

Der 32. Bundeswettbewerb Informatik für Jugendliche bis 21 Jahre.

Einsendeschluss ist der 25. November 2013.

Informationen und Unterlagen bitte anfordern bei:
BWINF, Wachsbleiche 7, 53111 Bonn
bwinf@bwinf.de, www.bundeswettbewerb-informatik.de

Grußwort

Liebe Teilnehmerinnen und Teilnehmer am
Bundeswettbewerb Informatik,

nur die Informatik schafft es, aus der 32 eine 100000 zu
machen, und so ist die diesjährige 32. Runde des Wett-
bewerbs eine binär ganz besondere. Erst in 32 Jahren gibt es
wieder eine binär-glatte Zahl, wenn dann der „1000000.“
Informatikwettbewerb gefeiert werden kann.

Das allein ist schon ein Grund, jetzt teilzunehmen, noch mehr
locken die einzelnen Aufgaben, die – so verspricht es die
Titelseite des Ausschreibungsflyers – mit dem krimino-
logischen Spürsinn eines modernen Sherlock Holmes gelöst
werden müssen.

Der Bundeswettbewerb Informatik bietet Jugendlichen die Möglichkeit eines geistigen
Kräftemessens besonderer Art. Dinge zu strukturieren, komplexe Systeme in überschaubare
Teile zu zerlegen, zu formalisieren und zu interpretieren – diese Fähigkeiten sind gewisser-
maßen das Handwerkszeug des Junginformatikers. Auch die Abstraktionsfähigkeit und das
Erfassen logischer Zusammenhänge, die Modellbildung und natürlich Sorgfalt, Genauigkeit
und Ausdauer sind Eigenschaften, die zum Erfolg bei diesem Wettbewerb führen werden.

Unsere moderne Gesellschaft ist ohne vernetzte Informations- und Kommunikations-Systeme
nicht mehr denkbar, unser Kommunikationsverhalten kann nicht mehr von der Informatik
abgekoppelt werden – man denke nur an die Selbstverständlichkeit des Internets, der Tablet-
PCs, der sozialen Netzwerke sowie Navigationssysteme und Steuerungssysteme. Deshalb
brauchen wir Menschen, die sich für Informatik begeistern. Wir benötigen kluge Köpfe, die
sich neuen Entwicklungen verschreiben, die forschen, reflektieren, Denkanstöße geben und
die für Informatik werben, indem sie bei Mitmenschen jeden Alters Verständnis für die sich
abzeichnenden Entwicklungen wecken und Lust auf neue Ideen und Wege machen.

Der Bundeswettbewerb Informatik richtet sich mit anspruchsvollen und aufregenden Aufga-
ben an alle, die daran interessiert sind, kreative Problemlösungen zu finden, die dem Alltag
oder zumindest einer realen Situation entstammen.

Ich wünsche dem Wettbewerb eine jubiläumsreife Teilnehmerzahl und allen, die mitmachen,
viel Freude am Knobeln und vor allem viel Erfolg!



Stephan Dorgerloh
Präsident der Kultusministerkonferenz 2013
Kultusminister des Landes Sachsen-Anhalt



Die Träger

Gesellschaft für Informatik e.V. (GI)

Gesellschaft
für Informatik



Die Gesellschaft für Informatik e.V. (GI) ist mit rund 20.000 Mitgliedern die größte Fachgesellschaft der Informatik im deutschsprachigen Raum. Ihre Mitglieder kommen aus allen Sparten der Wissenschaft, aus der Informatikindustrie, aus dem Kreis der Anwender sowie aus Lehre, Forschung, Studium und Ausbildung. In der GI wirken Männer und Frauen am Fortschritt der Informatik mit, im wissenschaftlich-fachlich-praktischen Austausch in etwa 120 verschiedenen Fachgruppen und mehr als 30 Regionalgruppen. Ihr gemeinsames Ziel ist die Förderung der Informatik in Forschung, Lehre und Anwendung, die gegenseitige Unterstützung bei der Arbeit sowie die Weiterbildung. Die GI vertritt hierbei die Interessen der Informatik in Politik und Wirtschaft. Im Web: www.gi.de

Fraunhofer-Verbund IuK-Technologie

 **Fraunhofer**
IUK-TECHNOLOGIE

Als größter europäischer Forschungsverbund für Informations- und Kommunikationstechnik (IuK) versteht sich der Fraunhofer-Verbund IuK-Technologie als Anlaufstelle für Industriekunden auf der Suche nach dem richtigen Ansprechpartner in der anwendungsorientierten IT-Forschung. Die Vernetzung der 5000 Mitarbeiter in bundesweit 18 Instituten ermöglicht die Entwicklung übergreifender branchenspezifischer IT-Lösungen, oft zusammen mit Partnern aus der Industrie, sowie anbieterunabhängige Technologieberatung. Entwickelt werden IuK-Lösungen für die Geschäftsfelder Medizin, Automotive, Produktion, Digitale Medien, E-Business, E-Government, Finanzdienstleister, Sicherheit sowie IT und Kommunikationssysteme. InnoVisions – Das Zukunftsmagazin des Fraunhofer Verbundes IuK-Technologie informiert Sie über aktuelle Forschungsprojekte auf www.innovisions.de. Weitere Informationen über den Fraunhofer IuK-Verbund gibt es auf www.iuk.fraunhofer.de.

Max-Planck-Institut für Informatik

 **max planck institut**
informatik

Eine der größten Herausforderungen der Informatik ist die robuste und intelligente Suche nach Information, die grundlegendes Verständnis und automatische Organisation der gewünschten Inhalte voraussetzt. Das Max-Planck-Institut für Informatik widmet sich seit seiner Gründung 1990 diesen Fragestellungen. Das Spektrum der Forschung reicht von allgemeinen Grundlagen der Informatik bis hin zu konkreten Anwendungsszenarien und umfasst Algorithmen und Komplexität, Automatisierung der Logik, Bioinformatik und Angewandte Algorithmik, Computergrafik, Bildverarbeitung und multimodale Sensorverarbeitung sowie Datenbanken und Informationssysteme. Das Max-Planck-Institut für Informatik unterstützt nachhaltig junge Forscher, die am Institut die Möglichkeit bekommen, ihr eigenes Forschungsgebiet und ihre eigene Gruppe zu entwickeln. Das Institut wirkt seit mehr als 20 Jahren auf Weltklassenniveau durch Publikationen und Software und durch seine jetzigen und ehemaligen Forscher, die Führungsrollen in Wissenschaft und Industrie übernommen haben. Web: www.mpi-inf.mpg.de

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

Unter der Schirmherrschaft des Bundespräsidenten

**Von der Kultusministerkonferenz empfohlener
Schülerwettbewerb**

Die Partner

Zusätzlich zur Förderung durch das Bundesministerium für Bildung und Forschung und seine Träger erfährt der Bundeswettbewerb Informatik und damit die Initiative BWINF (Bundesweit Informatiknachwuchs fördern) weitere Unterstützung durch viele Partner. Sie stiften Preise und bieten vor allem spannende Informatik-Workshops für Wettbewerbsteilnehmer an.

Die BWINF-Partner wünschen allen Teilnehmerinnen und Teilnehmern des 32. Bundeswettbewerbs Informatik viel Erfolg!



Triff BwInf-Teilnehmer in der Community
auf einstieg-informatik.de!



[BwInf.Informatik.erleben](https://www.facebook.com/BwInf.Informatik.erleben)

Bundeswettbewerb Informatik

Der Bundeswettbewerb Informatik (BwInf) wurde 1980 von der Gesellschaft für Informatik e.V. (GI) auf Initiative von Prof. Dr. Volker Claus ins Leben gerufen. Ziel des Wettbewerbs ist, 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 Kultusministerien der Länder empfohlen werden. Gefördert wird er vom Bundesministerium für Bildung und Forschung und steht unter der Schirmherrschaft des Bundespräsidenten. Der BwInf ist Kern von „Bundesweit Informatiknachwuchs fördern“ (BWINF), einer gemeinsamen Initiative von GI, Fraunhofer-Verbund IuK-Technologie und Max-Planck-Institut für Informatik. Die Gestaltung des Wettbewerbs und die Auswahl der Sieger obliegen dem Beirat; Vorsitzende: Prof. Dr. Nicole Schweikardt, Universität Frankfurt. Die Auswahl und Entwicklung von Aufgaben und die Festlegung von Bewertungsverfahren übernimmt der Aufgabenausschuss; Vorsitzender: Prof. Dr. Peter Rossmanith, RWTH Aachen. Die Geschäftsstelle des Wettbewerbs mit Sitz in Bonn ist für die fachliche und organisatorische Durchführung zuständig; Geschäftsführer: Dr. Wolfgang Pohl.

Drei Runden

Der Wettbewerb beginnt jedes Jahr im September, dauert etwa ein Jahr und besteht aus drei Runden. In der ersten und zweiten Runde sind die Wettbewerbsaufgaben zu Hause selbstständig zu bearbeiten. Dabei können die Aufgaben der ersten Runde mit guten grundlegenden Informatikkenntnissen gelöst werden; die Aufgaben der zweiten Runde sind deutlich schwieriger. In der ersten Runde ist Gruppenarbeit zugelassen und erwünscht. In der zweiten Runde ist dann eigenständige Einzelarbeit gefordert; die Bewertung erfolgt durch eine relative Platzierung der Arbeiten. Die bis zu dreißig bundesweit Besten der zweiten Runde werden zur dritten Runde, einem Kolloquium, eingeladen. Darin führt jeder Gespräche mit Informatikern aus Schule und Hochschule und analysiert und bearbeitet im Team zwei Informatik-Probleme.

Juniorliga

Um die Teilnahme jüngerer Schülerinnen und Schüler am BwInf zu fördern, gibt es in der ersten Runde die Juniorliga. Für Schüler, die nicht älter als 16 Jahre sind, werden zwei leichtere Aufgaben gestellt, die Junioraufgaben. Stammt eine Einsendung von Schülern, die alle die Altersgrenze für Junioraufgaben erfüllen, nehmen die darin bearbeiteten Junioraufgaben in der Juniorliga teil (wenn auch andere Aufgaben bearbeitet sind, nimmt die Einsendung zusätzlich mit allen Aufgabenbearbeitungen in der Hauptliga teil). Die Juniorliga wird getrennt bewertet, Preise werden separat vergeben.

Die Chancen

Preise

In allen Runden des Wettbewerbs wird die Teilnahme durch eine Urkunde bestätigt. In der ersten Runde werden auf den Urkunden erste und zweite Preise sowie Anerkennungen unterschieden; mit einem Preis ist die Qualifikation für die zweite Runde verbunden. In der zweiten Runde gibt es erste, zweite und dritte Preise; jüngere Teilnehmer haben die Chance auf eine Einladung zu einer Schülerakademie. Ausgewählte Gewinner eines zweiten Preises erhalten einen Buchpreis des Verlags O'Reilly; erste Preisträger werden zur dritten Runde eingeladen, die im Herbst 2014 an der Universität Lüneburg ausgerichtet wird.

Die dort ermittelten Bundessieger werden in der Regel ohne weiteres Aufnahmeverfahren in die Studienstiftung des deutschen Volkes aufgenommen. Zusätzlich sind für den Bundessieger, aber auch für andere besondere Leistungen Geld- und Sachpreise vorgesehen.

Informatik-Olympiade

Ausgewählte Teilnehmerinnen und Teilnehmer können sich in mehreren Trainingsrunden für das vierköpfige deutsche Team qualifizieren, das an der Internationalen Informatik-Olympiade 2015 in Kasachstan teilnimmt.

Informatik-Workshops etc.

Informatik-Workshops exklusiv für TeilnehmerInnen werden in Baden-Württemberg, vom Hasso-Plattner-Institut, von Hochschulen wie der RWTH Aachen, der TU Braunschweig, der LMU München (gemeinsam mit der QAware GmbH) und dem Max-Planck-Institut für Informatik (2. Runde) veranstaltet. Bei einigen von Fraunhofer-Instituten veranstalteten „Talent Schools“ gibt es reservierte BwInf-Plätze. Die Firma Google lädt ausgewählte Teilnehmerinnen zum „Girls@Google Day“ ein.

Ausgewählte Endrundenteilnehmer werden im Herbst 2014 vom Bundesministerium für Bildung und Forschung zum „Tag der Talente“ eingeladen.

Eine Einsendung zur zweiten Runde kann in vielen Bundesländern als besondere Lernleistung in die Abiturwertung eingebracht werden.

Preise für BwInf-Schulen

Für eine substantielle Beteiligung am Wettbewerb werden Schulpreise vergeben: An mindestens 3 vollwertigen Einsendungen (also mit je mindestens 3 bearbeiteten Aufgaben) zur 1. Runde – oder an 2 vollwertigen Einsendungen und 2 weiteren Einsendungen in der Juniorliga – müssen mindestens 10 Schülerinnen und Schüler einer Schule, darunter bei gemischten Schulen mindestens 2 Jungen und mindestens 2 Mädchen, beteiligt sein. Schulen, die diese Bedingung erfüllen, werden als „BwInf-Schule 2013/2014“ ausgezeichnet: sie erhalten ein entsprechendes Zertifikat, ein Label zur Nutzung auf der Schul-Website und einen Gutschein im Wert von **300 Euro** für Bücher oder andere für den Informatikunterricht benötigte Dinge.

Die Regeln

Teilnahmeberechtigt

... sind Jugendliche, die nach dem 25.11.1991 geboren wurden. Sie dürfen jedoch zum 1.9.2013 noch nicht ihre (informatikbezogene) Ausbildung abgeschlossen oder eine Berufstätigkeit begonnen haben. Personen, die im Wintersemester 2013/14 an einer Hochschule studieren, sind ausgeschlossen, falls sie nicht gleichzeitig noch die Schule besuchen. Jugendliche, die nicht deutsche Staatsangehörige sind, müssen wenigstens vom 1.9. bis 25.11.2013 ihren Wohnsitz in Deutschland haben oder eine staatlich anerkannte deutsche Schule im Ausland besuchen.

Junioraufgaben dürfen von bis zu 16-Jährigen bearbeitet werden (geboren nach dem 25.11.1996) bzw. von Gruppen mit mindestens einem solchen Mitglied.

Weiterkommen

An der zweiten Runde dürfen jene teilnehmen, die allein oder mit ihrer Gruppe wenigstens drei Aufgaben der ersten Runde weitgehend richtig gelöst haben. Für die dritte Runde qualifizieren sich die besten ca. 30 Teilnehmer der zweiten Runde. In der Juniorliga gibt es noch keine zweite Runde.

Einsendungen

... enthalten Bearbeitungen zu mindestens einer Aufgabe und werden von Einzelpersonen oder Gruppen abgegeben. Eine Einsendung besteht aus Dokumentation und (bei Aufgaben mit Programmierauftrag) Implementierung. Zu jeder bearbeiteten Aufgabe enthält die **Dokumentation** eine Beschreibung der Lösungsidee und Beispiele, die die Korrektheit der Lösung belegen. Ist ein Programm gefordert, sollen außerdem die Umsetzung der Lösungsidee in das Programm erläutert und die wichtigsten Teile des Quelltextes hinzugefügt werden. Achtung: eine gute Dokumentation muss nicht lang sein! Die **Implementierung** enthält für Aufgaben, in denen ein Programm gefordert ist, das lauffähige Programm selbst und den kompletten Quelltext des Programms.

Die **Einsendung** wird über das Online-Anmeldesystem als Dateiarchiv im ZIP-Format abgegeben. Dieses Archiv muss auf oberster Ebene enthalten:

- > die Dokumentationen: am besten in einem PDF-Dokument für alle Aufgaben oder in je einem PDF-Dokument pro Aufgabe;
- > zu jeder bearbeiteten Aufgabe einen Ordner mit den Implementierungsdateien: Programm (nach Möglichkeit eigenständig lauffähig) und Quelltextdatei(en).

Anmeldung

Die Anmeldung ist bis zum Einsendeschluss möglich, und zwar online unter

www.bundeswettbewerb-informatik.de/anmeldung

Wettbewerbsteilnehmer können sich dort eigenständig anmelden oder **von Lehrkräften angemeldet** werden.

Einsendung: Online im Rahmen des Anmeldesystems (s.o.)

Einsendeschluss: 25.11.2013

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.

Beispiellösung: Skyline

Das Wahrzeichen von Boomtown ist die Needle, ein schlanker, 100m hoher Turm. Nach ihrem Bau wurde einst beschlossen, dass kein Gebäude in Boomtown höher als die Needle sein darf. Nun sollen viele neue Gebäude in Boomtown errichtet werden, und die Bauherren wollen höher hinaus.

Der Rat von Boomtown überlegt, dass Gebäude, die weiter von der Needle weg sind, höher sein dürfen. So können die Bauherren befriedigt werden, und gleichzeitig bleibt die Needle als markantes Wahrzeichen der Stadt erhalten. Der Rat erlässt also die Vorschrift, dass mit jedem 100m Abstand von der Needle 1m höher gebaut werden darf. Also ist die maximal erlaubte Höhe bei einem Abstand von 0m bis 99m: 100m, bei einem Abstand von 100m bis 199m: 101m usw.

Nun soll für die vielen Bauanfragen entschieden werden, wie hoch das Gebäude jeweils sein darf. Zum Glück hatten die Stadtväter von Boomtown einst ein Koordinatensystem mit der Einheitslänge 1m eingeführt, in dem die Needle die Koordinaten (0,0) hat. Für die Grundrisse aller Gebäude gelten folgende Regeln:

- sie sind rechteckig;
- ihre Eckpunkte haben ganzzahlige Koordinaten;
- ihre Seiten liegen parallel zu den Achsen des Koordinatensystems.

Das Bild unten zeigt drei Gebäude mit den Grundrissen (angegeben durch die Eckpunkte)

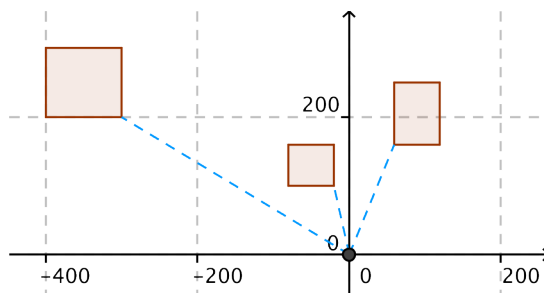
$(-80, 100); (-20, 100); (-20, 160); (-80, 160)$
 $(60, 160); (120, 160); (120, 250); (60, 250)$
 $(-400, 200); (-300, 200); (-300, 300); (-400, 300)$

Aufgabe

Schreibe ein Programm, das für eine Liste von Grundrissen die erlaubten Höhen der Gebäude berechnet.

Für das Beispiel im Bild lautet die richtige Ausgabe:

101
101
103



Lösungsidee

Ein Gebäudegrundriss ist ein Rechteck. Ein Rechteck ist durch zwei diagonal gegenüberliegende Eckpunkte genau festgelegt. Wir gehen also davon aus, dass für jedes Rechteck die Koordinaten des Eckpunkts unten links (x_{ul}, y_{ul}) und des Eckpunkts oben rechts (x_{or}, y_{or}) bekannt sind. Wir nennen diese Eckpunkte auch Haupteckpunkte.

Wir suchen den Punkt auf dem Rand des Rechtecks, der dem Ursprung bzw. Nullpunkt des Koordinatensystems am nächsten liegt. Haben wir diesen Punkt gefunden, können wir den

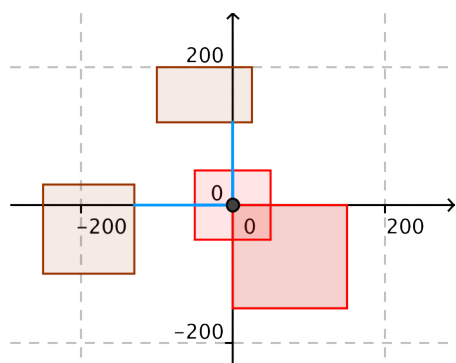
Abstand d des Punktes zum Nullpunkt berechnen und dann die erlaubte Höhe so ermitteln: $100 + \text{abrunden}(d/100)$.

Bei der Bestimmung des nächsten Punktes müssen wir verschiedene Fälle unterscheiden:

Fall 1: Das Rechteck liegt in genau einem Quadranten des Koordinatensystems; weder schneidet noch berührt es eine der Achsen (wie die Rechtecke im Beispiel aus der Aufgabenstellung). Dann kann der Abstand zum nächstgelegenen Eckpunkt (x, y) gemessen werden, und zwar mit der Formel $\sqrt{x^2 + y^2}$ (vergleiche den Satz von Pythagoras).

Fall 2: Das Rechteck schneidet oder berührt eine Koordinatenachse (siehe die beiden Rechtecke links und oben im folgenden Bild). Dann muss der Abstand zum näheren Schnittpunkt mit der jeweiligen Achse berechnet werden, wie die blau gepunkteten Linien im Bild zeigen. Dafür kann folgende Formel verwendet werden („min“ liefert den kleinsten von zwei Werten): $\min(|x_{ul}|, |x_{or}|)$ bzw. $\min(|y_{ul}|, |y_{or}|)$. Die beiden Rechtecke im Bild haben die Haupteckpunkte $(-250, -100)$ und $(-130, 30)$ bzw. $(-100, 120)$ und $(25, 200)$. Die Abstände sind also 130 bzw. 120, die erlaubten Höhen jeweils 101.

Fall 3: Das Rechteck schneidet oder berührt beide Koordinatenachsen (siehe die roten Rechtecke im Bild unten). Dann enthält es den Nullpunkt, evtl. als Randpunkt. In diesem Fall liegt ein Fehler vor, denn das Gebäude soll nicht auf die Needle gebaut werden.



Umsetzung

Die Lösungsidee wird in ein Programm in der Sprache Python umgesetzt. Die Unterscheidung der drei Fälle erledigt die Funktion `berechne_abstand`, die genau die vier Hauptkoordinaten `x_ul`, `y_ul`, `x_or` und `y_or` als Parameter erwartet. Der Fehlerfall 3 liegt vor, wenn `x_ul <= 0 <= x_or` und `y_ul <= 0 <= y_or`. Gilt nur eine dieser beiden Ungleichungen, dann liegt Fall 2 vor, ansonsten der „Normalfall“ 1.

Das Programm erwartet die Eingabedaten in einer Datei. Diese enthält in der ersten Zeile die Anzahl der in der Datei beschriebenen Grundrisse. Jede weitere Zeile beschreibt dann einen Grundriss und enthält die Koordinaten aller seiner Eckpunkte, allerdings in vereinfachter Form: `x1 y1 x2 y2 x3 y3 x4 y4`

Die Funktion `hauptkoordinaten` nimmt genau so eine Zeile und gibt eine Liste mit den vier Hauptkoordinaten zurück: `[x_ul, y_ul, x_or, y_or]`.

Die zentrale Funktion des Programms, `skyline`, liest die Zeilen der Eingabedatei nacheinander ein, besorgt sich für jede Koordinatenzeile die `hauptkoordinaten` und ruft damit `berechne_abstand` auf. Je nach Rückgabewert wird eine Fehlermeldung (Fall 3) oder die erlaubte Höhe (Fälle 1 und 2) ausgegeben.

Beispiele

Zunächst wird das Programm mit den Beispieldaten aus der Aufgabenstellung getestet. Die Eingabedatei `bwinf.in` sieht so aus:

```
3
-80 100 -20 100 -20 160 -80 160
60 160 120 160 120 250 60 250
-400 200 -300 200 -300 300 -400 300
```

Nun wird `skyline` mit dieser Datei aufgerufen:

```
>>> skyline('bwinf.in')
Erlaubte Höhe: 101
Erlaubte Höhe: 101
Erlaubte Höhe: 103
```

Ein zweiter Beispiellauf zeigt, dass das Programm auch die Fälle 2 und 3 richtig bearbeitet. Die folgende Eingabedatei `sonderfaelle.in` entspricht dem obigen Bild: zuerst kommen die beiden Fehlerfälle, dann das Rechteck über der x-Achse und zuletzt das über der y-Achse:

```
4
-50 -50 50 -50 50 50 -50 50
0 -150 150 -150 150 0 0 0
-250 -100 -130 -100 -130 30 -250 30
-100 120 25 120 25 200 -100 200
```

Als Ausgaben sind also zwei Fehlermeldungen und zweimal der Höhenwert 101 zu erwarten.

```
>>> skyline('sonderfaelle.in')
Fehler: Gebäude liegt über der Needle!
Fehler: Gebäude liegt über der Needle!
Erlaubte Höhe: 101
Erlaubte Höhe: 101
```

Quelltext

```
import math
```

```
def hauptkoordinaten(zeile):
    """Eingabe: Zeile der Eingabedatei, also: 'x1 y1 ... x4 y4'.
    Rückgabe: Koordinaten der Eckpunkte unten links und
    oben rechts als Liste [x_ul, y_ul, x_or, y_or]."""
    x_koordinaten = [] # an den Positionen 0, 2, ... der Zeile
    y_koordinaten = [] # an den Positionen 1, 3, ... der Zeile
    for i, k in enumerate(zeile.split()):
        # "enumerate" ermöglicht zwei Laufvariablen:
        # i als Zähler, k für die Listenelemente
        if i%2 == 0:
            x_koordinaten.append(int(k))
        else:
            y_koordinaten.append(int(k))
    return [min(x_koordinaten), min(y_koordinaten),
            max(x_koordinaten), max(y_koordinaten)]

def abstand(x,y):
    """Berechnet Abstand des Punktes (x,y) zum Ursprung (0,0)."""
    return math.sqrt(x*x + y*y)
```

```

def berechne_abstand(x_ul, y_ul, x_or, y_or):
    """Der kleinste Abstand eines Rechtecks zum Nullpunkt
    wird berechnet. Gegeben sind die Koordinaten der Eckpunkte
    unten links und oben rechts: x_ul, y_ul, x_or, y_or"""
    if (x_ul <= 0 <= x_or):
        # senkrechte Seiten auf beiden Seiten der y-Achse und ...
        if (y_ul <= 0 <= y_or):
            # ... waagerechte Seiten auf beiden Seiten der x-Achse:
            # Fall 3, Fehler!
            return -1 # negativer Wert als Fehlercode
        else: # Fall 2: nur senkrechte Seiten
            # auf beiden Seiten der y-Achse
            return min(abs(y_ul), abs(y_or))
    elif (y_ul <= 0 <= y_or):
        # Fall 2: nur waagerechte Seiten
        # auf beiden Seiten der x-Achse
        return min(abs(x_ul), abs(x_or))
    else: # Fall 1: Rechteck liegt komplett in einem Quadranten
        return min(abstand(x_ul,y_ul),
                   abstand(x_ul,y_or),
                   abstand(x_or,y_ul),
                   abstand(x_or,y_or))

def hoehe(distanz):
    """Berechnet die erlaubte Höhe für eine (minimale) Distanz."""
    return (100 + int(distanz/100))

def skyline(dateiname):
    """Hauptfunktion"""
    with open(dateiname) as eingabe:
        anz_gebaeude = int(eingabe.readline())
        for i in range(0, anz_gebaeude):
            # lies die nächste Zeile und berechne die Hauptkoordinaten
            geb_koord = hauptkoordinaten(eingabe.readline())
            # berechne den Abstand
            abstand = berechne_abstand(geb_koord[0], geb_koord[1],
                                       geb_koord[2], geb_koord[3])

            # Ausgabe
            if abstand < 0:
                print("Fehler: Gebäude liegt über der Needle!")
            else:
                print("Erlaubte Höhe:", hoehe(abstand))

```

Songwriter

Das Duo „Fake that“ ist der aktuelle Stern am Pop-Himmel. Sie bringen einen Hit nach dem anderen heraus, aber nach einiger Zeit dämmert es selbst ihren größten Fans, dass die Texte ihrer Silben-Songs (so nennt das Duo seinen Stil) sich alle sehr ähneln. Das letzte Album hieß wohl nicht umsonst „It's Always The Same“.

Nellie, ein mittlerweile etwas gelangweilter Fan der beiden, hat festgestellt, wie die Texte der Silben-Songs funktionieren:

- > Eine *Silbe* wird aus einem Konsonanten und einem Vokal gebildet. Beispiele: 'do', 'nu', 'la'.
- > Eine *Zeile* besteht aus einer ungeraden Anzahl von Wiederholungen einer Grundsilbe, wobei derjenigen in der Mitte ein 'p di' angehängt wird. Beispiele: 'sup di', 'da dap di da', 'ne ne nep di ne ne'.
- > Eine *Strophe* besteht aus mindestens zwei Zeilen. Die Menge der Konsonanten und Vokale, aus denen die Grundsilben einer Strophe gebildet werden, ist immer sehr klein, z. B. {s, u, a}. Die Anzahl der Silbenwiederholungen ist für alle Zeilen einer Strophe gleich (Nellie spricht deshalb von der Silbenzahl einer Strophe). Am Ende einer Strophe steht gelegentlich ein markiger *Call* wie 'yeah!', 'yo man', 'fake that!' oder ähnlich pseudo-cooles Zeug.
- > Ein *Song* besteht aus mindestens zwei Strophen. Die Zeilenzahlen der Strophen eines Songs folgen einem Muster, z. B. „immer 3 Zeilen“, „abwechselnd 4 und 6 Zeilen“ usw. Auch die Silbenzahlen der Strophen folgen ähnlichen Mustern.

Nellie zweifelt nun an der Kreativität von „Fake that“. Solche Songtexte kann man sich bestimmt vom Computer schreiben lassen!

Junioraufgabe 1

Schreibe einen „Songwriter“, also ein Programm, das einen nach Nellies Regeln aufgebauten Text eines Silben-Songs erzeugen kann. Stelle dabei so weit wie möglich sicher, dass die von deinem Programm erzeugten Texte sich voneinander unterscheiden. Gib in der Dokumentation mindestens drei unterschiedliche Songtexte an, die dein Programm erzeugt hat.

Zollstock

Die Auszubildende Raluca hat einen 2 m langen Zollstock, der aus zehn gleich langen starren Segmenten besteht und an einem Ende ein Loch zum Aufhängen hat. Am Ende der Arbeitszeit hat sie es immer sehr eilig: Der Zollstock wird rasch irgendwie so zusammengefaltet, dass er in Ralucas Stoffbeutel passt. Dieser ist 50 cm lang und kann damit gerade zwei Segmente des Zollstocks in Verlängerung voneinander aufnehmen; aber er ist zu schmal für ein „quer stehendes“ Zollstocksegment.

Als Ralucas Verlobter diese lustige Gewohnheit bemerkt, sagt er zu ihr: „Morgen bist du schon zwei Monate im Betrieb. Bestimmt hast du den Zollstock an zwei Tagen in genau derselben Weise gefaltet“. Aber Raluca erwidert: „Das muss nicht sein. Es ist durchaus möglich, dass sich bisher keine Faltung wiederholt hat“.

Junioraufgabe 2

Finde heraus, wer von den beiden recht hat, und beschreibe deine Vorgehensweise.

Rationalisierung

Obwohl der Lehrer es nicht erlaubt, benutzt Anna bei ihren Mathe-Hausaufgaben immer den Taschenrechner. Als Ergebnis erhält sie dabei häufig eine Fließkommazahl. Damit ist der Lehrer aber nicht zufrieden; er erwartet als Ergebnis einen Bruch, also eine rationale Zahl p/q .

Einmal zeigte der Taschenrechner, nach einer langen Rechnung, die Zahl $x = 1,857143$ an. Anna gab ihr Ergebnis so an: $x = 1857143/1000000$. Aber das akzeptierte ihr Lehrer nicht; er sagte, das richtige Ergebnis sei $x = 13/7$. Da erinnerte sich Anna daran, dass in den meisten Schulaufgaben die Ergebnisbrüche kleine Zähler und Nenner haben.

Für die Zukunft sucht Anna also immer eine rationale Zahl, die „möglichst nahe“ am Taschenrechner-Ergebnis ist und „möglichst kleine“ Zähler und Nenner hat.

Aufgabe 1

Schreibe ein Programm, das Anna hilft.

Beschreibe dafür genauer, was mit den Zielsetzungen „möglichst klein“ und „möglichst nahe“ gemeint sein könnte und wie diese verschiedenen Ziele miteinander vereinbart werden können. Was gibt dein Programm für die folgenden Fließkommazahlen aus: 1,293982; 3,141593; 1,414214 ?

Tutorien

Zu vielen großen Vorlesungen mit teilweise hunderten Studenten werden Übungsgruppen eingerichtet, deren Größe etwa mit der einer Schulklasse verglichen werden kann (ca. 25). In diesen können Fragen zur Vorlesung diskutiert und Übungsaufgaben besprochen werden. Eine derartige Übungsgruppe wird Tutorium genannt und die Leiter dieser Tutorien heißen Tutoren. Die Tutorien finden einmal wöchentlich statt. Jeder Student der Vorlesung soll an genau einem dieser Tutorien teilnehmen.

Wir nehmen an, dass eine Universität für eine Vorlesung n Tutoren beschäftigt, die jeweils ein Tutorium leiten. Den Studenten, die die Tutorien besuchen sollen, werden jedoch zunächst $n + 1$ Termine vorgeschlagen. Sie melden sich über eine Internetseite für einen dieser $n + 1$ Termine an und geben zusätzlich 1-2 Ausweichtermine an.

Nach Abschluss der Befragung wird der am wenigsten nachgefragte Termin gestrichen, so dass nur die verbleibenden n Tutorien stattfinden. Die für das gestrichene Tutorium angemeldeten Studenten werden auf einen ihrer Ausweichtermine verteilt.

Komplizierter wird das Ganze dadurch, dass die Tutoren ihrerseits andere Vorlesungen hören und deswegen nur zu bestimmten Zeiten zur Verfügung stehen. Deshalb gibt es nur eine begrenzte Anzahl t von überhaupt möglichen Terminen.

Die Liste, die den Studenten zur Anmeldung vorgelegt wird, muss deshalb in folgendem Sinne „brauchbar“ sein: Egal, welcher unter den $n + 1$ Terminen der Anmeldeliste gestrichen wird, steht für jeden der übrigen n Termine ein Tutor zur Verfügung. Dabei muss jeder Tutor genau ein Tutorium leiten.

Beispiel: Es gibt 6 mögliche Termine und 4 Tutoren; 5 Termine sollen den Studenten zur Anmeldung vorgelegt werden ($t = 6, n = 4$). Die Tutoren stehen zu den in der Tabelle angekreuzten Terminen zur Verfügung:

	Di 08:00	Di 13:00	Mi 08:00	Mi 13:00	Do 08:00	Do 13:00
Helge	X		X			
Andrea		X	X		X	
René		X		X	X	
Denniz				X		X

Eine brauchbare Anmeldeliste für die Studenten ist:
Di 08:00, Di 13:00, Mi 08:00, Mi 13:00, Do 13:00

Nicht brauchbar ist diese Anmeldeliste: Di 08:00, Di 13:00, Mi 08:00, Mi 13:00, Do 08:00.
Der Grund: Sollte der Termin Mi 13:00 gestrichen werden, stehen für die übrigen vier Termine nur noch drei Tutoren zur Verfügung – Denniz leider nicht.

Aufgabe 2

Schreibe ein Programm, das Angaben über die Verfügbarkeit jedes Tutors an jedem der t möglichen Termine als Eingabe erwartet und eine brauchbare Anmeldeliste von $n + 1$ aus diesen t möglichen Terminen berechnet (falls dies möglich ist). Beschreibe die genauen Eingabe- und Ausgabeformate deines Programms und zeige seine Wirkungsweise anhand mehrerer aussagekräftiger Beispieleingaben.

Erstelle außerdem ein Testprogramm, mit dem überprüft werden kann, ob die Anmeldeliste von $n + 1$ Terminen wirklich brauchbar ist.

Vortänzer

In einem Roboter-Wettbewerb gibt es eine neue Disziplin: Dance-Challenge. Dabei soll ein Roboter des Typs Dancemaster 2000 so programmiert werden, dass er einen vorgegebenen Tanz imitiert.

Der Dancemaster 2000 orientiert sich an einem Raster: Bei einem Schritt bewegt er sich von einem Rasterpunkt zu einem der benachbarten Punkte (auch diagonal). Der Roboter startet immer am gleichen Punkt im Raster und mit gleicher Ausrichtung.

Den Dancemaster 2000 kann man mit den folgenden Bewegungsbefehlen programmieren:

F	1 Schritt vorwärts
B	1 Schritt rückwärts
l	Drehung um 45° nach links
r	Drehung um 45° nach rechts
-	Pause: mache nichts

Die Ausführung eines Bewegungsbefehls dauert 1 Sekunde.

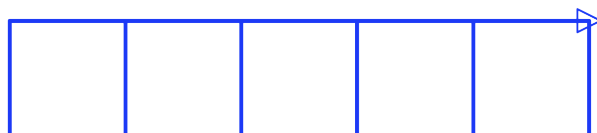
Zur Abkürzung von Dancemaster-Programmen gibt es einen Wiederholungsbefehl:

***n* Befehlsfolge .**

bedeutet: Führe die Befehlsfolge n -mal aus ($1 \leq n \leq 9$).

Zum Beispiel lassen die Programme **FrrFrrFrrF** und **4Frr.** den Dancemaster je ein Quadrat tanzen. **54Frr.F.** tanzt fünf Quadrate nebeneinander (s. Bild A; aus Platzgründen ist das Bild um 90° im Uhrzeigersinn gedreht).

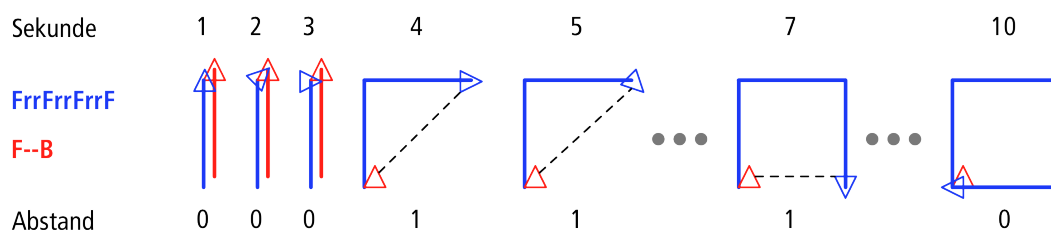
Bild A: **54Frr.F.**



Ziel des Wettbewerbs ist es, ein Tanzprogramm T mit einem anderen, möglichst kurzen Programm I möglichst genau zu imitieren und so für dieses „Imitat“ I möglichst wenig Strafpunkte zu erhalten. Weil der Befehlsspeicher knapp ist, kostet jedes Zeichen von I genau 3 Strafpunkte. Außerdem wird der Abstand zwischen den durch T bzw. I erreichten Positionen nach jeder Sekunde bestimmt und zu den Strafpunkten addiert. Der Abstand zwischen zwei Rasterpunkten ist die kleinste Anzahl von Schritten (ohne Drehung), um vom einen zum anderen Punkt zu gelangen.

Beispiel: Im Vergleich mit dem Tanzprogramm **FrrFrrFrrF** kostet das Imitat **4Frr.** 15 Strafpunkte; **F--B** kostet 18 Strafpunkte (s. Bild B).

Bild B:



Aufgabe 3

Bestimme für die Tanzprogramme in der Tabelle unten jeweils ein Imitat I , das höchstens so viele Strafpunkte kostet wie in der Tabelle angegeben. Kandidaten für I kannst du von Hand finden. Schreibe zu deiner Hilfe ein Programm, das für ein Tanzprogramm T und ein Imitat I die Strafpunkte für I berechnet.

Tanzprogramm	Strafpunktobergrenze
4FFFFrr.	21
FrrFllFrrFrrrFF	20
lFrrFFllFFFrrFllFrrFF	50

Fotoforensik

Ein Mitarbeiter hat sich in der Kaffeepause selbst fotografiert, um zu dokumentieren, wie toll er mit der Kaffeekanne jonglieren kann. Leider klappte das aber nicht, und die Kaffeekanne zerbrach, was auf dem Foto zu erkennen ist. Da das Foto mit einem Smartphone aufgenommen wurde, wurde es zudem sofort automatisch auf den Server hochgeladen.

Jetzt fragt sich die Chefin, wer die Kaffeekanne zerbrochen hat. Der Übeltäter meldet sich nicht, weil er sich seines Missgeschicks zu sehr schämt. Er soll aber gar nicht bestraft werden; die Chefin stellt sich nur aus akademischem Interesse die Frage, ob anhand des gemachten Bildes die zugehörige Kamera bestimmt werden könnte. Alle Mitarbeiter haben einige Bilder auf der Firmenwebseite veröffentlicht, die alle in der gleichen Auflösung vorliegen.

Aufgabe 4

Enthält ein solches Bild einen eindeutigen „Fingerabdruck“ seiner Kamera? Können kamera-spezifische Unterschiede in der Lichtempfindlichkeit der individuellen Pixel des CMOS-Sensors jeder Kamera zu statistisch signifikanten Eigenheiten führen? Gehe dieser Frage nach! Es stehen Dir neben dem Tatfoto auch eine Menge Fotografien zur Verfügung, die fünf einzelnen Kameras zugeordnet sind. Welche dieser Kameras gehört dem erfolglosen Kaffeekannensjongleur?

Unter www.bundeswettbewerb-informatik.de findest du alle Bilder mit einer Beschreibung des verwendeten Formats.

Versteigerung

Die Herstellung des Materials Sehrteurium ist aufwendig, und das Verfahren ist streng geheim. Die beiden Abnehmer Alphasoft und Betahard benötigen das Material dringend für ihre Produktion. Der Hersteller von Sehrteurium versteigert am Ende jeden Monats die monatliche Produktionsmenge von 20 Gramm nach folgenden Regeln:

Die Vertreter von Alphasoft und Betahard müssen zur Versteigerung mit je 1000 Unzen Gold erscheinen. Sie bieten dann gleichzeitig eine frei wählbare Anzahl ihrer Goldunzen auf die ersten 2 Gramm Sehrteurium. Nach der Abgabe werden die Gebote für beide sichtbar. Den Zuschlag erhält, wer die meisten Unzen geboten hat; bei Gleichstand erhält jeder Bieter 1 Gramm. Beide Bieter müssen ihren jeweiligen gebotenen Betrag bezahlen – auch der unterlegene. Ein Gebot von 0 Unzen ist erlaubt. Das Bieten auf jeweils zwei Gramm wird solange wiederholt, bis der Vorrat an Sehrteurium komplett versteigert ist.

Jeder Bieter möchte selbstverständlich eine größere Menge ersteigern als sein Konkurrent. Deshalb hat die Firma Alphasoft ein Computerprogramm entwickelt, das für jede Situation ein Gebot berechnet.

Firma Betahard möchte ebenfalls ein solches Programm besitzen und schreibt ein Turnier aus, in dem Programme in simulierten Versteigerungen gegeneinander antreten. In einer Versteigerung gewinnt das Programm, das in der Lage ist, mehr Sehrteurium zu ersteigern als das andere. Bei Gleichstand gewinnt das Programm, das mehr Gold übrig behält.

Aufgabe 5

Schreibe ein Programm, welches an solch einer Versteigerung teilnehmen kann, und erlautere seine Strategie.

Miss dein Programm im BWINF-Turniersystem (turnier.bundeswettbewerb-informatik.de) mit anderen Programmen; auch das Alphasoft-Programm ist dort zu finden. Du hast die Aufgabe erfolgreich gelöst, wenn dein Programm das Alphasoft-Programm meistens schlägt. Dieses bietet immer den gleichen Betrag: 100 Unzen.

Nach dem Einsendeschluss wird ein Turnier durchgeführt, in dem jedes Programm gegen jedes andere zweimal antritt. Die Bestplatzierten des Turniers erhalten einen Sonderpreis.

Teilnehmen

Einsendeschluss ist der 25. November 2013

Anmelden und Einsenden

online unter

www.bundeswettbewerb-informatik.de

Fragen zu den Aufgaben?

per Telefon:

0228 378646

zu üblichen Bürozeiten

per E-Mail:

bwinf@bwinf.de

Diskutiere über die Aufgaben mit den Mitgliedern der **EI Community**:

www.einstieg-informatik.de

Einsenden – was und wie?

Für jede bearbeitete Aufgabe solltest du im schriftlichen Teil deiner Einsendung (der **Dokumentation**)

- > deine **Lösungsidee** beschreiben;
- > die **Umsetzung** der Idee in ein Programm (falls gefordert) erläutern;
- > mit genügend **Beispielen** zeigen, dass und wie deine Lösung funktioniert; und
- > die wichtigsten Teile des **Quelltextes** einfügen.

Achtung: eine gute Dokumentation muss nicht lang sein – aber unbedingt Beispiele enthalten.

Bei Aufgaben mit Programmierung umfasst die **Implementierung** den kompletten Quelltext und das ausführbare Programm (Windows, Linux, MacOS oder Android).

Beispiele für Aufgabenbearbeitungen findest du auf den BwInf-Webseiten.

Eine Gruppe gibt gemeinsam eine **Einsendung** ab. Die Einsendung wird über das Online-Anmeldesystem als Dateiarchiv im ZIP-Format abgegeben. Dieses Archiv muss auf oberster Ebene enthalten:

- > die Dokumentationen: ein PDF-Dokument für alle Aufgaben oder je ein PDF pro Aufgabe;
- > zu jeder bearbeiteten Aufgabe einen Ordner mit den Implementierungsdateien (Programm, Quelltextdatei(en)).

Tipps

Unter www.bundeswettbewerb-informatik.de/tipps findest du

- > genauere Hinweise zur Einsendung;
- > Beispiele für Aufgabenbearbeitungen;
- > Hinweise auf nützliche fachliche Informationen.

Deine Chancen

Mit einer Teilnahme am Bundeswettbewerb Informatik kannst du nur gewinnen. In allen Runden gibt es **Urkunden** für Teilnahme und besondere Leistungen; zum Dank gibt es kleine **Geschenke** für alle.

Wer sich für die zweite Runde qualifiziert, kann mit Einladungen zu **Informatik-Workshops** rechnen: zum Jugendforum Informatik in Baden-Württemberg, dem Camp „Fit for BwInf“ des Hasso-Plattner-Instituts, den Informatiktagen der RWTH Aachen, der TU Braunschweig, der LMU München mit QAware GmbH und weitere mehr. Einige Teilnehmerinnen werden von Google zum **Girls@Google Day** eingeladen.

Nach der zweiten Runde winken die **Forschungstage Informatik** des Max-Planck-Instituts für Informatik. Ausgewählte Gewinner eines zweiten Preises in der zweiten Runde erhalten einen Buchpreis von O'Reilly. Eine Einsendung zur zweiten Runde kann in vielen Bundesländern als **besondere Lernleistung** in die Abiturwertung eingebracht werden.

Die Besten erreichen die **Endrunde**; dort werden Bundessieger und Preisträger ermittelt, die mit **Geldpreisen** belohnt werden. Bundessieger werden in der Regel ohne weiteres Auswahlverfahren in die **Studienstiftung** des deutschen Volkes aufgenommen.

www.bundeswettbewerb-informatik.de/chancen