIN
- C) WIRSTDUAUCHDASEIN
- D) ORTDESTREFFENSISTWO

T	R	E	F
F	E	N	M
I	T	B	I
L	L	Y	U
M	6	#	#

Antwort B ist richtig:

Um herauszufinden, welcher Text mitgeteilt wird, muss man die Methode umdrehen: Man schreibt die Zeichen der geheimen Nachricht wieder in eine Tabelle, diesmal aber so, wie sie herausgenommen wurden: Spalte für Spalte von links nach rechts, in jeder Spalte von oben nach unten.

Liest man die Zeichen nun so, wie sie hineingeschrieben wurden (Zeile für Zeile), erhält man diese Nachricht: OKICHWERDEDORTSEIN

Die # am Ende gehören nicht zur Nachricht.

Das ist Informatik!

Nachrichten, die man über ein Computernetz sendet, können leicht abgefangen werden. Doch nur der gewünschte Empfänger soll die Nachricht verstehen und niemand anderes. Dazu muss man den Text der Nachricht verschlüsseln und in eine geheime Nachricht umwandeln. Wenn nur der gewünschte Empfänger weiß, wie man die geheime Nachricht wieder entschlüsseln und den ursprünglichen Text zurückgewinnen kann, ist die Geheimhaltung geglückt.

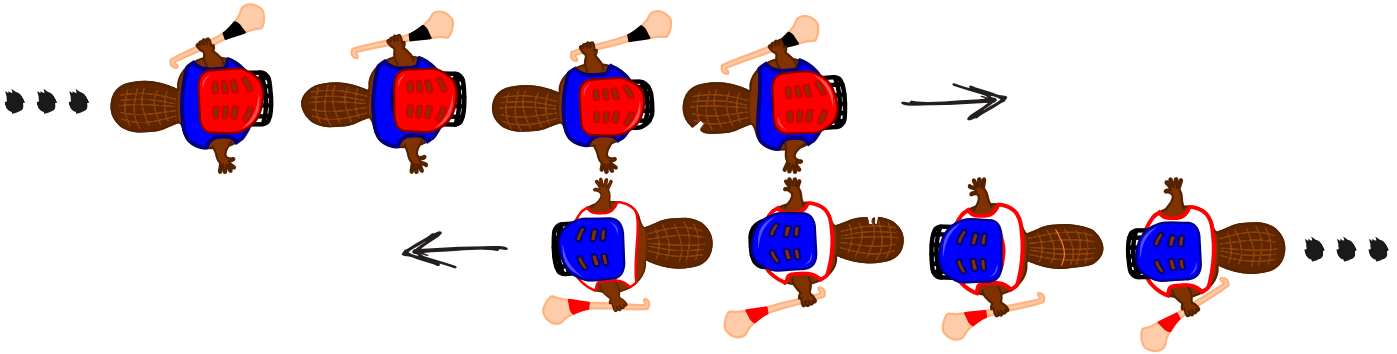
Seit Jahrtausenden haben Menschen Gründe, ihre Nachrichten geheim zu halten. Es gibt deshalb viele Verfahren zur Verschlüsselung (und Entschlüsselung). Das Verfahren in dieser Aufgabe verwendet das Prinzip der Transposition. Alle Buchstaben des Textes bleiben erhalten, nur ihre Reihenfolge wird geändert. Nachrichten, die mit einem Transpositionsverfahren verschlüsselt wurden, können aber relativ leicht auch von unerwünschten Empfängern entschlüsselt werden.

Die Kryptographie ist die Wissenschaft der Verschlüsselung und liegt an der Grenze zwischen Mathematik und Informatik. Moderne Verschlüsselungsverfahren werden mit dem Computer durchgeführt. Dabei ist die Verschlüsselung leicht zu berechnen, aber die Entschlüsselung praktisch unmöglich, wenn man nicht der gewünschte Empfänger ist.



Hände schütteln

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



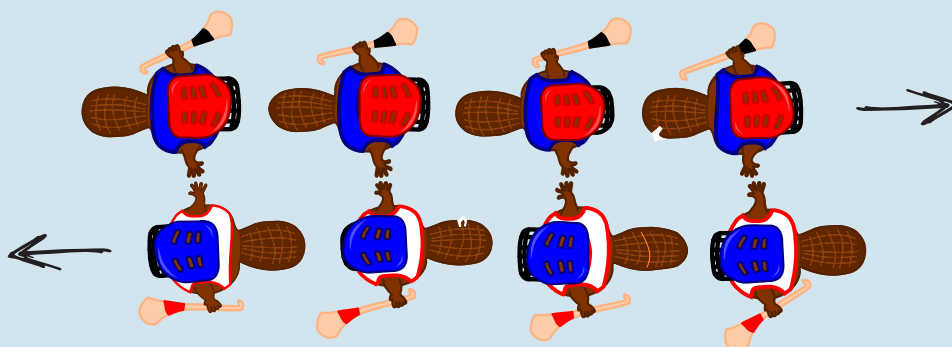
Das Händeschütteln geht so: Zuerst schütteln sich die beiden ersten Spieler die Hände. Dann schütteln die ersten Spieler den zweiten Spielern der jeweils anderen Mannschaft die Hände (siehe Bild). Dies geht so weiter, bis auch die beiden letzten Spieler sich die Hände geschüttelt haben.

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

Wie viele Sekunden dauert das Händeschütteln der beiden Mannschaften insgesamt?

**29 ist richtig:**

Wenn der erste Spieler den letzten Spieler erreicht und ihm die Hände geschüttelt hat, sind genau so viele Sekunden vergangen, wie jede Mannschaft Spieler hat. Bei Mannschaften mit vier Spielern sind das vier Sekunden, und die Spieler stehen dann so:



Die letzten Spieler haben dann dem ersten Spieler der jeweils anderen Mannschaft die Hände geschüttelt. Sie müssen danach noch allen weiteren Spielern der anderen Mannschaft die Hände schütteln. Bei vier Spielern pro Mannschaft dauert das noch einmal vier weniger eins gleich drei Sekunden.

Allgemein dauert das Händeschütteln in Sekunden also zweimal die Anzahl der Spieler einer Mannschaft weniger eins. Bei 15 Spielern pro Mannschaft dauert es folglich:

$$(2 \cdot 15) - 1 = 29 \text{ Sekunden.}$$

Das ist Informatik!

Für Hurling-Teams mit 15 Spielern konnten wir genau berechnen, wie lange ihr Händeschüttel-Algorithmus abläuft. 29 Sekunden sind für die Zuschauer gut auszuhalten. Doch wie sieht diese „Laufzeit“ z.B. bei Eishockey-Teams mit insgesamt 22 Spielern aus? Ist das Hurling-Händeschütteln dann immer noch brauchbar, oder würde es zu lange dauern? Es wäre gut, eine allgemeine Einschätzung der Laufzeit eines Algorithmus zu haben, anstatt Laufzeiten für bestimmte Werte einzeln zu überlegen.

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

Hinter dem Laufzeitausdruck $2n - 1$ steckt eine lineare Funktion. Damit gehört der Hurling-Händeschüttel-Algorithmus zur Klasse der Algorithmen mit linearer Laufzeit, die man auch als $O(n)$ bezeichnet. Ein Händeschüttel-Algorithmus, bei dem jeder jedem anderen nacheinander die Hand gibt, wäre weniger schnell. Er gehörte zur Klasse $O(n^2)$, und mit einem Händeschüttel-Algorithmus dieser Klasse könnten die Hurling-Teams in der Größenordnung von $15^2 = 225$ Sekunden benötigen. Das sind fast vier Minuten – gäh! Es lohnt sich also darüber nachzudenken, ob man Dinge parallel, also gleichzeitig erledigen kann – wie die Hurler beim Händeschütteln.



IP-Adressen

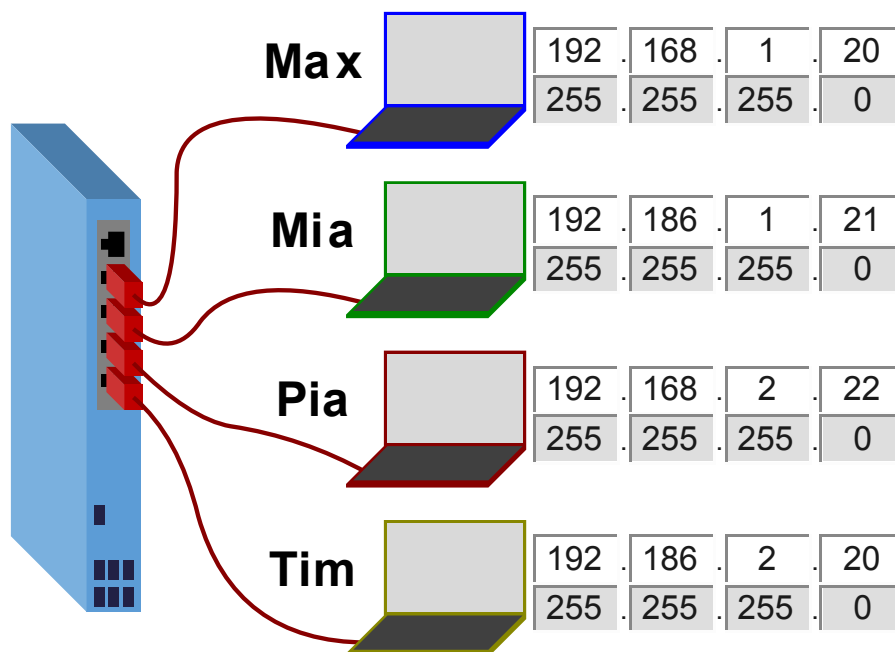
In einem Netzwerk hat jeder Computer eine eindeutige IP-Adresse. Die besteht aus vier Teilen mit Zahlenwerten zwischen 1 und 254. Ein Beispiel: 192.168.1.25

Auch die Netzmaske besteht aus vier Teilen, in denen meist 255 oder 0 steht. Ein Beispiel: 255.255.0.0. Die Netzmaske trennt die Adressen im Netzwerk in den Netzanteil und den Geräteanteil. Eine 255 bedeutet: Dieser Teil gehört zum Netzanteil. Eine 0 bedeutet: Dieser Teil gehört zum Geräteanteil.

Damit die Geräte in einem Netzwerk kommunizieren können, dürfen ihre Adressen sich nur im Geräteanteil unterscheiden. Der Netzanteil muss immer gleich sein.

Ein Beispiel mit Netzmaske 255.255.0.0: Zwei Geräte mit IP-Adressen 172.16.0.10 und 172.16.0.11 können untereinander kommunizieren. Ein weiteres Gerät mit IP-Adresse 172.31.0.12 kann mit keinem der beiden anderen kommunizieren.

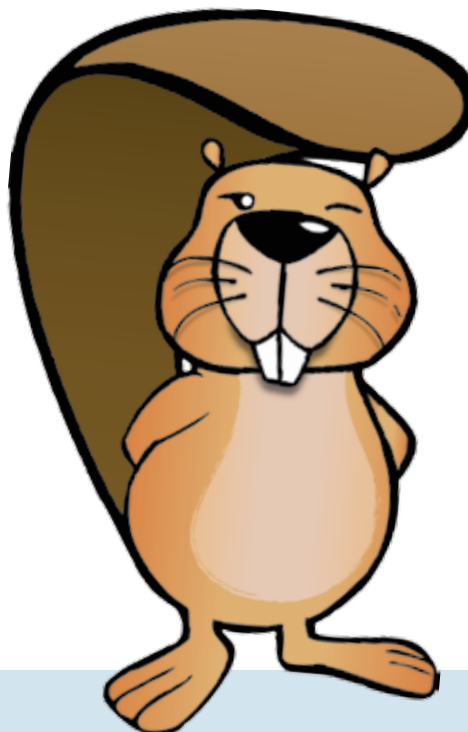
Max, Mia, Pia und Tim wollen ein gemeinsames Multiplayer-Spiel starten. Mithilfe von Netzkabeln verbinden sie ihre Computer mit einem sogenannten Switch. Als Netzmaske wird 255.255.255.0 eingestellt. Doch nachdem sie ihre IP-Adressen gesetzt haben, können keine zwei Computer miteinander kommunizieren. Was ist los?



Ändere die IP-Adressen der Computer so, dass alle vier Computer untereinander kommunizieren können. Es gibt viele richtige Möglichkeiten; eine genügt.



Nützliche Informationen zum Thema Einstieg in die Programmierung findest du unter:
www.coding4you.at



So ist es richtig:

Die Netzmaske 255.255.255.0 gibt vor, dass die ersten drei Teile der vier Adressen identisch sein müssen. Im vierten Teil müssen sich die Adressen aber alle unterscheiden, und alle Werte müssen im Bereich von 1 bis 254 liegen. Eine richtige Möglichkeit ist also:

Max: [192.168.1.20]

Mia: [192.168.1.21]

Pia: [192.168.1.22]

Tim: [192.168.1.30]

Das ist Informatik!

Das „Internet Protocol“ IP wurde 1974 zum ersten Mal beschrieben. Mit Hilfe von IP sollten Computer flexibel und notfalls auf unterschiedlichen Wegen Daten untereinander austauschen können. Zusammen mit TCP (Transmission Control Protocol) bildet IP die Grundlage des Internet. Damit ein Gerät mit anderen im Internet vernetzten Geräten kommunizieren kann, muss sein Betriebssystem TCP/IP beherrschen – und das Gerät muss eine IP-Adresse haben.

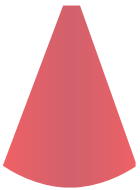
Die hier vorgestellte Struktur von IP-Adressen entspricht der aktuell noch überwiegend genutzten Version des Internet Protocol, IPv4. IPv4-Adressen bestehen also aus 4 Bytes. In jedem Byte sind 254 verschiedene Werte erlaubt. Deshalb kann es höchstens $254 \cdot 254 \cdot 254 \cdot 254 = 4.162.314.256$, also gut vier Milliarden unterschiedliche IPv4-Adressen geben. Längst aber sind viel mehr Geräte mit dem Internet verbunden. Wie gut, dass Geräte in einem lokalen Netzwerk (z.B. in der Schule oder zu Hause) hinter einem so genannten Router „versteckt“ werden können. Nur der Router muss eine universell eindeutige Adresse haben; die Adressen der anderen Geräte müssen nur im lokalen Netz eindeutig sein.

Trotzdem gibt es mittlerweile zu wenige IPv4-Adressen. In der Nachfolgeversion IPv6 dieses Internet Protocols wird eine neue Adress-Struktur festgelegt, die sehr viel mehr unterschiedliche Adressen ermöglicht. Die Informatik hat dafür gesorgt, dass beide IP-Versionen im Internet parallel verwendet werden können. Es ist auch deshalb nicht abzusehen, dass IPv4-Adressen irgendwann verschwinden.

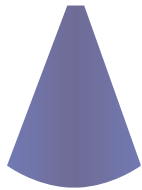


Karten und Kegel

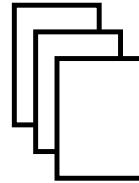
rot



blau

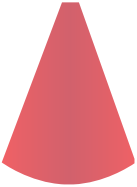


gelb



Der Biber spielt mit Karten, auf die er Zahlen schreibt.
Die Karten stapelt er unter farbigen Kegeln. Wenn er einen Kegel anhebt,
sieht er nur die oberste Karte des Stapels.

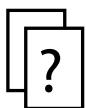
Der Biber schreibt in dieser Form auf, was er tut:

rot <--- 5;	rot <--- 3;
rot  5	rot  3

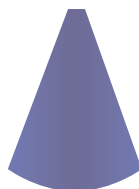
Nun bist du dran. Der Biber schreibt:

rot <--- 3; gelb <--- 5; rot <--- 6; gelb <--- 8; blau <--- 1; gelb <--- 3;

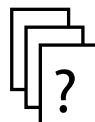
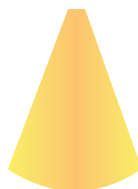
rot



blau



gelb



Welche Karten sieht der Biber, wenn er die Kegel anhebt?

**So ist es richtig:**

rot 6, blau 1, gelb 3

Der Kartenstapel unter dem roten Kegel besteht von unten nach oben aus den Karten 3 und 6.

Der Kartenstapel unter dem blauen Kegel besteht aus einer Karte 1.

Der Kartenstapel unter dem gelben Kegel besteht von unten nach oben aus den Karten 5, 8 und 3.

Nur die obersten Karten der drei Stapel sind sichtbar.

Das ist Informatik!

Daten brauchen ihren Platz. Für verschiedene Arten von Daten haben die Menschen immer schon überlegt, wie man Daten diesen Platz geben kann. Einige Beispiele: Texte und Bilder kann man auf Papier drucken, Musik als Rillen in Vinylscheiben prägen, Videobilder durch Belichtung von Zelluloidfilmrollen speichern. Ob Papier, Vinyl oder Zelluloid: Für verschiedene Arten von Daten unterscheiden sich diese Daten-Speicherplätze. Und sind die Daten erst einmal gespeichert, kann man sie nur noch schwer verändern.

Beide Einschränkungen verschwinden, wenn Daten in Computern ihren Platz finden. Zum einen: Daten, die an einem Platz im Computerspeicher abgelegt sind, können beliebig verändert werden. Wie bei den Kegeln: Man legt einfach eine neue Karte unter den Kegel, und sofort wird die zuletzt abgelegte Karte nicht mehr lesbar – und damit vergessen. Anders als bei den Karten unter den Kegeln werden auf einem Platz im Computerspeicher die alten Daten aber nicht aufbewahrt.

Zum anderen: Alle Arten von Daten können ihren Platz im Computerspeicher finden. Unter die Kegel passen nur Karten, aber in den Computerspeicher passen Texte, Bilder, Musik, Videos – und insbesondere Programme, die den Computer steuern können. Das liegt daran, dass es für alle Datenarten Methoden gibt, sie in die Sprache des Computers umzuwandeln: den Binärcode. Letztlich gibt es im Computer nur 0 und 1, aus und an, nein und ja. Und so könnte man prinzipiell versuchen, einen Text als Video anzusehen oder einen Song als Programm laufen zu lassen. Vermutlich kommt nichts Vernünftiges dabei heraus – aber wer weiß.



3-4: schwer

5-6: leicht

7-8: –

9-10: –

11-13:



Käsegänge

Eine Maus steht oben vor fünf langen Gängen.

Zwischen den langen Gängen gibt es kurze Quergänge.

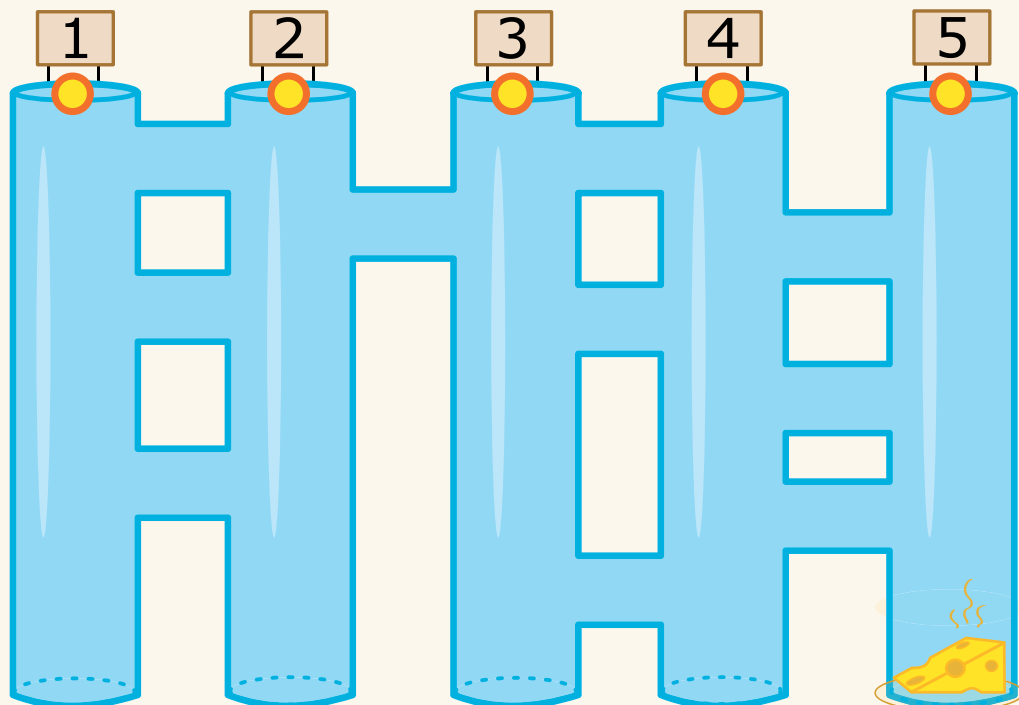
Die Maus möchte zum Käse unten am Ende vom langen Gang ganz rechts.

Auf ihrem Weg durch die Gänge befolgt die Maus immer abwechselnd diese Anweisungen:

Laufe durch den langen Gang nach unten, bis ein Quergang kommt.

Laufe durch den Quergang.

In welchen langen Gang muss die Maus hineinlaufen, damit sie zum Käse kommt?





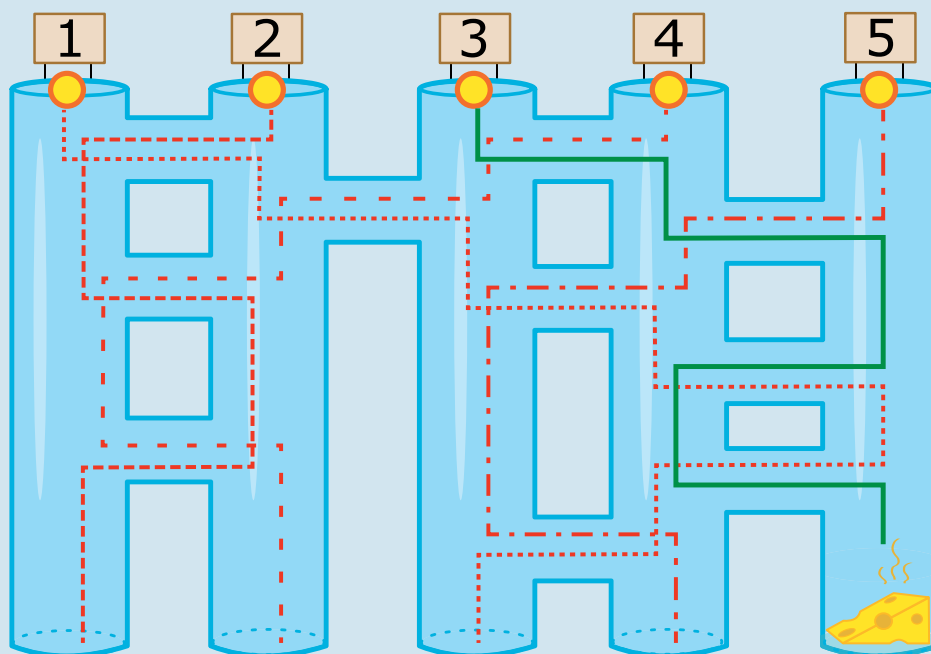
3-4: schwer

5-6: leicht

7-8: –

9-10: –

11-13:

**So ist es richtig:**

Die Maus muss in Gang 3 hineinlaufen, damit sie zum Käse kommt. Die grüne Linie zeigt, wie sie läuft.

Wenn die Maus in einen anderen Gang hineinläuft, kommt sie nicht zum Käse.

Die roten gestrichelten Linien zeigen die falschen Wege:

Läuft sie in Gang 1 hinein, kommt sie ans Ende von Gang 3.

Läuft sie in Gang 2 hinein, kommt sie ans Ende von Gang 1.

Läuft sie in Gang 4 hinein, kommt sie ans Ende von Gang 2.

Läuft sie in Gang 5 hinein, kommt sie ans Ende von Gang 4.

Das ist Informatik!

Die Maus hat klare Anweisungen, wie sie durch die Käsegänge laufen soll. Die Anweisungen sagen der Maus, wie sie auf bestimmte Ereignisse reagieren soll. Das kann man leicht erkennen, wenn die Anweisungen anders ausgedrückt werden:

Wenn das Ereignis eintritt, dass du in einen langen Gang hineinkommst, laufe den langen Gang hinunter.

Wenn das Ereignis eintritt, dass du an einen Quergang kommst, laufe durch den Quergang.

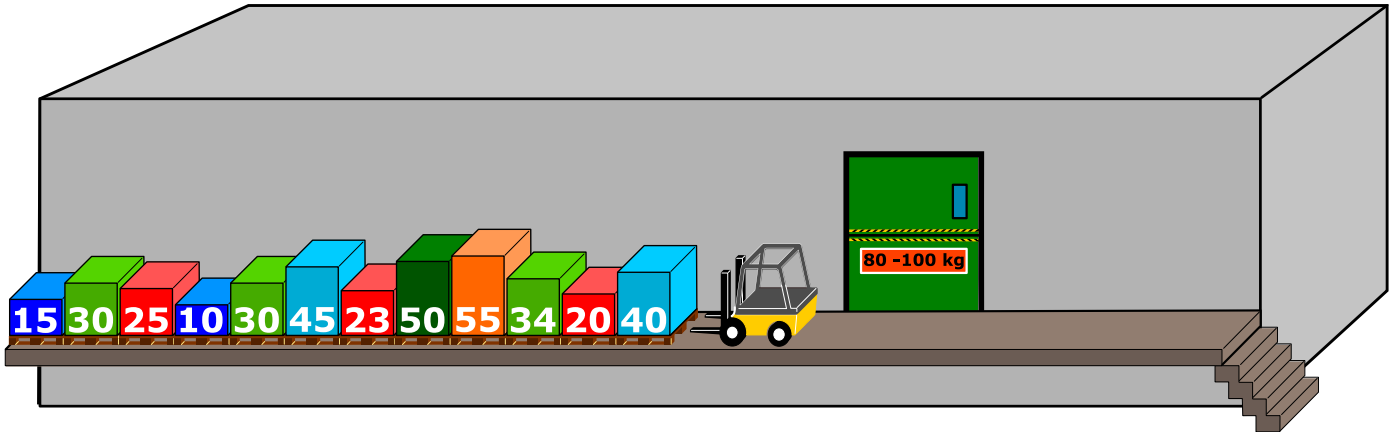
Viele Computerprogramme müssen auf Ereignisse reagieren. Besonders wichtig ist das bei dem Teil eines Computerprogramms, das seine Benutzung durch Menschen regelt: der Mensch-Computer-Schnittstelle, auf Englisch meist Graphical User Interface (GUI) genannt. Dieser Programmteil muss festlegen, was passiert, wenn mit der Maus oder dem Finger etwas angeklickt wird, auf dem Bildschirm gezeigte Objekte bewegt werden oder mit der Tastatur ein Buchstabe getippt wird.

Da für sehr viele Programme ein GUI benötigt wird, hat die Informatik einen passenden Programmierstil entwickelt, das ereignis-getriebene Programmieren (englisch: event-driven programming). Programmierwerkzeuge, die diesen Stil unterstützen, helfen Programmiererinnen und Programmierern dabei, gute GUIs zu erstellen.



Kistenlift

Von einer Laderampe aus führt ein Lift in einen Keller. Auf einer Seite der Laderampe stehen eine Reihe Kisten. Die Kisten wiegen, von links nach rechts in kg (Kilogramm):
15, 30, 25, 10, 30, 45, 23, 50, 55, 34, 20, 40.



Die Kisten sollen in den Lift geladen und damit in den Keller gebracht werden. Der Lift macht eine Fahrt, sobald seine Ladung mindestens 80 kg und höchstens 100 kg wiegt. Er kommt danach leer wieder zurück.

Die Laderampe ist so eng, dass Kisten nicht aneinander vorbei gehoben werden können. Deshalb wird immer die vorderste Kiste der Reihe als nächste bearbeitet. Diese Kiste wird nur in den Lift geladen, falls danach dessen Ladung höchstens 100 kg wiegt. Andernfalls wird die Kiste auf der anderen Seite der Laderampe in einer neuen Reihe abgestellt.

Wenn alle Kisten der Reihe bearbeitet worden sind, werden die Kisten, die nun auf der anderen Seite der Laderampe in einer Reihe stehen, genau so bearbeitet.

Alle Kisten werden nun auf die beschriebene Weise mit dem Lift in den Keller gebracht.

Welche der folgenden Aussagen ist richtig?

- A) Bei der zweiten Fahrt des Lifts wiegt seine Ladung 98 kg.
- B) Es werden keine Kisten auf der anderen Seite der Laderampe abgestellt.
- C) Bei einer Fahrt des Lifts wiegt seine Ladung 100 kg.
- D) Der Lift macht insgesamt fünf Fahrten.
- E) Es ist unmöglich, alle Kisten auf die beschriebene Weise in den Keller zu bringen.



3-4: –

5-6: –

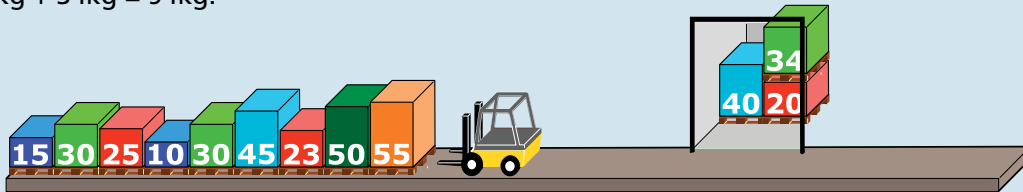
7-8: –

9-10: schwer

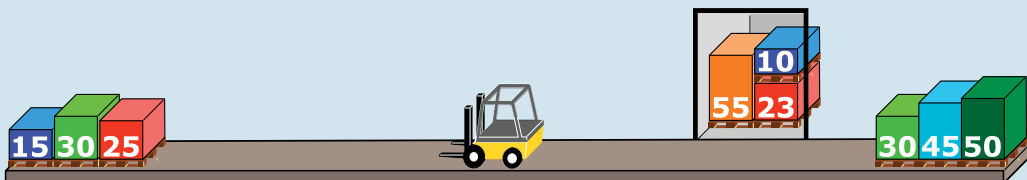
11-13: mittel

**Antwort C ist richtig:**

Bei der ersten Fahrt besteht die Ladung aus den ersten drei Kisten und wiegt $40\text{kg} + 20\text{kg} + 34\text{kg} = 94\text{kg}$.



Für die zweite Fahrt wird zuerst die nächste Kiste (55 kg) in den Lift geladen. Die folgende Kiste (50 kg) kann nicht in den Lift geladen werden ($55\text{ kg} + 50\text{ kg} > 100\text{ kg}$), und wird auf der anderen Seite der Laderampe abgestellt. Die nächste Kiste (23 kg) wird in den Lift geladen; die Ladung ist aber noch zu leicht: $55\text{ kg} + 23\text{ kg} = 78\text{ kg}$. Die nächste Kiste (45 kg) kann nicht in den Lift geladen werden (zu schwer) und wird auf der anderen Seite der Laderampe abgestellt. Dasselbe geschieht mit der nächsten Kiste (30 kg). Schließlich kann die nächste Kiste (10 kg) in den Lift geladen werden. Der Lift macht seine zweite Fahrt mit einer Ladung von $55\text{ kg} + 23\text{ kg} + 10\text{ kg} = 88\text{ kg}$.



Die letzten drei Kisten der Reihe (25 kg + 30 kg + 15 kg = 70 kg) werden ebenfalls in den Lift geladen. Nun werden die Kisten auf der anderen Seite der Laderampe bearbeitet. Die erste Kiste dieser Reihe (30 kg) wird in den Lift geladen, der mit einer Ladung von genau 100 kg seine dritte Fahrt macht.



Zu diesem Zeitpunkt sind nur noch zwei Kisten auf der anderen Seite der Laderampe übrig: $45\text{ kg} + 50\text{ kg} = 95\text{ kg}$. Der Lift macht mit diesen Kisten seine vierte Fahrt. Die Ladung der dritten Fahrt wiegt 100 kg; die Aussage C ist also richtig. Die Aussagen A, B, D und E sind nicht richtig, wie die Erklärung zeigt.

Das ist Informatik!

In der Informatik gibt es einige elementare Datenstrukturen, die sich immer wieder für die Modellierung einer Anwendung gut eignen. Eine Datenstruktur, die man auf jeden Fall kennen sollte, ist der „Stapel“ (englisch „stack“). Ein Stapel kann beliebig viele Elemente enthalten; besonders wichtig ist das oberste Element.

In dieser Biberaufgabe gibt es zwei Stapel. Man erkennt sie vielleicht auf den ersten Blick nicht, weil sie quer liegen. Der eine Stapel ist die erste Kistenreihe, die zu Beginn alle 12 Kisten enthält. Ihr oberstes Element ist die Kiste, die 40 kg wiegt. Der zweite Stapel ist die Kistenreihe auf der anderen Seite der Laderampe, die sich beim Verladen bildet. Zu Beginn ist dieser Stapel leer, er enthält keine Elemente und ist darum noch unsichtbar.

An einem Stapel sind nur zwei Handlungen erlaubt:

- 1) Lege ein Element oben auf den Stapel.
- 2) Nimm das oberste Element vom Stapel weg.

Sie werden traditionell gerne PUSH und POP genannt. Hier findet man die Begründung warum: [en.wikipedia.org/wiki/Stack_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Stack_(abstract_data_type))

Auf der Laderampe werden PUSH und POP vom Gabelstapler ausgeführt.



Kix

In den Biberlanden sind die Postleitzahlen vierstellig und enthalten Buchstaben und Ziffern. Es gibt sogar einen eigenen Strichcode für die Postleitzahlen, den Kix-Code. In jedem Zeichen des Kix-Codes gibt es einen oberen Teil, zwei lange und zwei kurze Balken, und einen unteren Teil, ebenfalls zwei lange und zwei kurze Balken. In der Mitte überdecken sich die Balken.

In der Tabelle siehst du die Kix-Code-Zeichen für 0, 7, G und Y.

Für die biberländische Postleitzahl G7Y0 steht also dieser Kix-Code:



Für welche Postleitzahl steht dieser Kix-Code?



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

**BC16 ist richtig:**

Der gefragte Kix-Code besteht aus vier Kix-Code-Zeichen:



Das Bild zeigt die Tabelle aus der Aufgabenstellung, aber nun mit den Kix-Code-Zeichen für alle Buchstaben und Ziffern. In der Tabelle kann man die Bedeutung der vier Zeichen des gefragten Kix-Codes nachschauen.

	0	1	2	3	4	5
1						
2						
3						
4						
5						
6						

Das ist Informatik!

Maschinenlesbare grafische Codes ermöglichen es, vieles automatisch zu erledigen, wo vorher Menschen an Fließbändern arbeiteten; zum Beispiel um Briefe und Pakete zu sortieren. Der Barcode („Bar“ ist das englische Wort für „Balken“) kommt unter anderem in Supermärkten an den Laserscanner-Kassen zum Einsatz. Der QR-Code wurde für die Autoindustrie entwickelt, um Bauteile zu kennzeichnen. Mittlerweile ist der QR-Code auch überall in der Werbung zu finden. Es gibt kostenlose Smartphone-Apps zum Scannen der Codes und Ausgeben deren Botschaften.



Die Informatik befasst sich mit der Verarbeitung und damit auch der Übertragung von Information – wie der Name schon sagt. Information muss aber nicht nur von Computer zu Computer übertragen werden, sondern auch aus der realen Welt in die virtuelle Welt der Computer. Ob Kix-Code, Barcode, oder QR-Code: Grafische Codes machen diese Übertragung einfacher – und reichhaltiger. An der Supermarkt-Kasse muss kein Preis mehr vom Preisschild abgetippt werden, seitdem es Barcodes und Barcode-Scanner gibt. Und da der Barcode mehr Information enthält als nur den Preis, weiß der Supermarkt-Computer gleich, welche Ware verkauft wurde.

Übrigens: In den Niederlanden wird der Kix-Code tatsächlich bei der Post verwendet, allerdings mit ein paar Zeichen mehr.



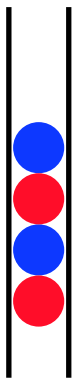
Kugelspiel

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

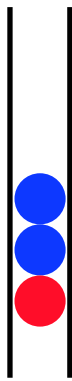
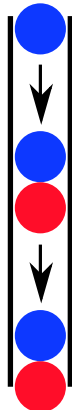
Nach einem Klick auf eine Taste fallen immer die beiden unteren Kugeln aus der Röhre. Außerdem fallen von oben neue Kugeln in die Röhre. Dabei gibt es zwei Möglichkeiten, abhängig von der Farbe der untersten Kugel vor dem Klick:

Falls die unterste Kugel rot war, fällt von oben eine blaue Kugel in die Röhre:

vorher

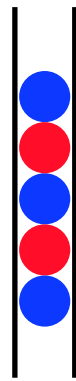


nachher

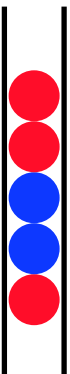
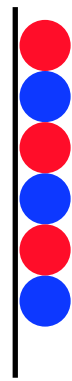


Falls die unterste Kugel blau war, fallen drei Kugeln mit den Farben rot, blau und rot von oben in die Röhre:

vorher



nachher



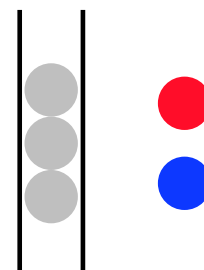
Solange mindestens drei Kugeln in der Röhre sind, klickt Emil immer wieder auf die Taste. Das Spiel endet, wenn sich weniger als drei Kugeln in der Röhre befinden.

Ein Beispiel: Wenn Emil mit dieser Kugelfolge beginnt, bleiben nach fünf Klicks nur zwei blaue Kugeln übrig, und das Spiel ist beendet.

Emil findet heraus, dass es zu Spielbeginn Kugelfolgen geben kann, so dass das Spiel niemals endet, egal wie oft er klickt. Er nennt solche Kugelfolgen „Endlos-Folgen“.

Bilde eine Endlos-Folge aus drei Kugeln.

Ziehe dazu von rechts passende Farben auf die drei Kugelplätze.





3-4: –

5-6: –

7-8: –

9-10: schwer

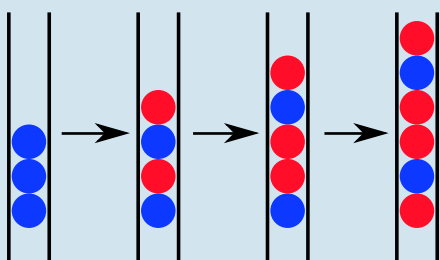
11-13: mittel

**So ist es richtig:**

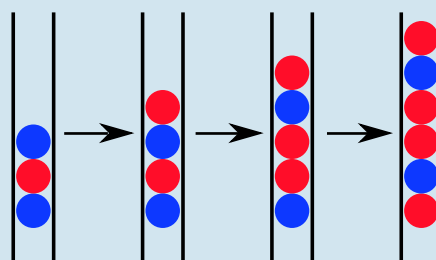
Wenn die unterste Kugel einer Dreierfolge rot ist, endet das Spiel schon nach dem ersten Klick: Dann sind nur noch zwei Kugeln in der Röhre.

Hingegen ist jede Dreierfolge, deren unterste Kugel blau ist, eine Endlos-Folge. Mit höchstens vier Klicks wird nämlich aus jeder der vier möglichen Dreierfolgen mit unterster blauer Kugel die Sechserfolge rot-blau-rot-rot-blau-rot:

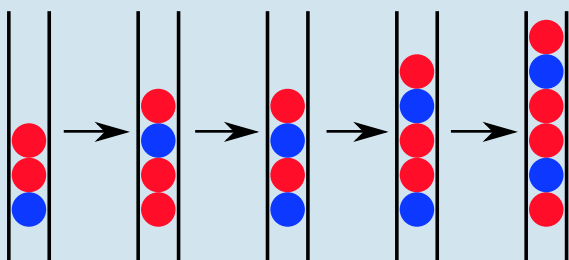
1)



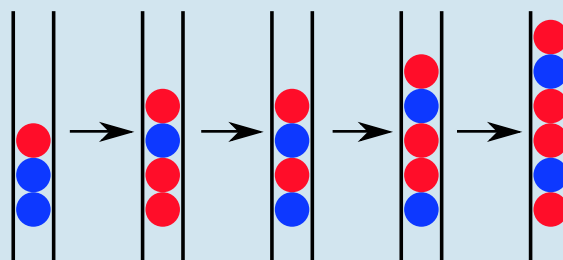
2)



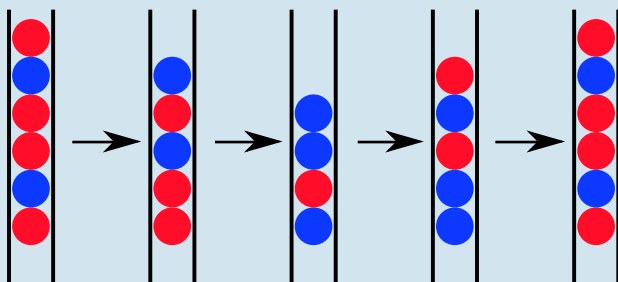
3)



4)



Nach weiteren vier Klicks entsteht dann wieder diese Folge, und so geht es endlos weiter.

**Das ist Informatik!**

Emil Leon Post (1897-1954) war ein polnischer Mathematiker und Logiker, der viele wissenschaftliche Beiträge zur theoretischen Informatik auf dem Gebiet der Aussagenlogik veröffentlichte. Das in dieser Biberaufgabe vorgestellte Spiel wurde von Post als Beispiel verwendet, um zu zeigen, dass es bei Systemen, in denen Symbolketten anhand bestimmter Regeln manipuliert werden, zu nicht abbrechenden Prozessen kommen kann.

In solchen Systemen werden Teile von Symbolketten durch andere Symbolketten ersetzt. Sie werden in der theoretischen Informatik auch als Ersetzungssysteme bezeichnet. Die möglichen Ersetzungen werden durch eine endliche Menge von Regeln vollständig bestimmt. Die Umformungen, die ein Ersetzungssystem ermöglicht, können als eine Art von Berechnung angesehen werden – und damit werden Sie für die Informatik wichtig. Für die von Post erfundenen Ersetzungssysteme wie auch für andere Arten von Ersetzungssystemen, etwa die so genannten Semi-Thue-Systeme, konnte nachgewiesen werden, dass sie – als Berechnungsmechanismus – gleichwertig zu Turingmaschinen sind. Das bedeutet aber auch, dass es Fragen gibt, die für Ersetzungssysteme nicht entschieden werden können: Das Halteproblem der Turingmaschine entspricht hier dem Wortproblem; beide sind nicht entscheidbar.



Laut oder Leise

Karaoke ist ein großer Spaß. Ein Mensch bemüht sich mit Melodie und Text. Eine Musik-App spielt die Begleitmusik. Laut oder leise. So, wie auf den vier Bildern jeweils die Lautstärke eingestellt ist.

Auf welchem Bild ist die Musik-App am lautesten eingestellt?



Antwort C ist richtig:

Bei dem Bild mit dem von links (leise) nach rechts (laut) dicker werdenden grünen Balken ist die als Lautstärken-Anzeige dienende rote Kugel auf sehr leise eingestellt.

Bei dem Bild mit der gelben Lautstärken-Anzeige von unten (leise) nach oben (laut) ist ungefähr die halbe Lautstärke eingestellt.

Bei dem Bild mit den größer werdenden bunten Rechtecken, von blau (leise) bis rot (laut), ist der senkrechte schwarze Schieber ganz rechts auf volle Lautstärke eingestellt.

Bei dem Bild mit dem durchkreuzten Lautsprecher-Symbol ist der Lautsprecher auf stumm geschaltet. Die Mutige singt ein Solo.

Das ist Informatik!

Auf unseren kleinen und großen Computerbildschirmen ist massenhaft grafische Symbolik zu sehen. Manche zeigt Programmzustände an. Manche steht für Programmfunktionen, die benutzt werden können.

Die Bedeutung grafischer Symbole muss von den Benutzern gelernt und verstanden werden. Das ist mühsam und teuer. Darum benutzen die Hersteller von Mensch-Maschine-Interaktion oft auch im sonstigen Leben Gebräuchliches – so wie in dieser Biberaufgabe die „Lautstärke einer akustischen Ausgabe“. Die Informatik hat darüber hinaus Betriebssysteme entwickelt, die dem einen Benutzer zur Arbeitseffizienz und dem anderen zum ästhetischen Genuss komplett durchgestylt sind. Dabei geht man „paradigmatisch“ vor. Typische „Paradigmen“ sind zum Beispiel „Büroschreibtisch mit Ordnern, Dokumenten und Papierkorb“ oder „Cockpit mit Instrumenten, Schaltern und Anzeigen“ oder „Intelligente digitale Handtasche mit Apps und Daten-Börse“.



Medianfilter

Ein Bild wird als Tabelle gespeichert, mit Grauwerten von 1 bis 5 für jedes Pixel. Der Grauwert 1 steht für Schwarz, der Grauwert 5 steht für Weiß, die Grauwerte von 2 bis 4 stehen für die heller werdenden Grautöne dazwischen.

Ein Algorithmus „Medianfilter“ berechnet aus einem ursprünglichen Bild ein gefiltertes Bild gleicher Größe.

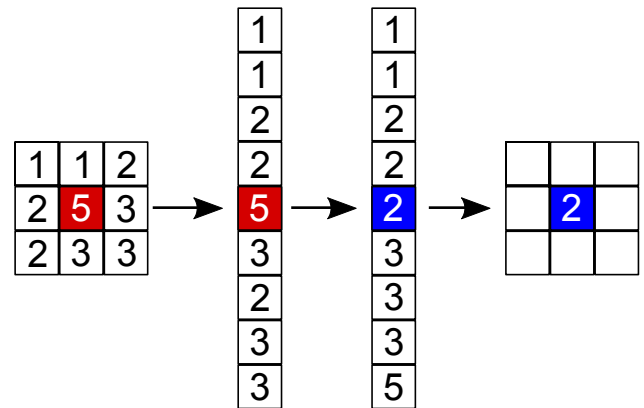
Dazu werden für jedes Pixel aus dem ursprünglichen Bild ...

... sein Grauwert und die acht Grauwerte seiner acht Nachbar-Pixel ...

... aufsteigend sortiert in einer Reihe aufgeschrieben ...

... und der mittlere, also fünfte Wert von neun, ...

... als Grauwert des gleichen Pixels im gefilterten Bild genommen.



Das Bild zeigt, wie der Medianfilter aus dem Grauwert 5 eines Pixels im ursprünglichen Bild den Grauwert 2 für das gleiche Pixel im gefilterten Bild berechnet.

Wie wird das gefilterte Bild aussehen, wenn der Medianfilter dieses ursprüngliche Bild verarbeitet hat?



A)



B)



C)



D)



Antwort A ist richtig:

Bei diesem Medianfilter werden quadratische Bildausschnitte mit neun Pixeln verarbeitet. Das Pixel in der Mitte bekommt einen neuen Wert. Wenn man das Beispiel rechts betrachtet, fällt auf, dass von den neun Pixeln nur drei schwarz sind (die das Pixel in der Mitte selber vor dem Anwenden des Filters). Der Median der sortierten Grautonfolge ist folglich heller als das ungefilterte Original. Aus Schwarz wird ein Grauton. In dem Bild in der Aufgabe sind in allen quadratischen 9-Pixel-Ausschnitten des Bildes die schwarzen Pixel in der Minderheit. Deshalb enthält das gefilterte Bild kein einziges schwarzes Pixel. Das ist nur bei Bild A der Fall.

Das ist Informatik!

Beim Bearbeiten von Fotos möchte man schnell mal bestimmte Effekte erzielen. Häufig wünscht man sich mehr Schärfe oder lebendigere Farben. Manchmal möchte man auch künstlerische Effekte erzielen, um einem Bild eine besondere Note zu geben. Diese Effekte kann man mit Hilfe von Bildfiltern erzeugen. Ein solcher Filter ist der Median-Filter. Er wird beispielsweise dafür verwendet einzelne Pixelfehler, die etwa aufgrund eines Defekts des Bildsensors entstanden sind, auszugleichen. Der Effekt ist, dass das Bild ein wenig geglättet wirkt und kein einzelnes Pixel mehr hervorsteht. Gewisse Formen von Bildrauschen können so verringert werden.



Mittagspause

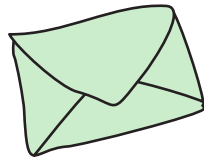
Während ihrer Mittagspause (12:00 Uhr-13:00 Uhr) möchte Alexandra Folgendes erledigen:

- ein Buch in der Buchhandlung kaufen
- eine Flasche Milch im Lebensmittelgeschäft kaufen
- einen Brief per Post versenden
- einen Kaffee trinken in der Cafeteria.

Für jede dieser vier Aktivitäten weiß Alexandra, wie lange sie dauern – aber nur außerhalb der Stoßzeiten. Alexandra will die Stoßzeiten vermeiden.

Aktivität	Dauer	Stoßzeiten
Buch kaufen 	15 Minuten	12:40 Uhr–13:00 Uhr
Milch kaufen 	10 Minuten	12:00 Uhr–12:40 Uhr
Brief versenden 	15 Minuten	12:00 Uhr–12:30 Uhr
Kaffee trinken 	20 Minuten	12:30 Uhr–12:50 Uhr

Bringe die Aktivitäten in eine Reihenfolge, bei der Alexandra die Stoßzeiten vermeidet.



**So ist es richtig:**

Die richtige Reihenfolge ist:

Kaffee trinken

Buch kaufen

Brief versenden

Milch kaufen.

Dieses Problem weist einige Einschränkungen auf. Wenn man die in einer Tabelle darstellt, sieht das so aus (rot: Stoßzeit):

Aktivität	Dauer	12:00	12:05	12:10	12:15	12:20	12:25	12:30	12:35	12:40	12:45	12:50	12:55
Buch kaufen	15 Minuten					X	X	X					
Milch kaufen	10 Minuten											X	X
Brief versenden	15 Minuten								X	X	X		
Kaffee trinken	20 Minuten	X	X	X	X								

Alexandra muss vor 12:40 Uhr in der Buchhandlung gewesen sein. Sie muss das Lebensmittelgeschäft nach 12:40 Uhr besuchen. Die Post kann sie erst nach 12:30 Uhr besuchen. Sie muss die Cafeteria vor 12:30 Uhr besuchen, weil sie nach 12:50 Uhr nicht mehr genügend Zeit für ihre Pause hat. Der einzige mögliche Zeitplan ist also (in der Tabelle oben mit „X“ markiert): Kaffee trinken: 12:00 Uhr-12:20 Uhr; Buch kaufen: 12:20 Uhr-12:35 Uhr; Brief versenden: 12:35 Uhr-12:50 Uhr; Milch kaufen: 12:50 Uhr-13:00 Uhr

Das ist Informatik!

Ein Problem ist nur dann gut gelöst, ein Auftrag ist nur dann wirklich erledigt, wenn alle gegebenen Einschränkungen (Constraints) eingehalten wurden. Je nach Anwendungsfall kann alles Mögliche eine Einschränkung sein.

Für einen mobilen Roboter kann es ein Gegenstand auf seinem Weg sein, für einen Manager kann es das seinem Projekt zugeteilte Geld sein. In dieser Biberaufgabe sind für Alexandra die Zeiträume einschränkend, zu denen eine besonders große Zahl anderer Kunden zu erwarten ist — die Stoßzeiten.

Für die Informatik ist das Lösen von Problemen unter Einschränkungen ein spannendes Thema.

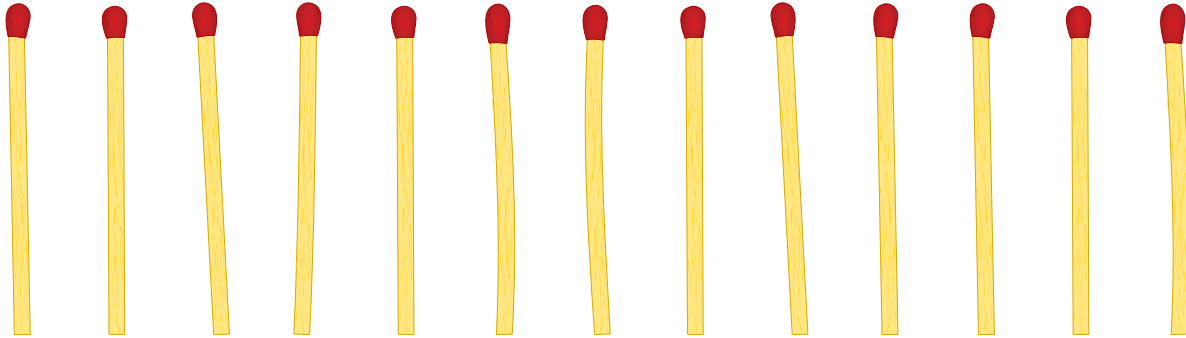
Die unterschiedlichen Anwendungsfälle fordern immer wieder zu kreativen Planungen heraus: Wenn es mehrere Lösungen gibt, welche ist die optimale? Wenn es keine Lösung gibt, welche minimalen Änderungen der Einschränkungen würde eine maximale Teillösung erlauben?

Es kommt noch hinzu, dass die reale Welt nicht still steht und die bisherigen Einschränkungen sich ändern können. Wenn viele Kunden sich wie Alexandra verhalten, verschieben sich die Stoßzeiten. Für alles Planen gilt (nicht nur in der Informatik): Erwarte stets das Unerwartete.



Nim

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



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

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

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

Antwort A ist richtig:

Wenn Beat 1 Hölzchen weg nimmt, bleiben 12 Hölzchen übrig. Beats Freund kann dann 1, 2 oder 3 Hölzchen wegnehmen.

Darauf nimmt Beat so viel weg, dass 8 Hölzchen übrig bleiben. Beats Freund kann dann wieder 1, 2 oder 3 Hölzchen weg nehmen. Darauf nimmt Beat wieder so viel weg, dass 4 Hölzchen übrig bleiben. Nun kann sein Freund nicht mehr gewinnen.

Allgemein gilt: Wenn Beat 2 oder 3 Hölzchen weg nimmt, kann sein Freund stets so reagieren, dass ein Vielfaches von 4 übrig bleibt. Dann kann Beat nicht mehr gewinnen. Wer so viele Hölzchen wegnehmen kann, dass ein Vielfaches von 4 Hölzchen übrig bleibt, der kann nicht mehr verlieren.

Das ist Informatik!

Spielregeln sind Regeln, die bei einem Spiel gelten, bei dem es um Verlieren und Gewinnen geht. Spielregeln erlauben es, die Konsequenzen eines eigenen Spielzugs und die nächsten Züge der Gegenspieler im Voraus zu durchdenken – und die darauf hin möglichen eigenen Gegengegengzüge – und die Gegengegengzüge der Gegenspieler – und so weiter.

Beim Nim ist das sehr überschaubar. In der Informatik beschäftigt man sich aber auch mit Spielen, bei denen alle möglichen Zugfolgen bis zu jeder gewünschten Länge zwar vollständig aufzählbar sind, dies aber das Gedächtnis selbst der geübtesten Menschen weit übersteigt, etwa bei Go und Schach.

Noch interessanter wird es für die Informatik, wenn der Gegenspieler zum Beispiel ein komplexes Problem aus der realen Welt ist und sich an jede Regel nur mit gewisser Wahrscheinlichkeit hält.

Dann gilt es eventuell, weniger auf Sieg zu spielen, sondern Strategien zu entwickeln, die ein Verlustrisiko möglichst klein halten. Etwa bei Wettervorhersagen.

Selten geht in der realen Welt das Vorhersagen so leicht wie in dieser Biberaufgabe, wo jedes Nim nach spätestens 13 Zügen entschieden ist.



Partygäste

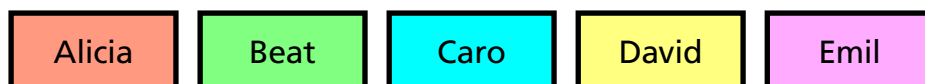
Sarah plant eine Party. Sie möchte ihre Freunde Alicia, Beat, Caro, David und Emil dazu einladen. Aber sie kennt das schon: Manche Freunde wollen gerne wissen, ob bestimmte andere Freunde auch zur Party kommen werden. Nämlich:

- David will wissen, ob Alicia zur Party kommt.
- Beat will wissen, ob Emil zur Party kommt.
- Caro will wissen, ob Beat und David zur Party kommen.
- Alicia will wissen, ob Beat und Emil zur Party kommen.

Wenn Sarah mit David sprechen möchte, muss sie also vorher mit Alicia sprechen. Und so weiter.

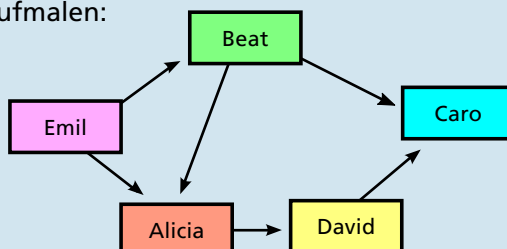
In welcher Reihenfolge muss Sarah mit ihren Freunden sprechen?

Ziehe die Namen in eine passende Reihenfolge.



So ist es richtig:

Zur Begründung kann man sich die „will wissen, ob der andere zur Party kommt“-Beziehungen zwischen Sarahs Freunden mit Pfeilen so aufmalen:



Von z.B. Alicia zu David geht ein Pfeil, weil David ja eben wissen will, ob Alicia zur Party kommt; die anderen Pfeile haben eine entsprechende Bedeutung. Damit entsteht das Bild eines Graphen, mit den Personen als Knoten und den „Bedingungen“ als gerichtete Kanten. Um ihr Reihenfolge-Problem zu lösen, muss Sarah einen Weg durch den Graphen finden, der von einem Knoten zum anderen in Pfeilrichtung an den Kanten entlang geht (damit berücksichtigt sie die Wissensbegier ihrer Freunde) und auf dem sie alle Knoten besucht (sie will ja alle ihre Freunde ansprechen). Die oben gezeigte Abfolge Emil → Beat → Alicia → David → Caro ist so ein Pfad, und dies ist außerdem der einzige Pfad.

Das ist Informatik!

Es passiert im Leben häufig, dass man Dinge erst tun kann, wenn man die Voraussetzungen dafür geschaffen hat. Ein gutes Beispiel ist das Kochen: Spagetti sollte man erst dann ins Wasser geben, wenn man es zum Kochen gebracht und Salz hinein geschüttet hat. Die Bolognese sollte man erst dann auf die Spagetti geben, wenn die Spagetti gar sind. Aber ob man die Spagetti und die Bolognese nacheinander oder gleichzeitig kocht, ist egal (vorausgesetzt, man hat genug Kochstellen auf dem Herd). Die Party-Einladung und das Kochen kann man selbst planen. Bei Abläufen mit vielen Schritten und vielen Bedingungen verlieren Menschen aber leicht die Übersicht. Da hilft die Informatik. Computerprogramme für Ablaufplanung (englisch: Scheduling) werden beispielsweise in der Industrie eingesetzt. Komplexe Abläufe wie der Bau eines Autos können so möglichst effizient, also zum Beispiel ohne unnötiges Warten auf Bauteile, gestaltet werden. Damit spart man viel Geld, weil man kein großes Lager mehr benötigt und Maschinen möglichst pausenlos arbeiten können.



Puzzlesteine

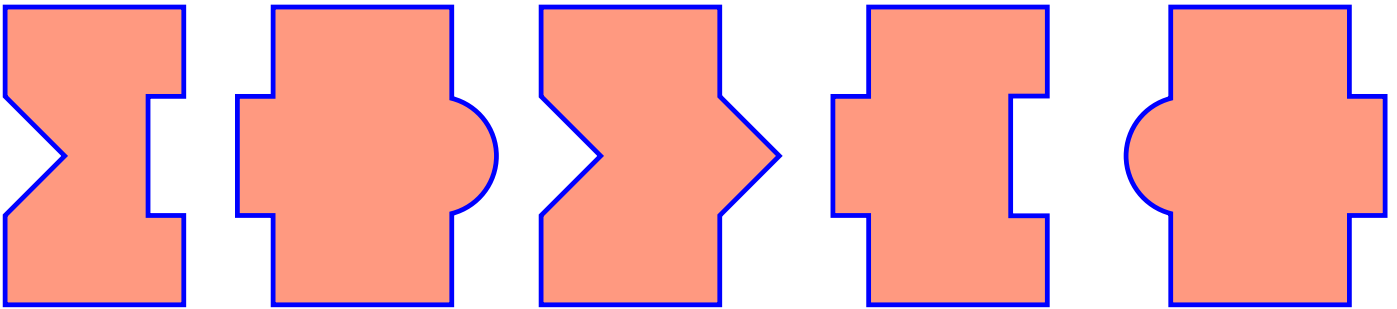
Die Biber haben ein neues Puzzle.

Die fünf Puzzlesteine haben an den Seiten besondere Formen.

Du kannst die Steine zu Pärchen zusammenschieben, wenn ihre Formen ineinander passen.

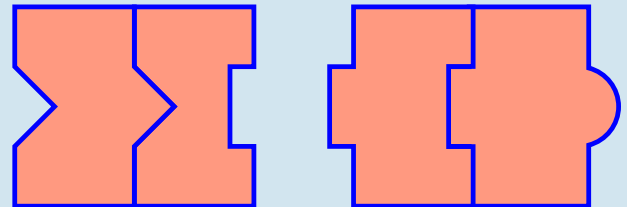
Wie viele Pärchen kannst du gleichzeitig bilden?

Bilde so viele Pärchen wie möglich!



So ist es richtig:

Diese beiden Pärchen kann man gleichzeitig bilden:



Aus fünf Steinen können höchstens zwei Pärchen gleichzeitig gebildet werden.

Es ist verführerisch, aus den beiden Steinen links sofort ein Pärchen zu bilden. Doch dann bleibt dieses das einzige Pärchen.

Das ist Informatik!

Viele Menschen machen Dinge gerne optimal: sie laufen so schnell wie möglich, suchen den kürzesten Weg von einem Ort zum anderen, verdienen möglichst viel Geld – oder bilden so viele Puzzlestein-Pärchen wie möglich. Weil es bei den Puzzlesteinen nicht so viele Möglichkeiten gibt, Pärchen zu bilden, kann man in Ruhe alle ausprobieren.

Wenn Computer für die Suche nach optimalen Lösungen verwendet werden, müssen sie meistens viele Daten verarbeiten. Dann würde die Methode „alles ausprobieren“ viel zu lange dauern. Die Informatik kennt deshalb noch viele andere Methoden zur Suche nach optimalen Lösungen. Eine davon heißt „greedy“, auf Deutsch: gierig. Die Greedy-Methode führt immer denjenigen nächsten Schritt aus, der am meisten Zugewinn bei der Lösung verspricht. Bei den Puzzlesteinen würde die Greedy-Methode die beiden Steine links zusammenschieben.

Das schöne an der Informatik ist, dass es meistens nichts nutzt, gierig zu sein: die optimale Lösung wird „greedy“ nicht gefunden. Das ist auch bei den Puzzlesteinen so: Man muss alle sinnvollen Möglichkeiten betrachten, um zwei Pärchen bilden zu können.



3-4: –

5-6: leicht

7-8: –

9-10: –

11-13: –



Regel-Regal

Beatrix schafft Ordnung auf ihrem Regal.

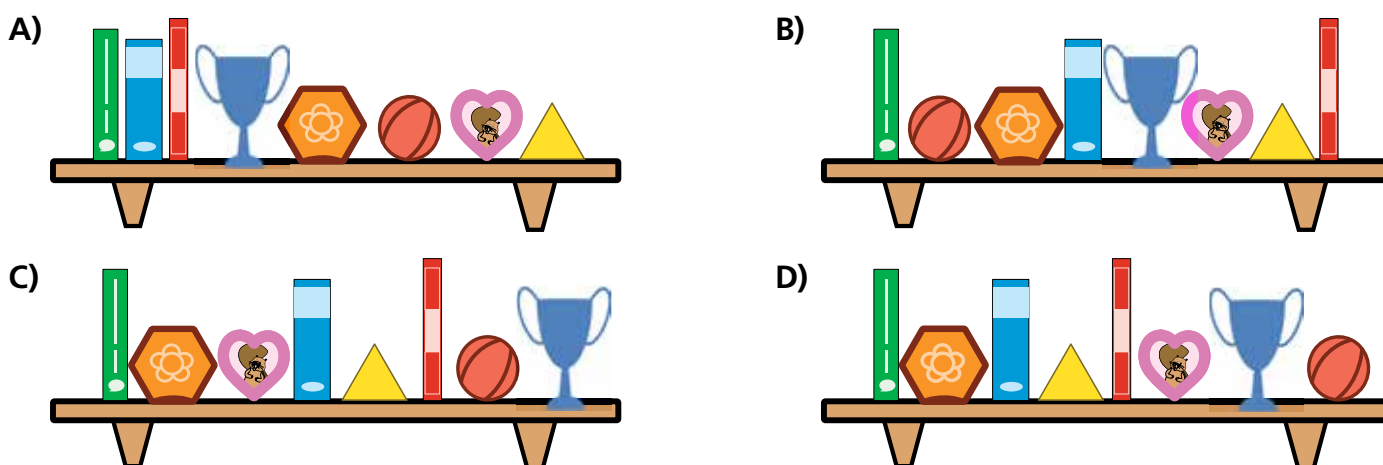
Ihre Ordnung hält sich an zwei Regeln:

1. Rechteckige Gegenstände dürfen **NICHT** nebeneinander sein.
2. Kugelförmige Gegenstände dürfen **NICHT** neben rechteckigen sein.

Die Bilder zeigen Regale mit verschiedenen Ordnungen.

Nur auf einem Regal hält sich die Ordnung an beide Regeln zugleich.

Das ist das Regel-Regal von Beatrix.



Antwort D ist richtig:

Regal A hält sich NICHT an Regel 1: Ganz links sind drei rechteckige Gegenstände nebeneinander.

Regal A hält sich nur an Regel 2: Es sind keine kugelförmigen Gegenstände neben rechteckigen.

Regal B hält sich nur an Regel 1: Es sind keine rechteckigen Gegenstände nebeneinander.

Regal B hält sich NICHT an Regel 2: Ganz links ist ein kugelförmiger Gegenstand neben einem rechteckigen.

Regal C hält sich nur an Regel 1: Es sind keine rechteckigen Gegenstände nebeneinander.

Regal C hält sich NICHT an Regel 2: Halbrechts ist ein kugelförmiger Gegenstand neben einem rechteckigen.

Nur Regal D hält sich an beide Regeln: Das ist das Regel-Regal von Beatrix.

Es sind keine rechteckigen Gegenstände nebeneinander und es sind keine kugelförmigen Gegenstände neben rechteckigen.

Es gibt noch weitere Gegenstände mit den Formen sechseckig, dreieckig, herzförmig und pokalförmig.

Zu diesen gibt es keine Regeln. Also können sie auch gegen keine Regeln verstoßen, egal wo sie im Regal sind.

Das ist Informatik!

Es gibt viele Arten von Regeln; unter anderem können Regeln Strukturen oder Abläufe festlegen.

Wie bei einer Verkehrsampel: Sie hat drei Lichter; oben ein rotes, in der Mitte ein gelbes, unten ein grünes. In vielen Ländern (aber nicht in allen!) leuchten die Lichter in dieser Folge: rot, rot und gelb, grün, gelb, rot, usw. Eine Ampel, die anders funktioniert, würde in Ländern mit diesen Ampel-Regeln zu Unfällen führen.

Regeln für Strukturen oder Abläufe sind in der Informatik sehr wichtig. Ein Beispiel: Computerprogramme sind darauf angewiesen, dass die Daten, die sie bearbeiten, einer vorgegebenen „Syntax“ folgen, also einer Menge von Regeln. In der Informatik ist sehr gut erforscht, mit welcher Art von Syntax Computer gut zurechtkommen. Eine Syntax mit sauber gepaarten Klammern etwa macht Computern die Arbeit leicht. Deshalb schreibt u.a. die Syntax der Sprache HTML (Hypertext Markup Language), in der die Dokumente des World-Wide Web (WWW) geschrieben sind, für (fast) alle Sprachelemente öffnende und schließende Klammern vor.



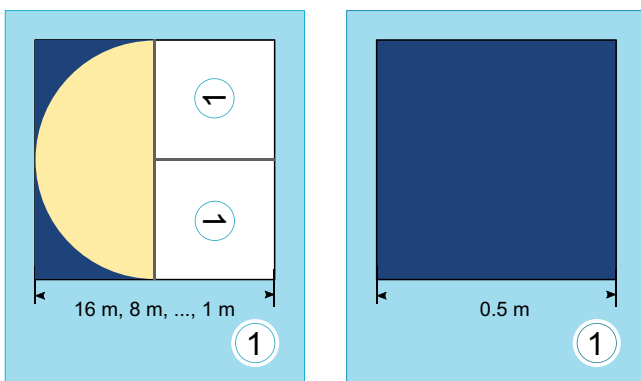
Rekursive Malerei

Tina und Ben helfen bei der Vorbereitung einer Sonderausstellung im Informatik-Museum. Auf den Boden eines Ausstellungsraums sollen sie ein 16 x 16 Meter großes Bild malen. Vom Künstler bekommen sie einen Satz Malanweisungskarten in dessen berühmter Malkartensprache, mit Hinweisen zu den Bildelementen, Maßen und Drehungen. Auf manchen Malanweisungskarten sind nummerierte Felder, die auf andere Malanweisungskarten verweisen.

Hier ein Beispiel aus einem früheren Malkartenprojekt. Wenn man diese drei Malanweisungskarten richtig ausführt, entsteht ein Bild des Bibers:

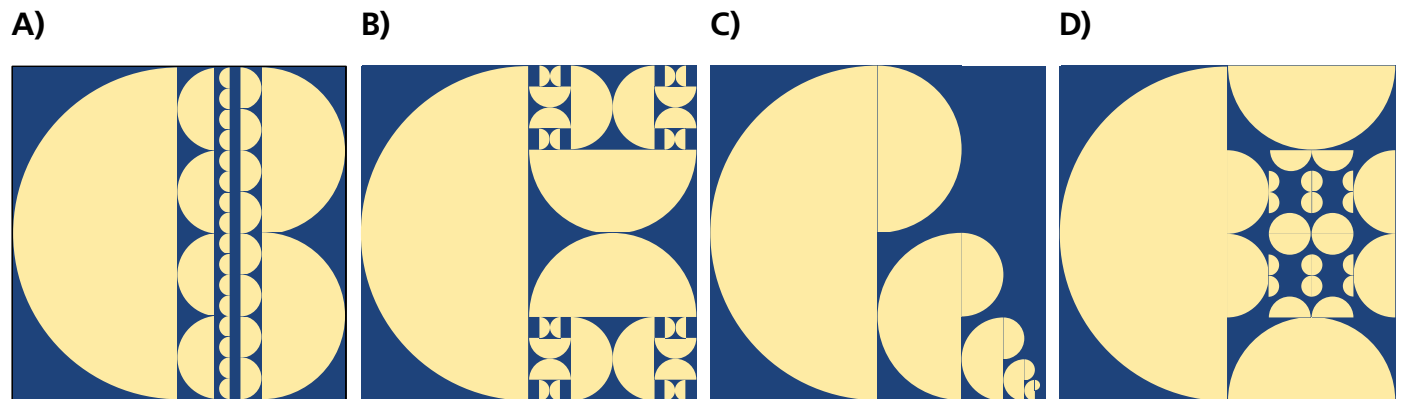


Für die Sonderausstellung bekommen Tina und Ben nun die folgenden zwei Malanweisungskarten:



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

Wie wird der Boden des Ausstellungsraums aussehen?



Antwort B ist richtig:

Die linke Malanweisungskarte besagt, dass die linke Hälfte des Bodens mit einer Halbkreisfläche gefüllt werden soll, deren runde Seite nach links zeigt. Für die rechte Hälfte soll dieselbe Malanweisungskarte zwei Mal verwendet werden. Die Orientierungen der Bildelemente muss den Orientierungen der Einsen entsprechen.

Bei dem einen Bildelement ist die Eins um 90° nach links gedreht. Deshalb muss das Bildelement ebenfalls nach links gedreht sein und die Rundung der Halbkreisfläche zeigt nach innen.

Bei dem anderen Bildelement ist die Eins um 90° nach rechts gedreht. Deshalb muss das Bildelement ebenfalls nach rechts gedreht sein und auch die Rundung dieser Halbkreisfläche zeigt nach innen.

Das ist allein bei Antwort B der Fall.

Das ist Informatik!

In der Informatik werden Anweisungen, die sich selbst aufrufen, als „rekursiv“ bezeichnet. Der Begriff kommt von lateinisch „recurrere“ (deutsch: zurücklaufen). Rekursion ist ein mächtiges Konzept.

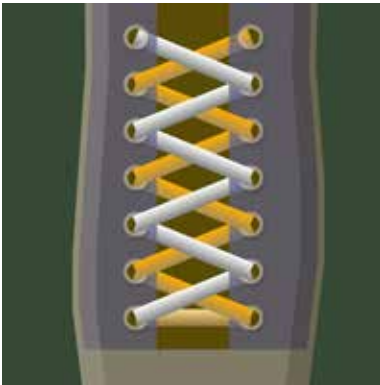
Für manche komplexen Aufgaben kann man kurz und überschaubar eine rekursive Anweisung zu ihrer Lösung formulieren.

Eine rekursive Anweisung muss eine Bedingung enthalten, die festlegt, wann die Rekursion abgebrochen werden soll. Sonst arbeitet die Rekursion solange weiter, bis irgendeine Ressource erschöpft ist. Etwa der Computerspeicher oder die Geduld des Benutzers.



Schuhbänder

Falco liebt seine Schuhbänder, die zur Hälfte orange und zur Hälfte weiß sind. Er lässt sich immer neue, komplizierte Schnürungen einfallen und findet sich damit sehr cool. Damit er keine Schnürung vergisst, schreibt er auf, wie er sie hergestellt hat. Dazu verwendet er eine Programmiersprache, die genau sagt, wie das Band durch die Ösen laufen soll.



Hier ist eines von Falcos Schnürungs-Programmen: (6 Zeilen lang):

```
01 orange: von_unten
02 weiss:  von_unten
03 {
04 orange: hoch wechseln von_unten
05 weiss:  hoch wechseln von_unten
06 }
```

Dieses Programm erzeugt die Schnürung im Bild.

Wie bei jeder Schnürung geht es bei den Ösen in der untersten Reihe los, mit weiß auf der linken Seite und orange auf der rechten Seite.

Für die Schnürungs-Programmiersprache hat Falco sich insgesamt folgende Befehle und Strukturen überlegt:

orange:	Das Folgende gilt für die orange Hälfte des Schuhbands.
weiss:	Das Folgende gilt für die weiße Hälfte des Schuhbands.
von_unten	Das Band wird von unten durch die aktuelle Öse gesteckt.
von_oben	Das Band wird von oben durch die aktuelle Öse gesteckt.
hoch	Das Band geht eine Öse nach oben (gleiche Seite, nächste Reihe).
wechseln	Das Band geht zur Öse auf der anderen Seite, in der gleichen Reihe.
{ . . . }	Was zwischen den Klammern steht, wird so oft wie möglich wiederholt.
*N { . . . }	Was zwischen den Klammern steht, wird genau N-mal wiederholt.



3-4: –

5-6: –

7-8: –

9-10: –

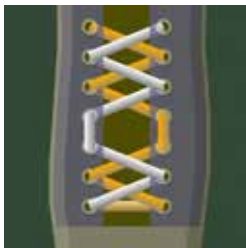
11-13: schwer



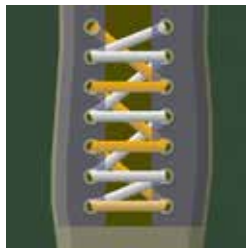
Welche Schnürung erzeugt dieses Programm? (Es ist 12 Zeilen lang)

```
01 orange: von_unten
02 weiss: von_unten
03 *2 {
04 orange: hoch wechseln von_unten
05 weiss: hoch wechseln von_unten
06 }
07 orange: hoch von_oben
08 weiss: hoch von_oben
09 {
10 orange: hoch wechseln von_unten
11 weiss: hoch wechseln von_unten
12 }
```

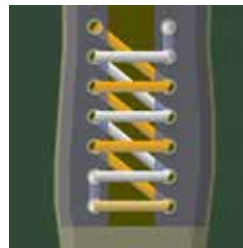
A)



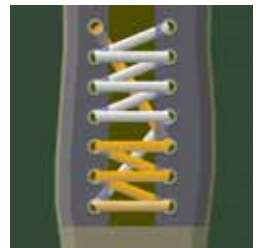
B)



C)



D)



Antwort A ist richtig:

Das Programm funktioniert für die unteren drei Ösen wie im Beispiel (Zeilen 1 bis 6).

Dann wandern beide Hälften des Bandes ohne Seitenwechsel eine Öse hoch (Zeilen 7 bis 8).

So entsteht das typische Auge, das nur bei Antwort A zu sehen ist.

Danach geht es weiter wie im einfachen Beispiel.

Unter www.coding4you.at/schuhband/ ist ein Programm installiert, mit dem man Schnürungen entwerfen kann.

Das ist Informatik!

Die Schnürungs-Programmiersprache bietet Strukturen, wie sie in fast jeder Programmiersprache vorkommen:

- + Ausführen einer Folge von Befehlen nacheinander (Sequenz, englisch „sequence“).
- + Wiederholen von Befehlsfolgen (Schleife, englisch „loop“).

Andere gebräuchliche Strukturen fehlen, weil sie für die Schnürungen nicht gebraucht werden:

- + Benutzung von Variablen.
- + Bedingte Verzweigung (englisch „case“, „if-then-else“).
- + Bilden benannter, parametrisierter Programmstücke (englisch „function“, „procedure“, „object“, „module“).

In der Informatik kommt es oft vor, dass auf Grundlage einer universellen Programmiersprache mit allen genannten Strukturen für eine bestimmte Anwendung eine einfachere Sprache mit passenden Befehlen (hier sind es acht) bereitgestellt wird. Die in der Anwendungssprache geschriebenen Programme sind dann aber im Allgemeinen nicht für andere Zwecke brauchbar. Es wäre wohl erfolglos, mit unserer Schnürungs-Programmiersprache einen industriellen Schweißroboter zu programmieren.



Spielernummern

Die Bilder zeigen zwei Teams mit je 15 Spielern.

Die Trikots der Spieler sind mit den Spielernummern bedruckt.

Die Spieler von Team 1 stehen nach ihren Spielernummern sortiert:



Die Spieler von Team 2 stehen nicht sortiert:



Nun erhältst du den Auftrag festzustellen, wie viele Spielernummern sowohl in Team 1 als auch in Team 2 vorkommen.

Wie kannst du den Auftrag möglichst schnell erledigen?

- A) Du durchläufst Team 1 von links nach rechts. Bei jeder Spielernummer (1, 4, 5, ...) schaust du bei Team 2 nach, ob die Spielernummer auch dort vorkommt.
- B) Du durchläufst Team 2 von links nach rechts. Bei jeder Spielernummer (8, 28, 12, ...) schaust du bei Team 1 nach, ob die Spielernummer auch dort vorkommt.
- C) Es ist egal, welches Team du durchläufst und bei welchem du nachschaust. Beides geht gleich schnell.
- D) Zuerst stellst du fest, wie viele Spielernummern nicht in beiden Teams vorkommen. Dann ziehst du diese Zahl von 15 ab und erhältst die gesuchte Zahl.



3-4: –

5-6: –

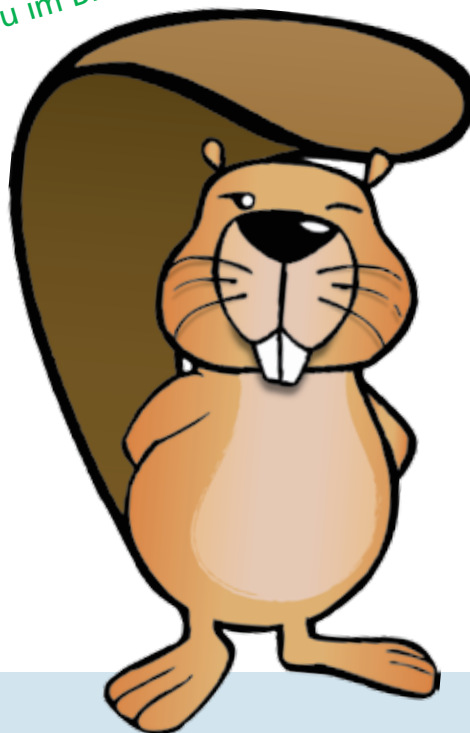
7-8: –

9-10: mittel

11-13: leicht



Auch Informatik ist ein Teamsport.
Beim Biber der Informatik dürfen Zweierteams mitmachen.
Solche Biber-Teams kannst du im Biber-Video sehen:
youtu.be/oHITRebIXDA

**Antwort B ist richtig:**

Durchläufst du Team 2, kannst du jede Spielernummer in der sortierten Folge der Spielernummern von Team 1 nachschauen. In einer sortierten Folge findet man Werte schneller als in einer nicht sortierten Folge. Dafür gibt es mehrere Gründe. Zum Beispiel kann man die Suche abbrechen, sobald man dem ersten Wert begegnet, der gemäß der Sortierung hinter dem gesuchten Wert liegt (bei Zahlen: der größer ist als die gesuchte Zahl). Der gesuchte Wert kann dann in der sortierten Folge nicht mehr vorkommen. Die Spielernummern in der nicht sortierten Folge von Team 1 nachzuschauen würde also länger dauern. Deshalb dauert die in Antwort A vorgeschlagene Methode länger als die von Antwort B, und deshalb ist auch Antwort C falsch.

Bei der Methode von Antwort D geht die Suche nach nicht doppelt vorkommenden Spielernummern etwa gleich schnell wie die Suche nach doppelt vorkommenden Spielernummern von Antwort B (wenn du Team 2 durchläufst und im sortierten Team 1 nachschaust). Aber die Zeit für die Subtraktion kommt hinzu.

Das ist Informatik!

Wie lässt sich eine sortierte Folge besonders schnell durchsuchen? Die Informatik kennt die Antwort auf diese Frage. Nimm an, du suchst z.B. die Nummer 9 auf den Trikots von Team 1. Du schaust zuerst auf den mittleren Spieler. Er hat die Nummer 17. Die gesuchte Nummer 9 ist kleiner. Also suchst du in der linken Hälfte der Mannschaft weiter. In deren Mitte steht Nummer 7. Also suchst du rechts von 7 weiter und hast die Nummer 9 rasch gefunden. Der Trick ist, dass du die zu durchsuchende Folge bei jedem Schritt in zwei Teile teilst und dann nur noch in einem der Teile weitersuchst – indem du ihn wieder in zwei Teile teilst, usw. Dieses Verfahren nennt die Informatik binäre Suche. „Binär“ kommt vom lateinischen Wort „bini“, das bedeutet „je zwei“. Mit binärer Suche kann man eine Folge mit z.B. einer Million Werten in 20 Schritten durchsuchen – statt in bis zu einer Million Schritten, die man bräuchte, wenn man einfach von vorne nach hinten sucht.

Das Durchsuchen von Datenmengen ist eine der häufigsten Aufgaben von Computern. Deshalb kennt die Informatik viele Verfahren zum Sortieren und Suchen – und weiß genau, welche Verfahren unter welchen Bedingungen am schnellsten sind.



Trambahn (трамвай)

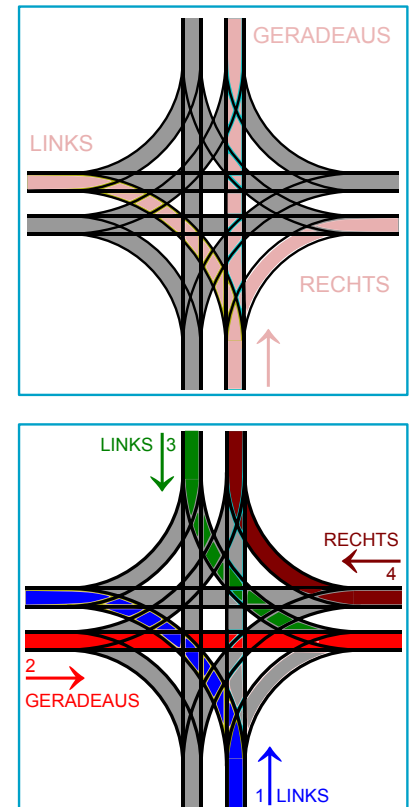
Auf den Straßen von Sankt Petersburg (Russland) fährt die Trambahn. An einer Trambahnkreuzung dort ist alles möglich: Die Trambahnen können aus allen vier Richtungen herkommen und auf alle möglichen Arten weiterfahren. Die Stellung der Weichen bestimmt, wie die Trambahnen weiterfahren. Die Weichenstellung wird so beschrieben: Im Uhrzeigersinn wird für jede Herkunftsrichtung angegeben, wie die Trambahnen weiterfahren: GERADEAUS, nach RECHTS oder nach LINKS.

Ein Beispiel: LINKS-GERADEAUS-LINKS-RECHTS bedeutet, dass die Trambahnen aus Richtung 1 nach links abbiegen, die Trambahnen aus der im Uhrzeigersinn nächsten Richtung 2 geradeaus fahren, und so weiter.

Allerdings kann es bei dieser Weichenstellung leicht zu Zusammenstößen kommen: von Bahnen aus den Richtungen 1 und 2, oder von Bahnen aus den Richtungen 2 und 3.

Auch bei einer der folgenden Weichenstellungen kann es leicht zu Zusammenstößen kommen. Bei welcher?

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



Antwort B ist richtig:

Trambahnen aus Richtung 1 (die nach rechts abbiegen) können mit Trambahnen aus Richtung 3 (die nach links abbiegen) zusammenstoßen. Ebenso können Trambahnen aus Richtung 2 (die nach rechts abbiegen) mit Trambahnen aus Richtung 4 (die nach links abbiegen).

Bei den anderen Weichenstellungen fahren die Trambahnen kollisionsfrei aneinander vorbei:

Bei Weichenstellung A fahren alle Trambahnen nur in den äußeren Kurven.

Bei den Weichenstellungen C und D fahren immer zwei Trams in parallelen Kurven aneinander vorbei. Die links abbiegenden Trambahnen in den inneren Kurven stören sich nicht. Das nennt man „tangenciales“ oder auch „amerikanisches“ Linksabbiegen.

Das ist Informatik!

Die Trambahnschienen – vor allem im Bereich von Kreuzungen – sind ein Beispiel für Ressourcen, die von mehreren Akteuren gemeinsam genutzt werden. Die Koordination und Optimierung der gemeinsamen Nutzung von Ressourcen durch parallel laufende Prozesse ist ein wichtiges Gebiet der Informatik. Wie bei der Trambahn muss man auch in der Digitaltechnik potenzielle Risiken beachten und Regeln für den sicheren Zugriff auf Datenbestände oder gemeinsam genutzte Geräte finden. Sonst droht der Verlust von Daten oder die Zerstörung von Hardware.

Im Fall der Trambahnkreuzung ist es eine gute Idee, nur sichere Weichenstellungen zuzulassen. Oder man sichert die Schienenstücke, auf denen ein Zusammenstoß passieren kann, durch Signale ab.

Die Informatik kennt ebenfalls Signale: sogenannte „Semaphoren“ (engl. für „Signal“) zeigen an, dass bestimmte Ressourcen belegt sind, und schützen sie vor kritischer gleichzeitiger Nutzung.



3-4: schwer

5-6: mittel

7-8: –

9-10: –

11-13: –



Vegetarische Schnitzeljagd

Biber Kati möchte einen Bibersalat machen. Kati muss fünf Zutaten in die Schüssel tun.

Im Garten findet Kati die Zutaten. Bei jeder Zutat zeigt ein Schild, welche Zutat sie als nächste in die Schüssel tun muss. Zum Beispiel muss sie nach dem Apfel den Mais in die Schüssel tun.



Welche Zutat muss Kate als erste in die Schüssel tun?

A)



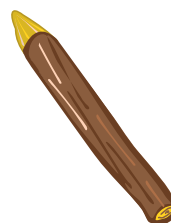
B)



C)



D)



E)



Antwort E ist richtig:

Kate muss als erste Zutat die Rübe  in die Schüssel tun. Die erste Zutat kann nur diejenige sein, die nicht auf einem Schild steht. Denn dann gibt es keine Zutat, die vor ihr in die Schüssel muss. Wenn beispielsweise der Apfel  die erste Zutat wäre, dürfte der Apfel nicht auf dem Schild bei der Rübe  stehen.

Das ist Informatik!

Zu dieser Biberaufgabe passt eine in der Informatik sehr gebräuchliche Datenstruktur. Sie ist leicht zu verstehen und wird in der bilderreichen Fachsprache der Informatik „einfach verkettete Liste“ genannt. Eine solche Liste hat Listenelemente. Hier sind es fünf, je eines zur Beschreibung einer Bibersalat-Zutat. Zur Festlegung ihrer Reihenfolge sind die Listenelemente vom ersten bis zum letzten aneinander gekettet. Dazu hat (fast) jedes Element einen Zeiger (hier: ein Holzschild) auf seinen Nachfolger in der Liste. Das letzte Element in der Liste hat keinen Nachfolger, also keinen Zeiger, also kein Holzschild. Das Listenelement, auf das in einer einfach verketteten Liste kein Nachfolger-Zeiger zeigt, muss in dieser Liste das erste sein.



Wer gewinnt?

Julie bekommt diese Werbung mit einer E-Mail:



Sie will unbedingt auch gewinnen. In der Schule hat sie aber gehört, dass nur Erwachsene im Glücksspiel mitmachen dürfen, dass man nur selten gewinnt und dass obendrein die persönlichen Daten der Spieler missbraucht werden können.

Was soll sie machen?

- A) Sich als die eigene Mutter ausgeben und mit deren persönlichen Daten teilnehmen?
- B) Einfach doch mit den eigenen persönlichen Daten teilnehmen?
- C) Die E-Mail löschen?
- D) Den älteren Bruder bitten, mit seinen persönlichen Daten teilzunehmen?

Antwort C ist richtig:

Julie sollte die E-Mail sofort löschen, es ist Spam. Bei den Antworten A, B und D müsste sie vor dem Teilnehmen persönliche Daten preisgeben. Diese Daten können dann für unlautere Zwecke missbraucht werden. Zum Beispiel zum gezielten Versenden von noch mehr Spam – aber auch Schlimmerem.

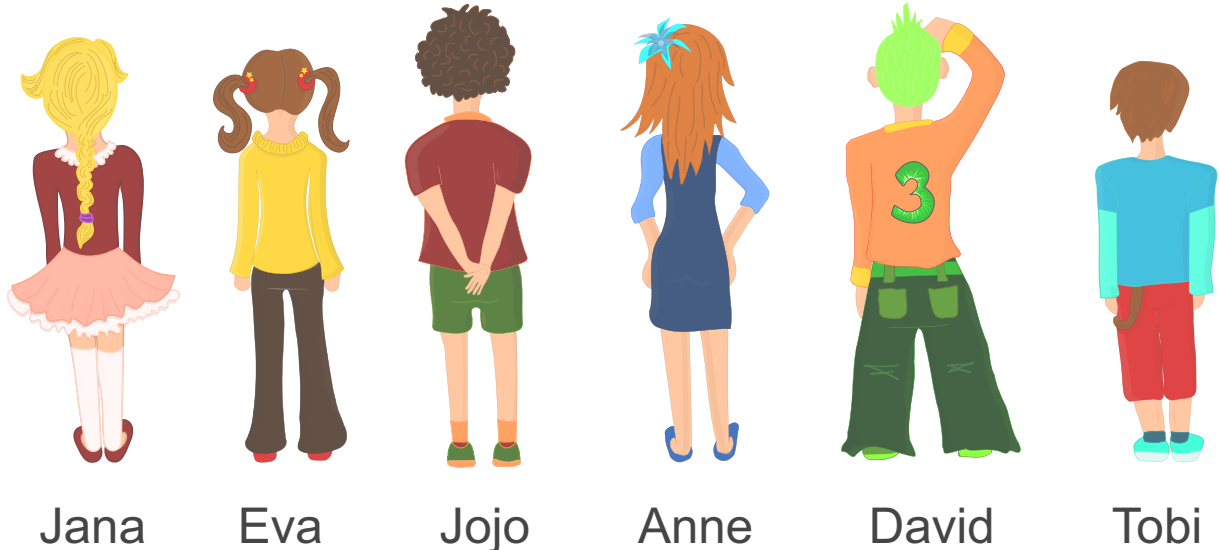
Das ist Informatik!

Spam ist teuer. Weil Millionen Benutzer jeden Tag Arbeitszeit mit dem Lesen und Löschen von Spam verplempern, entsteht ein beträchtlicher volkswirtschaftlicher Schaden. Spam ist ein Massenmedium zur Verbreitung von Schadsoftware. Viele Benutzer bemerken nicht, dass ihre Computer bereits per Fernsteuerung zum Versenden von Spam missbraucht werden. Nur weil sie einmal auf eine Spam-Mail geklickt haben, aber keine aktuelle Schutzsoftware auf dem Computer installiert ist.

Die Informatik bemüht sich nicht nur um die Verbesserung von automatischen Spam-Filtern und anderer Schutzsoftware, sondern auch um die Aufklärung der Bevölkerung über die Risiken und Vorsorge-Pflichten ihrer Teilnahme am Internet. Alle Jugendlichen sollten zum Beispiel frühzeitig erfahren, dass es ihnen aus gutem Grund verboten ist, an Internet-Glücksspielen teilzunehmen. Warum? Weil es dumm ist. Man kann dort der fairen Zufälligkeit des Spielablaufs nicht vertrauen und wird mit Sicherheit betrogen.

Wer war's?

Sechs Kinder spielen draußen mit dem Ball.



Es scheppert und klirrt! Ich laufe zum Fenster und sehe durch die zerbrochene Scheibe ein Kind wegrennen. Es hat lange Haare und trägt eine lange Hose.

Wer war's?

A) Jana B) Eva C) Jojo D) Anne E) David F) Tobi

Antwort B ist richtig:

Drei Kinder haben lange Haare: Jana, Eva und Anne.

Drei Kinder tragen lange Hosen: Eva, David und Tobi.

Also muss es Eva sein, denn nur für sie trifft beides zu.

Das ist Informatik!

In dieser Biberaufgabe geht es darum, Informationen zu kategorisieren. Informationen mit gemeinsamen Attributen (Eigenschaften) können so tabellarisch in einer Datenbank gespeichert werden. Mögliche gemeinsame Attribute der Kinder sind hier die Haarlänge, Haarfarbe, Farbe des Shirts, Länge der Hose, etc. Jedes Attribut hat für jedes Kind einen bestimmten Wert. Zum Beispiel haben Jana und Eva beide eine Haarfarbe, aber Janas Haarfarbe hat den Wert „blond“ und Evas Haarfarbe hat den Wert „braun“.

In der Informatik ist es immer wieder eine Herausforderung an Fachwissen, Allgemeinwissen und Lebenserfahrung, für Datenbanken die relevanten Attribute festzulegen und mit diesen Attributen dann gewünschte Informationen aus der Datenbank zu erfragen (zu selektieren).

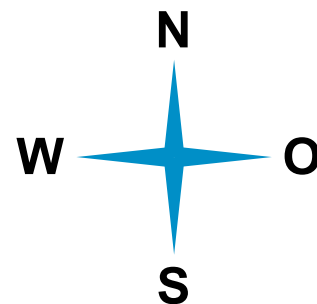
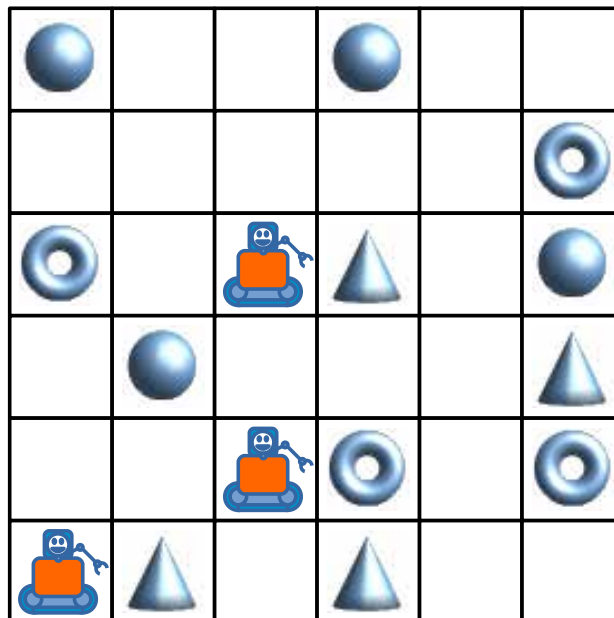


Zugleich

Drei Roboter arbeiten als Team zusammen. Du kannst das Team mit Richtungsbefehlen steuern: N, S, O oder W. Mit einem Richtungsbefehl steuerst du alle drei Roboter gleich: um ein Feld weiter in diese Richtung.

Du sollst die Roboter zu den Dingen steuern, die sie am Ende nehmen sollen. Damit sie nichts Falsches nehmen, musst du sie vorher um andere Dinge herum steuern.

Ein Beispiel: Du steuerst die Roboter mit diesen Befehlen: N, N, S, S, O.
Dann nehmen die Roboter am Ende zwei Kegel und einen Ring.



Die Roboter sollen einen Ball, einen Ring und einen Kegel nehmen.

Mit welchen Befehlen musst du sie steuern?

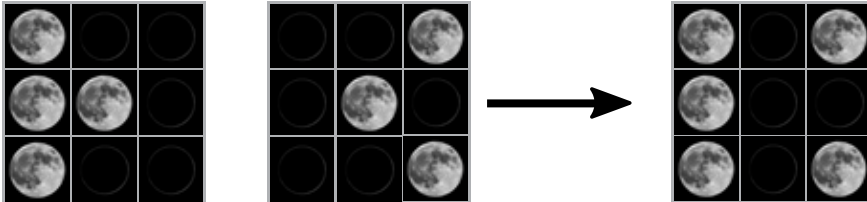
- A) N, O, O, O
- B) N, O, O, S, O
- C) N, N, S, O, N
- D) N, O, O, S, W



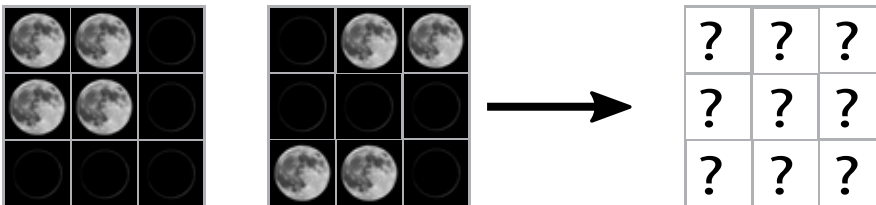
Zwei Möglichkeiten

Der weiße Anaxagoras besitzt Mondzauberkarten, die für jedes ihrer neun Felder nach einer festen Regel funktionieren. Bei zwei Karten gibt er in jedes Feld ein Mondsymbol ein: Neumond oder Vollmond. Sofort erscheinen auch Mondsymbole auf einer dritten Karte.

Hier ein Beispiel, links die zwei Eingabekarten, rechts die Ausgabekarte:

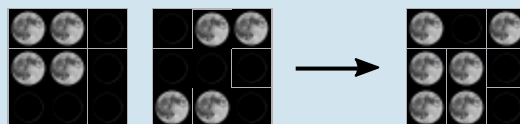


Verstehst Du die Regel? Dann klicke auf die Felder der Ausgabekarte rechts, bis alle das richtige Mondsymbol zeigen.



So ist es richtig:

Die Regel ist: Wenn auf den beiden Eingabekarten im gleichen Feld genau einmal Vollmond vorkommt, dann zeigt das entsprechende Feld der Ausgabekarte das Mondsymbol Vollmond, andernfalls das Mondsymbol Neumond.



Das ist Informatik!

In allen Bereichen der Informatik wird gerne zweiwertige Logik benutzt, um über die Wahrheit und Falschheit von Aussagen innerhalb des Modells einer Anwendung zu entscheiden. Das geschieht immer unter der Voraussetzung, dass es im Modell dafür genau zwei Möglichkeiten (zwei Werte) gibt. Eine dritte Möglichkeit ist ausgeschlossen: tertium non datur.

Einige Beispiele für solche Zwei-Werte-Mengen in Modellen sind: {zu auf}, {kalt heiß}, {ungerade gerade}, {leer voll}, {schwarz weiss}, {nein ja}, {0 1}, {Neumond Vollmond}. Auf einer solchen Grundlage kann man dann Aussagen innerhalb des Modells logisch miteinander verknüpfen. Eine logische Funktion liefert entweder den Wahrheitswert „falsch“ oder den Wahrheitswert „wahr“. Weil man logische Funktionen miteinander zu neuen logischen Funktionen verknüpfen kann, gibt es davon im Prinzip beliebig viele. In dieser Biberaufgabe benutzen wir die Funktion XOR („ausschließendes Oder“, englisch: „exclusive or“), weil sie besonders häufig vorkommt und leicht zu verstehen ist. XOR funktioniert so: Wenn von einer Anzahl Aussagen genau eine „wahr“ ist und alle anderen „falsch“, dann liefert XOR den Wahrheitswert „wahr“. In allen anderen Fällen liefert XOR den Wahrheitswert „falsch“. Im Beispiel zu dieser Aufgabe wird die logische Funktion XOR neunmal (Felder) mit zwei Aussagen (Eingabekarten) vorgeführt. Übrigens: Anaxagoras hat als erster erkannt, dass der Mond nicht von sich aus leuchtet. Nun trägt ein Mondkrater den Namen mit dem versteckten XOR.

BIBER DER INFORMATIK 2016



OESTERREICHISCHE[®]
COMPUTER GESELLSCHAFT
AUSTRIAN
COMPUTER SOCIETY