

1993

Das Aufgabenblatt



Unter der Schirmherrschaft von Bundespräsident Richard von Weizsäcker.

---

Eine gemeinsame Initiative von GI und GMD.

## Musteraufgabe:

# Reiterhofprobleme

Leider sind nicht alle Pferde auf dem Reiterhof so brav, daß sie von jedem Anfänger geritten werden können. Einige verlangen gute oder sogar sehr gute Reiter. Nicht genug damit. Die Reiterinnen und Reiter haben auch noch spezielle Wünsche.

### Reiterinnen und Reiter:

Anja	Anfängerin, mag Nicki, Pucki, Wittchen und Anex
Bertram	guter Reiter ohne Wunsch
Christa	sehr gute Reiterin, bevorzugt Hurrikan
Doris	Anfängerin, mag Anex, Wittchen und Pucki
Emil	Anfänger, mag alle Pferde
Fritz	sehr guter Reiter, mag Sturmwind und Hurrikan
Gabi	Ist gut, mag Hurrikan, Anex, Tornado, Sturmwind

### Pferde:

Anex	braves Anfängerpferd etwas schlitzhörnig, nur für gute Reiter
Nicki	temperamentvolles Pferd, nur für gute Reiter
Hurrikan	liebes Schmuspferd, für Anfänger geeignet
Pucki	schwierig und ungestum, verlangt sehr gute Reiter
Tornado	liebes Anfängerpferd bockig aber leistungsstark, nur für gute Reiter
Wittchen	müde gewordenes altes Pferd, für Anfänger eigenwillig und schwierig, nur für sehr gute Reiter
Sturmwind	
Zausel	
Zickzack	



### Lösungsidee:

Es wird versucht, jeden Reiter auf ein Pferd zu setzen. Dabei wird für jedes Paar (Reiter/Pferd) geprüft, ob diese Kombination möglich ist. Aus zwei Gründen kann das Besetzen eines Pferdes j mit einem Reiter i nicht möglich sein:

1. das Pferd j ist schon von einem anderen Reiter besetzt, oder
2. das Können von Reiter i reicht für Pferd j nicht aus.

In beiden Fällen muß versucht werden, ein anderes Pferd für Reiter i zu finden. Hat man eine Möglichkeit gefunden, jeden Reiter auf ein Pferd zu setzen, wird gezählt, wieviele Wünsche bei dieser Zuordnung erfüllt werden. Eine Zuordnung wird dann ausgegeben, wenn sie wenigstens so viele Wünsche erfüllt wie jede der bisher gefundenen Zuordnungen.

Wenn noch nicht alle Kombinationen (Reiter/Pferd) ausprobiert wurden, wird weitergesucht, ob es eine weitere, wenigstens genauso gute Zuordnung wie die zuletzt ausgegebene gibt.

## Programm-Text

### program Reiterhof:

```

const
  maxPferde = 10;
  maxReiter = 10;

type
  tier = record
    name: string;
    grad: integer;
  end;

  person = record
    name: string;
    koennen: integer;
    wunsch: array [1..maxPferde] of boolean;
  end;

```



```

var
  pferd: array [1..maxPferde] of tier;
  reiter: array [1..maxReiter] of person;
  Reitpferd: array [1..maxReiter] of integer;
  AnzReiter, AnzPferde, MaxWunsch: integer;
  ausgegeben: integer;

```

### procedure Eingabe:

```

var
  i, j: integer;
  jaNein: char;

begin
  write('Wieviele Reiter? (max. 10) '); readln(AnzReiter);
  write('Wieviele Pferde? (max. 10) '); readln(AnzPferde);
  for i := 1 to AnzPferde do with pferd[i] do begin
    write('Name von Pferd Nr. ', i, ': '); readln(Name);
    writeln('1 für Anfängerpferd, 2 normales, 3 schwieriges');
    write('Schwierigkeitsgrad: '); readln(Grad);
  end;
  for i := 1 to AnzReiter do with reiter[i] do begin
    write('Name von Reiter Nr. ', i, ': '); readln(Name);
    writeln('1 für Anfänger, 2 guter Reiter, 3 sehr guter');
    write('Können: '); readln(koennen);
    write('Wünsche: ');
    for j := 1 to AnzPferde do begin
      write('Ist ', pferd[j].Name, ' Wunschpferd? (j/n)'); readln(jaNein);
      reiter[i].Wunsch[j] := (jaNein='j');
    end;
  end;
end;
writeln;
for i := 1 to AnzReiter do Reitpferd[i] := 0;
MaxWunsch := 0;
end; (Eingabe)

```



### procedure Ausgabe:

```

var i, Wuensche: integer;

begin
  Wuensche := 0;
  for i := 1 to AnzReiter do
    if reiter[i].Wunsch[Reitpferd[i]] then
      Wuensche := Wuensche + 1;
  if Wuensche >= MaxWunsch then begin
    ausgegeben := ausgegeben + 1;
    MaxWunsch := Wuensche;
    for i := 1 to AnzReiter do
      writeln(reiter[i].Name, ' sitzt auf ',
              pferd[Reitpferd[i]].Name);
      writeln('Erfüllte Wünsche: ', Wuensche);
    end;
  end; (Ausgabe)
end; (Ausgabe)

procedure Besetze(i: integer);
var j: integer;

function Besetzbar(i, j: integer): boolean;
var
  moeglich: boolean;
  k: integer;

begin
  moeglich := true;
  for k := 1 to i-1 do
    if Reitpferd[k] = j then moeglich := false;
  if reiter[i].Koennen < pferd[j].Grad then moeglich := false;
  Besetzbar := moeglich;
end; (Besetzbar)

begin (Besetze)
  if i > AnzReiter then Ausgabe
  else begin
    for j := 1 to AnzPferde do
      if Besetzbar(i, j) then begin
        Reitpferd[i] := j;
        Besetze(i-1);
      end;
    end;
  end; (Besetze)
end; (Besetze)

begin
  Eingabe;
  ausgegeben := 0;
  Besetze(1);
  if ausgegeben = 0 then writeln('Schade, keine Lösung!')
  else writeln('ausgegeben, ', ausgegebene Losungen, ');
end;

```





## Bundeswettbewerb Informatik

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 und Wissenschaft. Er steht unter der Schirmherrschaft des Bundespräsidenten. Die Träger des Wettbewerbs sind die Gesellschaft für Informatik e.V. (GI) und die Gesellschaft für Mathematik und Datenverarbeitung mbH (GMD). Die Gestaltung des Wettbewerbs und die Auswahl der Sieger werden vom Auswahl Ausschuss vorgenommen; Vorsitzender: Prof. Dr. Rüdiger Loos, Universität Tübingen. Die Auswahl und Entwicklung von Aufgaben und die Festlegung von Bewertungsverfahren übernimmt ein Aufgabenausschuss; Vorsitzende: OStR Vera Reineke. Eine Geschäftsstelle erledigt die fachlichen und organisatorischen Arbeiten; Geschäftsführerin: Dr. Gabriele Reich, Tübingen.

### 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 selbständig mit dem Computer 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 drei 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 richtig gelöst haben, teilnehmen. In der zweiten Runde ist dann selbständige 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 informatische Probleme.

### Wer ist teilnahmeberechtigt?

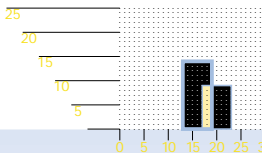
Teilnehmen können Jugendliche, die nach dem 12.11.1971 geboren wurden. Sie dürfen jedoch zum 1.9.1993 noch nicht ihre Ausbildung abgeschlossen oder ihr erstes Studiensemester beendet oder eine Berufstätigkeit aufgenommen haben. Falls sie nicht deutsche Staatsangehörige sind, müssen sie wenigstens vom 1.9. – 12.11.93 ihren Wohnsitz in Deutschland haben.

### 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 zu einem Kolloquium der dritten Runde im September 1994 eingeladen. Die dort ermittelten Bundessieger werden nach den geltenden Regelungen 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'95) in den Niederlanden teilnimmt.



## Aufgabe 2:

# Hast du Töne?

Die zwölf Töne in der westlichen Musik sind wie folgt angeordnet:

C Cis|Des D Dis|Es E Eis|F Fis|Ges G Gts|As  
A Ais|B H

Die Bezeichnungen links und rechts der senkrechten Striche stellen dabei in der wohltemperierten Stimmung jeweils den gleichen Ton dar. Das Intervall zwischen je zwei Tönen dieser Zwölferreihe heißt Halbtonschritt (H); zwei Halbtonschritte ergeben einen Ganztonschritt (G). Eine Dur-Tonleiter, bestehend aus 8 Tönen, wird nun, ausgehend von einem bestimmten Grundton, nach dem Schema G-G-H-G-G-G-H gebildet, wobei beim Erreichen des Endes in der obigen Liste wieder an Ihrem Anfang begonnen wird. Von den Noten mit doppeltem Namen enthält jede Dur-Tonleiter entweder nur solche, die links vom Strich stehen oder nur solche rechts vom Strich. Außerdem kommen in keiner Tonleiter zwei Noten mit gleichem Anfangsbuchstaben vor (B entspricht Hes). Das Intervall von einem Ton in einer Tonleiter zum selben Ton heißt Prim, das Intervall von einem Ton zum nächsten Sekund, zum übernächsten Terz, dann Quart, Quint, Sext, Septim und Oktav.

### Aufgabe:

Schreibe ein Programm, welches nach Eingabe eines Grundtons zunächst die Dur-Tonleiter dieses Grundtons ausgibt und anschließend Intervalle berechnet. Als Grundtone braucht das Programm nur C, G, D, A, E, H, Des, As, Es, B und F zu akzeptieren. Eingabe eines Punktes führt zur Aufforderung der Eingabe eines neuen Grundtons; wird hierfür auch ein Punkt eingegeben, so ist das Programm beendet.

### Beispiel:

Grundton: A  
A-Dur Tonleiter: A H Cis D E Fis Gis A  
Intervall: Cis plus Quart  
Cis plus Quart = Fis  
Intervall: Es minus Terz  
„Es“ kommt in dieser Tonleiter nicht vor!  
Intervall: .  
Grundton: Es  
Es-Dur Tonleiter: Es F G As B C D Es  
Intervall: Es minus Terz  
Es minus Terz = C  
Intervall: .  
Grundton: .

Sende drei Protokolle von Dialogen entsprechend dem Beispiel, darunter eines für die Eingabe:

D  
Fis plus Terz  
E minus Sext  
Cis plus Quint  
F plus Sekund

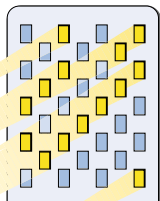
## Aufgabe 3:

# Hotelschlüssel

Wenn im Hotel „Graf Eberhard“, benannt nach dem Gründer der Universität Tübingen, ein Gast eintrifft, wird für ihn ein neuer Schlüssel hergestellt. Dieser besteht aus einer Karte mit Lochern, die von einem Lichtstrahl im Türschloß abgetastet werden. Das Türschloß eines Zimmers wird jeweils passend eingestellt.

### Beispiel:

## ZIMMER 3



Die Schlüssel müssen folgende Regeln erfüllen:

- Von den 32 möglichen Positionen sind wenigstens 5 mit einem Loch zu versehen. In unserem Beispiel sind es 14.
- Es darf keine der Spalten, Zeilen oder Diagonalen vollständig gelocht sein (damit der Schlüssel nicht so leicht bricht).
- Keine zwei Schlüssel dürfen identisch oder zueinander symmetrisch sein. Dadurch kann kein Schlüssel in einem anderen Zimmerschloß passen.

### Aufgabe:

Schreibe ein Programm, das in der Lage ist, verschiedene Schlüssel zu entwerfen. Sende uns zehn verschiedene Entwürfe.

## Aufgabe 1:

# Skyline

Der Rat des neuen Staates Utopolis möchte ein Stadtwappen einführen, welches unter anderem stilliert, aber maßstabsgerecht die Silhouette der Stadt enthält. Alle hohen Gebäude der Stadt sind auf der Ebene errichtete Quader. Jemand hat sich die Mühe gemacht, die Koordinaten der linken und der rechten oberen Ecke aller Gebäude der Stadt in einem 80 x 25-Raster zu bestimmen und aufzuschreiben. Beispiel siehe unten.

### Aufgabe:

Hilf dem Stadtrat, indem Du ein Programm schreibst, welches die ungeordnete Punktfolge der linken und rechten Ecken einliest, die Skyline als geordnete Punktfolge ausgibt und die Rechtecke und die Skyline als Graphik druckt (Textzeichen für die Graphik sind erlaubt). Die Eingabe-Punktfolge soll auf Zulässigkeit geprüft werden und eventuell zurückgewiesen werden.

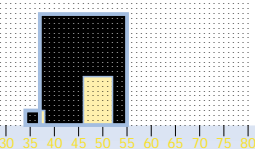
Im Beispiel wird die Skyline der ersten beiden Hochhäuser ausgegeben als Punktfolge (13,0), (13,14), (19,14), (19,9), (23,9), (23,0).

Schicke uns fünf Ergebnisse des Programms, darunter das zu der Punktfolge:

(6,9), (16,9), (72,16), (79,16), (7,12), (13,12), (9,14), (11,14), (17,7), (24,7), (19,4), (22,4), (17,10), (22,10), (24,7), (28,7), (31,12), (39,12), (42,15), (57,15), (41,15), (44,15), (54,12), (60,12), (58,5), (60,5), (35,8), (43,8)

### Beispiel:

(17,9), (23,9), (13,14), (19,14), (46,10), (52,10), (37,23), (55,23), (34,3), (38,3)



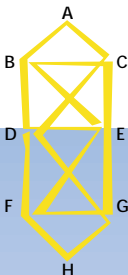
#### Aufgabe 4:

# Das Haus des Nikolaus

Für das Zeichnen des Haus-des-Nikolaus gelten die bekannten Regeln:

- Der Streckenzug muß ohne Absetzen gezeichnet werden.
- Jede Strecke wird genau einmal durchlaufen.

Nun steht das Haus-des-Nikolaus am Ufer eines Sees, in dem es gespiegelt wird. Es entsteht das folgende Bild:



#### Aufgabe:

Schreibe ein Programm, das alle möglichen Wege auslöst, die nach den oben angegebenen Regeln dieses Bild entstehen lassen! Gib dabei die Reihenfolge der durchlaufenen Punkte mit der oben angegebenen Bezeichnung aus!

#### Aufgabe 5:

Das Geheimnis der 3 Karten



Stelle dir einen Kartenstapel von  $2k$  ( $k \in \mathbb{N}$ ) Karten vor, die von oben nach unten durchnummeriert sind von 1 bis  $2k$ . Die oberste Karte ist also Karte 1. Der Kartenstapel wird nun gemischt nach folgendem Verfahren:  
Bei einer  $m$ -Mischung ( $m \in \mathbb{N}$  und  $m \leq k$ ) werden die obersten  $m$  Karten abgehoben und mit den nächsten  $m$  Karten abwechselnd neu aufgelegt, beginnend mit den nicht abgehobenen Karten. Der Rest des Stapels bleibt unverändert.  
Beispielsweise sieht für  $k = 8$  der Kartenstapel 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 nach einer 6-Mischung so aus: 7, 1, 8, 2, 9, 3, 10, 4, 11, 5, 12, 6, 13, 14, 15, 16

Bei einer vollständigen Mischung eines Kartenstapels aus  $2k$  Karten wird nacheinander eine 1-Mischung, eine 2-Mischung, eine 3-Mischung, ..., eine  $k$ -Mischung in dieser Reihenfolge ausgeführt.

In obigem Beispiel ergibt sich:

- 1 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16  
 2 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16  
 3 3, 2, 4, 1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16  
 4 1, 3, 5, 2, 6, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16  
 5 6, 1, 4, 3, 7, 5, 8, 2, 9, 10, 11, 12, 13, 14, 15, 16  
 6 5, 6, 8, 1, 2, 4, 9, 3, 10, 7, 11, 12, 13, 14, 15, 16  
 7 6, 5, 3, 6, 10, 8, 7, 1, 11, 2, 12, 4, 13, 14, 15, 16  
 8 1, 9, 11, 5, 2, 3, 12, 6, 4, 10, 13, 8, 7, 15, 16  
 9 4, 1, 10, 9, 13, 11, 8, 5, 14, 2, 7, 3, 15, 12, 16, 6

#### Aufgabe:

Schreibe ein Programm, das nach Eingabe von  $k$  ( $k \geq 2$ ) einen Kartenstapel mit  $2k$  Karten vollständig mischt. Während der Mischung sollen einige Informationen protokolliert werden, so daß das Programm die folgenden Fragen beantwortet:

- Welches sind die drei obersten Karten nach der vollständigen Mischung?
- Welches sind die drei Karten, die während des Mischens am häufigsten oben lagen? (Bei gleicher Häufigkeit interessiert die niedrigere bzw. niedrigste Kartennummer.)
- Bei der wievielten Mischung lag die oberste Karte zum erstenmal oben?
- Wie oft lag die oberste Karte bei den Mischungen insgesamt oben?

Gib uns fünf Lösungen an für verschiedene, möglichst große  $k$ , darunter  $k = 500$ .

#### Allgemeine Hinweise

Zu jeder Aufgabe sende uns folgendes:

#### 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.

#### Programmablauf-Protokoll:

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

#### Programm-Text:

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

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 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 Hefttrand zur Hand hast, stecke bitte jeweils alles, was zu einer Aufgabe gehört, in eine Sichthülle. 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

Sand 13

72076 Tübingen

Für Fragen zu den Aufgaben gibt es eine Hotline: Jeweils von 17 – 19 Uhr am 21., 22., 23., 29.9., am 8.10., 3. und 5.11. Telefon: 0 70 71- 653 92

#### Einsendeschluß ist der 12.11.93

(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.

#### Buchtip

Im Klett-Verlag ist folgende Buchreihe erschienen (ISBN 3-12-710750-1): Peter Heyderhoff (Hrsg.): Bundeswettbewerb Informatik Aufgaben und Lösungen, Bände 1, 2, 3, 4, 5

## 12. Bundeswettbewerb Informatik

### 1. Runde

Bitte in Druckschrift ausfüllen und Zutreffendes ankreuzen.



Name

Vorname

Geburtsdatum



Straße

männlich

weiblich

Postleitzahl Wohnort

Telefon: Vorwahl / Durchwahl

Bundesland

Name der (ehemaligen) Schule

Schultyp

Informatiklehrer/in

Straße der Schule

Postleitzahl Schulort

Klasse

Namen anderer Gruppenmitglieder

Wieviele Stunden haben Sie gebraucht ?



Programmiersprache ?

Rechner ?

eigener

Haben Sie an anderen Wettbewerben teilgenommen ?



Informatik



Mathematik



sonstige

Diese Daten werden an niemanden weitergegeben und haben keinen Einfluß auf die Bewertung, sondern 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

