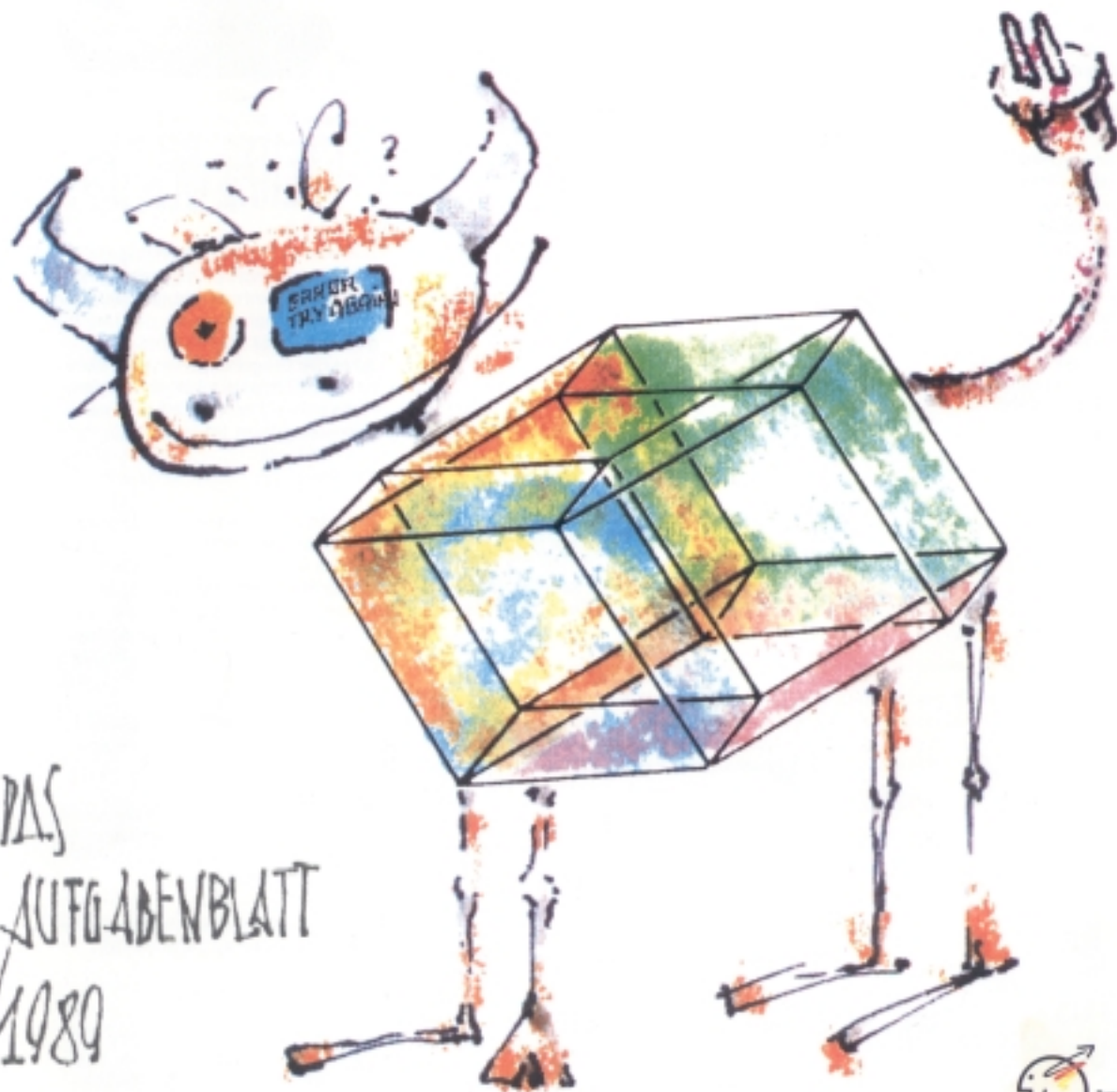


Achter Bundeswettbewerb

INFORMATIK



PAS
AUFGABENBLATT
1989



Aufgabe 1:

DER FLUCH DES PHARAO

Im Königsdreieck, das sich zwischen drei Pyramiden erstreckt, befindet sich irgendwo unter dem Staub der Jahrtausende der Eingang zur Grabkammer des Pharaos Tutramses. Schon viele Schatzsucher haben sich aufgemacht, das Grab zu finden und nach den kostbaren Grabbeigaben zu schürfen. Vergeblich!

Der Fluch des Pharaos bewirkt, daß sich der Schatzsucher, sobald er sich im Königsdreieck befindet, immer nur gradlinig auf eine der Pyramidenspitzen zu bewegen kann. Dabei schafft er jeweils genau die Hälfte der Strecke bis zur nächsten Pyramide und muß eine Weile rasten, um dann von neuem irgendeine der drei Pyramiden anzu- steuern. Gibt es Stellen im Königsdreieck, die ein Schatzsucher niemals erreichen kann, so daß der Eingang zur Grabkammer verborgen bleibt und der Pharaos seine ewige Ruhe behalten wird?

Aufgabe:

Schreibe ein Simulationsprogramm für die Schatzsuche, das nur die Rastpunkte auf dem Bildschirm anzeigt. Gib dazu die Koordinaten für die drei Eckpunkte des Königsdreiecks sowie für den Startpunkt des Schatzsuchers vor.

Schicke als Lösung mindestens fünf Grafiken ein. Bei einer davon sollen der Startpunkt und die drei Eckpunkte genau in den vier Ecken des Bildschirms liegen und 5000 Iterationen durchgeführt werden.



Aufgabe 2:

KROKUFANT

Ein Kinderbuch hat dreiteilige Seiten, deren Teile einzeln umgeblättert werden. Auf jeder Seite sind dreisilbige Tiernamen so angeordnet, daß jeweils eine Silbe auf einem Teil der Seite steht, z.B. E-le-fant, Kro-ko-dil, Kän-gu-ruh. Durch Umblättern einzelner Seitenteile entstehen neue Zusammensetzungen, z.B. Kro-ko-fant, Kän-le-dil.

Aufgabe:

Gib ein Buch mit einigen dreisilbigen Tiernamen vor.

Schreibe ein Programm

- mit dessen Hilfe weitere dreisilbige Tiernamen hinten ans Buch angefügt werden können,
- das herausfindet und protokolliert, wie man von einem vorgegebenen Tiernamen möglichst rasch zu einem anderen vorgegebenen Tiernamen blättern kann,
- das für b angibt, auf wieviele verschiedene Arten dies möglich ist.

Schicke uns fünf Protokolle von Programmläufen, wobei eins zeigt, wie man für das Buch

Jaguar, Krokodil, Pelikan, Elefant, Ameise, Känguruh

von Krokofant zum Agudil blättert.



Aufgabe 3:

FREITAG DER 13.

Dem Abergläubigen gilt Freitag, der 13te, als besonderer Unglückstag. Tatsächlich scheint etwas daran richtig zu sein: Wenn man annimmt, daß Unglück gleichverteilt ist über alle Tage aller Jahre des Gregorianischen Kalenders, so kommt Unglück häufiger an einem Freitag, dem 13ten, als z.B. an einem Donnerstag, dem 13ten, vor!

Aufgabe:

Schreibe ein Programm, das für einen einzugebenden Zeitraum des Gregorianischen Kalenders eine Häufigkeitsverteilung für die Wochentage erstellt, auf die der Dreizehnte eines Monats fällt. Berücksichtige dabei die vollständige Schaltjahrregelung des Gregorianischen Kalenders.

Schicke uns die Ergebnisprotokolle Deines Programms für folgende Zeiträume:

- die erste volle Periode des Gregorianischen Kalenders, welche am Freitag, dem 15.10.1582, begann und 400 Jahre dauerte,
- die Zeit vom Beginn des Gregorianischen Kalenders bis Ende 1989,
- das Orwelljahr 1984.

Wie viele Tage umfaßt die längste Zeitspanne ohne Freitag, den 13ten, in der ersten vollen Periode des Gregorianischen Kalenders?

Aufgabe 4:

FISCHE IM AQUARIUM

Wer ein Aquarium mit exotischen Fischen einrichten will, kommt nicht ohne den Rat des Experten aus. Wer außerdem bei beschränktem Geldbeutel versucht, Fische möglichst vieler verschiedener Arten gemeinsam unterzubringen, fragt besser gleich den Computer. Der Grund ist einfach: Nicht jede Art verträgt sich mit jeder anderen und natürlich sollten keine Fische unverträglicher Arten zusammen im Aquarium leben und sich bekämpfen oder gar - bei Fischen eine durchaus gängige Methode - gegenseitig auffressen.

Aufgabe:

Schreib ein Programm, das als Eingabe

- die für den Fischkauf verfügbare Summe,
- die erhältlichen Fischarten mit Preisen für einzelne Fische, sowie
- die Unverträglichkeiten von Arten untereinander entgegennimmt.

Ein Einkaufszettel für Fische soll ausgegeben werden, auf dem die Höchstzahl der für die verfügbare Summe erhältlichen Fischarten, die alle miteinander verträglich sind, notiert ist.

Schicke uns Protokolle des Programmlaufs für wenigstens fünf Eingaben, darunter die folgende:

verfügbare Summe:	170.- DM
erhältliche Arten:	
Grüne Migräne	70.- DM
Korallenqualle	50.- DM
Schuppenschatulle	30.- DM
Breitmaulmolch	40.- DM
Prachtpiranha	40.- DM
Zitterling	30.- DM
Grottensprotte	20.- DM



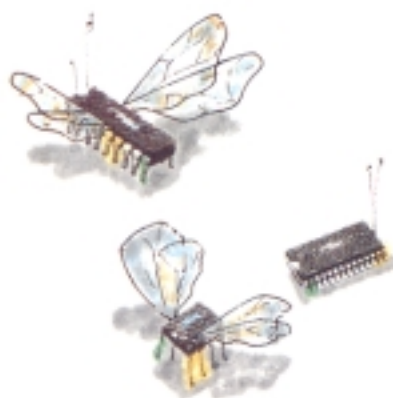
Unverträglichkeiten:

Grüne Migräne - Breitmaulmolch,
Grüne Migräne - Grottensprotte,
Schuppenschatulle - Breitmaulmolch,
Schuppenschatulle - Prachtpiranha,
Prachtpiranha - Grottensprotte,
Grottensprotte - Zitterling.

Aufgabe 5:

NAND OR NOR

Fritz findet auf dem Flohmarkt eine Kiste mit kleinen grauen Kästchen. Die Kästen haben jeweils zwei gelbe Buchsen, beschriftet mit "Eingang 1" und "Eingang 2" und eine grüne Buchse "Ausgang".



Neben der Kiste liegen Schildchen, die sich von den grauen Kästchen gelöst haben. Die Schildchen tragen Aufschriften: AND-Gatter, NAND-Gatter, NOR-Gatter, OR-Gatter, XOR-Gatter.

Fritz ist ein gewitzter Elektroniker: er belegt die Eingänge jedes Kästchens mit verschiedenen Kombinationen von 0 Volt und 5 Volt Spannung und mißt die Ausgangsspannung. Schon nach kurzer Zeit hat er alle Schildchen wieder den richtigen Kästchen zugeordnet.

Aufgabe:

Schreibe ein Simulationsprogramm, welches es interaktiv gestattet, die beiden Eingänge eines vom Programm zufällig gewählten Gatters mit 0 Volt oder 5 Volt zu belegen, und das dann die Ausgangsspannung anzeigt. Glaubt der Benutzer, das Gatter bestimmt zu haben, so gibt er den Gatternamen ein und erfährt, ob er richtig oder falsch geraten hat. Weiterhin soll das Programm alle Rateversuche protokollieren.

Schicke uns zu jedem der Gattertypen AND, NAND, OR, NOR, XOR mindestens ein Rateprotokoll.

Buchtip

Im Klett-Verlag ist folgendes Buch 1989 erschienen (ISBN 3-12-710750-1):

Dr. Peter Heyderhoff (Hrsg.):
Bundeswettbewerb Informatik.
Aufgaben und Lösungen
Band 1

Allgemeine Hinweise

Alle Einsendungen sollten aus den folgenden Teilen bestehen, getrennt nach Aufgaben:

Lösungsidee:

Eine Beschreibung der Lösungsidee. Die Form und die Begriffe der Lösungsidee müssen sich im Programm wiederfinden.

Programm-Dokumentation:

Eine Beschreibung des Programms. Hinweise auf Besonderheiten und Nutzungsgrenzen, Angaben zu Rechenzeiten und Speicherbedarf.

Programmablauf-Protokoll:

Kommentierte Probeläufe des Programms. Ein oder mehrere Protokolle. Bildschirm-Foto ist auch zulässig.

Programm-Text:

Das Programm selbst in einer der gängigen höheren Programmiersprachen wie z.B. Pascal. Keine Maschinensprache, keine Peeks und Pokes. Die Zeilen sollten numeriert sein.

Deine Einsendungen werden danach bewertet

- ob sie vollständig und richtig sind,
- ob die Ausarbeitungen gut strukturiert und verständlich sind,
- ob die Programmunterlagen übersichtlich und lesbar sind.

Bitte schicke Deine Ergebnisse auf einseitig bedrucktem DIN-A4-Papier. Alle Blätter sind rechts oben durchzunummerieren und mit Deinem Namen zu versehen. Die Texte sollen in deutscher Sprache abgefasst sein. Sende uns keine Disketten. Fülle das Begleitformular (Klappe des Aufgabenblattes oder eine Kopie davon) vollständig aus. Bei Gruppen muß jeder Teilnehmer ein Formular ausfüllen.

Sende alles an:

Bundeswettbewerb Informatik
GMD Schloß Birlinghoven
5205 Sankt Augustin 1

Einsendeschluß ist der 27.11.89

(Datum des Poststempels). Verspätete Einsendungen werden nicht geöffnet. Der Rechtsweg ist ausgeschlossen. Die Einsendungen werden nicht zurückgegeben. Der Veranstalter erhält das Recht, die Beiträge in geeigneter Form zu veröffentlichen.

Teilnehmen können Jugendliche, die nach dem 27.11.67 geboren wurden. Sie dürfen jedoch noch nicht im zweiten Semester studieren, ihre Ausbildung beendet oder einen Beruf ergriffen haben (siehe: Teilnahmeberechtigung).

Musteraufgabe:

ERBTEILUNG

Die Baronin von Birlinghoven hat ihren beiden Töchtern eine Truhe voller Goldmünzen hinterlassen. Ihr Testament bestimmt, daß das Gold einem benachbarten Kloster zukommt, falls es den Töchtern nicht gelingt, den Inhalt der Truhe wertmäßig genau in zwei Hälften untereinander aufzuteilen. Die Goldmünzen haben nur ganzzahlige Werte.

Beispiel:

Eine Truhe Goldmünzen mit den Werten 1, 9, 5, 3, 8 Taler könnten die Töchter in die Hälften 1, 9, 3 Taler und 5, 8 Taler teilen.

Eine Truhe Goldmünzen mit den Werten 1, 9, 7, 3, 8 Taler fiel an das Kloster, weil die Aufteilung gar nicht möglich ist.

Aufgabe:

Schreibe ein Programm, das bei Eingabe einer Folge ganzer Zahlen für die in der Truhe vorkommenden Werte die beiden Erbteile genau aufzählt, andernfalls das Erbe dem Kloster zuspricht, wenn eine Aufteilung nicht möglich ist.

Schicke uns mindestens 5 Beispiele mit verschiedenen Truheninhalten. Der Inhalt einer Truhe sei: 15, 27, 7, 23, 56, 13, 22, 5, 42, 34 Taler.

Musterlösung: Nr. 3 vom Aufgabenblatt 1988

Lösungsidee

Da Computer Routineaufgaben sehr schnell lösen können, bietet es sich hier an, ein Programm zu entwerfen mit dem alle Kombinationsmöglichkeiten der Reihe nach in einem rekursiven Verfahren mit Versuch und Rückzug durchprobiert werden. Um dem Computer unnütze Arbeit zu ersparen, werden einige einfache Kriterien angewandt, die sicherstellen, daß nur aussichtsreiche Versuche unternommen werden. Solche Kriterien sind:

1. Der Wert des Gesamterbes ist ganzzahlig durch zwei teilbar.
2. Im Verlauf eines Versuchs ist der Wert der ausgewählten Münzen plus dem Restwert, nämlich dem Wert der noch nicht betrachteten Münzen, größer oder gleich dem halben Erbe.
3. Im Verlauf eines Versuchs ist der Wert der ausgewählten Münzen kleiner als das Ziel, nämlich das halbe Erbe.

Programm-Dokumentation

Im Programm wird eine Liste der Münzwerte, der Restwerte und eine Schalterliste verwendet, die angibt, ob eine Münze ausgewählt ist.

Halbformale Beschreibung:

```
eingabe;
IF kriterium 1 THEN teile ab (1);
ausgabe.
```

```
teile ab (i):
  IF weitere münzen und
  kriterien 2 und 3
  THEN
    wähle i-te münze;
    erhöhe wert um i-te münze;
    IF wert = ziel
    THEN erhöhe anzahl erbtelle;
      zeige gewählte münzen
    ELSE teile ab (i+1)
    mache einen Rückzug;
    teile ab (i+1).
```

Programmablauf-Protokoll

Münzanzahl	<2 .. 20> : 10
1. Münze:	15
2. Münze:	27
3. Münze:	7
4. Münze:	23
5. Münze:	56
6. Münze:	13
7. Münze:	22
8. Münze:	5
9. Münze:	42
10. Münze:	34
/ Gesamterbe: 244, Halbes Erbe: 122	
Erbteil:	15 7 23 13 22 42
Erbteil:	27 56 5 34

Anzahl möglicher Teilungen: 1



Programm-Text

PROGRAM erbtellung ;

```
CONST max = 20 ;
VAR wert, ziel, erbtelle, n : INTEGER ;
rest, muenze : ARRAY [1 .. max] OF INTEGER ;
wahl : ARRAY [1 .. max] OF BOOLEAN ;
```

PROCEDURE teile_ab (i : INTEGER) ;

```
VAR j : INTEGER ;
BEGIN
  IF (i <= n) AND
  (wert < ziel) AND
  (wert + rest [i] >= ziel) THEN
    BEGIN
      wahl [i] := TRUE ;
      wert := wert + muenze [i] ;
      IF wert = ziel THEN
        BEGIN
          erbtelle := erbtelle + 1 ;
          writeln ; write ( ' Erbteil : ' ) ;
          FOR j := 1 TO n DO IF wahl [j] THEN write ( muenze [j] : 3 )
          END ( ' wert = ziel * ) ;
        ELSE teile_ab (i+1) ;
        wahl [i] := FALSE ;
        wert := wert - muenze [i] ;
        teile_ab (i+1)
      END
    END ( ' teile * ) ;
```

PROCEDURE eingabe ;

```
VAR i : INTEGER ;
BEGIN
  REPEAT writeln ; write ( ' Münzanzahl <2 .. ', max : 2, '> : ' ) ; read (n)
  UNTIL (2 <= n) AND (n <= max) ;
  FOR i := 1 TO n DO BEGIN
    wahl [i] := FALSE ;
    REPEAT write (i : 3, ' Münze : ' ) ; read ( muenze [i] )
    UNTIL (1 <= muenze [i]) AND (muenze [i] <= 99)
  END ( ' FOR * ) ;
  rest [n] := muenze [n] ;
  FOR i := n-1 DOWNTO 1 DO rest [i] := rest [i+1] + muenze [i] ;
  ziel := rest [1] DIV 2 ; wert := 0 ; erbtelle := 0 ;
  writeln ; write ( ' Gesamterbe : ', rest [1], ', Halbes Erbe : ', ziel )
  END ( ' eingabe * ) ;
```

PROCEDURE ausgabe ;

```
VAR i : INTEGER ;
BEGIN
  writeln ; writeln ;
  IF erbtelle = 0 THEN writeln ( ' Das Erbe fällt an das Kloster. ' )
  ELSE writeln ( ' Anzahl möglicher Teilungen : ', erbtelle DIV 2 ) ;
  readln ; readln
  END ( ' ausgabe * ) ;
```

```
BEGIN ( ' Hauptprogramm * )
  eingabe ; IF rest [1] MOD 2 = 0 THEN teile_ab (1) ; ausgabe
  END.
```