

Das Aufgabenblatt

Der 15. Bundeswettbewerb Informatik



## Allgemeine Hinweise

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 der theoretisch zu lösenden Aufgabe 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.

Sende alles an:

**Bundeswettbewerb Informatik**  
**Ahrstraße 45**  
**53175 Bonn**

Für Fragen zu den Aufgaben gibt es eine Hotline:  
Jeweils von 17 – 19 Uhr

am 18. 9., 26. 9., 11. 10., 21. 10., 30. 10., 4. 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, zu erreichen über  
[http://www.gi-ev.de/wettbewerbe\\_und\\_preise.html](http://www.gi-ev.de/wettbewerbe_und_preise.html)

**Einsendeschluß**  
**ist der 11.11.96**

(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**

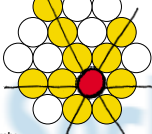
Unter der  
Schirmherrschaft  
des  
Bundespräsidenten

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

## Musteraufgabe:

In einem sechseckigen Feld von dicht gepackten Kreisen sollen möglichst viele Kreise gefärbt werden, allerdings so, daß auf jeder zu einer Sechsecksseite parallelen Reihe höchstens ein Kreis eingefärbt ist.

Die Abbildung zeigt ein Feld mit Kantenlänge 3 und einem rot eingefärbten Kreis. Die gelben Kreise geben die Reihen an, auf die der rote Kreis ausstrahlt. In diesen Reihen können also keine weiteren Kreise gefärbt werden.



### Aufgabe:

Schreibe ein Programm, welches die Kantenlänge des Feldes einliest und dazu die maximale Anzahl der Kreise bestimmt und einschließlich ihrer Position ausgibt, die auf diesem Feld rot eingefärbt werden können, ohne daß eine zu einer Sechsecksseite parallele Reihe mehr als einen roten Kreis enthält.

► Sende uns mindestens drei Beispiele mit unterschiedlichen Kantenlängen.

(aus Platzgründen entfallen die Programmdokumentation und die Bildschirm Ausgabe)

### Lösungsdeide:

Zunächst muß das Sechseck konstruiert werden: die einzelnen Punkte des Sechsecks werden als Objekte implementiert und kennen sowohl ihre Koordinaten (relativ zum mittleren Punkt, der die Koordination 0,0 erhält) als auch ihre Nachbarn. Das heißt, die Punkte werden sowohl gemäß ihren Positionen in einer Matrix abgelegt, was bei der Konstruktion und Ausgabe des Sechsecks nützlich ist, als auch untereinander mit Zeigern verbunden, was die Färbung entlang der Sechsecksdiagonalen sehr einfach macht.

Um dann die eigentliche Lösung zu finden, werden die Steine mit Hilfe eines Backtracking-Algorithmus auf freie Felder platziert. Das Programm beginnt mit der Platzierung eines Steines auf dem Feld 0,0. Nach der Platzierung werden die Felder in der gleichen Reihe und den Diagonalen markiert (gefärbt). Nun wird von Punkt 0,0 aus die weitere Steine zu setzen. Vorausgesetzt dafür ist, daß mindestens ein unmarkiertes Feld vorhanden ist. Ist dies nicht der Fall, werden der zuletzt gesetzte Stein und die durch ihn verursachten Markierungen entfernt und der Stein, wenn möglich, anders platziert.

Das Programm nummeriert die Felder bei ihrer Erzeugung durch und verwendet diese Nummern im weiteren Verlauf bei der Reihenfolge, in der die Felder besetzt werden. Prinzipiell ist die Reihenfolge, in der die Felder besetzt werden, aber beliebig, sie hat keinen Einfluß auf die maximal setzbare Anzahl Steine.

Die Steine werden auf die Felder in der Reihenfolge wachsender Nummern gesetzt. Dies reicht aus, alle übrigen möglichen Reihenfolgen (absteigend, alternierend) werden nicht ausprobiert.

Das Programm bricht ab, wenn es alle möglichen Positionierungen geprüft oder 2 \* Anzahl Kantenlänge - 1 Steine platziert hat. (Das ist die maximal mögliche Zahl.)

### Halbformale Programmbeschreibung:

Lies die Kantenlänge ein  
Erzeuge den mittleren Punkt des Sechsecks  
(rekursiver Aufruf: Wenn ein Punkt erzeugt wird, werden auch alle neuen Nachbarkpunkte, die noch im Sechseck liegen, erzeugt)

### wiederhole:

Ist die Anzahl der gesetzten Steine größer als die bisher best. Lösung?

Wenn ja, übermittle dies als neue Lösung und gib sie aus  
(nicht unbedingt erforderlich)

Für alle Felder:

(In der vorgegebenen Reihenfolge)

Könnte auf dieses Feld ein Stein gesetzt werden?

Wenn ja:

Setze den Stein, markiere die Zeile und Diagonalen  
Rufe die Schritte (wiederhole) rekursiv auf

Entferne den Stein und die von ihm erzeugten Markierungen

bis alle Kombinationen getestet oder das Maximum erreicht wurde.

### Programmtext: (zur Abweichung in C++)

// Sechseck.cpp

```
#include <stdio.h>
#include <stdlib.h>
```

class Sechseck

class Punkt

```
{
protected:
    int IX, IY, IZaehler;
    Punkt* Rechts, *Links, *RechtsOben,
        *LinksUnten, *RechtsUnten, *LinksOben;

    void SetzeTyp(int INeuertyp);
    void EntferneTyp(int INeuertyp);

public:
    Punkt(int X,int Y);

    void BildeVerbindungen(Sechseck *Basis);
    void SetzePunkt();
    void EntfernePunkt();
    int LiefererTyp();
}

typedef Punkt* PunktZeiger;

class Sechseck
{
protected:
    PunktZeiger *Punkte;
    PunktZeiger* PunktListe;
    int IKantenLaenge, IBreite, INPunkte;

public:
    Sechseck(int IKante);
    ~Sechseck();
    int NPunkte();
    Punkt* PunktNr(int INr);
    Punkt* LiefererPunkt(int IX,int IY);
    Punkt* LiefererOderErzeugePunkt(int IX,int IY);
}
```

```
Punkt::Punkt(int X,int Y)
{IX = X; IY = Y; IY := 0; IZaehler = 0}
```

```
void Punkt::BildeVerbindungen(Sechseck *Basis)
{
    Rechts = Basis->LiefererOderErzeugePunkt(IX+1,IY);
    if (Rechts != NULL) Rechts->Links = this;
    Links = Basis->LiefererOderErzeugePunkt(IX-1,IY);
    if (Links != NULL) Links->Rechts = this;
    RechtsOben = Basis->LiefererOderErzeugePunkt(IX+1,IY+1);
    if (RechtsOben != NULL) RechtsOben->LinksUnten = this;
    LinksUnten = Basis->LiefererOderErzeugePunkt(IX-1,IY-1);
    if (LinksUnten != NULL) LinksUnten->RechtsOben = this;
    RechtsUnten = Basis->LiefererOderErzeugePunkt(IX+1,IY-1);
    if (RechtsUnten != NULL) RechtsUnten->LinksOben = this;
    LinksOben = Basis->LiefererOderErzeugePunkt(IX-1,IY+1);
    if (LinksOben != NULL) LinksOben->RechtsUnten = this;
}
```

```
void Punkt::SetzeTyp(int INeuertyp)
{
    if (IY := INeuertyp) IZaehler++;
    else
        IY := INeuertyp;
    IZaehler = 1;
}
```

```
void Punkt::EntferneTyp(int INeuertyp)
{
    if (IZaehler < 2)
    {
        IZaehler = 1;
        IY := INeuertyp;
        IZaehler = IZaehler--;
    }
}
```

```
int Punkt::LiefererTyp()
{return IY}
```

```
void Punkt::SetzePunkt()
```

```
Punkt *I;
I = Rechts;
while (I != NULL)
{I->SetzeTyp(0); I = I->Rechts; }
I = Links;
while (I != NULL)
{I->SetzeTyp(2); I = I->Links; }
I = RechtsOben;
while (I != NULL)
{I->SetzeTyp(2); I = I->RechtsOben; }
I = RechtsUnten;
while (I != NULL)
{I->SetzeTyp(2); I = I->RechtsUnten; }
I = LinksOben;
while (I != NULL)
{I->SetzeTyp(2); I = I->LinksOben; }
I = LinksUnten;
while (I != NULL)
{I->SetzeTyp(2); I = I->LinksUnten; }
SetzeTyp(1);
}
```

```
void Punkt::EntfernePunkt()
```

```
Punkt *I;
I = Rechts;
while (I != NULL)
{I->EntferneTyp(0); I = I->Rechts; }
I = Links;
while (I != NULL)
{I->EntferneTyp(0); I = I->Links; }
I = RechtsOben;
while (I != NULL)
{I->EntferneTyp(0); I = I->RechtsOben; }
I = RechtsUnten;
while (I != NULL)
{I->EntferneTyp(0); I = I->RechtsUnten; }
I = LinksOben;
while (I != NULL)
{I->EntferneTyp(0); I = I->LinksOben; }
I = LinksUnten;
while (I != NULL)
{I->EntferneTyp(0); I = I->LinksUnten; }
}
```

```
while (I != NULL)
{I->EntferneTyp(0); I = I->LinksOben; }
I = LinksUnten;
while (I != NULL)
{I->EntferneTyp(0); I = I->LinksUnten; }
SetzeTyp(0);
}
```

```
Sechseck::Sechseck(int IKante)
```

```
{
    int IA;
    IKantenLaenge = IKante;
    IBreite = 2 * IKante - 1;
    IBreite = 2;
    INPunkte = 1;
    for (IA = 0; IA < IKante; IA++)
        INPunkte += 6 * IA;
    PunktListe = new PunktZeiger[INPunkte];
    INPunkte = 1;
    Punkte = new PunktZeiger[IBreite * IBreite];
    for (IA = 0; IA < IBreite; IA++)
        Punkte[IA] = NULL;
    PunktListe[0] = new Punkt(0,0);
    (PunktListe[0]Basis = this;
    PunktListe[0]Basis = BildeVerbindungen(this);
}
```

```
Sechseck::~Sechseck()
```

```
{
    int IA;
    for (IA = 0; IA < IBreite; IA++)
        delete PunktListe[IA];
}
```

```
Punkt* Sechseck::LiefererPunkt(int IX,int IY)
```

```
{
    if (abs(IX) >= 2 * IKantenLaenge || abs(IY) >= IKantenLaenge) return NULL;
    if (abs(IX) >= (2 * IKantenLaenge - abs(IY))) return NULL;
    return PunktListe[IX + IKantenLaenge * IBreite + IX + IKantenLaenge];
}
```

```
Punkt* Sechseck::LiefererOderErzeugePunkt(int IX,int IY)
```

```
{
    int IA;
    if (abs(IX) >= 2 * IKantenLaenge || abs(IY) >= IKantenLaenge) return NULL;
    if (abs(IX) >= (2 * IKantenLaenge - abs(IY))) return NULL;
    IA = IX + IKantenLaenge;
    IB = IY + IKantenLaenge;
    if (Punkte[IB]Breite + IA == NULL)
    {
        (Punkte[IB]Breite + IA) =
            new Punkt(IX,IY);BildeVerbindungen(this);
        PunktListe[INPunkte++] = Punkte[IB]Breite + IA;
        return Punkte[IB]Breite + IA;
    }
}
```

```
Punkt* Sechseck::PunktNr(int INr)
```

```
{return PunktListe[INr]}
```

```
Sechseck *Form;
```

```
int IBest, IMaximum, IKantenLaenge, INPunkte;
```

```
void ZeigeSechseck()
```

```
{
    int IX,IY;
    for (IY = -IKantenLaenge+1; IY < IKantenLaenge; IY++)
    {
        for (IX = -(IKantenLaenge-1)+2;
            IX <= IKantenLaenge-1; IX++)
        {
            if (Form->LiefererPunkt(IX,IY) == NULL) printf(" ");
            else
            {
                switch (Form->LiefererPunkt(IX,IY)-LiefererTyp()) {
                    case 0: printf(" "); break;
                    case 1: printf("O"); break;
                    case 2: printf("+"); break;
                }
            }
            printf("\n");
        }
    }
}
```

```
int Suche(int IPunkte,int IStart)
```

```
{
    int IA;
    Punkt *P;
    if (PunkteBest)
    {
        IBest = IPunkte;
        printf("%i\n", IBest);
        ZeigeSechseck();
        if (IBest == IMaximum) return 1;
    }
    for (IA = IStart; IA < IKantenLaenge; IA++)
    {
        if (P := Form->PunktNr(IA))-LiefererTyp() == 0)
        {
            P->SetzePunkt();
            if (Suche(IPunkte+1,IA+1)) return 1;
            P->EntfernePunkt();
        }
        return 0;
    }
}
```

```
int main()
```

```
{
    printf("Kantenlaenge: ");
    scanf("%i",&IKantenLaenge);
    IBest = 0;
    IMaximum = 2 * IKantenLaenge - 1;
    Form = new Sechseck(IKantenLaenge);
    INPunkte = Form->NPunkte();
    Suche(0);
    delete Form;
    return 0;
}
```



Dr. Jürgen Rüttgers

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

Neue Informations- und Kommunikationstechniken zu beherrschen, gehört zu den wichtigsten Schlüsselqualifikationen in modernen Industrienationen. Junge Menschen müssen daher möglichst früh lernen, mit solchen Techniken umzugehen. Die künftige wirtschaftliche Position unseres Landes im internationalen Wettbewerb ist in hohem Maße von Spitzenleistungen auch auf dem Gebiet der Informatik abhängig. Das Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie fördert den Bundeswettbewerb Informatik, um Schüler dazu anzuapornen, ihre Kenntnisse und Fähigkeiten auf dem Gebiet der Informatik auf hohem Niveau weiterzuentwickeln.

In dem Wettbewerb haben sie die Chance, Aufschluß über ihr eigenes Können zu erhalten und konstruktive Neugier auf ein wissenschaftliches Problem zu konzentrieren - beides wichtige Voraussetzungen für spätere Arbeit im Beruf. Ich hoffe, daß der Wettbewerb den Teilnehmern Freude bereitet.



Karl-Heinz Reck,  
Sachsen-Anhalt

### **Grußwort des Präsidenten der Kultusministerkonferenz**

Die neuen Medien, insbesondere „Multimedia“ und „Telekommunikation“, sind derzeit ein herausragendes Thema in der Öffentlichkeit und im gesamten Bildungswesen: Die neuen Technologien und die durch sie ermöglichten neuen Formen globaler Information und Kommunikation werden unser gesellschaftliches Leben in vielen Bereichen und in zunehmendem Maße merklich verändern. Der Schule kommt in diesem Zusammenhang die Aufgabe zu, die Entwicklungen und die Wirkungen der neuen Medien verstärkt zum Gegenstand sachlicher und kritischer Behandlung im Unterricht zu machen, um ihre Möglichkeiten nutzen zu können, zugleich aber auch zu einem verantwortungsvollen Umgang mit ihnen zu erziehen.

Die neuen Medien und die Informatik sind daher mittlerweile ebenfalls fester Bestandteil auch des Bildungsauftrags der Schulen geworden. Dabei nimmt die Informatik seit längerem einen gesicherten Platz im Spektrum der Unterrichtsfächer ein. Mit der Weiterentwicklung der technischen Voraussetzungen und Möglichkeiten wird die Bedeutung der informationstechnischen Bildung in den kommenden Jahren noch weiter zunehmen. 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 -senatoren der Länder in der Bundesrepublik seit langem einen wertvollen Beitrag. Er spornt mit seinen Aufgaben Jugendliche an, sich mit den Inhalten und Methoden der Informatik sowie 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 18 000 persönlichen und die 400 korporativen Mitglieder kommen aus der Informatik-Forschung, -Ausbildung, -Industrie, -Anwendung und -Studentenschaft. Träger der wissenschaftlichen Arbeit innerhalb der GI sind die über 100 Fachausschüsse und Fachgruppen, die in folgenden neun Fachbereichen zusammengeschlossen sind: Grundlagen der Informatik, Künstliche Intelligenz, Software-technologie und Informationssysteme, Technische Informatik und Architektur von Rechensystemen, Informationstechnik und technische Nutzung der Informatik, Informatik in der Wirtschaft, 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 (IT), Informatik und der Fachinformation sowie auf solchen Gebieten der Mathematik, die für den Fortschritt der Informationstechnik von besonderer Bedeutung sind. 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 Auswahlausschuß 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 11.11.1974 geboren wurden. Sie dürfen jedoch zum 1.9.1996 noch nicht ihre Ausbildung abgeschlossen oder eine Berufstätigkeit aufgenommen haben. Ebenfalls ausgeschlossen sind Personen, die zum Wintersemester 1996/97 oder früher ihr Studium an einer Hochschule/Fachhochschule aufnehmen bzw. aufgenommen haben. Jugendliche, die nicht deutsche Staatsangehörige sind, müssen wenigstens vom 1.9. - 11.11.96 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, im September oder Oktober 1997 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'98) in Portugal teilnimmt.



# Bunte Reihe

Ein Schiff soll mit Wimpeln geschmückt werden. Wir haben eine bestimmte Anzahl verschiedener Farben und von jeder Farbe eine bestimmte Anzahl von Wimpeln. Die Wimpel sollen so aufgehängt werden, daß keine zwei benachbarten Wimpel dieselbe Farbe haben.

## Aufgabe:

Lies zunächst die Anzahl verschiedener Farben und die Namen der Farben ein und dann für jede Farbe die entsprechende Anzahl von Wimpeln. Gib dann alle Möglichkeiten aus, wie die Wimpel entsprechend der Vorschrift aufgehängt werden können. Falls es mehr als 10 Möglichkeiten gibt, gib nur die Anzahl der Lösungen und drei Wimpelfolgen aus.

## Beispiele:

1. Anzahl der Farben: 3

Farbe 1: rot

Farbe 2: grün

Farbe 3: blau

Anzahl der roten Wimpel: 2

Anzahl der grünen Wimpel: 1

Anzahl der blauen Wimpel: 1

Reihenfolgen:

rot, grün, rot, blau

rot, grün, blau, rot

rot, blau, rot, grün

rot, blau, grün, rot

grün, rot, blau, rot

blau, rot, grün, rot

2. Anzahl der Farben: 4

Farbe 1: rot

Farbe 2: grün

Farbe 3: blau

Farbe 4: schwarz

Anzahl der roten Wimpel: 1

Anzahl der grünen Wimpel: 2

Anzahl der blauen Wimpel: 6

Anzahl der schwarzen Wimpel: 1

Geht nicht!!

Sende vier Beispiele von Programmlaufen ein, darunter zwei mit den folgenden Angaben: 6 Farben (rot, grün, blau, schwarz, weiß, gelb) und folgende Wimpel:

1. 3 rote, 2 grüne, 1 blauer, 8 schwarze,

2 weiße, 2 gelbe

2. 2 rote, 1 grüner, 1 blauer, 8 schwarze,

1 weißer, 1 gelber

# Zweihundvierzig

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

## Beispiel:

42 zweihundvierzig

425 vierhundertfünfundzwanzig

1300 eintausenddreihundert

24201 vierundzwanzigtausend-

zweihunderteins

101000 einhunderttausend

400016 vierhunderttausendsechzehn

## Aufgabe:

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

Erläutere ausführlich das Konstruktionsprinzip.

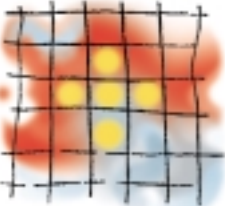


# Flutlicht

Ein überdachter Freizeitpark wird von in luftiger Höhe angebrachten Flutern lauschtig beleuchtet. Die lichtdurchflutete Fläche ist quadratisch und in 5 x 5 gleich große quadratische Beleuchtungsfelder eingeteilt. Über jedem Beleuchtungsfeld hängt genau ein Fluter.

Die Fluter werden von einer Zentrale aus gesteuert. Auf einem Monitor befindet sich ein maßstäblich verkleinertes Abbild der Beleuchtungsfelder. Die Fluter werden durch einen Mausklick auf das jeweilige Quadrat ein- und ausgeschaltet. Leider hat die Steuerung der Flutlichtanlage einen irreparablen Schaden, es werden nämlich gleichzeitig mit einem Fluter auch noch alle benachbarten Fluter geschaltet und zwar eingeschaltet, wenn sie ausgeschaltet waren, und umgekehrt. Zwei Fluter werden dabei genau dann als benachbart betrachtet, wenn sie in einer Zeile oder Spalte unmittelbar nebeneinander liegen.

Da sich die Gemeinde wegen akuten Geldmangels in absehbarer Zeit keine neue Steuerung leisten kann, braucht sie deine Hilfe, und deshalb hier deine



## Aufgabe:

Schreibe ein Programm, das einen beliebigen gegebenen Zustand der Beleuchtungsfelder, eine sogenannte „Beleuchtungskonfiguration“, einliest und dann einen „Klickplan“ ausgibt, d.h. einen Plan, welche Beleuchtungsfelder in welcher Reihenfolge angeklickt werden müssen, so daß nach Ausführung des Klickplans so wenige Fluter wie möglich eingeschaltet sind.

Teste dein Programm an folgenden Beleuchtungskonfigurationen:

- Alle Fluter sind eingeschaltet.
- Genau der mittlere Fluter ist eingeschaltet.
- Die Fluter in den 4 Ecken sind eingeschaltet.



#### Aufgabe 4:

## bezügliches

Der folgende Satz macht eine Aussage über sich selbst: er ist also „selbstbezüglich“:

„Dieser Satz enthält den Buchstaben **e** sechsmal, den Buchstaben **n** fünfmal.“

Es läßt sich schnell nachzählen, daß diese Behauptung falsch ist. Der Buchstabe **e** findet sich neunmal und der Buchstabe **n** siebenmal in dem oben genannten Satz.

Doch gibt es auch für diesen Satz Zahlwörter, die den Satz zu einer wahren Aussage werden lassen, wobei die **e**s und **n**s der ausgeschriebenen Zahlwörter natürlich mitgezählt werden müssen.

#### Aufgabe:

- Analysiere das Problem und beschreibe die Schwierigkeiten, die zum Finden einer Lösung zu bewältigen sind.
- Beschreibe eine Strategie, mit der alle möglichen Lösungen für einen wahren selbstbezüglichen Satz der obigen Form gefunden werden können.
- Erläutere, wie sichergestellt ist, daß alle Lösungen gefunden worden sind.

#### Aufgabe 5:

## Auf dem Sprung

Drei Damen aus dem beliebten und allseits bekannten 8-Damen-Problem ist es zu langweilig geworden; sie möchten gerne auch einmal andere Figuren zur Gesellschaft auf ihrem Schachbrett haben. Am interessantesten fänden sie es, wenn noch einige Springer hinzukämen. Natürlich möchten sie aber nicht von den Springern aus dem Feld geschlagen werden. Die Frage ist nun, wie viele Springer sie sich zur Gesellschaft einladen können.

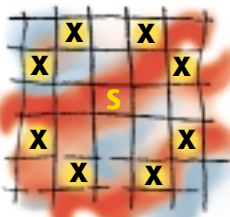
#### Aufgabe:

Schreibe ein Programm, das ermittelt, wie viele Springer höchstens auf einem Schachbrett zusammen mit 3 Damen untergebracht werden können, so daß keine der Figuren eine andere bedroht. Das Programm soll außerdem angeben, wie viele verschiedene Lösungsmöglichkeiten es gibt. Lösungen, die sich durch Drehung oder Spiegelung aus einer bereits gefundenen Lösung ergeben, sollen dabei unberücksichtigt bleiben. Wenigstens eine der Lösungen soll ausgegeben werden.

#### ► Hinweis für Schachkundige:

Eine Dame bedroht alle Felder in der gleichen Reihe, Spalte oder Diagonale, in der sie selbst steht.

Ein Springer (S) bedroht alle mit X markierten Felder, siehe Skizze:



# 15. Bundeswettbewerb Informatik 1996/97 1. Runde

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 ?



Informatik

Mathematik

sonstige

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