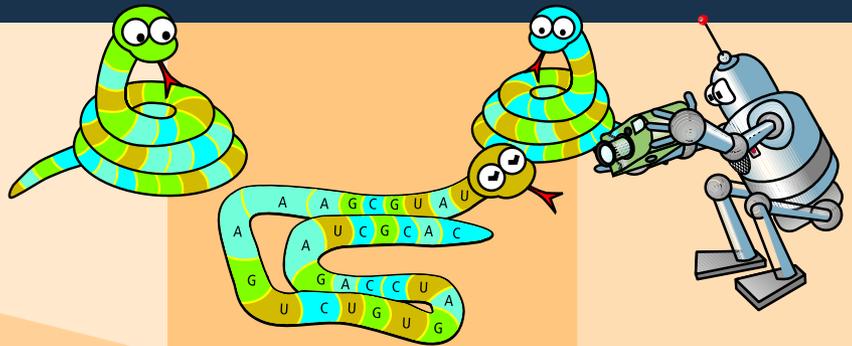


■ AUFGABE 1: RIBO-NATTER

Die Ribo-Natter (Kosename: RNA) ist eine lange Schlange mit vielen farbigen Ringeln, die in folgenden Farben vorkommen:
Aquamarin, Umbra, Grasgrün und Cyan

Diese Schlangenart ist ganz besonders eitel und möchte dem Betrachter ein möglichst gefälliges Aussehen präsentieren. Daher legt sie sich ganz flach so auf den Boden, dass – ihrer Meinung nach – eine farblich besonders harmonische Schlängelung entsteht. Dabei gibt es vier Punkte zu beachten:

- 1) Sie möchte dem Betrachter gerne viele Harmoniestellen zeigen. Das sind Stellen, an denen **A** mit **U** bzw. **C** mit **G** in Berührung kommt.
- 2) Ein Ringel darf zu höchstens einer Harmoniestelle gehören.
- 3) Damit die Schlängelung nicht zu geknickt aussieht, müssen zwischen zwei Ringeln, die miteinander eine Harmoniestelle bilden, mindestens drei Ringel sein, die zu keiner Harmoniestelle gehören.
- 4) Besonders schön sind Harmonieabschnitte, d. h. mehrere direkt aufeinander folgende Harmoniestellen.



Für diese Aufgabe nehmen wir an, dass die Ringel einer Schlange fortlaufend nummeriert sind, und zwar beginnend mit 1 vom Kopf her. Die oben abgebildete Schlange kann also wie folgt beschrieben werden:

Ringel:	U	A	U	G	C	G	A	A	G	U	C	U	G	U	G	A	U	C	C	A	G	A	U	C	G	C	A	C	
Nr.:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29

Eine Harmoniestelle entspricht dann einem Nummernpaar, z. B. (3,28); ein Harmonieabschnitt kann durch ein Paar von Nummernbereichen beschrieben werden: (3-7,28-24).

Eine komplette Schlängelung wird durch die entstandenen Harmonieabschnitte beschrieben; die Schlängelung aus der Abbildung lautet also (3-7,28-24) (12-14,22-20).

» Aufgabe

1. Schreibe ein Programm, das eine Schlange als Folge von Farbringeln einliest und ihren längsten Harmonieabschnitt findet. Demonstriere das Programm an drei Beispielen.
Eines dieser Beispiele soll die folgende Schlange sein:
**GGGAGCGUAGCUCAGUGCGGAGAGCGCCUCUUGCAGCAGAGGUCUGCGGUUCGAUC
CCGCGCGUCCCCACCA**
2. Eine Schlängelung mit möglichst vielen Harmoniestellen wäre der Ribo-Natter am liebsten. Beschreibe, wie eine Schlange vorgehen könnte, um die maximale (Farb-)Harmonie zu finden.

■ AUFGABE 2: WO BIN ICH?

Labormaus Ludwig erwacht aus der Narkose. Um sich herum fühlt er Sperrholzwände, und er weiß: Wieder hat die nette Verhaltensforscherin ihn irgendwo in das Labyrinth gesetzt, das er nun schon so gut kennt. Immer an derselben Stelle steht ein Napf mit etwas Futter. Die Verhaltensforscherin wird ihm außerdem noch mehr Futter geben, wenn er am Napf ist, und zwar umso mehr, je eher er dort ist – so hat Ludwig sie inzwischen dressiert.

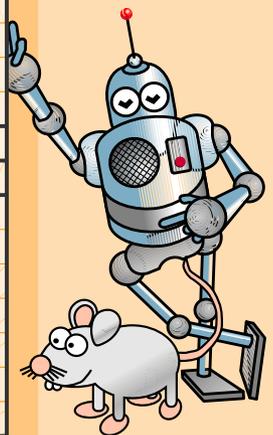
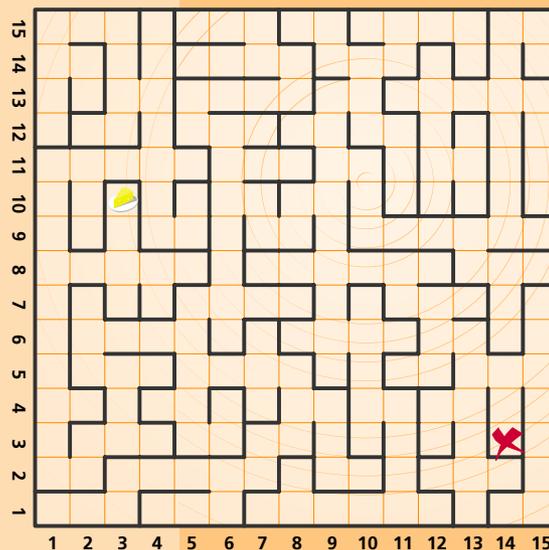
Ludwig ist hungrig wie immer, und er weiß genau, was er tun muss: Am schnellsten kommt er zum Futter, wenn er zunächst herausfindet, wo exakt er im Labyrinth ist, und von dort den kürzesten Weg zum Futternapf rennt. Da es immer dasselbe Labyrinth ist, hat er die Wegekarte klar im Kopf. (Ludwig entstammt einer alten Dynastie von Labormäusen, die über viele Generationen gelernt haben, sich Labyrinth einzuprägen.) Er kann ferner das Rauschen der Klimaanlage an der Nordwand des Laborraums hören, so dass er jederzeit weiß, wo Norden liegt. Da es dunkel ist im Labyrinth, kann er leider nur Wände unmittelbar um ihn herum wahrnehmen und nicht in Gänge oder Abzweige hineinschauen.

Seine Position im Labyrinth bestimmt Ludwig so: Er läuft die Labyrinthgänge entlang zu Kreuzungen, Einmündungen, Ecken und Sackgassen so lange, bis er an den Ganglängen und an der Form und Lage all der passierten Kreuzungen nach seiner Karte im Kopf eindeutig erkennen kann, wo er sich befindet. Manchmal kommt er zufällig schon während der Positionsbestimmung am Futternapf an: Das spart Lauferei, denn dann kann er direkt fressen.

Das Labyrinth ist abgeschlossen, Ludwig kann es also nicht verlassen. Alle Orte darin sind von allen Orten aus zu erreichen. Die Gänge des Labyrinths laufen in die Richtungen Nord, Ost, Süd oder West. Manche Gänge enden in Sackgassen. Die Futterstelle liegt am Ende einer Sackgasse.

» Aufgabe

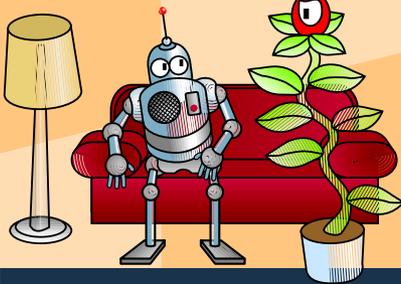
Die Positionen in einem Labyrinth liegen auf einem Raster mit Spalten und Zeilen, die zur einheitlichen Benennung mit Zahlen markiert sind. Die folgende Abbildung zeigt ein Labyrinth mit je 15 Spalten und Zeilen; der Futternapf steht an Position (3,10).



Schreibe ein Programm, das Ludwigs Methode zur Positionsbestimmung nachvollzieht. Gib das Verhalten der Maus Ludwig an drei Beispielen mit zufällig gewählten Startpunkten wieder. Das abgebildete Labyrinth muss mindestens in einem der Beispiele verwendet werden; dabei soll sich der Startpunkt an Position (14,3) befinden.

■ AUFGABE 3: GUTE STUBE

Die Cyber-Künstlerin EKAI will all die öden Chatrooms möblieren. Sie hat schon hübsche Bilder für die Objekttypen „Sessel“, „Couch“, „Zimmerpflanze“, „Kunstobjekt“, „Sitzkissen“, „Fahrrad“, „Lagerfeuer“, „Teddybär“, „Stehlampe“ und „Haustier“ entworfen. EKAI will, dass Chatter die vorkommenden Gegenstände und ihre Lage zueinander gemeinsam beschreiben und dann alle das Foto ihres Chatrooms auf dem Bildschirm sehen können. Die möglichen Lagen zueinander seien „links von“, „rechts von“, „neben“, „vor“, „hinter“, „über“ und „unter“.



► Aufgabe

- Schreibe für EKAI ein Programm, das
 - aus einer Folge von Sätzen die gemeinten Gegenstände und Lagebeschreibungen herausfiltert,
 - alle Mehrdeutigkeiten, Unvollständigkeiten und Widersprüche in der Gesamtdescription selbstständig irgendwie entscheidet und
 - ein 400x300 Pixel großes Bitmap (Foto) mit einer grafischen Umsetzung der Gesamtdescription erzeugt (ob in Schwarzweiß oder Farbe, ist uns egal). Benutze dazu digitale grafische Darstellungen (Grafiken, Fotos) für die Objekttypen von EKAI.
- Erläutere kurz, welche Mehrdeutigkeiten, Unvollständigkeiten und Widersprüche dein Programm in Gesamtdescriptionen erkennen kann und nach welchen Prinzipien es sie entscheidet. Diese Prinzipien müssen nicht begründet werden – künstlerische Freiheit ist hier sowieso unvermeidlich.

3. Schicke uns drei Satzfolgen (als unformatierte Textdateien) und die dazu erzeugten Bitmaps (als GIF-, JPEG- oder BMP-Dateien). Eine der drei Satzfolgen sei:

- Ich hätte gerne das Fahrrad#1 links von Stehlampe#1.
- Der Sessel#1 steht links von Sessel#2.
- Couch#1 befindet sich vor Zimmerpflanze#2.
- Sessel#1 ist neben dem Lagerfeuer#1.
- Ein Sessel#2 steht hinter Lagerfeuer#1.
- Sitzkissen#1 unter Haustier#1.
- Kunstobjekt#1 schwebt über Lagerfeuer#1.
- Die Zimmerpflanze#1 hinter Teddybär#1.
- Fahrrad#1 ist irgendwo rechts von Sessel#2.
- Stehlampe#1 sieht man links von Sessel#1.
- Doch die Zimmerpflanze#2 steht hinter dem Fahrrad#1.

■ AUFGABE 4: VERSTEHST DU BAHNHOF?

Jeden Tag treffen ein Güterzug aus Produdorf und einer aus Hersthallen in Mittelstadt ein, wo sie zu zwei neuen Zügen zusammengestellt werden, von denen einer nach Konuweiler und einer nach Verberau fährt. Jeder Zug besteht aus einer Lok gefolgt von Waggons. Du möchtest dem Bahnhofsvorsteher von Mittelstadt helfen, das Auseinander- und Zusammenkoppeln der Waggons zu planen.

In Mittelstadt werden die Züge für Konuweiler und Verberau aus den eintreffenden Zügen entsprechend dem gemeldeten Tagesbedarf zusammengestellt. Das Auseinander- und Zusammenkoppeln der Waggons ist aufwändig und zeitaufwendig. Daher sollte die Anzahl der Koppelvorgänge minimiert werden, wobei ein Koppelvorgang zwei Waggons bzw. eine Lok und einen Waggon auseinander- oder zusammenkoppelt.

► Aufgabe

Schreibe ein Programm, das den Tagesbedarf von Konuweiler und den von Verberau sowie die in Mittelstadt eintreffenden Züge als Folgen von Waggontypen einliest und eine Folge von Koppelanweisungen ausgibt, mit der die beiden in Mittelstadt eintreffenden Züge mit möglichst wenig Koppelvorgängen zu den beiden abfahrenden Zügen zusammengestellt werden können.

Zug aus Produdorf:



Zug aus Hersthallen:

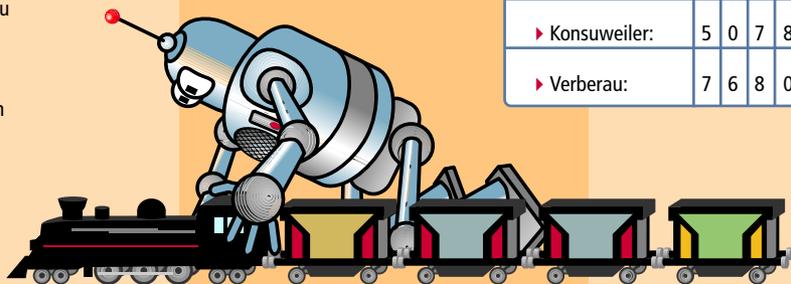


Die Waggons sind vom Typ A, B, C usw., je nachdem, ob sie Autoklaven, Bumerange, Celli usw. transportieren. Für jeden Typ wird der Tagesbedarf (eine ganze Anzahl Waggons) von Konuweiler und ebenso der von Verberau morgens den anderen Orten telefonisch mitgeteilt, und es wird dafür gesorgt, dass die Züge aus Produdorf und Hersthallen zusammen von jedem Typ genau die insgesamt geforderte Anzahl an Waggons enthalten. Ferner kann man davon ausgehen, dass in jedem dieser Züge alle Waggons eines Typs unmittelbar aufeinander folgen, wogegen das für die aus Mittelstadt abfahrenden Züge nicht gefordert wird.

In Mittelstadt stehen hinreichend viele Rangiergleise und Rangierloks zur Verfügung, deren Benutzung „gratis“ ist (Koppelvorgänge zwischen einer Rangierlok und einem Waggon werden nicht gezählt). Die aus Produdorf und Hersthallen ankommenden Loks müssen aber nach Konuweiler und Verberau weitergeführt werden.

Teste dein Programm anhand von drei Eingaben. Eine davon soll aus dem unten angegebenen Tagesbedarf und den oben abgebildeten Zügen bestehen.

Tagesbedarf an:	A	B	C	D	E
► Konuweiler:	5	0	7	8	3
► Verberau:	7	6	8	0	4



■ AUFGABE 5: DOSEN PACKEN

Ein Industrieroboter an einem Fließband muss so programmiert werden, dass er möglichst viele ankommende zylinderförmige Dosen in einer Schicht in eine 1 m x 1 m große Kiste packt. Die Durchmesser der Dosen sind unterschiedlich und liegen zwischen 10 cm und 20 cm. Der Roboter muss die Dosen der Reihe nach einpacken.

► Aufgabe

- Schreibe eine Funktion „verschiebe“, die eine teilweise gefüllte Kiste, die Position einer Dose in der Kiste (x- und y-Koordinaten) und eine Richtung (einen Winkel) als Eingabe nimmt und die Dose so lange in die gegebene Richtung verschiebt, bis sie an eine schon platzierte Dose oder an den Rand der Kiste stößt.

Die abschließende Position der Dose soll angegeben werden. Demonstriere die Funktion an drei Beispielen.

- Überlege dir und beschreibe in einem kurzen Text mindestens zwei Methoden, die die Funktion „verschiebe“ verwenden, um eine möglichst gute Packung zu erzielen.