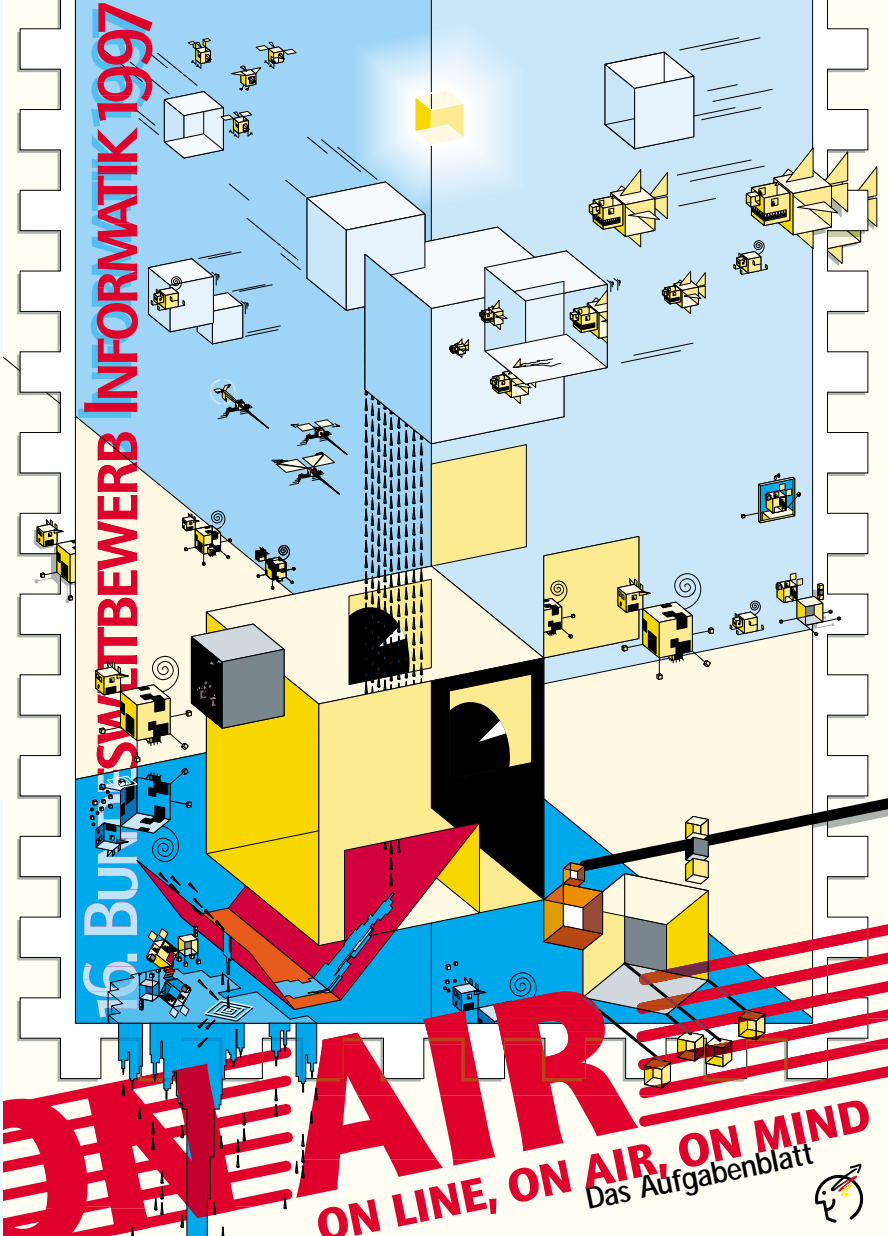


# 16. BUILD SWITBEBWERB INFORMATIK 1997



**ON LINE, ON AIR, ON MIND**  
Das Aufgabenblatt



Zu den Programmieraufgaben sende uns folgendes:

**Lösungsidee:**

Eine Beschreibung der Lösungsidee. Die Idee und die zu ihrer Beschreibung verwendeten Begriffe müssen sich im Programm wiederfinden.

**Programm-Dokumentation:**

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

**Programmablauf-Protokoll:**

Kommentierte Probelaufe des Programms. Mehrere unterschiedliche Beispiele, die die Lösung der Aufgabe verdeutlichen. Bildschirm-Fotos sind auch zulässig.

**Programm-Text:**

Das kommentierte Programm selbst in einer der gängigen höheren Programmiersprachen wie z.B. Pascal. Keine Maschinensprache, keine Peeks und Pokes.

Zu den theoretisch zu lösenden Aufgabenteilen sende uns bitte:

Eine knappe Beschreibung des Lösungsansatzes und eine ausführlichere Ausarbeitung dieses Ansatzes, die die gestellten Fragen beantwortet. Dabei sollen die Antworten auf die gestellten Fragen deutlich als solche erkennbar sein.

Einsendungen werden danach bewertet:

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

Bitte schicke Deine Arbeit nach Aufgaben geordnet und geheftet auf einseitig bedrucktem DIN-A4-Papier. Endlospapier schneide bitte entsprechend zu. Numeriere alle Blätter rechts oben und versieh sie mit Deinem Namen. Die Texte sollen in Deutsch abgefaßt sein. Falls Du einige DIN-A4-Klarsichthüllen **mit Heftrand** zur Hand hast, stecke bitte jeweils alles, was zu einer Aufgabe gehört, in eine Sichthülle. Andernfalls loche die Blätter bitte.

Fülle das Begleitformular (Klappe des Aufgabenblattes oder eine Kopie davon) **vollständig und leserlich!** aus. Insbesondere das **Geburtsdatum** muß angegeben sein, denn sonst kann die Einsendung nicht korrigiert werden. Bei Gruppen muß jeder Teilnehmer ein Formular ausfüllen, **Gruppenmitglieder ohne Teilnahmeformular werden nicht gewertet!**

Teilnehmer, die die Schule bereits verlassen haben, geben bei „Klassenstufe“ bitte an, was sie jetzt machen.

Wegen der ungeklärten rechtlichen Situation können wir **elektronische Einsendungen** leider **nicht** akzeptieren.

Sende alles an:

**Bundeswettbewerb Informatik**

**Ahrstraße 45**

**53175 Bonn**

Für Fragen zu den Aufgaben gibt es eine Hotline:

Jeweils von 17 – 19 Uhr

am 1. 9., 17. 9., 6. 10., 21. 10. und 5. 11.

Tel: 0228 - 37 86 46

oder email an: [bwinf@gmd.de](mailto:bwinf@gmd.de)

oder schriftlich an die obige Adresse

Informationen (FAQ) gibt's auch im Internet unter:

<http://www.bwinf.de>

**Einsendeschluß  
ist der 10.11.97**

(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.



**bmb+f**

**Unter der  
Schirmherrschaft  
des  
Bundespräsidenten**

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

# Zweihundvierzig

Banken verlangen häufig, daß Zahlen nicht nur in Ziffern, sondern auch in Worten angegeben werden.

Beispiel:	42	zweihundvierzig
	425	vierhundertfünfundzwanzig
	1300	eintausenddreihundert
	24201	vierundzwanzigtausendzweihunderteins
	101000	einhunderttausend
	400016	vierhunderttausendsechzehn

## Aufgabe:

Schreibe ein Programm, welches natürliche Zahlen zwischen 1 und 999999 korrekt in Worten ausgeben kann.

Erläutere ausführlich das Konstruktionsprinzip.

## Lösungsidee und Programmdokumentation:

Prinzipiell werden deutsche Zahlwörter gelegentlich regelmäßig gebildet:

Aus 735629 wird **7 hundert 5 und**

**30 tausend 6 hundert 9 und 20.**

Ausnahmen bilden dabei **elf**, **zwölf**, **sechzehn** (weil das **s** am Ende der 6 fehlt) und **siebzehn** (es heißt nicht **siebenzehn**).

Auch bei den Zehnern gibt es diverse Ausnahmen, nämlich **zwanzig**, **dreißig**, **sechzig** und **siebzig**.

Am einfachsten geht man mit diesen Ausnahmen um, indem man die Zahlen von 1 bis 20 und alle Zehner als Konstanten vereinbart.

Schließlich muß man noch auf die 1 aufpassen, denn bei auf 01 endenden Zahlen muß es **eins** heißen, innerhalb der Zahlwörter jedoch **ein**.

Für die Konstanten von 1 bis 19 verwenden wir ein Array *klein*, für die Zehner ein Array *zig*. Dann definieren wir drei Funktionen *zehner*, *hunderter* und *tausender*, die für die korrekte Umwandlung der Zahlen von 0 bis 99, von 100 bis 999 und von Zahlen ab 1000 sorgen.

Am einfachsten ist dabei die Funktion *tausender*, da sie nur wieder die Funktion *hunderter* aufruft und das Wort **tausend** einfügt.

In der Funktion *hunderter* wird mittels Division durch 100 berechnet, wieviele Hunderter in der Zahl enthalten sind und dafür gesorgt, daß nur dann das Wort **hundert** eingefügt wird, wenn es tatsächlich gebraucht wird. Der noch nicht in Buchstaben verwandelte Teil der eingegebenen Zahl wird mittels der Modulo-Funktion ermittelt und wiederum an die Funktion *hunderter* übergeben.

Wenn es im Rest keine Hunderter mehr gibt, wird die Funktion *zehner* aufgerufen.

In der Funktion *zehner* wird überprüft, ob für die umzusetzende Zahl eine Konstante vereinbart wurde. Wenn ja, wird diese eingesetzt, ansonsten wird die Zahl in ihren

Zehner- und ihren Einerteil aufgespalten (wieder mittels DIV und MOD) und die Wörter für diese beiden Teile werden, falls nötig, durch **und** verbunden.

Die Berechnung des Wortes für die eingegebene Zahl wird gestartet durch den Aufruf der Funktionen *tausender(zahl DIV 1000)* und *hunderter(zahl MOD 1000)*. Die Ergebnisse der Funktionsaufrufe werden aneinandergehängt.

Schließlich wird vor der Ausgabe noch geprüft, ob das ermittelte Wort auf **ein** endet, und wenn ja, wird ein **s** angehängt.

Die Eingabe von zu großen Zahlen führt nur zur Ausgabe **„zu groß!!!“**, danach können weitere Zahlen eingegeben werden. Beendet wird das Programm mit der Eingabe der Null. Die Eingabe von Zeichen statt Ziffern wird nicht abgefangen, sie führt zum Absturz des Programms.

## Halbformale Programmbeschreibung:

Das Programm ist so kurz, daß jede halbformale Programmbeschreibung, die mehr enthält als nur

Einlesen der Zahl  
Umwandeln in Zeichen  
Ausgeben

nur unwesentlich kürzer wäre als das Programm. Außerdem sollte die Vorgehensweise nach der ausführlichen Erläuterung der Lösungsidee anhand des Programmtextes leicht nachvollziehbar sein.

## BildschirmAusgabe:

0 <= Zahl <= 999999: 1001

eintausendeins

0 <= Zahl <= 999999: 666

sechshundertsechszehn

0 <= Zahl <= 999999: 777

siebenhundertsebenundsiebzig

0 <= Zahl <= 999999: 999999

neunhundertneunundneunzigtausendneunhundertneunundneunzig

0 <= Zahl <= 999999: 654321

sechshundertvierundfünfzigtausenddreihundertneunundzwanzig

0 <= Zahl <= 999999: 3432

dreitausendvierhundertzweiunddreißig

0 <= Zahl <= 999999: 75850

fünfundsechzigtausendachtundfünfzig

0 <= Zahl <= 999999: 1000000

zu groß!!!

0 <= Zahl <= 999999: 0

program zahlwoerter; (\* Lothar Oppor \*)

var

klein : array[1..19] of string[10];

zig : array[2..9] of string[10];

zahl : longint;

ergebnis : string;

procedure init;

begin

klein[1] := 'ein';

klein[2] := 'zwei';

klein[3] := 'drei';

klein[4] := 'vier';

klein[5] := 'fünf';

klein[6] := 'sechs';

klein[7] := 'sieben';

klein[8] := 'acht';

klein[9] := 'neun';

klein[10] := 'zehn';

klein[11] := 'elf';

klein[12] := 'zwölf';

klein[13] := 'dreizehn';

klein[14] := 'vierzehn';

klein[15] := 'fünfzehn';

klein[16] := 'sechzehn';

klein[17] := 'siebzehn';

klein[18] := 'achtzehn';

klein[19] := 'neunzehn';

zig[2] := 'zwanzig';

zig[3] := 'dreißig';

zig[4] := 'vierzig';

zig[5] := 'fünfzig';

zig[6] := 'sechzig';

zig[7] := 'siebzig';

zig[8] := 'achtzig';

zig[9] := 'neunzig';

end; (\* init \*)

function zehner(z : longint) : string;

begin

if z < 1 then zehner := ''

else if z <= 19 then zehner := klein[z]

else if (z mod 10) = 0 then

zehner := zig[z div 10]

else zehner := klein[z mod 10] + 'und'

+ zig[z div 10]

end; (\* zehner \*)

function hunderter(z : longint) : string;

begin

if z < 1 then hunderter := ''

else if z < 100 then hunderter := zehner(z)

else hunderter := klein[z div 100] +

'hundert' + hunderter(z mod 100)

end; (\* hunderter \*)

function tausender(z : longint) : string;

begin

if z < 1 then tausender := ''

else tausender := hunderter(z) + 'tausend'

end; (\* tausender \*)

begin

init;

zahl := 0;

while zahl >= 0 do

begin

write('0 <= Zahl <= 999999: ');

readln(zahl);

if zahl <= 0 then exit

else if zahl <= 999999 then

begin

ergebnis := tausender(zahl div 1000) +

hunderter(zahl mod 1000);

if copy(ergebnis, length(ergebnis)-2, 3) =

'ein' then

ergebnis := ergebnis + 's';

writeln(ergebnis)

end

else

writeln('zu groß!!!')

end (\* while \*)

end.

## Grußwort des Bundesministers für Bildung, Wissenschaft, Forschung und Technologie



Dr. Jürgen Rüttgers

Das Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie ruft in diesem Jahr zum 16. Mal zur Teilnahme am Bundeswettbewerb Informatik auf. Wir wollen damit Schülerinnen und Schüler anspornen, ihre Kenntnisse und Fähigkeiten auf dem Gebiet der Informatik weiterzuentwickeln. Das Aufgabenblatt des 16. Bundeswettbewerbs enthält wiederum sehr harte Nüsse. Junge Leute haben in diesem Wettbewerb die Chance, ihr Können zu testen und gleichzeitig mit Kreativität und Neugier Lösungen zu suchen. Beides sind wichtige Voraussetzungen für das spätere Berufsleben. Die neuen Informations- und Kommunikationstechniken prägen – direkt oder indirekt – 60 % unserer Arbeitsplätze. Sie sind Schlüsselfaktoren für unsere wirtschaftliche Leistungsfähigkeit und unsere

Wettbewerbsfähigkeit auf den Weltmärkten.

Wir stehen am Übergang der Industriegesellschaft zur Wissensgesellschaft.

Wer glaubt, daß die Wissensgesellschaft nur noch eine exklusive Gemeinschaft hochgezüchteter, theoretisierender Intellektueller sein wird, der irrt. Die Wissensgesellschaft schafft keine neuen Menschen. Schon gar nicht stuft sie den Menschen zurück.

Das Gegenteil ist richtig. Da nur der Mensch Produzent, Vermittler und Konsument von Wissen sein kann, rückt er in der Wissensgesellschaft – erstmalig in der Geschichte – ganz in den Mittelpunkt. Mehr als bei allen anderen technischen Errungenschaften haben wir deshalb Grund für Optimismus.

Die Wissensgesellschaft verlangt vom Bildungswesen insgesamt als auch von der Schule im besonderen neue Antworten. Die Vermittlung von Medienkompetenz spielt hierbei eine ganz entscheidende Rolle. Mit dem Projekt „Schulen ans Netz“ wollen wir dazu beitragen. Ziel ist, bis zum Jahr 2000 10.000 Schulen in Deutschland mit PCs auszustatten und ans Netz zu bringen.

Unser Land braucht helle Köpfe, die bereit sind, ihr Talent einzubringen und sich dem Wettbewerb zu stellen.

Ich wünsche allen Teilnehmern Freude und Engagement beim Wettbewerb. Ich bin gespannt, wer sich für die Endrunde qualifiziert.

## Grußwort des Präsidenten der Kultusministerkonferenz



Prof. Rolf Wernstedt

Die neuen Medien, insbesondere „Multimedia“ und „Telekommunikation“, sind schon seit längerer Zeit ein herausragendes Thema in der Öffentlichkeit und im gesamten Bildungswesen: Die neuen Technologien und die durch sie ermöglichten neuen Formen globaler Information und Kommunikation beeinflussen unser gesellschaftliches Leben bereits in vielen Bereichen. Der Schule kommt in diesem Zusammenhang die Aufgabe zu, zu einem verantwortungsvollen Umgang mit den neuen Medien zu erziehen.

Diese sind daher, neben der Informatik, mittlerweile fester Bestandteil des Bildungsauftrags der Schulen geworden. Die Schule übernimmt hier die Aufgabe, die Schülerinnen und Schüler zur Teilhabe an der globalen Information und Kommunikation zu befähigen und sie in die

Lage zu versetzen, Daten und Informationen selbständig zu gewinnen, zu überprüfen, zu verbreiten und zu bewerten.

In diesem Zusammenhang leistet der Bundeswettbewerb Informatik nach Auffassung der Kultusminister und Kultusenatoren der Länder in der Bundesrepublik Deutschland seit langem einen wertvollen Beitrag. Er spornt mit seinen Aufgaben Jugendliche an, sich mit den Inhalten und Methoden der Informatik, aber 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 angesprochen fühlen und in ihren Schulen die für eine erfolgreiche Teilnahme förderliche Unterstützung finden.

## Gesellschaft für Informatik e. V.

Die Gesellschaft für Informatik e.V. (GI) wurde 1969 als gemeinnütziger Verein zur Förderung der Informatik gegründet. Die 19 000 persönlichen und die 300 korporativen Mitglieder kommen aus der Informatik-Forschung, -Ausbildung, -Industrie, -Anwendung und -Studentenschaft. Träger der wissenschaftlichen Arbeit innerhalb der GI sind die Fachausschüsse und Fachgruppen, die in folgenden neun Fachbereichen zusammengeschlossen sind: Grundlagen der Informatik, Künstliche Intelligenz, Softwaretechnologie und Informationssysteme, Technische Informatik und Architektur von Rechensystemen, Informationstechnik und technische Nutzung der Informatik, Wirtschaftsinformatik, Informatik in Recht und öffentlicher Verwaltung, Ausbildung und Beruf, Informatik und Gesellschaft.

## 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.

Der Bundeswettbewerb Informatik wurde 1980 von der Gesellschaft für Informatik (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 vom Bundesminister für Bildung, Wissenschaft, Forschung und Technologie. Er steht unter der Schirmherrschaft des Bundespräsidenten. Die Träger des Wettbewerbs sind die GI (Gesellschaft für Informatik e. V.) und GMD - Forschungszentrum Informationstechnik GmbH. Die Gestaltung des Wettbewerbs und die Auswahl der Sieger werden vom Auswahl Ausschuss vorgenommen; Vorsitzender: Prof. Dr. Ingo Wegener, Universität Dortmund. Die Auswahl und Entwicklung von Aufgaben und die Festlegung von Bewertungsverfahren übernimmt ein Aufgabenausschuss; Vorsitzende: OStR Vera Reineke, Hannover. Eine Geschäftsstelle erledigt die fachlichen und organisatorischen Arbeiten; Geschäftsführerin: Dr. Gabriele Reich, Bonn.

### 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. Die Bearbeitungszeit beträgt circa zwei Monate. In der ersten Runde ist Gruppenarbeit zugelassen und erwünscht. An der zweiten Runde dürfen jene, die allein oder zusammen mit anderen wenigstens drei Aufgaben weitgehend richtig gelöst haben, teilnehmen. In der zweiten Runde ist dann selbstständige 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?

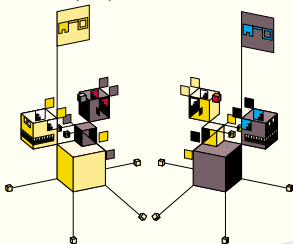
Teilnehmen können Jugendliche, die nach dem 10.11.1975 geboren wurden. Sie dürfen jedoch zum 1.9.1997 noch nicht ihre Ausbildung abgeschlossen oder eine Berufstätigkeit aufgenommen haben. Ebenfalls ausgeschlossen sind Personen, die zum Wintersemester 1997/98 oder früher ihr Studium an einer Hochschule/Fachhochschule aufnehmen bzw. aufgenommen haben. Jugendliche, die nicht deutsche Staatsangehörige sind, müssen wenigstens vom 1.9. - 10.11.97 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, einem Kolloquium, eingeladen, das im Oktober 1998 von der Firma sd&m, München, ausgerichtet werden wird. Die dort ermittelten Bundessieger werden in der Regel ohne weiteres Aufnahmeverfahren in die Studienstiftung des deutschen Volkes aufgenommen. Zusätzlich sind für besondere Leistungen Geld- und Sachpreise sowie Einladungen zu Sommerakademien vorgesehen.

### ... 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 (IOI'99) in der Türkei teilnimmt.



# Belagerung um Farnsworth Castle



AD 1314. Nicht mehr lange können die Ritter Königin Eleonores auf Farnsworth Castle die Angreifer des Blythethwick-Clans abhalten. Die einzige Hoffnung der Königin ist die Benachrichtigung ihres Sohnes John, der sich mit Julee, der Prinzessin des Clans, verlobt hat. Wäre sie auf Farnsworth, müßten die Blythethwicks verhandeln. Doch die Königin kann sich nicht sicher sein, daß ihr Bote nicht abgefangen wird. Würde ihr Plan bekannt, wäre Blutvergießen unvermeidbar. Eleonore schickt darum zwei Boten: Einen mit einer verschlüsselten Botschaft, den anderen mit einem Schlüssel. Bei der Verschlüsselung geht sie folgendermaßen vor: Sie schneidet eine quadratische Schablone aus Leder, die in quadratische Felder eingeteilt ist. Manche dieser Felder sind ausgeschnitten. Diese Schablone legt die Königin auf ein Stück Papier und schreibt die ersten Buchstaben ihrer Botschaft von links nach rechts, oben nach unten, durch die ausgeschnittenen Felder. Dann dreht sie die Schablone um 90 Grad im Uhrzeigersinn, schreibt weiter und wiederholt den Vorgang noch zwei weitere Male. Die Nachricht ist:

## KOMM UND BRING JULEE NACH FARNSWORTH

Schablone und verschlüsselte Nachricht sehen so aus:



BRKOA  
ERIMN  
NMSWN  
OGACJ  
UHRNU  
LFTH

- Welchen Anforderungen müssen Anzahl und Anordnung der Löcher genügen, damit die Schablone zur Verschlüsselung geeignet ist?
- Gib ein einfaches Verfahren für den Entwurf von geeigneten Schablonen an.
- Schreibe ein Programm, das anhand einer gegebenen Schablone eine Nachricht verdichtet und entschlüsseln kann. Wie sieht die mit der obigen Schablone verschlüsselte Fassung folgender Rückantwort aus?

## JULEE MIT STALLKNECHT DURCHGEBRANNT

# Partyvorbereitungen

Katrin Kafer möchte ein großes Fest feiern. Dafür sind etliche Aufgaben zu erledigen: Salate machen, Pizza vorbereiten, Zimmer ausräumen, Nachbarn warnen und dergleichen mehr. Katrin möchte nicht die ganze Arbeit alleine machen und überlegt sich daher, einige ihrer Freundinnen und Freunde zur Mitarbeit heranzuziehen. Nun stellt sich natürlich die Frage, wie viele Personen sie um Mithilfe bitten soll.

Etwas formaler stellt sich ihr Problem wie folgt dar: Es sind  $n$  Aufgaben zu erledigen. Die einzelnen Aufgaben sind nicht mehr teilbar. Für die  $i$ -te Aufgabe ist die Zeit  $t_i$  zu veranschlagen. Katrin allein würde also die Zeit  $t_1 + t_2 + \dots + t_n$  benötigen. Jede Person, die Katrin helfen könnte, kann nur einen Nachmittag, d.h. nicht mehr als 5 Stunden Zeit aufbringen.

Um die minimale Anzahl Helfer zu berechnen, müßte Katrin alle Möglichkeiten, Kombinationen von Aufgaben auf Helfer zu verteilen, ausprobieren. Bei den vielen zu erledigenden Aufgaben dauert ihr das aber zu lange.

### Aufgabe:

Entwirf eine Strategie, die folgendes leistet:

- Sie ermittelt nicht die optimale Lösung durch Probieren aller Kombinationen.
- Sie ordnet nicht einfach jeder Person genau eine Aufgabe zu.
- Sie verplant höchstens doppelt so viele Personen, wie im besten Fall nötig gewesen wären.

Beschreibe Deine Strategie und beweise, daß Katrin damit nicht mehr als doppelt so viele Helfer einspannt, wie nötig gewesen wären.

Schreibe ein Programm, das Deine Strategie realisiert.

Gib für  $n = 9$ ,  $t_1 = t_2 = \dots = t_7 = 45$  min,  $t_8 = 165$  min,  $t_9 = 105$  min und pro Person zur Verfügung stehender Zeit 300 min an, zu welchem Ergebnis Deine Strategie führt (nicht die von Hand erzielte optimale Lösung!). Wende Deine Strategie außerdem auf 10.000 mit dem Zufallszahlengenerator erzeugte Zahlen im Bereich 1 bis 20 und pro Person zur Verfügung stehender Zeit 25 an ( $n = 10.000$ ,  $t_i \in \{1, \dots, 20\}$ ) und gib an, wie viele Personen benötigt werden und wie lange sie im Durchschnitt arbeiten.

# Rekursive Besen

Heiner Huschelmutz, CAD-Spezialist, ist verzweifelt. Seine Frau war eine Woche auf Dienstreise und kommt morgen zurück. Huschelmutz möchte ihr die Wohnung in blitzblankem Zustand präsentieren: im Moment gleicht sie noch einem Schlachtfeld. Er will einen Besen holen, kann aber keinen finden: es ist nach Ladenschluß. Da kommt ihm eine Idee: In seiner Heimwerkstatt hat er eine computergetriebene Fräsmaschine. Warum soll sie ihm nicht einen Besen fräsen? Die Daten für den Besen sollen automatisch generiert werden. Zwecks Inspiration nimmt er drei Borsten eines Pinsels in die Hand, die er auf seinem Schreibtisch herumschiebt und anstarrt. Sie liegen so da:



Heiner Huschelmutz' Sicht verschwimmt, und er sieht immer mehr Borsten, erst so:



dann so:



dann so:



Noch etwas später hat er folgendes Bild vor Augen:



Anfänglich ist Huschelmutz verwirrt, doch bald erkennt er ein Prinzip hinter dem Spiel, das seine Augen mit ihm treiben.

- Nach welchem Prinzip ist Heiner Huschelmutz' Vision aus der ursprünglichen Borstenanordnung entstanden?
- Schreibe ein Programm, welches das Prinzip über beliebige viele Schritte fortsetzt und jeweils die Borstenanordnung auf dem Bildschirm anzeigt. Schicke uns Bilder der ersten sechs Stufen.
- Verändere die Ausgangslage der Borsten so, daß andere schöne Bilder entstehen. Schicke uns drei davon.



# Wetter in Quadratien

Quadratien ist ein quadratisches Gebiet aus quadratischen Feldern. Das Feld in der Nord-west-Ecke hat die Zellennummer 0 und die Spaltennummer 0.

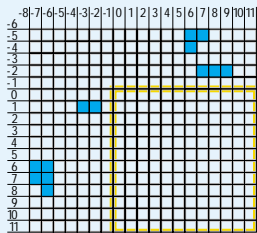
Das Wetter in Quadratien wird durch quadratische Wolken bestimmt, die genau ein Feld groß sind. Solche Wolken rücken getaktet über Quadratien vor, und zwar von Norden nach Süden 2 Felder pro Takt und, in einer anderen Höhe, von Westen nach Osten 3 Felder pro Takt. Es regnet überall dort, wo sich nach einem Vorrücken sowohl eine Nord-Süd- als auch eine West-Ost-Wolke befindet. Wolken, aus denen es regnet, lösen sich auf.

Die Wolkenvorhersage gibt an, an welchen Stellen (in der Form: Zellennummer,Spaltennummer) sich zum aktuellen Zeitpunkt Wolken befinden. Daraus läßt sich dann ermitteln, wo es in Quadratien regnen wird, denn es werden nur solche Wolken angegeben, die über Quadratien hinwegziehen werden.

## Beispiel:

Wolkenvorhersage:

-5/6 -4/6 -2/7 7/6 -2/9 -5/7  
1/-2 8/-6 6/-6 7/-7 6/-7 1/-3 -2/8.



Es wird jeweils einmal regnen an den Stellen 1, 6, 1, 7 und 8, 9.

## Aufgabe:

Schreibe ein Programm, welches folgendes leistet:

1. Einlesen der Größe von Quadratien (Anzahl Zeilen bzw. Spalten)
2. Einlesen einer Wolkenvorhersage
3. Ausgabe, wo in Quadratien wie oft Regen fällt

Sende uns 3 Beispiele, darunter eines für ein Quadratien der Größe 10 x 10 und folgendes Wolkenvorhersage:

2/-3 -5/5 -3/4 1/-4 6/-12 -3/5  
-7/5 3/-10 -6/6 6/-11 -7/4 3/-4  
-4/5 -3/3 -6/9 2/-4.



# Nach der Party

Das Fest bei Katrin Kafer war ein großer Erfolg. Nur eine Kleinigkeit trübte ihre Hochstimmung: Der von Katrin heimlich verehrte Robert hatte ihre Einladung zum Fest mit der Bemerkung abgelehnt, daß er keine Lust habe, seine Zeit auf so einer langweiligen Veranstaltung zu verschwenden. Mit der Verehrung ist es nach dieser Bemerkung natürlich vorbei, aber Katrin hofft nun, daß Robert von möglichst vielen verschiedenen Personen hört, und sich entsprechend ärgert. Wie oft Robert von dem Fest hört, läßt sich ausrechnen, wenn man weiß, wer alles auf dem Fest war, welche Personen im weitläufigen Bekanntenkreis von Katrin die Festgäste kennen, welche dieser Personen sich untereinander kennen und wer alles Robert kennt.

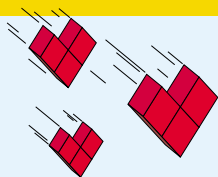
Se vermutet ganz richtig, daß jeder, der etwas von dem Fest hört, dies auch weiter erzählen wird. Andererseits wird aber niemand mehrfach der gleichen Person von dem Fest erzählen, auch wenn der Betreffende von mehreren verschiedenen Personen davon erzählt bekommt. Natürlich kennen sich alle Festgäste untereinander. Und natürlich wird sie selbst, obwohl sie Robert ja kennt, ihm niemals auch nur ein einziges Wort von ihrem Fest berichten. Auch ihren sonstigen nicht-eingeladenen Bekannten wird sie nicht von ihrem Fest vorschwärmen.

## Beispiel:

Auf dem Fest waren Freddy, Ina und Vera. Freddy kennt Robert und Lothar. Ina kennt Lothar und Hans. Vera kennt Robert und Hans. Robert kennt Lothar und Volker. Volker kennt Kirsten. Kirsten kennt Robert. Hans kennt Lothar und Robert.

(Wenn Robert Lothar kennt, dann kennt auch Lothar Robert, auch wenn dies nicht explizit angegeben wurde.)

In diesem Fall hört Robert viermal von dem Fest, nämlich von Freddy, Hans, Lothar und Vera.

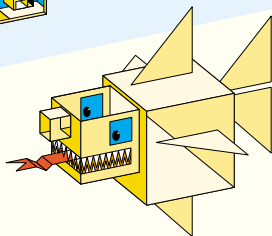
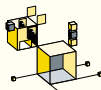
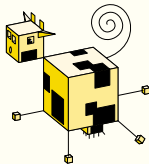


## Aufgabe:

Schreibe ein Programm, das nach Eingabe der Gäste und der Bekanntschaften im Bekanntenkreis von Katrin ermittelt, wie oft Robert von dem Fest hört. Die Eingabe der Bekanntschaften muß dabei nicht unbedingt in der Form zweier Namen erfolgen, sondern geeignete Kürzel sind ebenfalls möglich.

Sende uns drei Beispiele von Programmläufen, darunter eines mit folgenden Festgästen und Bekanntschaften:

Auf dem Fest waren Elisabeth, Christoph, Jochen und Gabby. Jochen kennt Muriel, Helga, Cornelia und Bettina. Bettina kennt Reinhard, Helga und Cornelia. Elisabeth kennt Peter, Michael, Muriel und Herbert. Robert kennt Ulrich, Herbert und Cornelia. Jochen kennt Herbert, Janine und Robert. Ulrich kennt Werner und Wolfgang. Wolfgang kennt Robert und Werner. Werner kennt Robert und Andrea. Andrea kennt Robert und Wolfgang. Janine kennt Reinhard und Robert. Christoph kennt Peter, Michael und Herbert. Peter kennt Robert und Herbert. Muriel kennt Janine, Reinhard und Robert. Michael kennt Robert.



Bitte in Druckschrift leserlich 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

Wieviele Stunden haben Sie gebraucht ?



Programmiersprache ?

Rechner ?

eigener

Haben Sie an anderen Wettbewerben teilgenommen?



Mathematik

Informatik

sonstige

Wie haben Sie vom Wettbewerb erfahren?



schon mal teilgenommen

Schule

ehemalige Teilnehmer

Messen

sonstiges

Diese Daten werden an niemanden weitergegeben, haben keinen Einfluß auf die Bewertung, aber dienen statistischen Zwecken.

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

Datum

Unterschrift