

### **Bsp 1:    Summensuche**

Erzeugen Sie ein Integer-Array mit einer von Tastatur eingelesenen Größe und initialisieren Sie die Werte des Arrays mit Zufallszahlen (siehe dazu Blatt 4 Beispiel 3). Lesen Sie eine weitere Zahl von Tastatur und ermitteln Sie, ob die Summe zweier unterschiedliche Zahlen des Arrays die eingegebene Zahl ergibt.

### **Bsp 2:    Spiel des Lebens**

Es gibt eine Spielwelt als Matrix von Zellen, die entweder leben können oder tot sind. Jede Zelle hat 8 Nachbarn. Randzellen haben die Zellen des gegenüberliegenden Randes als Nachbarn. Aus der momentanen Zellpopulation kann die Population der nächsten Generation durch folgende Regeln berechnet werden:

- Hat eine tote Zelle genau drei lebende Nachbar, erwacht sie zum Leben.
- Hat eine lebende Zelle zwei oder drei Nachbarn, bleibt sie am Leben.
- Alle anderen lebenden Zellen sterben.

Schreiben Sie ein Programm, das eine Zahl  $n$  sowie eine  $n \times n$ -Matrix aus Nullen und Einsen von einer Datei einliest. Die Matrix verkörpert die Initialpopulation: Eine Eins bedeutet lebende Zelle, eine Null eine tote Zelle. Nach jeder berechneten Generation soll die Spielwelt ansprechend ausgegeben und eine Zeile von Tastatur eingelesen werden. Ist die Eingabe `quit`, so soll das Programm beendet werden; ansonsten soll die nächste Generation berechnet werden.

### **Bsp 3:    Wörter identifizieren**

Lesen Sie einen Satz von Tastatur ein und geben sie alle Wörter einzeln untereinander aus. Die Satzzeichen sollen in der Ausgabe nicht enthalten sein.

### **Bsp 4:    Palindrome**

Palindrome sind Wörter und Sätze, die vor- und rückwärts gelesen identisch sind. Zum Beispiel: „Elly biss Sibylle.“ oder „Reit nie tot ein Tier.“. Bestimmen Sie, ob ein von Tastatur eingelesener Satz ein Palindrom ist. Ignorieren Sie dabei Leerzeichen sowie Satzzeichen.

### **Bsp 5:    Namen formatieren**

In einer Datei befindet sich eine Liste von Namen. Pro Zeile steht ein Name der aus Vorname, optionalem zweiten Vornamen und Nachname besteht. Formatieren Sie die Namen nach folgender Regel und schreiben diese in eine zweite Datei:

|                           |    |                        |
|---------------------------|----|------------------------|
| Martin Rolf Ammerbacher   | => | Ammerbacher, Martin R. |
| Roman Andreas Weizenhuber | => | Weizenhuber, Roman A.  |

### **Bsp 6:    URL zerlegen**

Lesen Sie eine URL von Tastatur und zerlegen Sie die URL in Protokoll, Host, Port, Pfad und Datei. Geben Sie die einzelnen Teile aus.

Zum Beispiel `http://www.orf.at:8080/content/aktuell/news.html` soll in `http`, `www.orf.at`, `8080`, `/content/aktuell/news.html` und `news.html` zerlegt werden.

Die Angabe des Ports ist optional.

**Bsp 7: Lauflängencodierung**

Die Lauflängencodierung ist eine Komprimierungstechnik, bei der jede Zeichenfolge, die aus mehr als zwei gleichen Zeichen besteht, durch das Zeichen und die Länge der Folge codiert wird. Die Eingabe von „ABBCCCDEEEEEE“ soll beispielsweise zu „ABB3CD5E“ werden. Schreiben Sie Methoden zur Komprimierung und Dekomprimierung eines Strings.

**Bsp 8: Mustersuche**

Geben Sie alle Zeilen einer Textdatei aus, die mit einem von Tastatur eingelesenen Muster übereinstimmen. Das Muster besteht aus einer beliebigen Zeichenfolge und einem optionalen \*-Zeichen, das als Platzhalter für beliebig viele beliebige Zeichen steht. Z.B. können mit dem Muster `Huber *` alle Personen mit Namen Huber der Namensregisterdatei aus Beispiel 5 ausgegeben werden.

Aus Gründen der Einfachheit ist nur ein \*-Zeichen als Platzhalter erlaubt.