

# Aufgabe 1

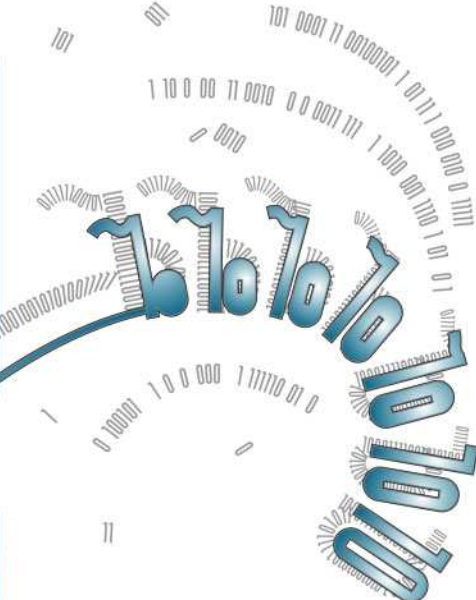
## Favorites First

Einige MP3-Player haben eine spezielle Zufalls- bzw. Shuffle-Funktion: Beim Anschalten berechnen sie eine zufällige Reihenfolge der abgespeicherten Stücke und spielen sie dann in dieser Folge ab.

Bo Hay, Marketingexperte eines Player-Produzenten, hat nun beobachtet, dass es Nutzerinnen und Nutzer gibt, die die Zufallsfunktion prinzipiell gerne einschalten, es aber schade finden, dass sie dabei ihre Lieblingsstücke relativ selten hören. Eine Ursache dafür ist, dass in der Regel nur die Anfangsteile solch zufälliger Abspielfolgen gehört werden. Bo Hay möchte, dass die Zufallsfunktion auf die folgende Weise abgewandelt wird: Unter den ersten 20 Stücken einer Abspielfolge sollen immer 5 vorkommen, die zufällig aus den 10 bis dahin meistgehörten Stücken ausgewählt werden. Dabei soll berücksichtigt werden, wenn ein Stück nach dem Anspielen übersprungen wird; es gilt dann als nicht gehört. Die Nutzerinnen und Nutzer sollen nicht wissen, dass der Zufall derart beeinflusst wird und sich nur über den für sie angenehmen Effekt freuen. Einen schönen Namen hat er für dieses neue Angebot auch schon gefunden: „Favorites First“.

### → Aufgabe

1. Realisiere eine Funktion (oder verwende eine von deiner Programmiersprache zur Verfügung gestellte Funktion), die die abgespeicherten Stücke in eine zufällige Reihenfolge bringen kann.
2. Überlege und beschreibe, wie Bo Hays Spezifikation von „Favorites First“ umgesetzt werden kann, und schreibe ein Programm, das deine Überlegungen möglichst einfach realisiert.
3. Simuliere den Effekt von „Favorites First“. Gehe davon aus, dass 250 Stücke auf dem Player abgespeichert sind, dass von jeder Abspielfolge nur die ersten L Stücke zumindest angespielt werden (wobei L eine zufällige Zahl zwischen 10 und 20 ist) und dass von diesen L Stücken eine zufällige Auswahl von einem Viertel (abgerundet) der Stücke übersprungen wird. Simuliere mindestens 200 Benutzungen des Players. Ist ein Effekt zu beobachten, den Bo Hay sich vielleicht nicht gewünscht hat? Wenn ja: was ist deiner Meinung nach der Grund für das unerwünschte Verhalten?



# Aufgabe 2

## Formel-Up

Der Forscher Schinkenfranz ist wieder einmal der Weltformel auf der Spur. Er hat drei neue Messgeräte „H-Wert“, „A-Wert“ und „M-Wert“ gebaut und lässt damit experimentieren. Die ganzzahligen positiven Messwerte werden in eine dreispaltige Messtabelle eingetragen. Jedes Experiment ergibt eine Zeile. Schinkenfranz sucht nach „schönen Formulierungen“ für seine Messtabellen, in der Form:

$$\langle \text{faktor-1} \rangle * \text{H-Wert} \langle \text{operator} \rangle \langle \text{faktor-2} \rangle * \text{A-Wert} = \langle \text{faktor-3} \rangle * \text{M-Wert}$$

$\langle \text{faktor-i} \rangle$  sei eine positive ganze Zahl (1, 2, 3, ...).  
 $\langle \text{operator} \rangle$  sei eine der zwei Grundrechenarten +, -.

Weil er weiß, dass kleine Messabweichungen unvermeidlich sind und seine völlig überforderten Assistenten gelegentlich auch einen groben Experimentierfehler machen, gibt er diesen drei Hinweise, wie sie eine für ihn „schöne“ Formel finden können:

1. Die Summe der absoluten Abweichungen der M-Werte von ihren durch die Formel geforderten Idealwerten sei klein.
2. Bis zu drei Zeilen, die überhaupt nicht in die Formel passen wollen, dürfen als „fehlerhaft ausgeführte Experimente“ aus der Tabelle gestrichen werden.
3. Kleine Faktoren machen Formeln schöner als große Faktoren.

Die Assistenten stöhnen auf. Da erbarmt sich Schinkenfranz und führt wenigstens noch ein kleines Beispiel vor:

Experiment	H-Wert	A-Wert	M-Wert
1:	22	5	13
2:	57	31	29
3:	44	31	19
4:	42	21	21
5:	128	1	85

Wenn man aus der obigen Messtabelle Experiment 2 als fehlerhaft streicht, dann ist  $2 * \text{H-Wert} - 1 * \text{A-Wert} = 3 * \text{M-Wert}$

eine schöne Formel, weil die Summe der absoluten M-Wert-Abweichungen 0 ist und die Faktoren (2 1 3) alle klein sind.

### → Aufgabe

1. Programmiere einen FORMULIERER, der zu einer beliebigen dreispaltigen Messtabelle drei schöne Formeln vorschlägt und seine Vorschläge mit Hilfe einfacher Textbausteine kurz begründet.
2. Nimm die Messtabelle unter [www.bwinf.de/aufgaben/material.php](http://www.bwinf.de/aufgaben/material.php) sowie vier selbst gemachte Messtabellen, um die sinnvolle Arbeit deines FORMULIERERS deutlich zu machen.
3. Gib dem FORMULIERER eine Tabelle mit Zufallszahlen. Was passiert?
4. Beantworte mit ein paar Sätzen: Wie entscheidet dein FORMULIERER, ob eine Formel schöner ist als eine andere? Warum entscheidet er gerade so? Gäbe es dazu vernünftige Alternativen?

# Aufgabe 3

## Zaras Zauberfolie

Die Handlungsreisende Zara Zackig wird von ihrer Firma häufig in Länder geschickt, von denen aus sie mit ihrer Firma nur per Fax in Kontakt treten kann. Für erfolgreiche Verkaufsverhandlungen benötigt Zara immer wieder Informationen, die die Kunden nicht sehen dürfen. Die Firma überlegt sich daher eine Methode, Zara per Fax verschlüsselte Nachrichten zu übermitteln. Dazu wird zunächst eine Transparenz-Folie erzeugt, die Zara mit auf Reisen nimmt. Auf die Nachrichten, die sie unterwegs erhält, legt sie jeweils diese Folie – und siehe da, sie weiß, was sie wissen muss.

Die zu übermittelnde Nachricht entsteht durch Überlagerung des Fax und der Folie als ein schwarz-weißes Rasterbild. Hierzu wird jedes der eigentlichen Rasterquadrate nochmals unterteilt in vier Unterquadrate und tritt sowohl auf der Folie als auch auf dem Fax nur in den folgenden zwei Formen auf: Form A: (entspricht  $\blacksquare$ ), Form B: (entspricht  $\blacklozenge$ ). Nur wenn A und B genau übereinanderliegen, entsteht ein schwarzes Rasterquadrat (entspricht  $\blacksquare$ ).

Auf der Folie werden Quadrate der Formen A und B zufällig verteilt. In Abhängigkeit von der Folie muss das Fax so gestaltet werden, dass beim Auflegen der Folie auf das Fax das, was Zara sehen soll, schwarz hervortritt.

(Stark vergrößertes) Beispiel mit 7 x 7 Rasterquadrate (also 14 x 14 Unterquadrate):



Ein Konkurrent kann sich Kopien mehrerer von Zara empfangener Faxe aneignen. Er möchte diese nun gerne entschlüsseln. Leider passt Zara gut auf ihre Originalmaske auf und er muss die Entschlüsselung ohne diese vornehmen.

### → Aufgabe

1. Überlege, ob und wie er die Entschlüsselung vornehmen kann.
2. Schreibe ein Programm, das aus den digitalisierten Kopien der verschlüsselten Faxe die Originale so gut rekonstruiert, dass sie in der Regel lesbar werden.

Dein Programm bekommt die eingescannten Bilder dabei in folgendem Format (PBM, Portable Bitmap):

```
P1
9 9
000010000
000010000
000101000
000101000
001000100
001111100
001000100
010000010
111000111
```

Die Datei beginnt stets mit P1, gefolgt von der Breite und Höhe in Pixeln als Dezimalzahl. Danach folgen alle Pixel, die durch 0 und 1 dargestellt werden. Obiges Beispiel zeigt natürlich den Buchstaben A.

3. Erläutere die Arbeitsweise deines Programms und entschlüssele sechs Faxe, welche du von [www.bwinf.de/aufgaben/material.php](http://www.bwinf.de/aufgaben/material.php) herunterladen kannst.

## Fehlzeitendatenbank

An der Zuse-Schule ist häufiges Fehlen und Zuspätkommen in letzter Zeit zu einem gravierenden Problem geworden. Deshalb hat die Schulkonferenz beschlossen, die Anwesenheit von Schülern und Lehrern elektronisch zu erfassen. Jedem Schüler und jedem Lehrer soll ein RFID-Chip im Nackenbereich implantiert werden. Alle Türrahmen von Unterrichtsräumen sollen mit Erfassungseinrichtungen ausgestattet werden, die Personen beim Betreten und beim Verlassen eines Unterrichtsraumes registrieren.

Zusätzlich fordert der Elternrat der Schule, es solle eine Möglichkeit geschaffen werden, die aktuelle Unterrichtsanwesenheit und die Fehlzeiten der eigenen Kinder und der ihre Kinder unterrichtenden Lehrer via Internet abzufragen.

### → Aufgabe

1. Beschreibe ein sinnvolles Verfahren für die Abwicklung von Entschuldigungen für Fehlzeiten. Beurteile die Praxistauglichkeit deines Verfahrens.
2. Erstelle ein Entity-Relationship-Modell für eine Datenbank, die eine Abfrage der Fehlzeiten ermöglicht. Beschreibe, in welcher Hinsicht dein Modell eine Vereinfachung der Realität darstellt. Rechtfertige deine Entscheidungen.
3. Gib, deinem Entity-Relationship-Modell entsprechend, ein System von Tabellen in Relationenschreibweise an. Ein Beispiel für die Schreibweise einer Tabelle:

**Schüler (Schüler-ID, Vorname, Nachname, Geburtsdatum)**

Unterstreiche Schlüsselattribute und kennzeichne Fremdschlüssel in geeigneter Weise.

4. Was hältst du von dem Beschluss der Schulkonferenz und dem Ansinnen des Elternrates? Begründe deine Meinung.

Hinweis:

Literatur und Material zu Datenbanken, Entity-Relationship-Modell und RFID-Chips findest du unter [www.bwinf.de/aufgaben/material.php](http://www.bwinf.de/aufgaben/material.php).

## Kleingeld

Clara ärgert sich über das viele Kleingeld, das ihr Portemonnaie so dick macht. Insbesondere die Cent-Münzen (also die Münzen mit den Werten 1, 2, 5, 10, 20 und 50 Cent) würde sie gerne los werden. Sie überlegt, wie viele Cent-Münzen sie durchschnittlich mit sich trägt, wenn sie stets darauf bedacht ist, die Münzanzahl zu minimieren. Du sollst ihr helfen, diese Frage zu beantworten. Dabei kannst du Folgendes annehmen:

→ Clara versucht, nach jedem Einkauf möglichst wenig Cent-Münzen im Portemonnaie zu haben.

→ Sie weiß, dass das Kassenpersonal überall freundlich ist und ihr möglichst wenig Cent-Münzen als Wechselgeld gibt.

→ Clara hat immer genügend 1-Euro-Münzen dabei. Wenn sie als einzige Cent-Münze ein 2-Cent-Stück im Portemonnaie hat, wird sie also bei einem Einkauf für 2,46 Euro diese Münze mit auf die Theke legen, um anschließend nur drei Münzen zu haben (50, 5 und 1 Cent) statt vier (50 und drei mal 2 Cent).

### → Aufgabe

1. Schreibe ein Programm, das die Entwicklung des Cent-Münzen-Gehalts von Claras Portemonnaie simulieren kann. Es soll von einem Anfangsbestand an Cent-Münzen ausgehen und eine Folge von zufälligen Beträgen erzeugen, die Clara bei ihren Einkäufen bezahlen muss. Für jeden Betrag soll es berechnen und geeignet anzeigen, welche und wie viele Cent-Münzen Clara bei der Bezahlung verwendet, wieder herausbekommt und anschließend im Portemonnaie hat.
2. Simuliere eine ausreichend hohe Anzahl von Einkäufen und bestimme die durchschnittliche Anzahl Cent-Münzen in Claras Portemonnaie (zu Beginn hat sie keine Cent-Münzen). Dokumentiere die Ausgaben deines Programms für die ersten 100 Einkäufe.
3. Wird eine Münzsorte besonders selten verwendet? Glaubst du, dass sie einfach weggelassen werden kann?

## Der Segelflug der Solinge

Ein Soling ist ein „Verwandter“ des Lemmings. Im Gegensatz zu einem Lemming tritt ein Soling aber immer alleine auf.

Bei diesem Spiel segeln auf einem gerasterten Bildschirm von oben nach unten derartige Solinge. Auf dem Weg nach unten befinden sich mehrere horizontale Barrieren, die jeweils an bis zu drei Stellen durchbrochen sind.

Die Solinge sollen vom Spieler möglichst schnell zur unteren linken Ecke des Bildschirms geführt werden. Der Spieler kann dabei die Bewegungsrichtung des Solings nach links oder rechts beeinflussen.

Anschließend segelt der nächste Soling auf den Bildschirm. Dabei verändern sich jeweils auch die Barrieren auf dem Spielfeld.

Ziel des Spieles ist: Möglichst viele Solinge sollen in einer bestimmten Zeit das Ziel erreichen.

### → Aufgabe

1. Die obige Spielbeschreibung lässt noch viele Punkte offen. Präzisiere diese Beschreibung lückenlos und eindeutig. Begründe dabei deine Entscheidungen.
2. Implementiere das Spiel entsprechend deiner Beschreibung.

Beispiel für ein Spielfeld:  
(siehe unten; das X stellt einen Soling dar)

```

=====
#START#                                     #
###  ##                                     #
#                                           #
#                                           #
#-----                                     #
#                                           #
#                                           #
#                                           #
#                                           #
#-----                                     #
#                                           #
#                                           #
#                                           #
#-----                                     #
#                                           #
#                                           #
#                                           #
#-----                                     #
#                                           #
#                                           #
#                                           #
#-----                                     #
##  ##                                     #
#ZIEL#                                     #
=====
    
```

