



25. Bundeswettbewerb Informatik 2006/2007

Die Aufgaben der zweiten Runde

Allgemeine Hinweise

An dieser Runde dürfen nur Einzelpersonen teilnehmen, die in der ersten Runde in drei Aufgaben mindestens 12 Punkte erreicht haben oder einer Gruppe angehört haben, der dieses gelungen ist. Gruppenarbeit ist in der zweiten Runde nicht zulässig. **Einsendeschluss ist der 16. April 2007**, wobei das Datum des Poststempels gilt. Bitte adressiere deine Einsendung wieder an den **Bundeswettbewerb Informatik, Ahrstraße 45, 53175 Bonn**.

Es gibt drei Aufgaben. **Wichtig:** Eine Einsendung darf nur Bearbeitungen zu höchstens zwei Aufgaben enthalten, deren Bewertung dann das Gesamtergebnis ausmacht. Sollte eine Einsendung Bearbeitungen zu allen drei Aufgaben enthalten, werden wir zwei davon zufällig auswählen und nur diese bewerten.

Die Bearbeitung einer Aufgabe sollte zunächst eine einfache, nachvollziehbare und vollständige Lösung aller Teilaufgaben enthalten. **Pluspunkte** für eine höhere Bewertung kannst du erreichen, wenn du die Aufgabe dort, wo es möglich und sinnvoll ist, eigenständig weiterentwickelst. Dabei bevorzugen wir inhaltliche Erweiterungen und Verbesserungen, etwa von Datenstrukturen und Algorithmen, und legen keinen Wert auf aufwändige Tricks, z.B. zur reinen Verschönerung der Bedienungsoberfläche. Begründe für jede Erweiterung, weshalb sie sinnvoll ist und ihre Realisierung eine eigene Schwierigkeit darstellt.

Denke bitte daran, dass zur Bewertung möglicherweise nur die Papier-Unterlagen herangezogen werden können. Diese sollten also einen lückenlosen und nachvollziehbaren Nachweis des Leistungsumfangs und der Funktionstüchtigkeit der Programme geben. Der Umfang der Einsendung soll sich in Grenzen halten; eine gute Dokumentation vermittelt kurz und präzise alles Nötige, insbesondere die wesentlichen Lösungsideen. Nötig ist alles, was Interessierte mit guten Informatikkenntnissen, die die Aufgabenstellung kennen, wissen müssen, um die Lösungsidee zu verstehen und die Realisierung dieser Idee nachzuvollziehen. Generell sind gute und originelle Ideen zwar entscheidend, aber die Dokumentation hat schon oft den Ausschlag für oder gegen das Weiterkommen gegeben.

Grundsätzlich gelten die Gliederungs- und Dokumentationsrichtlinien der 1. Runde weiter. Zu jeder Teilaufgabe gehört also die Lösungsidee und die Dokumentation der Lösung sowie

des dazu gehörigen Programms (eine Beschreibung, wie die Idee in konkrete Programmelemente umgesetzt wurde, Hinweise auf Nutzungsgrenzen, Besonderheiten usw.). Dabei sind (halb-)formale Notationen besser als Programmausschnitte. Für die geforderten Programme erwarten wir Programmablaufprotokolle, also kommentierte Probeläufe des Programms, aus denen ersichtlich wird, wie das Programm sich in unterschiedlichen Situationen verhält. Sende uns außerdem bitte (abgedruckt!) aussagekräftige Ergebnisse von Programmläufen mit unterschiedlichen Daten. Komplettiert wird das Papier-Material durch den Programmtext, wobei unwichtige und automatisch generierte Teile nicht ausgedruckt werden sollen.

Schicke uns alles in lesbarer Form auf Papier, Schriftgröße mindestens 10 Punkt, bei Quelltext mindestens 8 Punkt. Bitte schicke uns deine Unterlagen auf losen, gelochten Blättern im Format DIN A 4 (Hüllen mit Lochrand nur bei ausreichender Stabilität verwenden; keine Heftstreifen oder Mappen) und gib auf jedem Blatt Verwaltungsnummer, Vorname, Name und Seitennummer an. Die Verwaltungsnummer steht auf der Teilnahmebescheinigung der ersten Runde. Bitte gliedere deine Einsendung in (a) Allgemeines, (b) Unterlagen zur ersten bearbeiteten Aufgabe und (c) Unterlagen zur zweiten bearbeiteten Aufgabe.

Außerdem sende uns bitte die Programmtexte und lauffähigen Programme auf einer CD oder DVD (3,5“-Disketten sind möglich, aber wegen ihrer Anfälligkeit für Beschädigungen beim Transport nicht zu empfehlen). Bei der Bewertung können Programme unter Windows (98-XP), Linux und MacOS X (10.3, 10.4) ausgeführt werden.

Fragen zu den Aufgaben dürfen per E-Mail an bwinfo@bwinfo.de oder telefonisch unter 0228-378646 (zu üblichen Arbeitszeiten) gestellt werden. Die Antwort auf E-Mail-Anfragen kann sich leicht verzögern. Informationen zur 2. Runde finden sich auf den Webseiten des BWINF (www.bwinfo.de). In der newsgroup fido.ger.bwinfo wird sicher wieder über die Aufgaben diskutiert werden – ohne Lösungsideen auszutauschen.

Allen Teilnehmern der zweiten Runde wird Anfang Juni die Bewertung mitgeteilt. Die Besten werden zur Endrunde eingeladen, die im Herbst von und bei SAP ausgerichtet werden wird. Dort werden die Bundessieger und Preisträger ermittelt und am letzten Tag ausgezeichnet. Bundessiegern winkt die Aufnahme in die Studienstiftung des deutschen Volkes. Außerdem werden Geld- und Sachpreise vergeben. Der Rechtsweg ist ausgeschlossen.

Die folgenden Informationen können vielleicht zusätzlich dazu motivieren, sich an der zweiten Runde aktiv zu beteiligen: Es ist prinzipiell möglich, eine Einreichung zur zweiten Runde als so genannte besondere Lernleistung in die Abiturwertung einfließen zu lassen; Genaueres erfährst du an deiner Schule. Jüngere Zweitrundenteilnehmer haben die Chance, zu einer Schülerakademie eingeladen zu werden. Und sehr gute Leistungen, die nur knapp nicht zur Qualifikation für die Endrunde ausgereicht haben, werden mit Buchpreisen des Verlags O'Reilly prämiert.

Viel Spaß und viel Erfolg!

MCI-Sonderpreis

Beim Bundeswettbewerb Informatik spielt die Mensch-Computer-Schnittstelle oder Bedienschnittstelle eines eingesandten Programms bei der Bewertung prinzipiell keine Rolle. Für die Benutzbarkeit von Informatiksystemen ist diese Komponente aber von ganz entscheidender Bedeutung, und so wird bei einigen Einsendungen zum BWINF erhebliche Mühe auf den Interaktionsaspekt verwendet.

Diese Mühe soll belohnt werden: Der Fachbereich Mensch-Computer-Interaktion (MCI) der Gesellschaft für Informatik (GI) schreibt zum siebten Mal einen Sonderpreis für besonders gelungene Bedienschnittstellen aus. Verliehen wird dieser Preis auf der Tagung „Mensch & Computer 2007“, die vom 2. bis 5. September 2007 in Weimar stattfinden wird.

Sie haben die Möglichkeit, sich mit einem der von Ihnen eingesandten Programme um den MCI-Sonderpreis bewerben. Beschreiben Sie dazu in einem separaten Dokument die Bedienschnittstelle dieses Programms. Geben Sie aber nicht nur eine Bedienungsanleitung, sondern erläutern Sie vor allem Entwurfskonzept und -entscheidungen.

Das Programm und seine Bedienschnittstelle werden nach folgenden Kriterien bewertet:

Dialogkriterien: Der Dialog eines Benutzers mit einem System ist

aufgabenangemessen, wenn das System den Benutzer bei der Ausführung der Aufgaben unterstützt, d.h. bei seinen Arbeitsschritten sinnvoll führt, ohne Handlungen unnötig einzuschränken.

selbsterklärend, wenn jeder Dialogschritt durch Reaktionen des Systems sofort verständlich wird oder dem Benutzer auf Anforderung erklärt wird;

kontrollierbar, wenn das System in jeder Situation auf die BenutzerInnen reagiert und diese agieren, d.h. Richtung und Geschwindigkeit der Interaktion bestimmen können, bis ihr Ziel erreicht ist;

fehlertolerant, wenn trotz offensichtlicher Fehler in der Eingabe das beabsichtigte Ergebnis ohne bzw. mit nur geringem Eingreifen der Benutzer erzielt werden kann.

Präsentationskriterien: Die Präsentation von Information (also z.B. auch der Bezeichnung von Dialogelementen) ist

fachlich gut gestaltet, wenn zusammengehörige Informationen räumlich gruppiert sind, ein „aufgeräumter“ Eindruck entsteht;

übersichtlich, wenn die Menge der Informationen knapp und strukturiert dargestellt wird;

lesbar, klar und präzise, wenn die (visuelle) Darstellung der Information leicht zu lesen bzw. erkennen ist und der Informationsgehalt schnell und genau und ohne überflüssige Informationen vermittelt wird.

BWINF-Kriterien: Ein Programm und seine Bedienschnittstelle sind

originell, wenn die Bedienschnittstelle ungewöhnlich und mit eigenen Mitteln (aber dennoch ergonomisch) gestaltet ist;

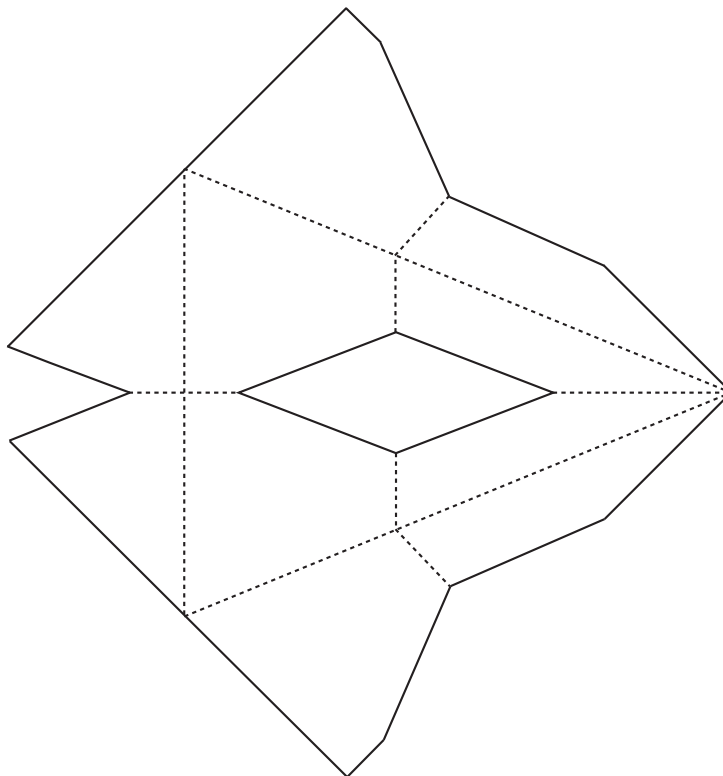
inspizierbar, wenn die Bedienschnittstelle vollen Zugang zur Funktionalität erlaubt, also z.B. Kontrollflüsse dargestellt werden oder Parameter beeinflusst werden können.

Aufgabe 1: Faltpolygone

Müde davon, immer nur Papierflieger zu bauen, hat die Schülerin Mia Giro in einer langweiligen Stunde damit angefangen, Papierbögen in neue, spannende Formen zu falten, sie dann mit einer Schere zu durchschneiden und sich nach dem Ausfalten über die schönen, symmetrischen Formen zu freuen. Ihre inzwischen auch begeisterte Lehrerin ist neugierig, welche Muster wohl mit Bögen gemacht werden können, die zehnmal oder mehr gefaltet werden. Da kein reales Papier dies mitmacht, müssen die entstehenden Objekte, die sie Faltpolygone nennen, im Rechner modelliert werden.

Aufgabe:

1. Beschreibe sinnvolle Operationen, die ein abstrakter Datentyp „Faltpolygon“ zur Verfügung stellen sollte. Faltpolygone fangen als rechteckige ungefaltete Bögen an und können danach wiederholt und in beliebiger Reihenfolge gefaltet, entlang einer Geraden (ganz) durchgeschnitten und ausgefaltet werden.
2. Schlage vor, wie ein Faltpolygon im Rechner dargestellt werden kann.
3. Programmiere entsprechend deiner Vorschläge in den Aufgabenteilen 1 und 2 eine Datenstruktur, die den Datentyp „Faltpolygon“ realisiert.
4. Zeige drei besonders schöne oder interessante ausgefaltete Faltpolygone, die mit deiner Datenstruktur erstellt werden können. Zeige auch die Folgen von Operationen, deren Ausführungen diese Faltpolygone erzeugen. Versuche, mit wenigen Schneideoperationen auszukommen.



Aufgabe 2: Bidoku

Sudoku ist ein sehr bekanntes Spiel. Ein 9×9 -Spielfeld besteht aus neun Zeilen und neun Spalten und ist zusätzlich in 3×3 quadratische Unterfelder der Größe 3×3 unterteilt. Das Ziel des Spieles besteht darin, in jedes Feld eine Ziffer zwischen 1 und 9 einzutragen, so dass jede Zeile, jede Spalte und jedes Unterfeld jeweils alle neun Ziffern von 1 bis 9 enthält.

Als Hommage an die Informatik beschäftigen wir uns im folgenden mit einer binären Variante dieses Spiels, welche wir *Bidoku* nennen wollen. Die Vorsilbe *Bi-* soll das Wort *binär* andeuten.

Bei *Bidoku* gibt es 16 Zeilen und 16 Spalten. Das Spielfeld ist außerdem in 4×4 Unterquadrate unterteilt, die wiederum aus 4×4 Feldern bestehen. Jedes Feld enthält anfangs eine 0 oder 1, oder es bleibt leer. Ziel des Spiels ist es, die Felder so mit Nullen und Einsen aufzufüllen, dass jede Zeile, jede Spalte und jedes Unterquadrat genau acht Einsen und acht Nullen enthält. Es darf aber nur eine Möglichkeit geben, die Felder so zu ergänzen.

0 1 1	0 1	0 0	0 1
0 0	0 1 1	1 0 1	0 1
0 0 0	0 1	0 1	0 1 0
1 1 1	1	0 1	1
1	0 0 1	0 0	0 1 0
1 0	0 0 0	0	1 0
0 1 1	0	1 0	1 0
1	1 0	0 0 0	1 0
0	0 0 1	0 0	0 0
0 1	0 0	1 0	0 0
1 0	1 1 0	1	1 0 0
0 0 0 0	1 1	0 0 1	
0	1	1	0
0 1 1	0	1 1	1 1
0 1 1 0		1 1 1	0 1
1	0	0 1 0	1 1

Ein Bidoku lässt sich im allgemeinen mit anderen Ideen als Sudoku lösen. In diesem Beispiel enthält die zehnte Spalte sechs Nullen. In ihr können wir also nur noch zwei Nullen hinzufügen. In der zweiten Spalte des vorletzten Unterquadrats enthält die Lösung also mindestens eine Eins, mit welcher dieses Unterquadrat schon acht Einsen enthält. Seine erste, dritte und vierte Spalte können daher mit Nullen gefüllt werden. Die elfte Spalte enthält dann acht Nullen und kann mit Einsen ergänzt werden.

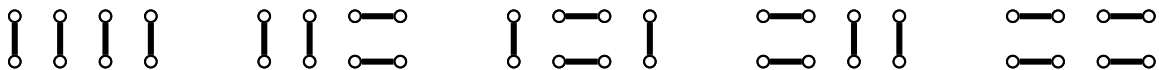
Aufgabe:

Schreibe ein Programm, das Bidoku-Rätsel erzeugt. Stelle sicher, dass die Lösung eindeutig ist und das Rätsel für den Spieler interessant ist. Insbesondere soll es nicht zu einfach zu lösen sein, wobei die Schwierigkeit der erzeugten Rätsel einstellbar sein soll.

Du kannst auch versuchen, ästhetisch schöne Muster aus vorbesetzten Feldern zu erzeugen. Besonders schön sind beispielsweise symmetrische Figuren.

Aufgabe 3: Folgenreiche Spielereien

Bei mathematischen Untersuchungen stößt man manchmal auf unbekannte unendliche Folgen von ganzen Zahlen, deren erste Glieder „von Hand“ berechnet werden können. Zählt man zum Beispiel, auf wie viele Arten $2n$ Menschen, die in zwei Reihen aufgestellt sind, sich mit jeweils einem Nachbarn (in derselben Reihe oder gegenüber) zu Paaren zusammenschließen können, so kommt man für $n = 1, 2, 3, 4, 5, \dots$ auf die Anzahlen $1, 2, 3, 5, 8, \dots$. Zum Beispiel gibt es für $n = 4$ die fünf Möglichkeiten



Häufig tritt eine solche Folge nicht zum ersten Mal in der Weltgeschichte auf, und es ist von großem Interesse zu erfahren, ob sie bereits bekannt ist und vielleicht einen Namen oder eine einfachere Definition hat. Es gibt ein Buch, gewissermaßen ein Folgenlexikon, das einem hilft, solche Fragen zu beantworten, und auch im Internet findet sich eine ähnliche Hilfe (<http://www.research.att.com/~njas/sequences/index.html>). Befragt man diese Quellen bezüglich der Folge $1, 2, 3, 5, 8, \dots$, so erfährt man, dass es sich, bis auf zwei fehlende erste Glieder, um die Fibonacci-Folge f_0, f_1, f_2, \dots handelt, die so definiert ist: $f_0 = 0$, $f_1 = 1$, und $f_n = f_{n-1} + f_{n-2}$ für $n \geq 2$.

Die Fibonacci-Folge ist durch eine Rekursionsformel gegeben, die jedes Folgenglied als Funktion von früheren Gliedern ausdrückt. Eine Folge kann auch durch Anwendung von Operatoren auf elementare Folgen entstehen. So erhält man die Folge $0, 1, 3, 6, 10, \dots$ der Anzahlen der paarweisen Begrüßungen, wenn n Freunde sich treffen, als $P(P(\mathbf{1}) - \mathbf{1})$, wobei $\mathbf{1}$ die konstante Folge $1, 1, 1, \dots$ und $-$ Subtraktion Glied für Glied bezeichnen und $P(a)$ für jede Folge a mit den Gliedern a_1, a_2, a_3, \dots die Folge $a_1, a_1 + a_2, \dots, \sum_{i=1}^n a_i, \dots$ der so genannten Präfixsummen von a ist.

Aufgabe:

Schreibe ein Programm, mit dem man ein derartiges (natürlich viel weniger umfangreiches) Folgenlexikon automatisch erstellen kann. Die oben erwähnten Quellen sind hauptsächlich durch eine aufwändige Literatursuche entstanden; hier soll stattdessen versucht werden, ein Folgenlexikon nach freier Wahl mit einem oder mit beiden oben erwähnten Ansätzen (Rekursionsformeln und Kombination von Elementarfolgen) aufzubauen. Gibt der Benutzer die ersten Glieder einer Folge an, soll das Programm sein Folgenlexikon nach Folgen durchsuchen, deren erste Glieder mit den angegebenen Zahlen übereinstimmen, und geeignete Beschreibungen oder eventuell weitere Glieder der gefundenen Folgen ausgeben. Beschreibe, wie dein Folgenlexikon aufgebaut wird, begründe die Form der Antworten deines Programms an den Benutzer und illustriere die Fähigkeiten, aber auch die Beschränkungen deines Programms durch fünf oder mehr geeignete Beispiele.