

19. Night.Dance.

Das Aufgabenblatt
Bundeswettbewerb Informatik



Grußwort des Präsidenten der Kultusministerkonferenz



Willi Lemke

Die neuen elektronischen Informations- und Kommunikationsmedien verändern unsere Gesellschaft in revolutionärer Weise. Mit ihnen vernünftig und geschickt, kreativ und einfallsreich umgehen zu können, ist eine zentrale Qualifikation für die moderne Arbeitswelt. Kinder und Jugendliche nutzen die neuen Techniken in Familie und Freizeit bereits ganz selbstverständlich. Genauso gehört die Vermittlung von Medienkompetenz heute fraglos zum Bereich der Allgemeinbildung und tritt in der Schule als moderne Kulturtechnik neben Lesen, Schreiben, Rechnen und das Erlernen von strukturiertem Denken.

In diesem Zusammenhang leistet der Bundeswettbewerb Informatik nach Auffassung der Kultusminister und -senatoren der Länder in der Bundesrepublik Deutschland seit langem einen wertvollen Beitrag. Er fordert mit seinen Aufgaben die Jugendlichen in Schule und Berufsausbildung in besonderer Weise heraus, sich mit Inhalten und Methoden der Informatik, mit Möglichkeiten des Einsatzes und der Anwendung von informationstechnischen Systemen und auch mit Fragen der Bewertung und des verantwortlichen Umgangs mit Informationen zu befassen. Im Interesse der Weiterentwicklung der informationstechnischen Bildung ist zu wünschen, dass sich viele Jugendliche von den Aufgaben des 19. Bundeswettbewerbs Informatik angesprochen fühlen und in ihren Schulen die für eine erfolgreiche Teilnahme förderliche Unterstützung finden.

Willi Lemke

Präsident der Ständigen Konferenz der Kultusminister in der Bundesrepublik Deutschland
Senator für Bildung und Wissenschaft der Freien Hansestadt Bremen

Gesellschaft für Informatik e.V.

Die Gesellschaft für Informatik e.V. (GI) ist mit rund 20.000 Mitgliedern die größte deutsche wissenschaftliche Fachgesellschaft in ihrem Bereich. In der GI wirken Männer und Frauen am Fortschritt der Informatik mit, vor allem im wissenschaftlich-fachlichen Austausch in etwa 200 verschiedenen Fachgruppen. Ihre Mitglieder kommen aus allen Sparten der Wissenschaft, der Informatikindustrie, aus dem Kreis der Anwender sowie aus Lehre, Studium und Ausbildung. Ihr gemeinsames Ziel ist die Förderung der Informatik in Forschung, Lehre und Anwendung, die gegenseitige Unterstützung bei der Arbeit sowie die Weiterbildung. Weitere Info: www.gi-ev.de

GMD - Forschungszentrum Informationstechnik GmbH

Die GMD - Forschungszentrum Informationstechnik GmbH, eine der 16 Großforschungseinrichtungen der Bundesrepublik Deutschland, betreibt mit ihren ca. 1200 wissenschaftlich-technischen Mitarbeitern Forschung, Entwicklung und Transfer im Bereich der Informationstechnik, Informatik, Kommunikationstechnik und Medien. Die Forschungs- und Entwicklungsaufgaben reichen von der Grundlagenforschung bis zur konkreten Zusammenarbeit mit der Industrie in gemeinsamen Kooperationsvorhaben.



bmb+f

Unter der
Schirmherrschaft
des
Bundespräsidenten

Träger des Wettbewerbs:
Bundesministerium für Bildung und Forschung,
Gesellschaft für Informatik e.V. (GI),
GMD-Forschungszentrum Informationstechnik GmbH

Allgemeine Hinweise

► Zu den Aufgaben sende uns jeweils Folgendes:

Lösungsidee:

Eine Beschreibung der Lösungsidee, unabhängig vom eingesandten Programm. Die Idee und die zu ihrer Beschreibung verwendeten Begriffe müssen sich aber im Programm wiederfinden.

Programm-Dokumentation:

Eine Beschreibung, wie die Lösungsidee im Programm und seinen Teilen realisiert wurde. Hinweise auf Besonderheiten und Nutzungsgrenzen, verlangte Eingabeformate usw.

Programm-Ablaufprotokoll:

Kommentierte Probeläufe des Programms, d.h. wann wird welche Eingabe erwartet, was passiert dann, welche Ausgabe erscheint usw. Mehrere unterschiedliche Beispiele, die die Lösung der Aufgabe und das Funktionieren des Programms verdeutlichen (teilweise in den Aufgabenstellungen vorgegeben). Bildschirm-Fotos sind auch zulässig.

Programm-Text:

Den kommentierten Quelltext des Programms in einer der gängigen höheren Programmiersprachen wie z. B. Pascal. Maschinensprache ist nicht zulässig. Den Programmtext bitte ausdrucken, dabei aber auf nicht selbst geschriebene Teile (z.B. verwendete Funktionen der Entwicklungsumgebung, automatisch generierter Programmtext) verzichten.

Programm:

Das lauffähige Programm selbst auf einer 3,5"-Diskette oder CD, die unter DOS/Windows95 gelesen werden kann. Die Diskette/CD muss auch den Programm-Text enthalten.

► Einsendungen werden danach bewertet,

- ob sie vollständig und richtig sind,
- ob die Ausarbeitungen gut strukturiert und verständlich sind und
- ob die (Programm-) Unterlagen übersichtlich und lesbar sind.

Für den Sonderpreis „**Mensch-Computer-Interaktion**“ wird unter anderem bewertet, in wie weit die Benutzungsschnittstelle eines eingesandten Programms ohne Erläuterung korrekt bedienbar ist und ob sie die Auswertung des Programms erleichtert. Originelle Ideen spielen eine große, rein grafischer Aufwand spielt hingegen keine entscheidende Rolle.

Bitte schicke deine Arbeit nach Aufgaben geordnet und geheftet auf einseitig bedrucktem DIN-A4-Papier. Endlospapier schneide bitte entsprechend zu. Nummeriere alle Blätter rechts oben und versieh sie mit deinem Namen. Die Texte sollen in Deutsch abgefasst sein. Falls du einige DIN-A4-Klarsichthüllen **mit Lochrand** zur Hand hast, stecke bitte jeweils alles, was zu einer Aufgabe gehört, in eine solche Hülle. Andernfalls **loche** die Blätter bitte. Beschrifte die Diskette oder CD, die die Programme und Programm-Texte enthält, mit deinem Namen. Fülle das Anmeldeformular (Klappe des Aufgabenblattes oder eine Kopie davon) **vollständig, korrekt und leserlich!** aus. Insbesondere das **Geburtsdatum** muss angegeben sein, denn sonst kann die Einsendung nicht korrigiert werden. Teilnehmer, die die Schule bereits verlassen haben, geben bei „Klassenstufe“ bitte an, was sie zur Zeit machen. Gib bitte auch an, ob und mit welchem Programm (nur eins) du dich um den Sonderpreis „Mensch-Computer-Interaktion“ bewerben möchtest. Gib für dieses Programm genau an, unter welchen Bedingungen (möglichst wenige) es lauffähig ist.

Bei Gruppen muss jeder Teilnehmer ein Formular ausfüllen, **Gruppenmitglieder ohne Anmeldeformular werden nicht gewertet!** Für die Anmeldung gibt es auf www.bwinf.de auch ein Online-Formular, das vor dem Einsendeschluss ausgefüllt werden kann. Bei der Online-Anmeldung wird eine Kennnummer vergeben; bei der Einsendung muss das Papierformular nur noch mit dieser Nummer, dem Namen und einer Unterschrift versehen werden. Wer sich online anmeldet, erhält nach der Einsendung eine Eingangsbestätigung per E-mail.

Einsendungen per E-mail oder nur auf Diskette/CD ohne Ausdruck können wir leider nicht akzeptieren.

► **Sende alles an:**

Bundeswettbewerb Informatik
Ahrstraße 45
53175 Bonn

Für Fragen zu den Aufgaben gibt es eine Hotline:

Telefonisch unter **0228 / 37 86 46** jeweils von 17-19 Uhr am 18.9., 4.10., 30.10. und 9.11.
oder E-mail an: bwinf@bwinf.de

oder schriftlich an die obige Adresse bzw. per Fax an 0228 / 37 29 000.

Informationen (FAQ) gibt es auch im Internet: www.bwinf.de

Meinungen und Fragen (aber keine Lösungsideen) zu den Aufgaben können auch in der BWINF-Newsgroup fido.ger.bwinf ausgetauscht werden.

Einsendeschluss ist der 13. November 2000 (Datum des Poststempels).

Verspätete Einsendungen können nicht berücksichtigt werden. Der Rechtsweg ist ausgeschlossen. Die Einsendungen werden nicht zurückgegeben. Der Veranstalter erhält das Recht, die Beiträge in geeigneter Form zu veröffentlichen.

Wer wissen möchte, ob seine Einsendung angekommen ist, kann eine an sich selbst adressierte und frankierte Postkarte mitschicken oder sich online mit E-mail-Adresse anmelden.

Dominiks

Domino-Desaster

Peter Pingel ist ein penibler Mensch, und in seinem Spielwarengeschäft geht ihm Ordnung über alles. Nun ist ausgerechnet das letzte Dominospiel seines Geschäftes aus dem Regal gefallen. Die Schachtel ist aufgegangen, und die 28 verschiedenen Steine sind auf dem Boden verstreut.

Dominik, der sein Praktikum in Peter Pingels Laden macht, soll sie wieder aufsammeln und in die flache Kiste legen. Doch Pingel ist nicht nur penibel, sondern auch eigenwillig, und so hat er sich eine ganz besondere Ordnung für die Dominosteine ausgedacht. Pingel gibt Dominik einen Zettel, auf dem er notiert hat, wie die einzelnen Steine nebeneinander in der Schachtel lagen. Dabei hat Pingel die numerischen Augenzahlen notiert. Als Dominik den Zettel genauer betrachtet, stellt er aber fest, dass die Abgrenzungen der Steine auf dem Zettel nicht mehr sichtbar sind. So weiß Dominik nicht, ob die Steine waagrecht oder senkrecht in der Schachtel lagen. So könnte der Stein  mit den Augenzahlen 4 und 2 zum Beispiel in horizontaler Position in der obersten Reihe, an zwei Stellen vertikal in der ersten Spalte oder vertikal in der zweiten Spalte liegen.

Der Zettel sieht so aus:

4	2	2	2	6	6	2	2
4	4	1	3	1	3	2	0
2	5	5	5	1	3	3	0
4	0	1	5	3	5	2	6
4	1	1	1	3	3	5	6
6	6	5	5	3	1	0	6
6	4	4	0	4	0	0	0

Aufgabe

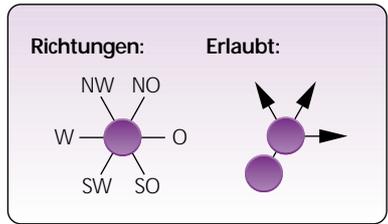
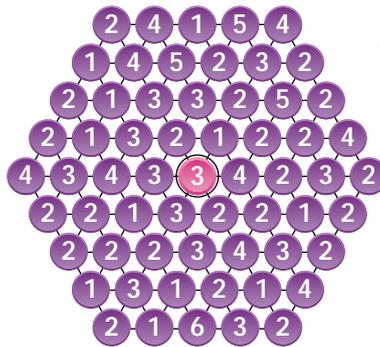
Schreibe ein Programm, das aus der Zeichnung von Peter Pingel eine Anordnung der Dominosteine in der Schachtel bestimmt. Ist das die Anordnung, die Peter Pingel gemeint hat? Begründung!



Zahlen-Hüpfen

Auf dem unten abgebildeten sechseckigen Spielbrett soll ein Spielstein so viele Felder in gerader Linie weiterhüpfen, wie die Zahl auf dem Feld, auf dem er steht, angibt. Begonnen wird auf dem Feld in der Mitte.

Der Anfangszug darf in eine beliebige der 6 möglichen Richtungen erfolgen. Bei jedem weiteren Zug darf jeweils nur in der selben Richtung weitergehüpft werden — oder in eine rechts oder links benachbarte Richtung. Die Hüpferei ist erfolgreich zu Ende, wenn das Ausgangsfeld wieder erreicht wird.



Aufgabe:

Schreibe ein Programm, welches einen erfolgreichen Hüpfweg sucht und ein Protokoll des Weges ausgibt (pro Hüpfen: die Richtung und Hüpfweite). Falls das Ausgangsfeld nicht erreicht werden kann, soll dies an Stelle des Protokolls ausgegeben werden.



Welche Sprache?

Das Übersetzungsbüro Eurobla fertigt Übersetzungen von verschiedenen europäischen Sprachen in andere europäische Sprachen an. Als Hilfe stehen den Übersetzerinnen und Übersetzern für einige Sprachen Softwarepakete zur Verfügung, die Rohübersetzungen anfertigen. Bevor man solch eine Software auf eine fremdsprachliche Textdatei anwenden kann, muss man natürlich wissen, um welche Sprache es sich handelt. Wünschenswert wäre es, die Entscheidung, in welcher Sprache ein vorliegender Text abgefasst ist, zu automatisieren.

Beispiele

- Kommt in einem Text häufig die Buchstabenkombination „th“ vor, könnte es sich um Englisch handeln.
- Kommt in einem Text das Zeichen „ß“ vor, handelt es sich mit Sicherheit um Deutsch.

Aufgabe

1. Entwickle Kriterien für die Sprachen **Englisch, Französisch, Italienisch, Spanisch, Türkisch** und **Deutsch**, die zur Entscheidung zwischen diesen Sprachen dienen können.
2. Schreibe ein Programm, das eine Textdatei einliest, sie nach den oben formulierten Kriterien analysiert und ausgibt, um welche Sprache es sich handeln könnte.
3. Schicke uns 3 Textbeispiele, die dein Programm richtig erkannt hat. Gib zusätzlich zu den Texten an, wie dein Programm zu den Ergebnissen gekommen ist.
4. Beschreibe mit Begründung, wie ein Text beschaffen sein könnte, den dein Programm nicht richtig erkennt.

Wichtig, wichtig



Die Firma Wo?Da! betreibt ein Mobilfunknetz und möchte den Nutzern ihres Kurznachrichtendienstes etwas ganz Besonderes bieten. Wichtige Leute haben ja meist jemanden, der die Post vorsortiert. Handy-Nutzer müssen aber bisher mit ihren Kurznachrichten ganz unsortiert klar kommen. Das soll sich ändern! Ein virtuelles Helferlein soll automatisch und individuell für jeden Kunden die Nachrichten nach Wichtigkeit sortieren. Kurznachrichten enthalten Angaben über den Absender, einen Betreff („subject“; max. 5 Worte) und den eigentlichen Text. Im Handy-

Display erscheint zunächst der Absender. Wird eine Nachricht angewählt, erscheint der Betreff, und erst nach weiterer Bestätigung der Text. Nachrichten können jeweils nach Anzeige des Absenders oder Betreffs gelöscht werden. Weiterhin kann die Nachricht nach dem Lesen gespeichert oder gelöscht werden. Für das Sortieren nach Wichtigkeit sollen die Nutzerreaktionen berücksichtigt werden. Dazu werden eingetragene Nachrichten laufend in Form einer Liste protokolliert. Ein Beispiel:

Eingetroffene Nachrichten:

DISPLAY Absender	Reaktion	DISPLAY Betreff	Reaktion	DISPLAY Inhalt	Reaktion
Alex	Weiter	I love you	Weiter	Speichern
Chef	Weiter	Allgemeine Mitteilung	Löschen		
Willi	Löschen				
Mama	Weiter	Sofort anrufen	Weiter	Löschen
Chef	Weiter	Bitte zu mir	Weiter	Löschen
Alex	weiter	Brauche Geld	Löschen		

Aufgabe:

- Entwickle und beschreibe eine Datenstruktur, mit der die einzelnen Kurznachrichten und die Nutzerreaktionen geeignet dargestellt werden können
- Schreibe ein Programm, das ausgehend von einer Reihe von Nutzerreaktionen auf bisherige Kurznachrichten die Wichtigkeit einer neuen Nachricht bestimmt. Zeige an drei Beispielen, wie neue Nachrichten nach ihrer Wichtigkeit sortiert werden können. Dabei muss eines der Beispiele das obige Beispielprotokoll mit den folgenden neuen Nachrichten verbinden:

Absender	Betreff
Chef	Bitte anrufen
Lala	Winkwinke
Willi	Sofort anrufen
Mama	Brauchst du Geld
Alex	Brauche Schatz

- Beschreibe kurz die Kriterien, nach denen dein Programm die Nachrichten sortiert, als Grundlage für einen Werbespot.
- Sollen die durch den Betrieb des Programms entstehenden persönlichen Informationen (wie Gewichtungskriterien, Nutzerreaktionen) auf dem Handy oder bei Wo?Da! verarbeitet werden? Begründe deine Entscheidung!

Tänzer ▶ ▼ Takt	T1	T2	T3	T4	T5
0	K1	K3	K5	K8	K7
1	K2	K4	K6		
2				K3	
3	K1		K5		K8
4		K6			

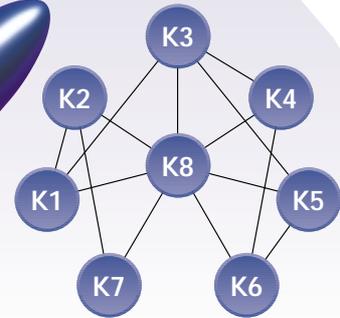
Night.Dance

Die Choreografin Uschi Paban markiert mögliche Positionen auf der Bühne durch Kreise. Sie zeichnet zwischen einigen dieser Kreise gerade Verbindungslinien. Die Aufgabe von Uschi Paban ist es nun, jedem Tänzer zu sagen, wann und über welche Verbindungslinien er sich bewegen soll.

Die Choreografin bestimmt eine Choreografie nach folgenden Regeln:

- ▶ In einigen, aber nicht in allen Kreisen steht ein Tänzer.
 - ▶ Tänzer springen immer zu Beginn eines Taktes.
 - ▶ Ein Tänzer kann sich in einem gigantischen Sprung von seinem Kreis entlang einer Verbindungslinie in einen anderen Kreis bewegen, der schon zu Beginn des Taktes frei sein muss.
 - ▶ Die während eines Taktes benutzten Verbindungslinien dürfen sich nicht kreuzen, um blutige Zusammenstöße in der Luft zu vermeiden.
 - ▶ Jeder Tänzer hat nach seinem Sprung mindestens einen Takt lang eine Pause.
 - ▶ In jedem Takt sollen sich möglichst viele Tänzer bewegen.
- Gibt es nach diesen Regeln mehr als eine Möglichkeit zu springen, so wird zufällig entschieden.

Die folgende Grafik zeigt beispielhaft eine von Uschi Pabans Bühnenmarkierungen:



Eine aus einer Bühnenmarkierung entwickelte Choreografie wird als Tabelle dargestellt, in der die berechneten Bewegungsabläufe notiert sind. Hier ein Beispiel, das auf der Bühnenmarkierung beruht, die die obige Grafik zeigt:

Aufgabe:

Schreibe ein Programm, das die Anfangspositionen der Tänzer (Takt 0) einliest und nach den obigen Regeln eine Choreografiertabelle erstellt. Vorgegeben sind die Kreise und die Verbindungslinien, wie sie in der Grafik dargestellt sind. Liefere für drei unterschiedliche Anzahlen und verschiedene Positionen von Tänzern Choreografiertabellen, die die Bewegungen der Tänzer während der ersten 20 Takte beschreiben. Die erste dieser Tabellen soll die Beispieltabelle auf 20 Takte erweitern.

19. Bundeswettbewerb Informatik 2000/2001. 1. Runde

Bitte in Druckschrift ausfüllen und Zutreffendes ankreuzen.



Geburtsdatum Name, Vorname

Straße

männlich

weiblich

Postleitzahl Wohnort

Telefon: Vorwahl / Durchwahl

Bundesland der Schule

Name der (ehemaligen) Schule

Schultyp

Informatiklehrer/in

Straße der Schule

Postleitzahl Schulort

Klassenstufe (1-13)

Namen anderer Gruppenmitglieder



Bewerbung um MCI-Sonderpreis / mit welcher Aufgabe?

Wieviele Stunden hast du gebraucht ?

Was hast du bei der Verarbeitung verwendet? Welche/s/n...

Programmiersprache? Betriebssystem? Computertyp? eigener

Hast du an anderen Wettbewerben teilgenommen?

Mathematik

Informatik

sonstige

Wie hast du vom Wettbewerb erfahren?

schon mal teilgenommen

Schule

ehemalige Teilnehmer

sonstiges

Diese Daten werden an niemanden weitergegeben, haben keinen Einfluss auf die Bewertung, aber dienen statistischen Zwecken. Sie werden ausschließlich für die Zwecke des Bundeswettbewerbs Informatik ausgewertet.

Ich bin mit der Computerspeicherung dieser Daten einverstanden und versichere, dass ich die Aufgaben selbstständig bzw. mit den angegebenen Gruppenmitgliedern bearbeitet habe.

Datum

Unterschrift

Bundeswettbewerb Informatik

Der Bundeswettbewerb Informatik wurde 1980 von der Gesellschaft für Informatik e.V. (GI) auf Initiative von Prof. Dr. Volker Claus ins Leben gerufen. Ziel des Wettbewerbs ist es, Interesse an der Informatik zu wecken und zu intensiver Beschäftigung mit ihren Inhalten und Methoden sowie den Perspektiven ihrer Anwendung anzuregen. Er gehört zu den bundesweiten Schülerwettbewerben, die von den Kultusministern der Länder unterstützt werden. Gefördert wird er von der Bundesministerin für Bildung und Forschung. Er steht unter der Schirmherrschaft des Bundespräsidenten. Die Träger des Wettbewerbs sind die GI und die GMD - Forschungszentrum Informationstechnik GmbH. Die Gestaltung des Wettbewerbs und die Auswahl der Sieger obliegen dem Beirat; Vorsitzender: Prof. Dr. Ingo Wegener, Universität Dortmund. Die Auswahl und Entwicklung von Aufgaben und die Festlegung von Bewertungsverfahren übernimmt der Aufgabenausschuss; Vorsitzende: StD Vera Reineke, Hannover. Die Geschäftsstelle des Wettbewerbs ist in Bonn und ist für die fachliche und organisatorische Durchführung zuständig; Geschäftsführer: Dr. Wolfgang Pohl.

Start und Ziel im September

Der Wettbewerb beginnt und endet im September, dauert etwa ein Jahr und besteht aus drei Runden. In der ersten und zweiten Runde sind fünf bzw. drei Aufgaben zu Hause selbstständig zu bearbeiten. Dabei können die Aufgaben der ersten Runde ohne größere Informatikkenntnisse gelöst werden; die Aufgaben der zweiten Runde sind deutlich schwieriger. In der ersten Runde ist Gruppenarbeit zugelassen und erwünscht. An der zweiten Runde dürfen jene teilnehmen, die allein oder zusammen mit anderen wenigstens drei Aufgaben weitgehend richtig gelöst haben. In der zweiten Runde ist dann eigenständige Einzelarbeit gefordert. Die Bewertung erfolgt durch eine relative Platzierung der Arbeiten. Die ca. dreißig bundesweit Besten werden zur dritten Runde, einem Kolloquium, eingeladen. Darin führt jeder ein Gespräch mit je einem Informatiker aus Schule und Hochschule und analysiert und bearbeitet im Team zwei Informatik-Probleme.

Wer ist teilnahmeberechtigt?

Teilnehmen können Jugendliche, die nach dem 13.11.1978 geboren wurden. Sie dürfen jedoch zum 1.9.2000 noch nicht ihre Ausbildung abgeschlossen oder eine Berufstätigkeit aufgenommen haben. Ebenfalls ausgeschlossen sind Personen, die zum Wintersemester 2000/2001 oder früher ihr Studium an einer Hochschule/Fachhochschule aufnehmen bzw. aufgenommen haben. Jugendliche, die nicht deutsche Staatsangehörige sind, müssen wenigstens vom 1.9. - 13.11.2000 ihren Wohnsitz in Deutschland haben oder eine staatlich anerkannte deutsche Schule im Ausland besuchen.

Als Anerkennung ...

Jeder Teilnehmer der ersten Runde erhält eine Urkunde für die Teilnahme bzw. die erfolgreiche Teilnahme bei richtiger Lösung von wenigstens drei der fünf Aufgaben. Erfolgreiche Teilnahme berechtigt zum Eintritt in die zweite Runde. Die Besten der zweiten Runde werden zur dritten Runde eingeladen, die im Herbst 2001 von der Deutsche Börse Systems AG ausgerichtet werden wird. Die dort ermittelten Bundessieger werden in der Regel ohne weiteres Aufnahmeverfahren in die Studienstiftung des deutschen Volkes aufgenommen. Zusätzlich sind für den Bundessieg, aber auch für andere besondere Leistungen Geld- und Sachpreise vorgesehen, unter anderem ein Aufenthalt an einer Sommerschule in den USA sowie Einladungen zu Schülerakademien in Deutschland.

... Teilnahme an der Informatik-Olympiade

Aus dem Teilnehmerkreis der Endrunde werden die Kandidaten festgelegt, aus denen im Laufe mehrerer Trainingsrunden das vierköpfige deutsche Team ausgewählt wird, das an der Internationalen Olympiade in Informatik 2002 in Korea teilnimmt.

Sonderpreis Mensch-Computer-Interaktion

Die Nützlichkeit von Informatiksystemen hängt im wesentlichen von der Qualität ihrer Kernbestandteile ab, wie sie u.a. die verwendeten Algorithmen und Datenstrukturen darstellen. Beim Bundeswettbewerb Informatik werden deshalb bei der Bewertung nur die Kernbestandteile der eingesandten Programme gewertet, während die Bedienschnittstelle keine Rolle spielt. Die Benutzbarkeit vieler Informatiksysteme hängt aber von dieser Komponente ab, die die Interaktion zwischen Mensch und Computer ermöglicht. In den letzten Jahren wurde auch bei den Einsendungen zum Bundeswettbewerb Informatik immer mehr Mühe auf die Mensch-Computer-Interaktion verwendet. Diese Mühe soll nun belohnt werden: Zum ersten Mal wird beim 19. Bundeswettbewerb Informatik in der ersten Runde ein Sonderpreis „Mensch-Computer-Interaktion“ für besonders gelungene Benutzerschnittstellen ausgeschrieben, in Zusammenarbeit mit dem Fachausschuss 2.3 „Mensch-Computer-Interaktion“ der Gesellschaft für Informatik e.V. Hauptgewinn ist ein Notebook, das von IBM Deutschland gestiftet wird. Weiterhin wird der Preis unterstützt von den Firmen humanIT, Sankt Augustin, und User InterfaceDesign, München.

Der Geschäftsführer und sein Handy

Trudchen Taste ist Assistentin von Otto Heitz, Geschäftsführer einer Firma für Sicherheitsschlösser, die erst kürzlich ihre Produktpalette durch elektronische Schlösser ergänzt hat. Heitz hat Schwierigkeiten mit dem neuen Handy, das Trudchen ihm empfohlen hat, damit er in Zukunft auch auf dem Golfplatz wichtige Telefonate führen kann. Das Handy verfügt über einen Rufnummernspeicher, der bei Eingabe eines Namens automatisch die richtige Nummer wählt. Der Name wird dabei über die Zifferntastatur eingegeben, die folgendermaßen belegt ist:



Da die Tasten mehrfach belegt sind, wird der jeweils richtige Buchstabe durch mehrmaliges Drücken der Taste ausgewählt. Die Buchstaben werden durch Eingabepausen voneinander getrennt. So ist z.B. Trudchens Nummer unter "Taste" gespeichert, also durch 8<Pause>2<Pause>7777<Pause>8<Pause>33 erreichbar.

Heitz ist das zu kompliziert: er fragt sich, warum es nicht reicht, jede Taste einmal zu drücken (also 82783 für Trudchen). Trudchen erklärt ihm geduldig, dass dann Mehrdeutigkeiten auftreten können. So hätte Heitz' Geschäftspartner Viktor Vapud dann die gleiche Tastenkombination für seine Nummer wie sie selbst. Um Heitz die Sache zu vereinfachen, gibt Trudchen jedoch eine Änderung der Software in Auftrag. Diese erlaubt es Heitz, tatsächlich jede Taste nur einmal zu drücken und dann aus den passenden Namen den richtigen durch das Drücken weiterer Tasten auszuwählen.

Aufgabe:

- ▶ Schreibe ein Programm, das eine Liste mit Namen und Telefonnummern (keine Umlaute oder ß, Groß-/Kleinschreibung spielt keine Rolle) einliest und dann Telefonnummern anhand einer Eingabe auf der numerischen Tastatur herausucht.
- ▶ Wird dein Programm langsamer, wenn die Liste deutlich länger wird? Beschreibe eine Datenstruktur, die in der Lage ist, die Suchzeiten bei größer werdender Liste gleich zu halten.

Lösungsidee:

Die Namen der Gesprächspartner werden durch Ersetzung der Buchstaben durch entsprechende Ziffern in die (nicht eindeutigen) Ziffernfolgen (auch Suchfolgen genannt) umgewandelt, die auf dem Handy bei der Suche verwendet werden sollen. In einer Hashtabelle werden die Zuordnung von jeder dieser Suchfolgen zu einem oder mehreren Namen (z.B. 82783 zu Vapud und Taste) und den zugehörigen Telefonnummern gespeichert. Bei der Suche werden alle Einträge, die einer eingegebenen Ziffernfolge zugeordnet sind, zusammen mit weiteren, diesmal eindeutigen Ziffern ausgegeben. Der Benutzer kann den gewünschten Eintrag durch Eintippen der passenden Ziffer(n) auswählen, worauf das Handy die richtige Nummer wählt.

Programm-Dokumentation:

Das Programm ist in Perl geschrieben, so dass für die Hashtabelle, die die Zuordnung zwischen Suchfolgen und Namen/Telefonnummern speichert, die entsprechenden in Perl eingebauten Mechanismen verwendet werden.

Nach dem Start (perl handy.pl) öffnet das Programm die Datei telnum.lst. Sie enthält die Namen und Telefonnummern der Gesprächspartner in der folgenden Form:

```
Trudchen Taste:766235
```

Die einzelnen Einträge in der Datei werden in Vorname, Nachname und Telefonnummer zerlegt. Aus den Buchstaben des Nachnamens wird die entsprechende Suchfolge gebildet. Jeder Eintrag wird unter seiner Suchfolge in der Hashtabelle %eintraege abgespeichert.

Danach kann der Benutzer eine Ziffernfolge eingeben. Alle unter dieser Suchfolge abgespeicherten Einträge werden aus der Hashtabelle ausgelesen. Wenn es mehrere sind, wird jedem Eintrag im nächsten Schritt eine eindeutige weitere Ziffernfolge zugewiesen. In einer temporären Hashtabelle wird dann die Telefonnummer des Eintrags unter dieser weiteren Ziffernfolge abgespeichert. Wenn der Benutzer nun seine Auswahl eingibt, kann die Telefonnummer aus der Hashtabelle ausgelesen und vom Handy gewählt werden.

Programm-Ablaufprotokoll

In allen Beispielen wird zuerst die Eingabedatei eingelesen, die u.a. folgende Einträge enthält:

```
Trudchen Taste:766235
Viktor Vapud:978614
Dunkel Kammer:752452
Steffen Jammer:880123
Beispiel 1:
Name (Kurzform) eingeben: 82783
```

Wählen Sie einen Gesprächspartner aus!

1. Trudchen Taste: 766235
2. Viktor Vapud: 978614

Ihre Wahl: 1

```
Wähle 766235
... und zwei weitere Beispiele
```

2. Teilaufgabe

Wird dein Programm langsamer, wenn die Liste deutlich länger wird? Beschreibe eine Datenstruktur, die in der Lage ist, die Suchzeiten bei größer werdender Liste gleich zu halten.

Das Programm verwendet für die Speicherung der Daten eine Hashtabelle. Die Daten werden unter einem Schlüssel abgelegt; hier ist die Suchziffernfolge der Schlüssel. Eine Hashtabelle basiert auf einem Feld mit M Speicherplätzen, die direkt adressiert werden können (ein Array also). Anhand des Schlüssels können Daten aus einer Hashtabelle ausgelesen werden. Der Schlüssel wird in eine Zahl k umgewandelt. Aus dieser Zahl wird durch weitere Rechenoperationen die Adresse ermittelt, unter welcher der gesuchte Datensatz gespeichert ist. Im einfachsten Fall entspricht die Adresse dem Divisionsrest aus k und M ($k \bmod M$). Die Speicheradresse eines gesuchten Datensatzes kann also auf direktem Wege ermittelt werden. Es spielt dabei fast keine Rolle, wie viele Einträge in der Hashtabelle gespeichert sind. Das Programm wird also fast nicht langsamer, wenn die Telefonliste länger ist. Warum „fast“? Weil es passieren kann, dass mehrere Schlüssel auf die gleiche Adresse abgebildet werden.

Dann muss man sich um Kollisionsbeseitigung kümmern, z.B. indem an einer Stelle in der Tabelle eine Liste mit den dort abgelegten Datensätzen verwaltet wird. Kollisionsbeseitigung passiert hier prinzipiell umso öfter, je mehr Daten abgelegt werden. Idealerweise sollte M also ziemlich groß sein (und eine Primzahl, damit bei der Modulo-Rechnung nur sehr wenige Zahlen auf die gleiche Adresse abgebildet werden). Noch etwas besser als eine Hashtabelle wäre die Verwaltung einer hierarchischen Struktur, nämlich eines Baumes, dessen Knoten jeweils 8 Nachfolger haben, einen für jede Ziffer, die für einen Buchstaben stehen kann. Bei einer vorgegebenen Suchfolge wie 82783 für „Taste“ startet man also am Wurzelknoten, besucht dessen 8. Nachfolger, dann wiederum dessen 2. Nachfolger usw. bis die Folge abgearbeitet ist. An dieser Stelle legt man dann den Datensatz ab bzw. liest ihn aus. Auch hier kann es Kollisionen geben („Taste“ und „Vapud“ hängen dann am gleichen Knoten), und je länger die Telefonliste, desto wahrscheinlicher ist es, dass zwei Namen die gleiche Suchfolge haben. Prinzipiell kann man aber sagen, dass es für jeden Knoten eine konstante maximale Zahl von Einträgen gibt, unabhängig von der Länge der Liste. Hier ist die Suchzeit also eher von der Länge der Suchfolge (also des Namens) abhängig, weil pro Ziffer eine weitere Ebene in der Hierarchie abgestiegen werden muss. Weil man in einem solchen Baum nur die Knoten anlegen muss, die man wirklich für die Eingabedaten braucht, ist der Speicherverbrauch im Gegensatz zur Hashtabelle übrigens recht günstig.

Programm-Text:

```
#!/usr/bin/perl

open DATEI, "<telnum.lst";
# Liste mit Telefonnummern öffnen
for (<DATEI>)
# Daten einlesen
{
($name, $tel) = split /\:/;
# Telefonnummer und Name trennen
chop $tel;
# \n-Zeichen entfernen
($vorn, $nachn) = split //, uc $name;
# Vorname und Nachname trennen
# (Großbuchstaben)
$nachn =~ tr/A-CD-FG-IJ-LM-OP-ST-VW-Z/2223334445556667778889999;
# Zeichen in entsprechende Ziffern umwandeln
$eintraege{$nachn} .= "$name:$tel";
# Eintrag unter dieser Nummer speichern
}

# (Nummer = zu drückende Handytasten)
close DATEI;
# Datei wieder schließen

print "\nName (Kurzform) eingeben: ";
chop ($eingabe = <STDIN>);
# Benutzereingabe einlesen
print "\nWählen sie einen Gesprächspartner aus!";

for (split /\&/, $eintraege{$eingabe})
# Für alle in Frage kommenden Einträge
{
($name, $tel) = split /\:/;
# Eintrag in Namen und Telefonnummer
# aufspalten
$$i++;
# Nummer des Eintrags erhöhen
print "\n$i. $name: $tel";
# Nummer, Name und Telefonnummer ausgeben
$auswahl[$i] = $tel;
# Telefonnummer in einem Hash speichern
}

print "\nIhre Wahl: ";
chop ($eingabe = <STDIN>);
# Benutzereingabe einlesen
print "\nWähle $auswahl{$eingabe}\n";
# ...und Telefonnummer wählen...
```

Lösung nach Markus Völker