



INFORMATIK-BIBER SCHWEIZ CASTOR INFORMATIQUE SUISSE CASTORO INFORMATICO SVIZZERA

Aufgaben 2025

Schuljahre 11/12/13



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

Herausgeber:

Susanne Thut, Nora A. Escherle,
Jean-Philippe Pellet

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SV!A

www.svia-ssie-ssie.ch
schweizerischervereinfürinformatikind
erausbildung//société suisse pour l'infor
matique dans l'enseignement//società sviz
zeraper l'informaticanell'insegnamento





Mitarbeit Informatik-Biber 2025

Masjar Babazadeh, Jean-Philippe Pellet, Andrea Maria Schmid, Giovanni Serafini, Susanne Thut

Projektleitung: Nora A. Escherle

Herzlichen Dank für die Aufgabenentwicklung für den Schweizer Wettbewerb an:

Patricia Heckendorf, Gymnasium Kirschgarten

Juraj Hromkovič, Regula Lacher: ETH Zürich, Ausbildungs- und Beratungszentrum für Informatikunterricht

Jens Hartmann, Stephan Koch, Dirk Schmerenbeck und Jacqueline Staub: Universität Trier, Deutschland

Die Aufgabenauswahl wurde erstellt in Zusammenarbeit mit den Organisatoren von Bebras in Deutschland, Österreich und Ungarn. Besonders danken wir:

Philip Whittington, Silvan Horvath: ETH Zürich, Ausbildungs- und Beratungszentrum für Informatikunterricht

Wolfgang Pohl, Karsten Schulz, Franziska Kaltenberger, Margaretha Schlüter, Kirsten Schlüter, Michael Weigend: Bundesweite Informatikwettbewerbe (BWINF), Deutschland

Wilfried Baumann: Österreichische Computer Gesellschaft

Gerald Futschek, Lukas Lehner: Technische Universität Wien

Zsuzsa Pluhár, Bence Gaal: ELTE Informatikai Kar, Ungarn

Die Online-Version des Wettbewerbs wurde auf cuttle.org realisiert. Für die gute Zusammenarbeit danken wir:

Eljakim Schrijvers, Justina Oostendorp, Alieke Stijf, Kyra Willekes: cuttle.org, Niederlande

Andrew Csizmadia: Raspberry Pi Foundation, Vereinigtes Königreich

Die Programmieraufgaben wurden speziell für die Online-Plattform erstellt und entwickelt. Wir danken herzlich für die Initiative:

Jacqueline Staub: Universität Trier, Deutschland

Dirk Schmerenbeck: Universität Trier, Deutschland

Dave Oostendorp: cuttle.org, Niederlande

Für den Support während der Wettbewerbswochen danken wir:

Eveline Moor: Schweizer Verein für Informatik im Unterricht

Für die Organisation und Durchführung des Finales 2024 an der ETH danken wir:

Dennis Komm, Hans-Joachim Böckenhauer, Angélica Herrera Loyo, Andre Macejko, Moritz Stocker, Philip Whittington, Silvan Horvath: ETH Zürich, Ausbildungs- und Beratungszentrum für Informatikunterricht

Für die Korrektur der Finalaufgaben:

Clemens Bachmann, Morel Blaise, Tobias Boschung, Davud Evren, Jay Forrer, Sven Grübel, Urs Hauser, Fabian Heller, Jolanda Hofer, Alessandra Iacopino, Saskia Koller, Richard Královič, Jan



Mantsch, Adeline Pittet, Alexander Skodinis, Emanuel Skodinis, Jasmin Sudar, Valerie Verdan, Chris Wernke

Für die Übersetzung der Finalaufgaben ins Französische:

Jean-Philippe Pellet: Haute école pédagogique du canton de Vaud

Christoph Frei: Chragokyberneticks (Logo Informatik-Biber Schweiz)

Andrea Leu, Sarah Beyeler, Maggie Winter: Senarcens Leu + Partner AG

Ganz besonderen Dank gilt unseren grossen Förderern Juraj Hromkovič, Dennis Komm, Gabriel Parriaux und der Haslerstiftung. Ohne sie würde es diesen Wettbewerb nicht geben.

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

Die französischsprachige Übersetzung wurde von Elsa Pellet und die italienischsprachige Übersetzung von Christian Giang erstellt.



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Der Informatik-Biber 2025 wurde vom Schweizerischen Verein für Informatik in der Ausbildung (SVIA) durchgeführt und massgeblich und grosszügig von der Hasler Stiftung unterstützt. Weitere Partner*innen und Wettbewerbssponsoren, die den Wettbewerb finanziell unterstützt haben, sind die Abraxas Informatik AG, das Amt für Kindergarten, Volksschule und Beratung (AKVB) des Kantons Bern, Amt für Wirtschaft AWI des Kantons Zürich, die CYON AG sowie die UBS.

Folgende Akademischen Partner unterstützen uns bei der Aufgabenerstellung: ETH Zürich, Ausbildungs- und Beratungszentrum für Informatikunterricht, Haute école pédagogique du canton de Vaud, La Scuola universitaria professionale della Svizzera italiana, Pädagogische Hochschule Luzern und die Universität Trier.

Dieses Aufgabenheft wurde am 10. Dezember 2025 mit dem Textsatzsystem \LaTeX erstellt. Wir bedanken uns bei Christian Datzko für die Entwicklung und langjährige Pflege des Systems zum Generieren der 36 Versionen dieser Broschüre (nach Sprachen und Schulstufen). Das System wurde analog zum Vorgänger-System neu programmiert, welches ab 2014 gemeinsam mit Ivo Blöchliger entwickelt wurde. Jean-Philippe Pellet danken wir für die Entwicklung der `bebras` Toolchain, die seit 2020 für die automatisierte Konvertierung der Markdown- und YAML-Quelldokumente verwendet wird.

Hinweis: Alle Links wurden am 1. Dezember 2025 geprüft.



Die Aufgaben sind lizenziert unter einer Creative Commons Namensnennung – Nicht-kommerziell – Weitergabe unter gleichen Bedingungen 4.0 International Lizenz. Die Autoren sind auf S. 22 genannt.



Vorwort

Der Wettbewerb «Informatik-Biber», der in verschiedenen Ländern der Welt schon seit über 20 Jahren bestens etabliert ist, will das Interesse von Kindern und Jugendlichen an der Informatik wecken. Der Wettbewerb wird in der Schweiz auf Deutsch, Französisch und Italienisch vom Schweizerischen Verein für Informatik in der Ausbildung SVIA durchgeführt und von der Hasler Stiftung unterstützt.

Der Informatik-Biber ist der Schweizer Partner der Wettbewerbs-Initiative «Bebras International Challenge on Informatics and Computational Thinking» (<https://www.bebras.org/>), die in Litauen ins Leben gerufen wurde.

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

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

Der Informatik-Biber 2025 wurde in fünf Altersgruppen durchgeführt:

- Stufen 3 und 4
- Stufen 5 und 6
- Stufen 7 und 8
- Stufen 9 und 10
- Stufen 11 bis 13

Jede Altersgruppe erhält Aufgaben in drei Schwierigkeitsstufen: leicht, mittel und schwierig. In den Altersgruppen 3 und 4 waren 9 Aufgaben zu lösen, mit je drei Aufgaben in jeder der drei Schwierigkeitsstufen. Für die Altersklassen 5 und 6 waren es je vier Aufgaben aus jeder Schwierigkeitsstufe, also 12 insgesamt. Für die restlichen Altersklassen waren es 15 Aufgaben, also fünf Aufgaben pro Schwierigkeitsstufe.

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

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



Dieses international angewandte System zur Punkteverteilung soll den Anreiz zum blossen Erraten der Lösung eliminieren.

Jede Teilnehmerin und jeder Teilnehmer hatte zu Beginn 45 Punkte (Stufen 3 und 4: 27 Punkte; Stufen 5 und 6: 36 Punkte) auf dem Punktekonto.

Damit waren maximal 180 Punkte (Stufen 3 und 4: 108 Punkte; Stufen 5 und 6: 144 Punkte) zu erreichen, das minimale Ergebnis betrug 0 Punkte.

Bei vielen Aufgaben wurden die Antwortalternativen am Bildschirm in zufälliger Reihenfolge angezeigt. Manche Aufgaben wurden in mehreren Altersgruppen gestellt. Diese Aufgaben hatten folglich in den verschiedenen Altersgruppen unterschiedliche Schwierigkeitsstufen.

Einige Aufgaben werden für bestimmte Altersgruppen als «Bonus» angegeben: sie haben keinen Einfluss auf die Berechnung der Gesamtpunktzahl. Diese Übungen dienen vielmehr dazu, bei mehreren TeilnehmerInnen mit identischer Punktzahl zu entscheiden, wer sich für eine mögliche nächste Runde qualifiziert.

Für weitere Informationen:

Schweizerischer Verein für Informatik in der Ausbildung

SVIA-SSIE-SSII

Informatik-Biber

Nora A. Escherle

<https://www.informatik-biber.ch/kontaktieren/>

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



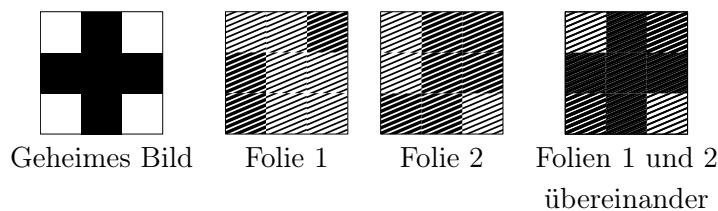
Inhaltsverzeichnis

Mitarbeit Informatik-Biber 2025	i
Vorwort	iv
Inhaltsverzeichnis	vi
1. Kurierdienst	1
2. Der Drachen ist weg!	2
3. Brennende Kerzen	3
4. Helligkeitskarte	4
5. Mehltransport	5
6. Schwarz - Weiss	6
7. Biber-Mail-Adressen	7
8. Biberone	9
9. ÖV	10
10. Seoul entdecken!	11
11. Bergseen	12
12. Parkplätze	13
13. Biber Jones	14
14. Prüfungsplan	16
15. Prüf-Biber	17
16. Schere, Stein, Papier	18
17. Hin und Her	21
A. Aufgabenautoren	22
B. Akademische Partner	24
C. Sponsoring	25



1. Kurierdienst

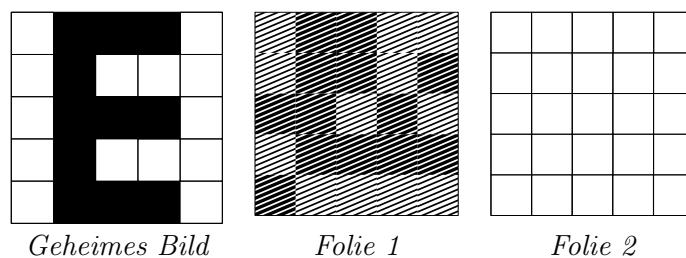
Ein geheimes Bild, das aus schwarzen und weissen Pixeln besteht, soll sicher übertragen werden. Hierfür erstellt der Kurierdienst auf transparenten Folien zwei Bilder aus dunklen und hellen Pixeln. Das geheime Bild wird erst dann erkennbar, wenn die beiden Folien übereinander gelegt werden.



Die Bilder für die beiden Folien werden so erstellt: Zuerst wird für Folie 1 ein zufälliges Muster aus dunklen und hellen Pixeln erzeugt. Die Pixel im Bild für Folie 2 werden dann nach der folgenden Regel gesetzt, abhängig von den Pixeln an der gleichen Stelle im geheimen Bild und in Folie 1:

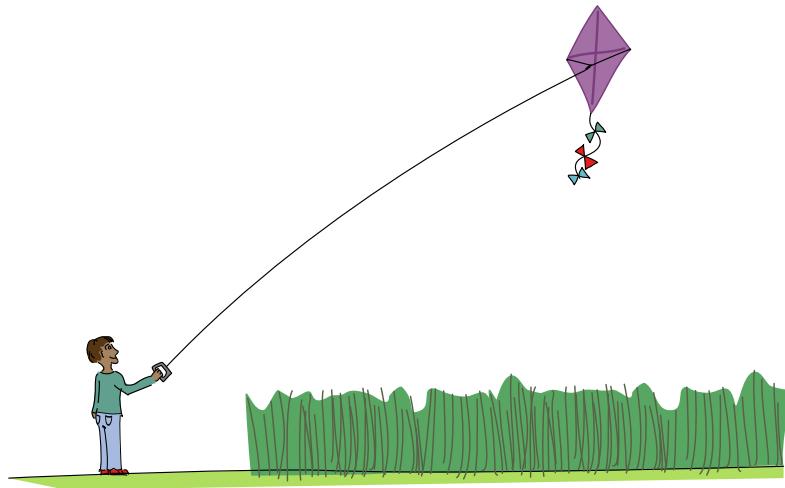
- Ist das Pixel im geheimen Bild schwarz , dann müssen die Pixel in Folie 1 und Folie 2 verschieden sein (das eine dunkel , das andere hell).
- Ist das Pixel im geheimen Bild weiss , dann müssen die Pixel in Folie 1 und Folie 2 gleich sein (beide oder beide).

Für das folgende geheime Bild wurde Folie 1 bereits erzeugt. Erstelle nun Folie 2.





2. Der Drachen ist weg!

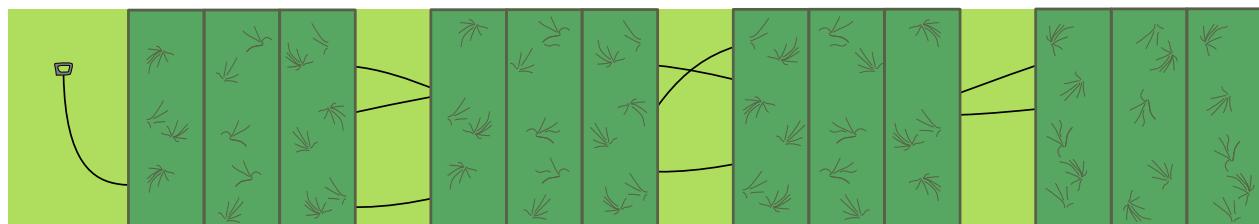


So ein Pech! Asterios hat seinen Drachen auf der Wiese verloren. Die Drachenschnur hat sich im hohen Gras verfangen, und es ist gar nicht so einfach, den Drachen wiederzufinden.

Die Wiese ist in 15 Felder unterteilt, die man einzeln durchsuchen kann.

Asterios hat schon 3 Felder der Wiese durchsucht. Er schaut sich genau an, wie die Schnur in diesen Feldern verläuft, und erkennt: Jetzt muss er nur noch ein weiteres Feld durchsuchen, um sicher zu wissen, wo der Drachen ist.

Welches Feld ist das?





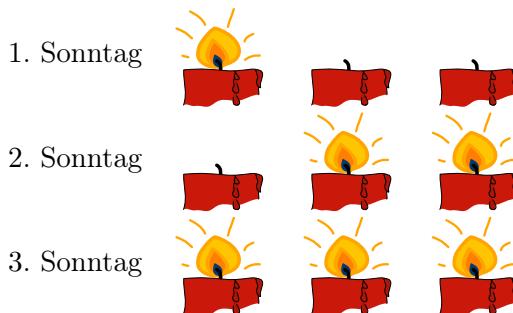
3. Brennende Kerzen

Es gibt eine Tradition, an den vier Sonntagen vor Weihnachten Kerzen anzuzünden: 1 Kerze am ersten Sonntag, 2 Kerzen am zweiten Sonntag und so weiter.

Chris liebt diese Tradition. Alle vier seiner Kerzen sind gleich lang. Chris' Weihnachtsfest wäre wunderschön, wenn auch nach dem letzten Sonntag alle Kerzen noch gleich lang wären. Dafür müsste er jede Kerze insgesamt gleich oft anzünden.

Leider findet Chris keine Möglichkeit, ein wunderschönes Weihnachtsfest zu haben.

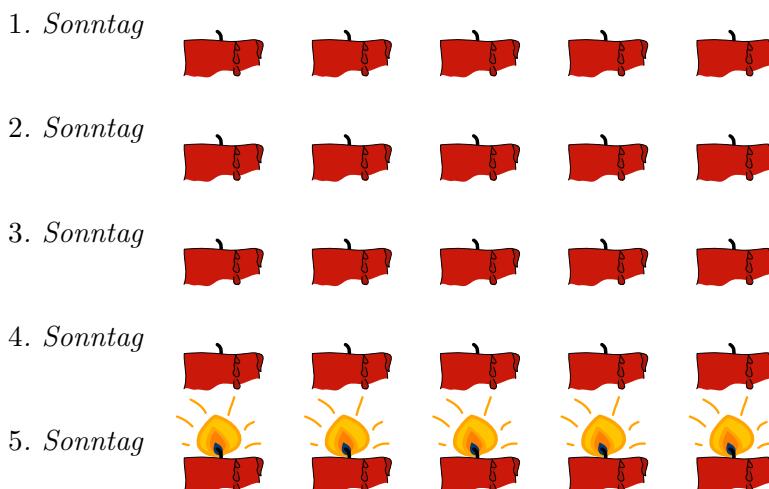
Wenn die Tradition nur drei Sonntage (und Kerzen) umfassen würde, wäre es möglich. Dann würde Chris jede Kerze genau zweimal anzünden:



Auch mit fünf Sonntagen (und Kerzen) wäre es möglich.

Zeige Chris, wie er jede Kerze gleich oft anzünden kann.

Für den fünften Sonntag haben wir die Kerzen bereits angezündet.



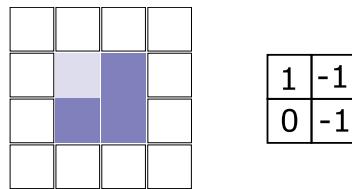


4. Helligkeitskarte

Digitale Bilder bestehen häufig aus Pixeln. Sandra erstellt Helligkeitskarten für solche Pixelbilder. Dazu legt sie um ein Bild zuerst einen Rahmen aus zusätzlichen weißen Pixeln. Dann bestimmt sie für jedes Pixel des Bildes einen Helligkeitswert, und zwar:

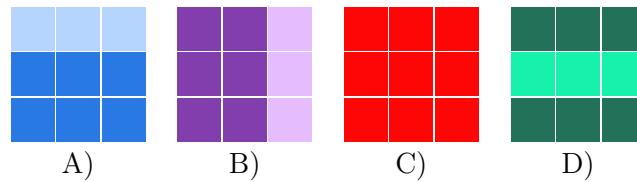
- | | |
|----|--|
| 1 | |
| 0 | |
| -1 | |
- 1, falls das Pixel heller ist als sein rechtes Nachbarpixel.
 - 0, falls das Pixel gleich hell ist wie sein rechtes Nachbarpixel.
 - 1, falls das Pixel dunkler ist als sein rechtes Nachbarpixel.

Hier siehst du ein Bild aus vier Pixeln (plus die zusätzlichen weißen Pixel) und seine Helligkeitskarte.



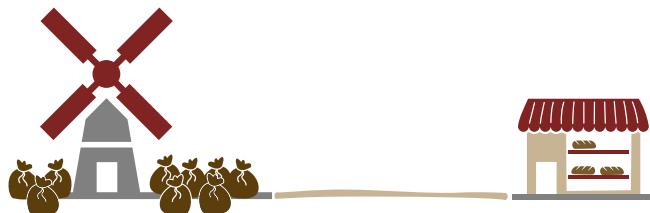
Unten siehst du vier Bilder mit je neun Pixeln. Genau drei davon haben die gleiche Helligkeitskarte.

Welches der Bilder hat als *einziges* eine **andere** Helligkeitskarte?





5. Mehltransport



Albert und Marco arbeiten in einer Bäckerei.

Sie müssen immer wieder Mehl von der Mühle holen gehen. Es darf aber immer nur einer gehen, damit der andere die Kunden bedienen kann.

Ihre Leistungen beim Mehl holen sind unterschiedlich:



Albert holt 13 kg Mehl in 1 Stunde.



Marco holt 5 kg Mehl in 30 Minuten.

Für jeden der beiden gilt aber: Nach dreimal Mehl holen sind 30 Minuten Erholung nötig. In dieser Zeit können Kunden bedient werden.

Albert und Marco möchten in 8 Stunden so viel Mehl wie möglich holen. Unter genau einer der folgenden Bedingungen schaffen sie das. Unter welcher?

- A) Albert muss zuerst gehen.
- B) Marco muss zuerst gehen.
- C) Marco muss zuletzt gehen.
- D) Albert darf nicht zuletzt gehen.
- E) Marco muss genau einmal gehen.



6. Schwarz - Weiss

Sarah möchte Folgen von schwarzen und weissen Kästchen mit Buchstaben beschreiben. Sie wendet dazu diesen Algorithmus auf eine Kästchen-Folge an:

- Wenn alle Kästchen der Folge weiss sind, schreibe **W**.
- Wenn alle Kästchen der Folge schwarz sind, schreibe **S**.
- Wenn die Folge schwarze und weisse Kästchen enthält, schreibe **x** und mache Folgendes:
 - Wende den Algorithmus auf die linke Hälfte der Folge an.
 - Wende den Algorithmus auf die rechte Hälfte der Folge an.

Hier siehst du für einige Kästchen-Folgen, welche Buchstaben-Beschreibung der Algorithmus ausgibt:

	W
	xWS
	xxSWS
	xSxWxSW

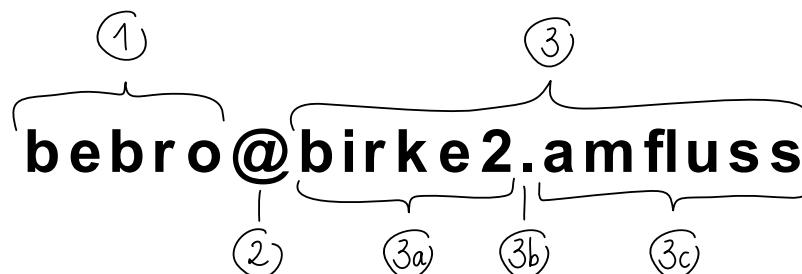
Welche Buchstaben-Beschreibung gibt Sarahs Algorithmus für diese Kästchen-Folge aus?





7. Biber-Mail-Adressen

Die IT-Biber brauchen ein System, das erkennt, ob eine Zeichenfolge eine Biber-Mail-Adresse ist. Eine Biber-Mail-Adresse besteht aus drei Teilen:



	Bezeichnung	Erläuterung	Zulässige Werte
①	Benutzername	eine beliebige, nicht leere Zeichenfolge aus Kleinbuchstaben und Ziffern	0–9, a–z
②	AT-Zeichen		@
③	Servername		
③a	Domänenname	eine beliebige, nicht leere Zeichenfolge aus Kleinbuchstaben und Ziffern	0–9, a–z
③b	Punkt		.
③c	Top-Level-Domänenname	eine beliebige, nicht leere Zeichenfolge aus Kleinbuchstaben und Ziffern	0–9, a–z

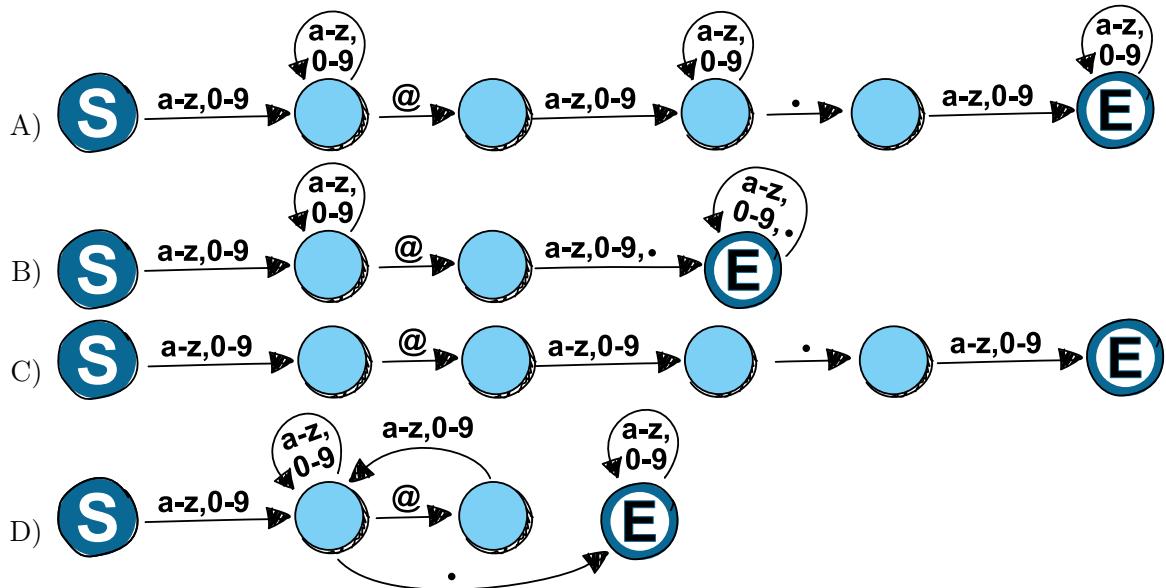
«Nicht leere» Zeichenfolgen bestehen aus mindestens einem Zeichen. «`bebro@birke2.amfluss`» ist ein Beispiel für eine Biber-Mail-Adresse.

Die IT-Biber überlegen sich vier Systeme und beschreiben sie mit Diagrammen aus Kreisen und Pfeilen. Ein Kreis steht für einen der Zustände, in denen das System sein kann. Ein Pfeil beschreibt den Wechsel zu einem nächsten, möglicherweise gleichen Zustand. Die Beschriftung des Pfeils sagt, zu welchen Zeichen dieser Zustandswechsel passt.

Ein System untersucht eine Zeichenfolge Zeichen für Zeichen, von links nach rechts, und ist zu Beginn im Zustand **S**. Der nächste Zustand hängt vom aktuell untersuchten Zeichen ab: Das System folgt dem Zustandswechsel, der zum aktuellen Zeichen passt. Ist das System nach dem letzten Zeichen im Zustand **E**, hat es die Zeichenfolge als Biber-Mail-Adresse erkannt.



Nur eines dieser Systeme erkennt alle möglichen Biber-Mail-Adressen korrekt. Welches?





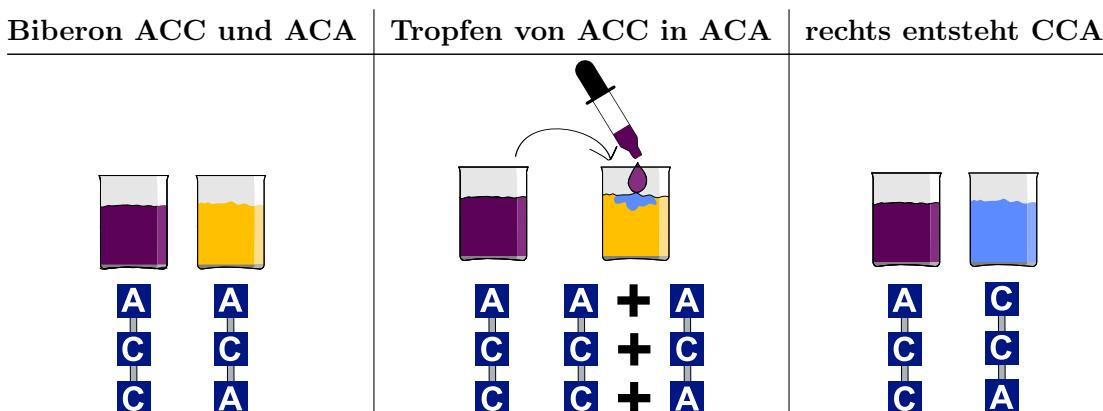
8. Biberone

Ein Biberon ist ein besonderer chemischer Stoff: Seine Moleküle sind jeweils eine Folge von genau drei Bausteinen der Typen A und C. Diese Folge wird auch als Name des Biberons verwendet, zum Beispiel: ACA.

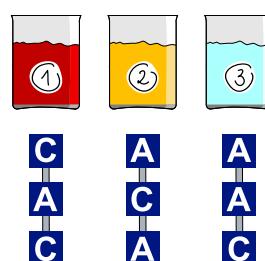
Wenn ein Biberon zu einem anderen Biberon hinzugefügt wird, entsteht wieder ein Biberon. Aus den Bausteinen in den Molekülen der beiden Biberone entstehen die Bausteine des neuen Biberons nach diesen Regeln:



Ein Beispiel: Wenn man einen Tropfen von ACC zu ACA hinzufügt, entsteht CCA:

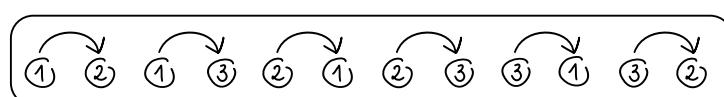


Im Labor gibt es drei Behälter 1, 2 und 3 mit den Biberonen CAC, ACA und AAC:



Mit einer solchen Anweisung kannst du bestimmen, dass ein Tropfen des Biberons in einem Behälter (z.B. Behälter 2) dem Biberon in einem anderen Behälter (z.B. Behälter 3) hinzugefügt wird.

Unten siehst du sechs verschiedene Anweisungen. Verwende möglichst wenige davon, um die Biberone in den Behältern 1 und 3 zu vertauschen: Am Ende soll in Behälter 1 AAC sein, in Behälter 2 bleibt ACA, und in Behälter 3 soll CAC sein.





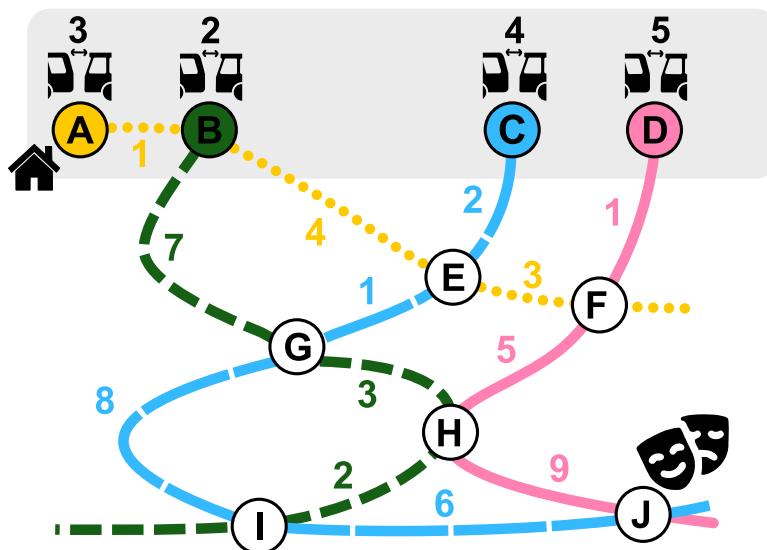
9. ÖV

In Marcus' Stadt gibt es vier Buslinien. Sie starten an den Haltestellen A, B, C und D. Ihre Fahrstrecken kannst du unten in der Karte sehen.

Die ersten Busse jeder Linie fahren zur gleichen Zeit von den Starthaltestellen (A, B, C, D) ab: das ist Minute 0. Danach fahren die Busse in den angegebenen Zeit-Abständen . Von Haltestelle A zum Beispiel fahren Busse alle 3 Minuten, also an den Minuten 0, 3, 6, 9 ...

Für jeden Streckenabschnitt zwischen zwei Haltestellen gibt eine Zahl an, wie viele Minuten ein Bus für den Abschnitt benötigt. Der erste Bus, der von A abfährt, hat also an Minute 0 + 1 + 4 + 3 = 8 die Haltestelle F erreicht.

Marcus wohnt bei Haltestelle A. Von dort aus möchte er mit dem ersten Bus zum Theater fahren. An Haltestellen, an denen sich Buslinien kreuzen, kann er innerhalb von 0 Minuten umsteigen. Er kann also mit jedem Bus weiterfahren, der zur gleichen Zeit oder später wie er an der Umsteige-Haltestelle ankommt. Marcus weißt auf welcher Route er fahren und umsteigen muss, damit er so früh wie möglich beim Theater ist.

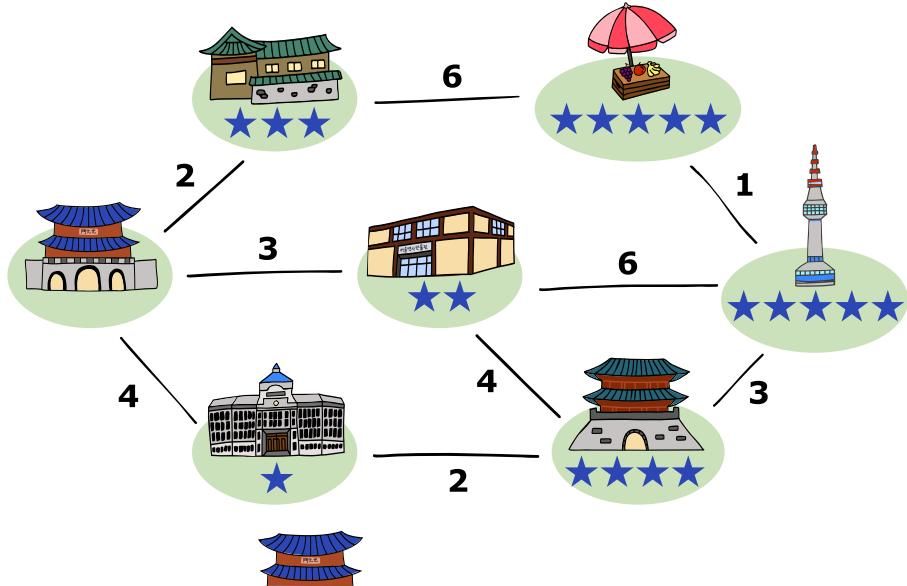


Welche Haltestellen liegen auf dieser Route?



10. Seoul entdecken!

In Seoul in Korea gibt es Busse für Touristen, die sehenswerte Orte miteinander verbinden. Das Bild zeigt die wichtigsten Orte von Seoul. Die Sterne sagen, wie beliebt die Orte sind. Die Linien zeigen die Busverbindungen. An jeder Linie steht, wie viele Kilometer lang die Verbindung ist.



Lotte besucht zuerst den Palast . Von dort aus möchte sie mit den Bussen weitere Orte besuchen. Lotte hat eine Fahrkarte, mit der sie höchstens 10 Kilometer weit fahren kann. Damit möchte sie über die Verbindungen Orte erreichen, die insgesamt möglichst viele Sterne haben! Sie besucht einen Ort natürlich nur einmal und muss nicht zum Palast zurück.

Welche Verbindungen muss Lotte mit ihrer Fahrkarte fahren, um möglichst viele Sterne zu sammeln?



11. Bergseen

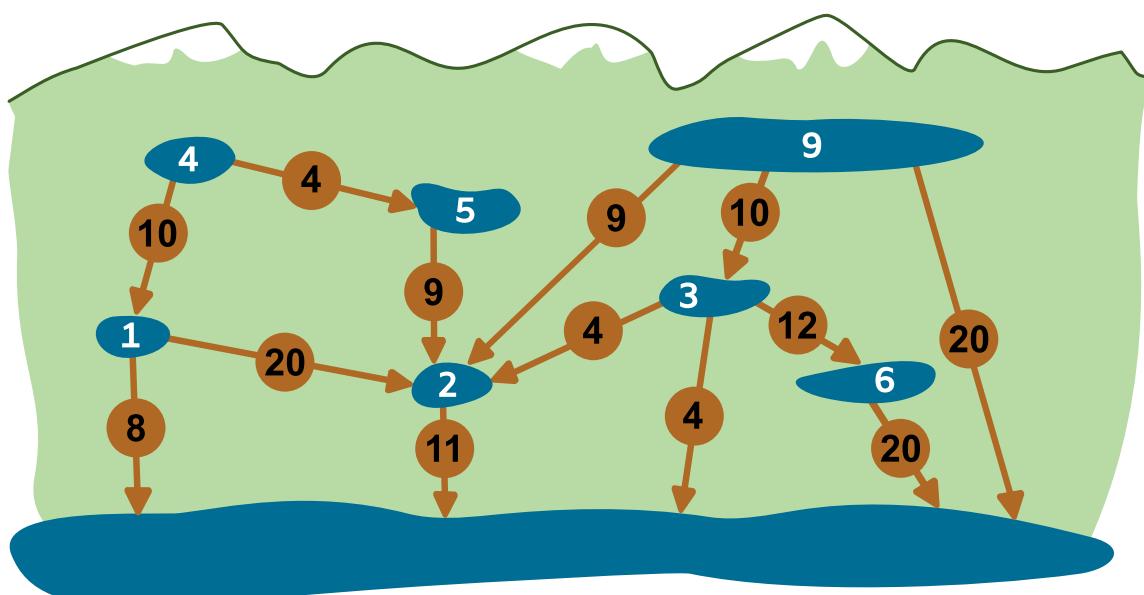
Am Bergmassiv hinter dem Stausee gibt es mehrere kleine Bergseen. Bei starkem Regen könnten sie überlaufen, und das ist gefährlich. Deshalb sollen zwischen einigen Seen Kanäle gebaut werden. Diese Kanäle sollen alles überschüssige Wasser aus den Bergseen in den Stausee ableiten können. Gleichzeitig soll ihr Bau möglichst wenig kosten.

Für jeden Bergsee gibt eine Zahl an, wieviel überschüssiges Wasser aus dem See abgeleitet werden muss.

An jeder Stelle zwischen zwei Seen, an der ein Kanal gebaut werden kann, ist ein Pfeil. Er zeigt, in welche Richtung ein Kanal das Wasser dort ableiten würde. Die Zahl an einem Pfeil gibt die Kapazität des Kanals an, also wieviel überschüssiges Wasser er ableiten kann. Die Kapazität bestimmt auch die Kosten für den Bau eines Kanals an dieser Stelle.

Beachte: Wenn ein Kanal Wasser von einem kleinen Bergsee in einen zweiten ableitet, sammelt sich im zweiten See das überschüssige Wasser aus beiden Seen.

An welchen Stellen sollen Kanäle gebaut werden?





12. Parkplätze

Zur Party kommen 9 Gäste mit ihren Autos. Vor dem Haus können 9 Autos so parkieren, dass in 3 Parkspuren jeweils 3 Autos hintereinander stehen. Die Gäste kommen in dieser Reihenfolge:

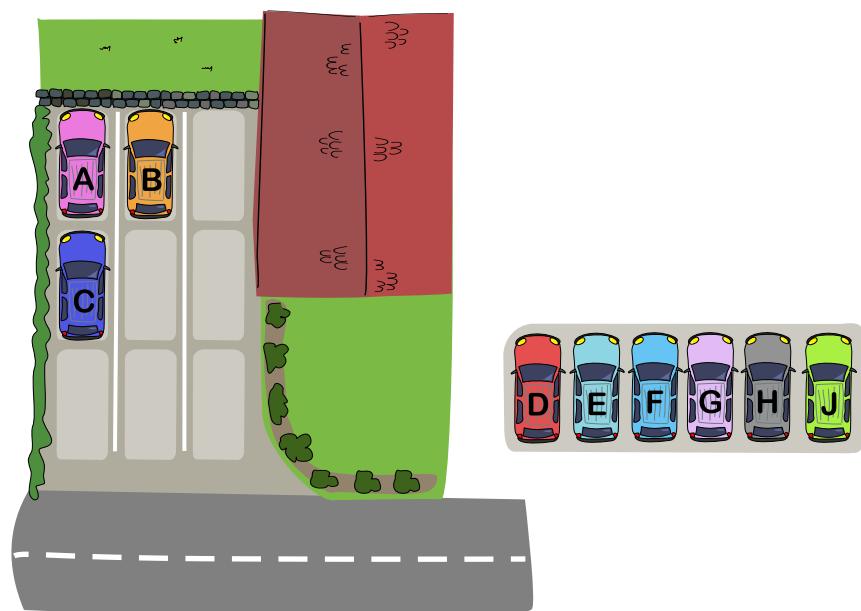
Anja, Beate, Clara, David, Elia, Frank, Gabi, Harald und zuletzt Julia.

Beim Einparkieren wählt jeder eine Parkspur aus und fährt darin so weit wie möglich nach vorne.

Die Gäste wollen in dieser Reihenfolge von der Party wegfahren:

Gabi, David, Beate, Elia, Julia, Clara, Harald, Anja und zuletzt Frank.

Die Autos von **Anja**, **Beate** und **Clara** sind bereits parkiert. Nun parkieren die anderen Gäste nach und nach ein. Sie wollen so parkieren, dass beim Wegfahren kein Auto von einem anderen blockiert ist, das später wegfährt.



Zeige den Gästen, wie sie so parkieren können!

Platziere die restlichen 6 Autos in den Parkspuren. Du musst die Reihenfolgen beim Ankommen **und** beim Wegfahren berücksichtigen.



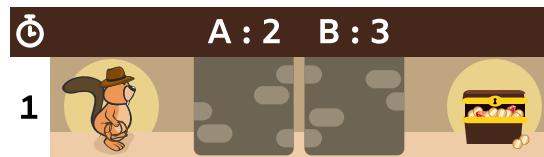
13. Biber Jones



Biber Jones ist in einer perfiden Pyramide mit gefährlichen Gängen. Am Ende jedes Ganges befindet sich ein sagenhafter Schatz. Jones will jeden Schatz *so schnell wie möglich* erreichen.

Jedoch ist jeder Gang durch eine Reihe von Fallblöcken gesichert. Am Anfang sind alle Blöcke unten. Sobald jemand den Gang betritt, beginnen die Blöcke sich zu bewegen. Jeder Block bewegt sich in festem Takt nach oben und unten. Zum Beispiel schnellt ein Block mit Takt 2 nach 2 Minuten hoch und kracht nach weiteren 2 Minuten wieder herunter.

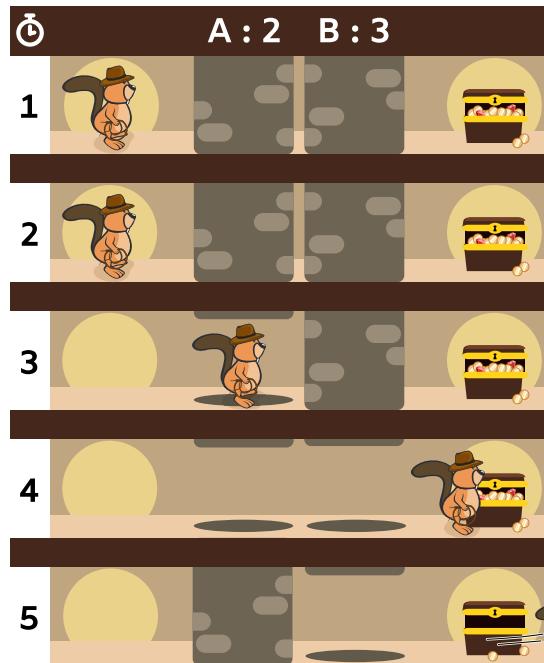
Jones steht vor seiner ersten Herausforderung:



Dieser Gang hat zwei Blöcke: Block A hat Takt 2, Block B Takt 3. Glücklicherweise hat Jones eine Schriftrolle mit Anweisungen gefunden, wie man so schnell wie möglich sicher den Schatz erreicht:

```
wait(2)  
goto_block(A)  
wait(1)  
goto_treasure
```

Jones befolgt die Anweisungen: Er wartet 2 Minuten, geht dann zu Block A, wartet dort 1 Minute und geht dann zum Schatz. Er erreicht den Schatz nach 3 Minuten:





Jones bemerkt, dass er auch mit weniger Anweisungen dem Schatz erreicht hätte, und zwar gleich schnell:

```
wait(3)  
goto_treasure
```

Er kommt zum nächsten Gang. Der hat vier Blöcke, mit Takt 3, 5, 8 und 4.

Was ist die kürzeste Folge von Anweisungen, mit denen Jones so schnell wie möglich den Schatz erreicht?





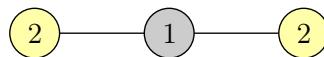
14. Prüfungsplan

Am Euler-Gymnasium stehen die Matura-Prüfungen an. Sie sollen an höchstens 5 verschiedenen Tagen geschrieben werden. Weil man nur eine Prüfung pro Tag mitschreiben darf, dürfen 2 Prüfungen, bei denen mindestens eine Person beide mitschreibt, nicht für den gleichen Tag geplant werden. Solche 2 Prüfungen haben einen «Tageskonflikt».

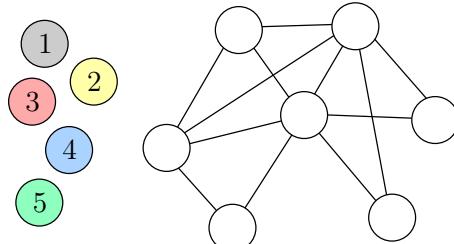
Um die Planung zu erleichtern, werden die Tageskonflikte in einem Diagramm aus Kreisen und Linien dargestellt:

- Für jede Prüfung wird ein Kreis gezeichnet.
- Zwischen 2 Kreisen wird genau dann eine Linie gezeichnet, wenn diese beiden Prüfungen einen Tageskonflikt haben, also nicht für den gleichen Tag geplant werden dürfen.

Hier ist ein Beispiel mit 3 Prüfungen: Die Prüfung in der Mitte hat 2 Tageskonflikte, nämlich mit jeder der beiden anderen Prüfungen. Die Nummern zeigen eine Möglichkeit, die Prüfungen auf 2 Tage (1 und 2) zu verteilen.



Innerhalb der nächsten 5 Tage sollen 7 Prüfungen geschrieben werden. Das Diagramm zeigt ihre Tageskonflikte.



Verteile die Prüfungen auf möglichst wenige der 5 Tage und beachte dabei die Tageskonflikte. Es gibt mehrere richtige Antworten.



15. Prüf-Biber

Der Biber-Boss setzt vier sogenannte *Nachrichten-Biber* ein: Sie halten Flaggen hoch, um Nachrichten zu senden. Jeder Nachrichten-Biber hält entweder eine rote  oder eine gelbe  Flagge hoch.

Manchmal passiert es, dass ein Biber die falsche Flagge hochhält. Das möchte der Biber-Boss erkennen können. Deshalb bestimmt er zusätzlich drei *Prüf-Biber*.

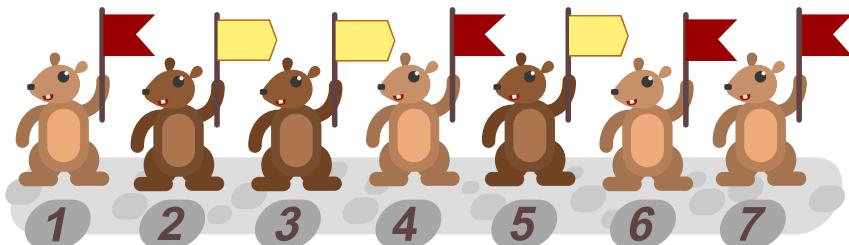
Jeder Prüf-Biber prüft drei Nachrichten-Biber. Wenn diese drei eine ungerade Anzahl an roten Flaggen hochhalten sollen, dann soll ihr Prüf-Biber auch eine rote Flagge hochhalten, sonst eine gelbe. Wenn alle die richtigen Flaggen hochhalten, dann halten ein Prüf-Biber und seine Nachrichten-Biber zusammen eine gerade Anzahl von roten Flaggen hoch.

Der Boss nummeriert die Nachrichten- und Prüf-Biber und ordnet sie so einander zu:

Nachrichten-Biber	Prüf-Biber
1, 2, 3	5
1, 2, 4	6
2, 3, 4	7

Insgesamt werden in einer Nachricht nun sieben Flaggen hochgehalten. Der Biber-Boss sieht die Nachricht unten. Er bemerkt, dass genau einer der sieben Biber die falsche Flagge hochhält.

Welcher Biber hält die falsche Flagge hoch?

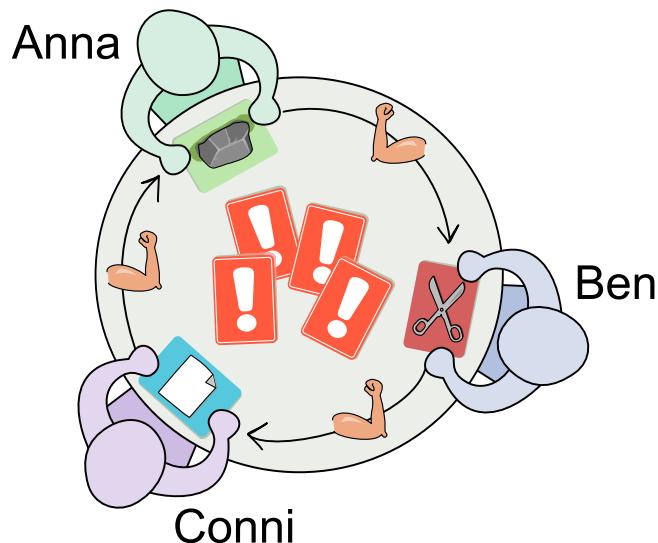




16. Schere, Stein, Papier

Anna, Ben und Conni spielen gemeinsam eine Runde Schere-Stein-Papier. Sie spielen gleichzeitig paarweise gegeneinander.

Jeder hat eine Spielkarte gezogen: Anna hat Stein , Ben hat Schere , und Conni hat Papier .



Es gelten die klassischen Regeln: Stein schlägt Schere, Schere schlägt Papier, Papier schlägt Stein. So wie die Karten jetzt verteilt sind, würde jeder einmal gewinnen und einmal verlieren. Zum Beispiel gewinnt Ben gegen Conni und verliert gegen Anna.

Bevor das Ergebnis für jedes Spielerpaar ausgewertet wird, muss aber noch 1 von 4 Aktionskarten gewählt und deren Aktion ausgeführt werden. Bei den Aktionen geht es darum, mehrfach Spielkarten zwischen je 2 Spielern zu tauschen. Bei jedem Tausch kann neu entschieden werden, welche 2 Spieler miteinander tauschen - es sei denn, die Aktion sagt etwas anderes.

Ben möchte unbedingt gegen Conni gewinnen, egal wie die Aktion auf der gewählten Aktionskarte ausgeführt wird.

Nur eine der vier Aktionen garantiert das. Welche?

- A) Ben und Conni tauschen ihre Karten eine ungerade Anzahl Male.
- B) Beliebig oft tauschen 2 Spieler ihre Karten, aber nie Ben mit Conni.
- C) Beliebig oft tauschen 2 Spieler ihre Karten, darunter Ben mindestens einmal mit Conni.
- D) Eine gerade Anzahl Male tauschen 2 Spieler ihre Karten.



Programmieraufgaben

Die folgenden Aufgaben zum Programmieren sind Bonusaufgaben des Wettbewerbs.

Für die Wettbewerbsaufgaben sind keine Vorkenntnisse notwendig. Diese Programmieraufgaben lassen sich jedoch mit Programmierkenntnissen einfacher lösen.

Da das Programmieren online viel mehr Spass macht und das Ergebnis direkt ausprobiert werden kann, sind diese Aufgaben unter folgendem QR-Code online zum Bearbeiten verfügbar.





17. Hin und Her

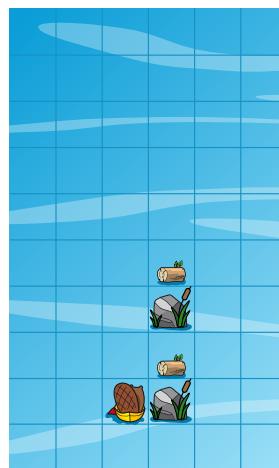
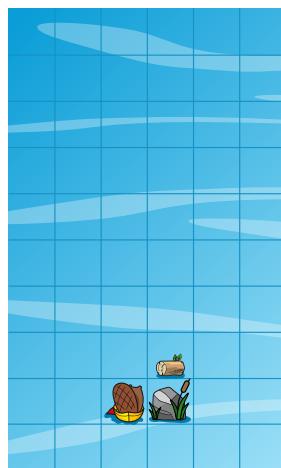
Biber Benno möchte Baumstämme im Seeland aufsammeln. Schreibe ein Programm, mit dem der Biber in allen Seen den Baumstamm einsammeln kann.

Verwende dazu die folgenden Anweisungen:

Anweisung	Beschreibung
<code>move()</code>	Benno bewegt sich genau ein Feld in Blickrichtung nach vorne.
<code>turnRight()</code> / <code>turnLeft()</code>	Benno dreht sich an Ort um 90 Grad nach rechts / links.
<code>removeLog()</code>	Benno entfernt den Baumstamm von dem Feld, auf dem er steht.
<code>while ...:</code> Anweisung Anweisung	Benno wiederholt nachfolgend eingerückte Anweisungen solange, wie eine Bedingung erfüllt ist.
<code>rockRight()</code>	<i>Bedingung:</i> Prüft, ob rechts von Benno ein Fels ist.
<code>rockLeft()</code>	<i>Bedingung:</i> Prüft, ob links von Benno ein Fels ist.

Im folgenden Beispiel bewegt ich Benno so lange ein Feld nach vorne, wie links von ihm ein Fels ist. Ist dies nicht (mehr) der Fall, fährt er mit der nächsten nicht eingerückten Anweisung – hier im Beispiel `turnLeft()` – fort:

```
while rockLeft():
    move()
    turnLeft()
```



Schreibe eine Anleitung, um in allen Seen den Baumstamm aufzusammeln.





A. Aufgabenautoren

James Atlas

Masiar Babazadeh

Leonardo Barichello

Wilfried Baumann

Susanne Berchtold

Maksim Bolonkin

Maria Cepeda

Špela Cerar

Gi Soong Chee

Anton Chukhnov

Darija Dasović

Christian Datzko

Justina Dauksaite

Diane Dowling

Nora A. Escherle

Gerald Futschek

Vernon Gutierrez

Silvan Horvath

Alisher Ikramov

Thomas Ioannou

Asterios Karagiannis

Blaž Kelvišar

David Khachatryan

Doyong Kim

Jihye Kim

Dong Yoon Kim

Vaidotas Kinčius

Jia-Ling Koh

Sophie Koh

Víctor Koleszar

Lukas Lehner

Gunwooong Lim

Yoshiaki Matsuzawa

Hamed Mohebbi

Mattia Monga

Anna Morpurgo

A-Yeong Park

Suchan Park

Jean-Philippe Pellet

Emmanuel Plan

Zsuzsa Pluhár

Wolfgang Pohl

Cesar F. Bolanos Revelo

Pedro Ribeiro

Rokas Rimkus

Kirsten Schlüter

Dirk Schmerenbeck

Jacqueline Staub

Alieke Stijf

Supawan Tasanaprasert

Susanne Thut

Christine Vender



 Michael Weigend

 Hsu Sint Sint Yee

 Kyra Willekes



B. Akademische Partner



Haute école pédagogique du canton de Vaud
<http://www.hepl.ch/>



AUSBILDUNGS- UND BERATUNGSZENTRUM
FÜR INFORMATIKUNTERRICHT

Ausbildungs- und Beratungszentrum für Informatikunterricht
der ETH Zürich
<http://www.abz.inf.ethz.ch/>

Scuola universitaria professionale
della Svizzera italiana



La Scuola universitaria professionale della Svizzera italiana
(SUPSI)
<http://www.supsi.ch/>

PÄDAGOGISCHE
HOCHSCHULE
ZÜRICH



Pädagogische Hochschule Zürich
<https://www.phzh.ch/>



Universität Trier
<https://www.uni-trier.de/>



C. Sponsoring

HASLERSTIFTUNG

Hasler Stiftung

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



Abraxas Informatik AG

<https://www.abraxas.ch>



Kanton Bern
Canton de Berne

Amt für Kindergarten, Volksschule und Beratung, Bildungs- und Kulturdirektion, Kanton Bern

<https://www.bkd.be.ch/de/start/ueber-uns/die-organisation/amt-fuer-kindergarten-volksschule-und-beratung.html>



Kanton Zürich
Volkswirtschaftsdirektion
Amt für Wirtschaft

Amt für Wirtschaft, Kanton Zürich

<https://www.zh.ch/de/volkswirtschaftsdirektion/amt-fuer-wirtschaft.html>



Informatik Stiftung Schweiz

<https://informatics-foundation.ch>

cyon

cyon

<https://www.cyon.ch>

senarc lens
leu+partner
strategische kommunikation

Senarc lens Leu & Partner

<http://senarc lens.com/>



Wealth Management IT and UBS Switzerland IT

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