

Liebe Schülerin, lieber Schüler,

wir freuen uns, Sie heute zum Landeswettbewerb des Sächsischen Informatikwettbewerbs 2017/2018 begrüßen zu können und wünschen Ihnen viel Erfolg, aber auch Freude bei der Lösung der Aufgaben.

Hier noch einige Hinweise:

- Für die Lösung der Aufgaben haben Sie 4 Stunden (240 min) Zeit.
- Als Hilfsmittel sind Standardsoftware, Programmierumgebungen sowie Taschenrechner und Tafelwerk zugelassen.
- **Standardsoftware:**
  - Office-Pakete (Microsoft Office; OpenOffice; LibreOffice)
  - Zeichenprogramme (Paint; Gimp; Inkscape)
- **Programmiersprachen (Entwicklungsumgebung):**
  - Pascal (Freepascal; Lazarus)
  - C, C++ (Dev C++)
  - Visual C (Microsoft Visual Studio; NetBeans)
  - Visual C++ (Microsoft Visual Studio; NetBeans)
  - Visual C# (Microsoft Visual Studio)
  - Visual Basic (Microsoft Visual Studio)
  - Python (Eclipse)
  - PHP (Eclipse; NetBeans; xampp [PHP])
  - Java (Eclipse; Java-Editor; NetBeans)
  - Java-Script (HTML Editor Phase 5)
  - Karol (Karol)
  - Scratch (Scratch)
- Der Start der Programme erfolgt über Startmenü / Alle Programme.
- Die Nutzung des Internet, eigener Datenträger und anderer Programme ist nicht gestattet.
- Vorgabedateien befinden sich im Ordner in **Z:\Vorgaben**.
- Speichern Sie alle Ergebnisse in den Ordnern **Z:\Ergebnisse\Aufgabe1** bzw. **Z:\Ergebnisse\Aufgabe2**. Sollten Sie ein anderes Arbeitsverzeichnis benutzen, müssen Sie nach Fertigstellung Ihrer Arbeit die Dateien (auch **\*.exe**, **\*.php** usw.) dorthin kopieren.
- Verwenden Sie als Dateinamen **<Anmeldename><aufgabe>.<typ>** (zum Beispiel **e04013A1b.doc**)
- Speichern Sie Ihre Lösungen der schriftlichen Aufgaben als Textdatei (z.B. mit Word)
- Benutzen Sie die bereitgestellten Zettel nur für Skizzen und ähnliches. Weitere Lösungszettel erhalten Sie bei Bedarf von der Aufsicht.

**Wir wünschen Ihnen viel Erfolg!**

### Hellseherautomaten

Hellseherautomaten (HA) kommunizieren mit ihrer Umwelt. Informationen der Umwelt (z.B. über das Auftreten von Frost oder ähnliches) bekommen sie über eine Eingabefolge von Nullen und Einsen. Die Eingabefolge entsteht dadurch, dass immer nach einer bestimmten Zeit eine neue Information (Null oder Eins) erfahren wird.

Grundsätzlich antwortet ein HA auch mit einer Ausgabefolge bestehend aus Nullen und Einsen, die aus Antworten resultierend aus der Eingabefolge entstehen.

Mit jedem Zeichen der Ausgabefolge versucht er das nächste kommende Signal aus der Umwelt (also aus der Eingabefolge) vorherzusagen. Würde also ein HA auf die Eingabefolge „0110010“ mit der Ausgabefolge „1100100“ oder „1100101“ antworten, wäre er perfekt, denn er hätte alle neu kommenden Informationen vollständig richtig vorhergesehen.

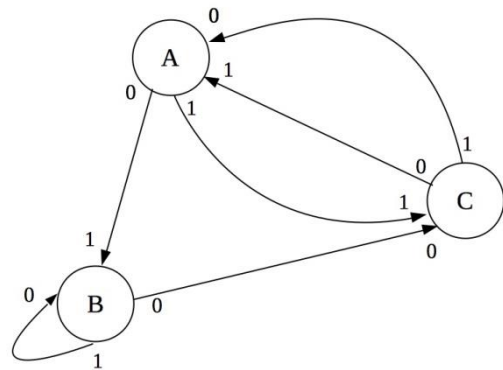


Abbildung 1

Die Abbildung 1 zeigt den speziellen HAL9000 und beschreibt, wie er die Ausgabefolge ermittelt. Das Einlesen beginnt immer im Zustand A. Die Beschriftung am Pfeilanzug steht für das gelesene Zeichen, der Pfeil zeigt auf den neuen Zustand und die Beschriftung an der Pfeilspitze steht für das Zeichen, wodurch der HA jeweils die Ausgabefolge ergänzt.

*Quelle: nach „Computerkurzweil“, Spektrum der Wissenschaft, 1988*

### Aufgaben:

- Begründen Sie, dass der HAL9000 bei der Eingabefolge „0111000010110“ die Ausgabefolge „1000011001000“ liefert. 2 Punkte
- Ermitteln Sie die Anzahl der in Aufgabe a) richtig vorhergesehenen Zeichen. Beschreiben Sie Ihr Vorgehen. 2 Punkte
- Ermitteln Sie eine mindestens 12-stellige Eingabeinformation, welche von HAL9000 vollständig richtig vorhergesehen wird. Beschreiben Sie Ihr Vorgehen. 3 Punkte
- Stellen Sie die Informationen aus der Abbildung 1 in einer Tabelle dar. 2 Punkte
- Erstellen Sie eine Computerlösung, welche nach Eingabe einer beliebigen Eingabefolge die entsprechende Ausgabefolge ausgibt. 4 Punkte
- Erweitern Sie Ihre Computerlösung so, dass diese jeweils die Anzahl der richtig vorhergesehenen Zeichen ausgibt. 2 Punkte

### Segelfetzen

Während eines großen Sturms wurde eine Gruppe von Matrosen schiffbrüchig und landete auf einer einsamen Insel im Mittelmeer. Da ihr Funkgerät kaputt ist, können sie nicht um Hilfe bitten und ihre einzige Überlebenschance besteht darin, ein Floß zu bauen. Allerdings brauchen sie ein Segel, um das Floß zu fortzubewegen.

Zum Glück haben sie das rechteckige Hauptsegel aus dem zerstörten Schiff gerettet, aber es ist in mehreren Fetzen gerissen. Daher müssen sie die Fetzen zusammennähen, um das ursprüngliche Segel wieder herzustellen.

Jeder der Fetzen ist ein Dreieck und alle Teile sind vorhanden. Weil auf dem Segel Streifen sind, ist es leicht, die korrekte Ausrichtung aller dreieckigen Teile zu identifizieren. Die Ausrichtung der Fetzen muss beibehalten werden, da sonst das Segel weniger widerstandsfähig gegen den Wind wäre und wieder zerreißen könnte.

Mehrere Stunden haben die Matrosen versucht, das Segel zusammenzusetzen, aber ohne Erfolg. Sie hoffen, dass der gerettete Schiffscomputer helfen kann.

*Quelle: Central European Olympiad in Informatics 2007, Brno*

#### Aufgaben:

- Stellen Sie den Sachverhalt mit mindestens 6 verschiedenen Segelfetzen zeichnerisch dar.  
2 Punkte
- Geben Sie eine Möglichkeit an, die Segelfetzen so zu beschreiben, dass ein Computerprogramm sie verarbeiten kann.  
2 Punkte
- Beschreiben Sie einen Algorithmus „Sturm“, mit dem ein (zufälliges) Zerreißen des Segels in mehrere dreieckige Fetzen simuliert werden kann. Die Größe des ursprünglichen Segels und die Anzahl der Fetzen sollen durch den Nutzer festgelegt werden können.  
5 Punkte
- Setzen Sie den Algorithmus aus c) in eine Computerlösung um. Das Ergebnis soll als Textdatei ausgegeben werden können. Beschreiben Sie den Aufbau dieser Datei. 6 Punkte
- Beschreiben Sie nun einen Algorithmus „Rettung“, mit dem das ursprüngliche Segel wieder zusammengesetzt werden kann. Beachten Sie, dass zwar die Ausrichtung aller Fetzen bekannt ist, nicht aber die Lage innerhalb des ursprünglichen Segels. 5 Punkte
- Setzen Sie den Algorithmus aus e) in eine Computerlösung um.  
Die Eingabedaten sollen in folgender Form aus einer Textdatei eingelesen (oder über die Zwischenablage kopiert) werden können.  
Zeile 1           Anzahl der Fetzen, Breite und Höhe des ursprünglichen Segels ( $n \times y$ )  
ab Zeile 2       Koordinaten der Eckpunkte eines Fetzens ( $x_A \ y_A \ x_B \ y_B \ x_C \ y_C$ )  
                  (mit korrekter Orientierung, aber nicht zwingend der richtigen Position)  
Mehrere Zahlen in einer Zeile werden immer durch ein Leerzeichen getrennt.  
Zum Testen Ihrer Lösung können Sie die Vorgabedateien `segel<x>.txt` verwenden.  
Die Lösung soll als Textdatei ausgegeben werden. Geben Sie an, welche Daten die Lösung enthalten muss und wie Ihre Lösungsdatei aufgebaut ist. 10 Punkte