



# 1000000.

## BUNDESWETTBEWERB INFORMATIK

  
Bundeswettbewerb  
Informatik

  
BUNDES  
WEIT  
INFORMATIK  
NACHWUCHS  
FÖRDERN

# 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 Wettbewerbs 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 kriminologischen Spürsinn eines modernen Sherlock Holmes gelöst werden müssen.

Der Bundeswettbewerb Informatik bietet Jugendlichen die Möglichkeit eines geistigen Kräfteressens besonderer Art. Dinge zu strukturieren, komplexe Systeme in überschaubare Teile zu zerlegen, zu formalisieren und zu interpretieren – diese Fähigkeiten sind gewissermaßen 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 Aufgaben 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)

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.

[www.gi.de](http://www.gi.de)

## Fraunhofer-Verbund 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](http://www.innovisions.de). Weitere Informationen über den Fraunhofer IuK-Verbund gibt es auf [www.iuk.fraunhofer.de](http://www.iuk.fraunhofer.de)

## Max-Planck-Institut für 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.

[www.mpi-inf.mpg.de](http://www.mpi-inf.mpg.de)



## Unter der Schirmherrschaft des Bundespräsidenten

Von der Kultusministerkonferenz empfohlener Schülerwettbewerb



GEFÖRDERT VOM



## 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!

# 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 Rossmann, 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](http://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.

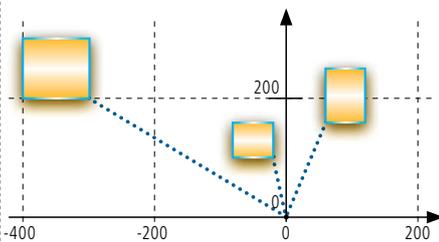
Das Bild unten zeigt drei Gebäude mit den Grundrissen (angegeben durch die Eckpunkte)  
(-80, 100); (-20, 100); (-20, 160); (-80, 160)  
(60, 160); (120, 160); (120, 250); (60, 250)  
(-400, 200); (-300, 200); (-300, 300); (-400, 300)

## 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 Hauptpunkte.

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:  $100 + \text{abrunden}(d/100)$ .

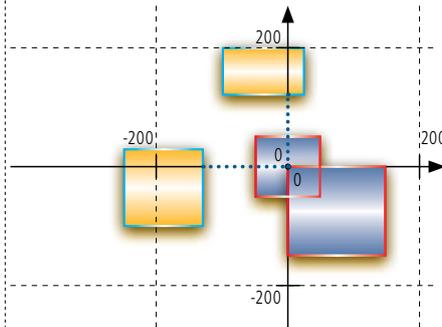
Bei der Bestimmung des nächsten Punktes müssen wir verschiedene Fälle unterscheiden:

**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“ liefere den kleinsten von zwei Werten):  $\min(|x_{ul}|, |x_{or}|)$  bzw.  $\min(|y_{ul}|, |y_{or}|)$ . Die beiden Rechtecke im Bild haben die Hauptpunkte (-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  $x_{ul} \leq 0 \leq x_{or}$  und  $y_{ul} \leq 0 \leq y_{or}$ . 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:  $x_1 y_1 x_2 y_2 x_3 y_3 x_4 y_4$

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
-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))
```