

Skyline

Das Wahrzeichen von Boomtown ist die Needle, ein schlanker, 100m hoher Turm. Nach ihrem Bau wurde einst beschlossen, dass kein Gebäude in Boomtown höher als die Needle sein darf. Nun sollen viele neue Gebäude in Boomtown errichtet werden, und die Bauherren wollen höher hinaus.

Der Rat von Boomtown überlegt, dass Gebäude, die weiter von der Needle weg sind, höher sein dürfen. So können die Bauherren befriedigt werden, und gleichzeitig bleibt die Needle als markantes Wahrzeichen der Stadt erhalten. Der Rat erlässt also die Vorschrift, dass mit jedem 100m Abstand von der Needle 1m höher gebaut werden darf. Also ist die maximal erlaubte Höhe bei einem Abstand von 0m bis 99m: 100m, bei einem Abstand von 100m bis 199m: 101m usw.

Nun soll für die vielen Bauanfragen entschieden werden, wie hoch das Gebäude jeweils sein darf. Zum Glück hatten die Stadtväter von Boomtown einst ein Koordinatensystem mit der Einheitslänge 1m eingeführt, in dem die Needle die Koordinaten (0,0) hat. Für die Grundrisse aller Gebäude gelten folgende Regeln:

- > sie sind rechteckig;
- > ihre Eckpunkte haben ganzzahlige Koordinaten;
- > ihre Seiten liegen parallel zu den Achsen des Koordinatensystems.

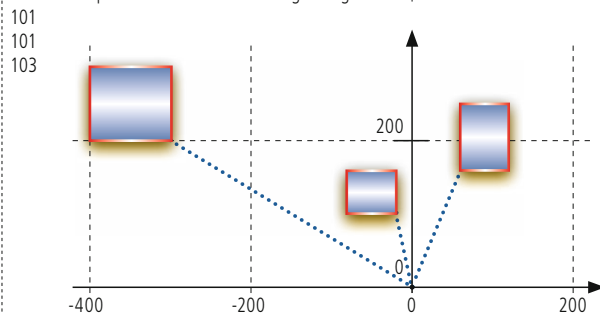
Das Bild unten zeigt drei Gebäude mit den Grundrissen (angegeben durch die Eckpunkte)

(-80, 100); (-20, 100); (-20, 160); (-80, 160)
(60, 160); (120, 160); (120, 250); (60, 250)
(-400,200); (-300, 200); (-300, 300); (-400, 300)

Junioraufgabe 1

Schreibe ein Programm, das für eine Liste von Grundrissen die erlaubten Höhen der Gebäude berechnet.

Für das Beispiel im Bild lautet die richtige Ausgabe:



Verben

Im Deutschunterricht wird gerade die Konjugation deutscher Verben behandelt. Mehmet denkt sich dabei: „Wozu braucht man da überhaupt einen Deutschlehrer? Das sind doch ganz einfache Regeln. Die programmiere ich mal fix; dann kann mein Smartphone hoffentlich die Hausaufgaben der nächsten Wochen erledigen.“ Nach einiger Zeit muss er aber zugeben, dass sein Programm immer noch nicht fertig ist und dass er nach fast jeder Deutschstunde irgendwelche kniffligen Fälle mit seinem Lehrer bespricht.

Junioraufgabe 2

Betrachte einen kleinen Teil von dem, was Mehmet vorhat: Schreibe ein Programm, das aus einer beliebigen Form eines deutschen Verbs durch Anwendung von Regeln versucht, die Grundform (den Infinitiv) des Verbs herauszufinden. Erhält das Programm z. B. die Eingabe „sagst“, soll es „sagen“ ausgeben.

Dein Programm muss zumindest bei Eingaben wie den folgenden die richtige Ausgabe produzieren:

- > sagst
- > leitete
- > geforscht
- > schweigend
- > trag

Zeige durch geeignete Eingabe-Ausgabepaare, was dein Programm kann. Zeige auch durch andere (vielleicht lustige) Beispiele, was es nicht kann.

Freiwillige Zusatzaufgabe:

Wenn du Lust hast, kannst du probieren, dein Programm so weiter zu entwickeln, dass es sich auch von hinterhältigen Eingaben wie „gebet“, „arbeite“, „genehmigt“, „gehört“ oder „mitgehalten“ / „mitgestalten“ nicht aus der Fassung bringen lässt. Das ist aber schwierig; und unregelmäßige Verben muss dein Programm sowieso nicht korrekt behandeln.

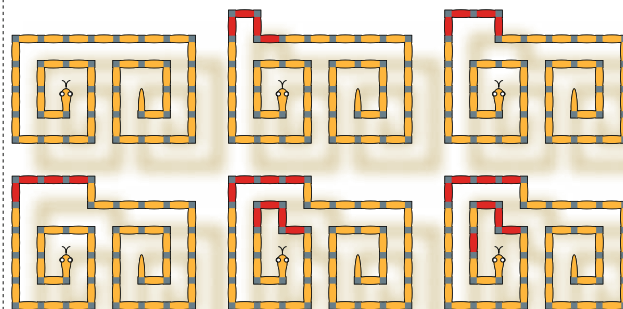
Frühspport

Die BWINF-Schlange Digitalil wacht nach einer durchrechneten Nacht mit heißen Algorithmen und zuviel Soft Ware ganz verknäult auf und möchte wieder ihr altes geradliniges Selbst werden. „Wie gut, dass ich in einer zwei- und nicht dreidimensionalen Welt lebe“, denkt sie, „denn sonst könnte ich echt ein Problem haben“. Auch so braucht sie aber deine Hilfe, sich zu entknäulen.

Digitalil besteht aus geraden Ringeln der Länge 1, die Gitterpunkte mit ganzzahligen Koordinaten verbinden. Sie ist normalerweise goldfarben; aber sie kann sich vorübergehend strecken und wird dann stellenweise rot: Aus einem oder zwei aufeinander folgenden goldenen Ringeln können doppelt so viele rote Ringel werden. Wenn sich die Schlange wieder zusammenzieht, werden aus zwei oder vier aufeinander folgenden roten Ringeln halb so viele goldene Ringel. Außerdem kann die Schlange ein oder zwei aufeinander folgende Ringel beliebiger Farbe verschieben, und zwei aufeinander folgende Ringel können ihre Farben tauschen. Diese akrobatischen Übungen sind allerdings alle nur möglich, wenn die dafür benötigten Gitterpunkte vorher frei waren und die Schlange zusammenhängend bleibt.

Aufgabe 1

Schreibe ein Programm, das eine beliebig verknäulte anfangs goldene Schlange einliest und eine Folge von Schritten der beschriebenen Art berechnet, nach deren Ausführung alle Ringel der Schlange auf einer Geraden liegen und (wieder) golden sind. Visualisiere die Ausführung der Schritte so, dass man sehen kann, wie sich die Schlange entknäult.



Geldtransporter

Zum Transport von Geld werden schwer bewachte Geldtransporter eingesetzt. In einem solchen Geldtransporter können Koffer mit Münzen transportiert werden. Die Koffer enthalten Münzen in unterschiedlicher Menge und mit unterschiedlichem Gesamtwert. Entsprechend unterscheiden sich die Koffer in Gewicht und Wert.

Da kommt so einiges an Gewicht zusammen. Es ist daher notwendig, den Transporter gleichmäßig zu beladen, so dass er nicht zu einer Seite umkippen kann.

Unser Transporter hat links und rechts je einen Kofferraum. Für jeden Kofferraum lassen sich Gesamtwert und Gesamtgewicht der darin enthaltenen Koffer bestimmen. Damit der Transporter keine Schlagseite bekommt, müssen die Koffer so eingeräumt werden, dass die Differenz zwischen den beiden Gesamtgewichten minimal ist. Aus versicherungstechnischen Gründen dürfen die beiden Gesamtwerte sich außerdem um höchstens 10.000 Euro unterscheiden.

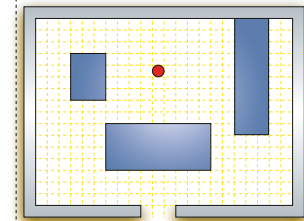
Aufgabe 2

Schreibe ein Programm zur Verteilung der Geldkoffer auf die beiden Kofferräume. Überprüfe dein Programm mit den auf www.bundeswettbewerb-informatik.de abgelegten Beispielen mit Angaben zu Werten und Gewichten der einzelnen Koffer.

Turn90

Der Robotertyp „Turn90“ ist in seiner Bewegungsfähigkeit eingeschränkt: Er kann sich nur vorwärts bewegen und um 90 Grad drehen. Außerdem hat er keinerlei visuelle Sensoren, kann also nicht „sehen“. Seine Kontaktsensoren und auch seine Rechenfähigkeiten sind aber sehr ordentlich.

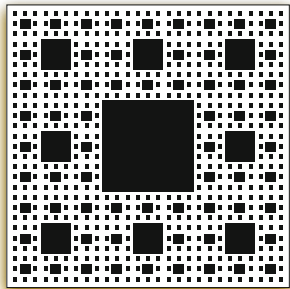
Ein neuer Auftrag zur Programmierung von Turn90-Robotern scheint darauf Rücksicht zu nehmen: Der Roboter soll befähigt werden, in einem Raum mit Hindernissen immer den einzigen Ausgang zu finden. Die Programmierer sind erleichtert, als sie hören, dass dem Raum ein Raster aus Quadraten zugrunde liegt. Die Wände füllen die äußeren Quadrate des Rasters, die Hindernisse füllen rechteckige Gruppen von Rasterquadraten aus. Der Roboter befindet sich zu Beginn in einem bestimmten Quadrat. Der Ausgang ist eine mindestens ein Quadrat breite Lücke in einer Raumwand. Das Bild unten zeigt einen Beispierraum mit Wänden und Hindernissen. Der Roboter ist als Kreis eingezeichnet.



Aufgabe 3

Schreibe ein Programm, mit dem ein Turn90-Roboter unter den genannten Bedingungen immer den Ausgang findet. Du kannst davon ausgehen, dass die Räume so gestaltet sind, dass es einen Weg vom Roboter zum Ausgang gibt. Schicke uns für die drei Räume, die auf den BwInF-Webseiten angegeben sind, und für zwei weitere Räume je ein grafisches Protokoll des Roboterweges. Lasse entweder dein Programm den Weg zeichnen, oder stelle ein schriftliches Protokoll des Weges (das du dann bitte mit-schickst) selbst grafisch dar.

SVG-Fraktale



Ein Fraktal ist eine selbstähnliche Figur. Es entsteht, indem, ausgehend von einer Grundfigur, geometrische Objekte schrittweise verfeinert werden. Ein bekanntes Fraktal ist zum Beispiel der Sierpinski-Teppich (s. Bild).

Scalable Vector Graphics (SVG) ist eine Sprache, in der sich geometrische Objekte mit leicht lesbaren Sprachelementen beschreiben lassen, um sie z. B. in Webbrowsern anzuzeigen. Eine Definition findet man unter www.w3.org/TR/SVG11/.

Für eine Lösung dieser Aufgabe genügen aber der „SVG-Rahmen“:

```
<?xml version="1.0"?>
<svg xmlns="http://www.w3.org/2000/svg"> elemente </svg>
```

und die folgenden Elemente:

```
<rect x="zahl" y="zahl" width="zahl" height="zahl" fill="farbe"/>
<circle cx="zahl" cy="zahl" r="zahl" fill="zahl"/>
<line x1="zahl" y1="zahl" x2="zahl" y2="zahl" stroke="farbe" stroke-width="zahl"/>
```

Die Grundfigur (erste Stufe) des Sierpinski-Teppichs, ein einfaches weißes Quadrat, wird in einer SVG-Datei so beschrieben:

```
<?xml version="1.0"?>
<svg xmlns="http://www.w3.org/2000/svg">
  <rect x="0" y="0" width="180" height="180" fill="white"/>
</svg>
```

Ein Verfeinerungsschritt besteht generell daraus, ein Objekt durch ein oder mehrere neue Objekte zu ersetzen. Beim Sierpinski-Teppich wird jedes weiße Quadrat durch neun kleinere, gleich große Quadrate ersetzt, von denen das mittlere schwarz und alle anderen weiß sind. Die zweite Stufe des Sierpinski-Teppichs wird also in einer SVG-Datei so beschrieben:

```
<?xml version="1.0"?>
<svg xmlns="http://www.w3.org/2000/svg">
  <rect x="0" y="0" width="60" height="60" fill="white"/>
  <rect x="60" y="0" width="60" height="60" fill="white"/>
  <rect x="120" y="0" width="60" height="60" fill="white"/>
  <rect x="0" y="60" width="60" height="60" fill="white"/>
  <rect x="60" y="60" width="60" height="60" fill="black"/>
  <rect x="120" y="60" width="60" height="60" fill="white"/>
  <rect x="0" y="120" width="60" height="60" fill="white"/>
  <rect x="60" y="120" width="60" height="60" fill="white"/>
  <rect x="120" y="120" width="60" height="60" fill="white"/>
</svg>
```

Das Bild oben zeigt die fünfte Stufe des Sierpinski-Teppichs.

Aufgabe 4

Wähle eine der auf www.bundeswettbewerb-informatik.de bereit gestellten Grundfiguren oder denke dir selbst eine Grundfigur aus. Schreibe ein Programm, das die Grundfigur oder eine bereits berechnete Verfeinerungsstufe aus einer SVG-Datei einliest und eine neue SVG-Datei erzeugt, die die nächste Stufe beschreibt. Die SVG-Dateien sollen von gängigen Web-Browsern (z. B. Firefox) angezeigt werden können.

Als Programmiersprache zur Verarbeitung von XML-Dialekten wie SVG eignet sich XSLT, aber auch andere Sprachen sind zugelassen. Es genügt, wenn eine Ausführung des Programms nur einen Verfeinerungsschritt vornimmt.

Kaffeerunde

Die Inhaberin der Firma Nash GmbH wünscht sich glückliche Mitarbeiter und möchte ihnen fürderhin kostenlos Kaffee anbieten. Hierzu wird Kaffee zur Verfügung gestellt sowie eine Kaffeemaschine mit einer Kanne. Die Mitarbeiter müssen die Maschine selbst bedienen (nach einer Einführungszeit von zehn Tagen, in denen ein Assistent der Inhaberin immer für eine volle Kanne sorgt). Abends wird die Kanne stets entleert.

Die Kanne fasst zehn Tassen und es gibt 15 Mitarbeiter. Jeder Mitarbeiter möchte gern drei Tassen täglich trinken, und zwar je eine vormittags (8–10 Uhr), mittags (12–14 Uhr) und nachmittags (15–17 Uhr). In jedem dieser zweistündigen Intervalle erscheinen die Kaffeetrinker in einer zufälligen Reihenfolge in der Kaffeeküche. Sie verhalten sich dann so:

Falls sich noch Kaffee in der Kanne befindet, nehmen sie sich eine Tasse daraus und gehen wieder. Ist die Kanne aber leer, dann hängt ihr weiteres Verhalten davon ab, ob sie glücklich sind. Falls ja, kochen sie eine neue volle Kanne und nehmen sich davon selbst eine Tasse. Sind sie aber nicht glücklich, dann haben sie keine Lust, für andere Kaffee mitzukochen, und gehen unverrichteter Dinge wieder weg.

Ob eine Person an einem bestimmten Tag glücklich ist, entscheidet sich direkt morgens nach dem Aufstehen und hängt davon ab, wie oft sie in den letzten zehn Tagen selbst Kaffee kochte und wie viele Tassen Kaffee sie in diesem Zeitraum trank. Die genaue Definition lautet:

Falls eine Person in den letzten zehn Tagen n Tassen trank und m -mal Kaffee kochte, dann ist sie genau dann glücklich, wenn $n \geq 10$ und $m/n < \alpha$ ist, wobei α ein persönlicher Schwellenwert ist.

Aufgabe 5

Schreibe ein Programm, das die Kaffeetrinker simuliert, wobei α der Einfachheit halber für alle identisch sein möge. Wie viele Tassen trinkt ein Mitarbeiter täglich im Durchschnitt auf lange Sicht, abhängig von verschiedenen Schwellenwerten α ?

Ist das Ergebnis so, wie du es erwartetest?

Teilnehmen

Einsendeschluss ist der 3. Dezember 2012
(Datum des Poststempels)

Einsendung an:

BWINF
Wachsbleiche 7, 53111 Bonn

Fragen zu den Aufgaben?

per Telefon:

0228 / 37 86 46
zu üblichen Bürozeiten

per E-Mail:

bwinf@bwinf.de

Diskutiere über die Aufgaben mit den Mitgliedern der EI Community:
www.einstieg-informatik.de

Anmeldung:

online unter

www.bundeswettbewerb-informatik.de/anmeldung

Hinweise zur Einsendung

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

- > deine **Lösungsidee** beschreiben;
- > die **Umsetzung** der Idee in ein Programm (falls gefordert) erläutern;
- > mit genügend **Beispielen** zeigen, dass und wie deine Lösung funktioniert; und
- > die wichtigsten Teile des **Quelltextes** anfügen.

Achtung: eine gute Dokumentation muss nicht lang sein – aber ausgedruckt werden (A4, einseitig, jede Aufgabe separat, am besten in je eine Hülle mit Lochrand; verwende keine Mappen!)

Bei Aufgaben mit Programmierung umfasst der **elektronische Teil** den kompletten Quelltext und das ausführbare Programm (Windows, Linux, MacOS oder Android).

Beispiele für Aufgabenbearbeitungen findest du auf den BwInf-Webseiten.

Sende uns die Dokumentationen und eine CD, DVD oder SD-Karte mit den elektronischen Teilen. Beschrifte alles großzügig mit deinem Namen und der Einsendungsnummer, die du bei der Anmeldung erhalten hast.

Eine Gruppe schickt ihre Einsendung gemeinsam.

Deine Chancen

Mit einer Teilnahme am Bundeswettbewerb Informatik kannst du nur gewinnen. In allen Runden gibt es **Urkunden** für Teilnahme und besondere Leistungen; zum Dank gibt es kleine **Geschenke** für alle.

Wer sich für die zweite Runde qualifiziert, kann mit Einladungen zu **Informatik-Workshops** rechnen: zum Jugendforum Informatik in Baden-Württemberg, dem Camp „Fit for BwInf“ des Hasso-Plattner-Instituts, den Informatiktagen der RWTH Aachen und dem BwInf-Workshop der TU Dortmund. Einige Teilnehmerinnen werden von Google zum **Girls@Google Day** eingeladen.

Nach der zweiten Runde winken die **Forschungstage Informatik** des Max-Planck-Instituts für Informatik. Ausgewählte Gewinner eines zweiten Preises in der zweiten Runde erhalten einen Buchpreis von O'Reilly. Eine Einsendung zur zweiten Runde kann in vielen Bundesländern als **besondere Lernleistung** in die Abiturwertung eingebracht werden.

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



Triff BwInf-Teilnehmer in der Community von www.einstieg-informatik.de!



BwInf.Informatik.erleben