

Parallelen

Martin und Joseph unterhalten sich über „Die zwei Parallelen“, ein Gedicht von Christian Morgenstern.

Martin: Ich habe etwas Erstaunliches entdeckt: Morgenstern hat eine geheime mathematische Struktur in sein Gedicht eingebaut, die der Menschheit bisher verborgen blieb.

Joseph: Wirklich? Was für eine Struktur denn?

Martin: Suche dir anfangs ein beliebiges Wort in der ersten Hälfte des Gedichts aus. Zähle die Anzahl n der Buchstaben in diesem Wort und springe entsprechend n Wörter im Text weiter. Zähle wieder die Buchstaben im erreichten Wort und springe entsprechend voran. Mache immer so weiter, bis es nicht mehr geht, weil nicht mehr genügend Wörter verbleiben. So endest du immer bei demselben Wort, egal wo du angefangen hast.

Joseph: Das glaube ich nicht. Irgendeinen Text zu schreiben, der nur diese Eigenschaft hat, halte ich noch für möglich, aber nicht, wenn er gleichzeitig Sinn, Versmaß und Reim in sich vereint.

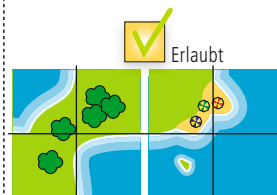
Junioraufgabe 1

Hat Martin Recht? Schreibe ein Programm, das seine Behauptung überprüft. Auf der [BWINF-Website](#) findest du Morgensterns Gedicht in einer Textdatei und weitere Hinweise zur Aufgabe.

Kacheln

Für ein Computerspiel sollen zweidimensionale Landschaften aus quadratischen Kacheln zusammengesetzt werden. Hierzu sind die rechts abgebildeten 16 Kacheln vorgegeben, von denen jede beliebig oft verwendet werden kann. Die Kacheln müssen nicht gedreht werden.

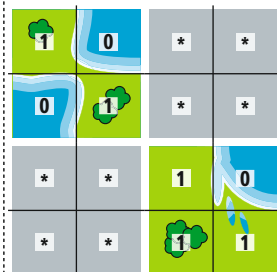
Die Kacheln sind so gestaltet, dass man jede in vier Quadrate aufteilen kann, deren Außenseiten entweder nur Wasser oder nur Land aufweisen. Damit die Landschaft natürlich aussieht, dürfen die Kacheln nur so aneinander gelegt werden, dass Land auf Land und Wasser auf Wasser trifft:



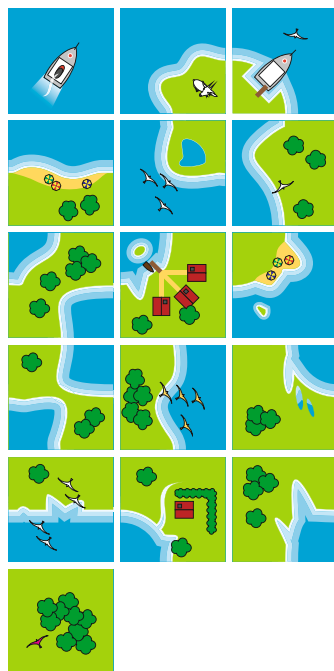
Damit Landschaften durch einen Computer generiert werden können, werden die Quadrate durch Bits und die Kacheln als 2x2-Bitmuster repräsentiert: 1 steht für ein Quadrat mit Land und 0 für ein Quadrat mit Wasser an den Außenseiten.



In dieser Aufgabe sollen nun unvollständige Landschaften vervollständigt werden, Beispiel:



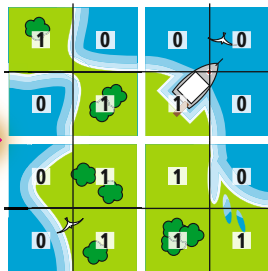
Die 16 Kacheln:



Junioraufgabe 2

Schreibe ein Programm, das eine unvollständige Landschaft einliest und unter Beachtung der obigen Regeln zu einer vollständigen Landschaft ergänzt. Selbst wenn unterschiedliche Ergänzungen möglich sind, soll nur eine vollständige Landschaft ausgegeben werden. Wenn eine Ergänzung ohne Regelverletzung gar nicht möglich ist, soll dies erkannt und angezeigt werden.

Wende dein Programm auf alle Beispiele an, die sich auf der [BWINF-Website](#) befinden. Dort sind auch die Kacheln und weitere Hinweise zu finden.



Blumenbeet

Im Blumenladen „Bunte Pracht“ sind kleine, rautenförmige Hochbeete der absolute Renner. Sie bestehen jeweils aus einer Kiste mit Erde, in die neun Blumenzwiebeln eingepflanzt werden. Es gibt nur diese Blumenfarben: blau, gelb, grün, orange, rosa, rot und türkis.

Die Hochbeete werden bisher mit einer willkürlich zusammengestellten Auswahl von Blumenzwiebeln verkauft, aber jetzt sollen individuelle Vorlieben von Kunden berücksichtigt werden. Hierzu gibt jeder Kunde an, wie viel verschiedene Farben (max. 7) das Beet haben soll. Außerdem kann er Paare von Farben, die ihm nebeneinander besonders gut gefallen, durch 1 bis 3 Bonuspunkte auszeichnen. Lea liebt rote neben blauen Blumen (3 Punkte), auch gefallen ihr rote neben türkisen Blumen (2 Punkte). Andere Kombinationen sind ihr egal. Das folgende Beet würde daher $2 + 3 + 2 = 7$ Punkte erhalten:



Der Blumenladen will die Beete so bepflanzen, dass die Vorstellungen des jeweiligen Kunden möglichst gut berücksichtigt werden.

Aufgabe 1

Schreibe ein Programm, das die Angaben eines Kunden einliest und dann eine Bepflanzung des Hochbeetes errechnet, die die größtmögliche Punktzahl erzielt. Als Nachbarblumen zählen dabei die Blumen links und rechts neben einer Blume sowie die bis zu vier Blumen schräg darunter oder darüber. Im obigen Beet hat die rosa Blume vier Nachbarblumen, die blaue Blume sechs und die rote Blume ganz oben nur zwei. Wende dein Programm auf alle Beispiele von Kundenangaben auf der [BWINF-Website](#) an.

Nummernmerker

Sarahs Mutter seufzt: „Puh, in diesem Formular muss ich schon wieder meine Mitgliedsnummer angeben. Jetzt muss ich sie suchen, ich kann mir solche Nummern einfach nicht gut merken!“

„Ich merke mir lange Nummern immer so, dass ich sie in kürzere Blöcke aufteile“, sagt Sarah. „Wenn’s geht, sollen diese aber nicht mit Null anfangen, weil ich sie mir dann nicht als Zahl merken kann. Meine Telefonnummer 01365400606 merke ich mir zum Beispiel so: 0136 5400 606.“

Aufgabe 2

Schreibe ein Programm, das Sarahs Mutter helfen kann, sich lange Nummern zu merken. Dein Programm soll eine Nummer, also eine Folge von Ziffern, einlesen und sie so in Blöcke aufteilen, dass gilt:

- > Jeder Block hat höchstens vier, aber mindestens zwei Ziffern.
- > Die Anzahl der Blöcke, die mit einer Null beginnen, ist minimal.

Du kannst davon ausgehen, dass eine einzulesende Nummer aus 2 bis 30 Ziffern besteht. Wende dein Programm auf alle Beispielnummern an, die sich auf der [BWINF-Website](#) befinden.

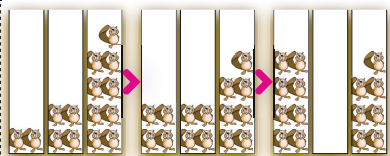


Telepaartie

In einem Werk der BWINF AG stehen drei Behälter mit Kuschelbibern. Die Behälter sind so groß, dass zwei Behälter ausreichen würden. Dafür müsste nur umgeschüttet werden. Ohne technische Hilfsmittel geht das aber nicht.

Zum Glück hat die Firmenleitung einen neuen *Telepaarteur* geschickt. Wenn man ihn auf zwei Behälter richtet und den Auslöser drückt, passiert eine *Telepaartie*. Sie verändert die Verteilung der Biber in den beiden Behältern so, als würde jeder Biber aus dem weniger vollen Behälter einen Partnerbiber aus dem anderen Behälter zu sich holen: Hat der weniger volle Behälter (bei Gleichstand: einer der beiden Behälter) vorher b Biber, hat er danach b Biber mehr (also $2b$ Biber), und der andere Behälter hat entsprechend b Biber weniger. Mit wiederholten Telepaartien ist es immer möglich, einen der drei Behälter leer zu bekommen, aber mit wie vielen?

Eine Verteilung von Bibern in den Behältern (eine *Biberverteilung*) lässt sich durch die Biberanzahlen in den drei Behältern angeben, z. B. $(2, 4, 7)$. Bei einer gegebenen Verteilung heißt die Anzahl der Telepaartien, die man mindestens braucht, um einen der Behälter leer zu bekommen, die *Leerlaufänge* (LLL) der Verteilung. Zum Beispiel ist die LLL der Verteilung $(2, 4, 9)$ genau 2, denn man kann damit anfangen, 2 Biber vom vollsten zum leersten Behälter zu telepaartieren:



Es gibt aber auch „schwierigere“ Verteilungen von 15 Bibern, deren LLL größer als 2 ist, zum Beispiel $(3, 5, 7)$. Für jede Gesamtbibierzahl n sei $L(n)$ die größte LLL von allen Verteilungen von insgesamt n Bibern auf die drei Behälter.

Aufgabe 3

Schreibe ein Programm, das eine Biberverteilung einliest und deren LLL ausgibt. Erweitere das Programm so, dass es eine Zahl n einlesen und die Werte $L(1)$, ..., $L(n)$ ausgeben kann. Zeige die Ausgabe für $n = 10$ und $n = 100$.

Urlaubsfahrt

Die Eltern von Justin, Esra, Julia und Vincent machen mit ihnen eine lange Urlaubsreise quer durch Europa mit einem Wohnmobil. Während der langen Fahrt müssen sie immer wieder halten, um zu tanken.

Justin sagt: „Lasst uns darauf achten, nicht so teuer zu tanken.“ Esra dagegen möchte möglichst schnell ankommen und meint: „Wir sind lange unterwegs. Ich möchte daher nicht oft anhalten; so groß sind die Preisunterschiede beim Diesel nun auch wieder nicht. Lass uns jedes Mal volltanken, dann kommen wir schneller an.“

Nach einiger Diskussion macht Julia den Vorschlag: „Wir machen es so: Wir tanken nur so oft, wie es unbedingt nötig ist; dafür suchen wir uns aber die günstigsten Tankstellen raus.“

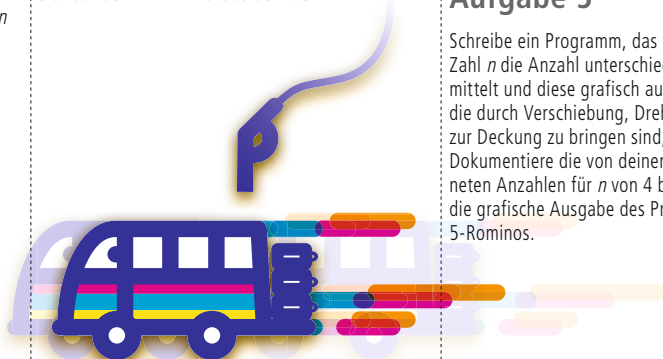
Vincent hat in der Zwischenzeit mittels einer App auf seinem Smartphone alle auf der Reiseroute liegenden Tankstellen mit ihren Entfernungen und Preisen für Diesel herausgefunden.

Aufgabe 4

Schreibe ein Programm, das den Verbrauch, die Tankgröße, die anfängliche Füllung des Tanks, die Länge der zu fahrenden Strecke und eine Liste von Tankstellen mit jeweiligen Entfernungen und Preisen einliest und berechnet, an welchen Tankstellen getankt werden soll und jeweils wie viel Diesel.

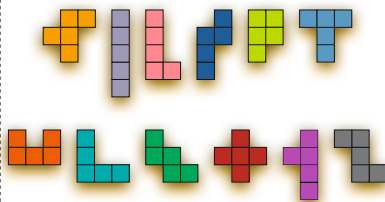
Es muss unbedingt so selten wie möglich getankt werden, selbst wenn häufigeres Tanken billiger sein könnte. Die Anzahl der Tankvorgänge muss also minimal sein, und sie sollen so geplant werden, dass insgesamt möglichst wenig Geld ausgegeben wird. Der Tank darf am Ziel leer sein.

Wende dein Programm auf alle Beispiele an, die sich auf der [BWINF-Website](#) befinden.



Rominos

Ruth hat zum Geburtstag eine Schachtel mit zwölf Figuren geschenkt bekommen: Sie heißen Pentominos und sind jeweils aus fünf gleich großen Quadraten gebildet. Es gibt genau zwölf verschiedene Pentominos, wie in Ruths Schachtel:



Ruth stellt fest, dass in einem Pentomino jedes Quadrat mit (mindestens) einer Seite bündig an der gesamten Seite eines anderen Quadrats anliegt.

Nun denkt sie über ähnliche Figuren aus fünf Quadraten nach, bei denen in mindestens einem Fall zwei Quadrate nur an einer Ecke aneinander anliegen, und zwar diagonal mit einem rechten Winkel; diese beiden Quadrate dürfen außerdem nicht an den Seiten ein und desselben dritten Quadrats anliegen. Ruth ist stolz auf ihre Erfindung und nennt diese neuen Figuren 5-Rominos, mit einem R für Ruth.

Ruth fragt sich, wie viele 5-Rominos es gibt. Ob dafür auch so eine kleine Schachtel genügen würde wie für die Pentominos? Bei 3-Rominos, also Rominos aus 3 Quadraten, ist es einfach. Es gibt genau drei unterschiedliche:



Aber schon bei 4-Rominos kommt Ruth ins Grübeln. Wie viele 4-Rominos gibt es?

Aufgabe 5

Schreibe ein Programm, das für eine eingelesene Zahl n die Anzahl unterschiedlicher n -Rominos ermittelt und diese grafisch ausgeben kann. Figuren, die durch Verschiebung, Drehung oder Spiegelung zur Deckung zu bringen sind, gelten als identisch. Dokumentiere die von deinem Programm berechneten Anzahlen für n von 4 bis 10 und außerdem die grafische Ausgabe des Programms für 4- und 5-Rominos.

Teilnehmen

Dieses Blatt enthält die Aufgaben der 1. Runde des 38. Bundeswettbewerbs Informatik (Bwlnf). Die Junioraufgaben sind gleichzeitig die Aufgaben der 3. und letzten Runde des Jugendwettbewerbs Informatik (Jwlnf) 2019.

Der Einsendeschluss ist für beide Wettbewerbe identisch: 25. November 2019.

Anmelden

online im BWINF-PMS: pms.bwinf.de

Sobald du im PMS registriert bist, kannst du dich dort zur Teilnahme anmelden: für den Jwlnf (3. Runde), den Bwlnf oder beides. Bei der Jwlnf-Anmeldung musst du deinen Benutzernamen für das Jwlnf-System angeben.

Bearbeiten

In der 3. Jwlnf-Runde bearbeitest du eigenständig die beiden Junioraufgaben. Im Bwlnf sind die Junioraufgaben SchülerInnen vor der Qualifikationsphase des Abiturs vorbehalten. Wer im Bwlnf in die 2. Runde kommen will, muss drei oder mehr Aufgaben bearbeiten, einzeln oder im Team.

Einsenden

Für jede bearbeitete Aufgabe sollst du im schriftlichen Teil deiner Einsendung (**Dokumentation**)

- > deine **Lösungsidee** beschreiben;
- > die **Umsetzung** der Idee in ein Programm erläutern;
- > an genügend **Beispielen** zeigen, wie deine Lösung funktioniert und was dein Programm jeweils ausgibt; und
- > die wichtigsten Teile des Quelltextes anfügen.

Achtung: eine gute Dokumentation muss nicht lang sein, aber unbedingt die **Beispiele** enthalten!

Der praktische Teil deiner Einsendung ist die **Implementierung** und umfasst den kompletten Quelltext und das ausführbare Programm (Windows, Linux, MacOS X oder Android).

Die **Einsendung** wird über das BWINF-PMS als ZIP-Dateiarchiv abgegeben. Ein Team gibt gemeinsam nur eine Einsendung ab.

Weitere Informationen unter: bwinf.de/teilnehmen

Jwlnf und Bwlnf: Jwlnf-Teilnehmende vor der Qualifikationsphase können ihre Bearbeitungen der Junioraufgaben auch zur 1. Bwlnf-Runde einsenden, gemeinsam mit Bearbeitungen mindestens einer weiteren Bwlnf-Aufgabe.

Fragen?

Wende dich an BWINF

- > per E-Mail: bundeswettbewerb@bwinf.de
- > oder Telefon: 0228 378646
- > Chat: bwinf.de/chat
- > Besuche das Diskussionsforum für den Wettbewerb bei der Einstieg-Informatik-Community: einstieg-informatik.de/community

Tipps und Infos

Unter bwinf.de/bundeswettbewerb/tipps findest du

- > genauere Hinweise zur Einsendung;
- > Beispiele für Aufgabenbearbeitungen;
- > Tipps zu Informatik und Programmierung.

Deine Chancen

Mit einer Teilnahme am Bundeswettbewerb Informatik kannst du nur gewinnen. In allen Runden gibt es **Urkunden** sowie kleine **Geschenke** für alle.

Bei erfolgreicher Teilnahme an der 1. Runde kannst du zu **Informatik-Workshops** eingeladen werden, die von vielen BWINF-Partnern wie dem Hasso-Plattner-Institut und der Deutschen Bundesbank ausgerichtet werden. Google lädt TeilnehmerInnen zum **Girls@Google Day** ein.

Nach deiner Teilnahme an der 2. Runde winken die **Forschungstage Informatik** des Max-Planck-Instituts für Informatik und einige Buchpreise. Die Einsendung zur 2. Runde kann in einigen Bundesländern als **besondere Lernleistung** in die Abiturwertung eingebracht werden.

Die Besten der 2. Runde erreichen die **Endrunde**. Dort werden die Bundessieger und Preisträger ermittelt, die mit **Geldpreisen** belohnt werden. Die Bundessieger werden in der Regel ohne weiteres Auswahlverfahren in die **Studienstiftung des deutschen Volkes** aufgenommen.

Siehe auch: bwinf.de/bundeswettbewerb/chancen



bwinf.de/teilnehmen



- > Triff Bwlnf-Teilnehmer und Jwlnf-Teilnehmer in der Community auf [einstieg-informatik.de!](https://einstieg-informatik.de/)

🐦 twitter.com/_BWINF 📷 instagram.com/bwinf