

# 33. Bundeswettbewerb Informatik

## Aufgabenblatt 1. Runde



Der 33. Bundeswettbewerb Informatik für Jugendliche bis 21 Jahre.

**Einsendeschluss ist der 24. November 2014.**

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

Online: [www.bundeswettbewerb-informatik.de](http://www.bundeswettbewerb-informatik.de)



### Grußwort



Digitale Technologien durchdringen unseren Alltag. Ein Leben ohne Handy und Computer, ohne Navigationssystem im Auto und ohne digitale medizinische Diagnostik ist heutzutage kaum noch vorstellbar. Auch für den Umbau der Energieversorgung und die Entwicklung integrierter Mobilitätssysteme brauchen wir Informatik.

Das Wissenschaftsjahr 2014 „Die digitale Gesellschaft“ befasst sich mit der Frage, was die Digitalisierung für unser Leben bedeutet. Dabei setzen wir uns sowohl mit den Chancen als auch mit den Herausforderungen auseinander, die mit der Entwicklung der digitalen Technologien verbunden sind. Ziel ist, technologische Zukunftsprozesse und gesellschaftliche Veränderungen mitzugestalten.

Die Bundesweiten Informatikwettbewerbe bieten eine gute Gelegenheit, auf besondere Weise in die faszinierende Welt der Informatik einzutauchen. Im Bundeswettbewerb Informatik können Schülerinnen und Schüler der Bedeutung von Informatik nachspüren und sich mit technologischen Zusammenhängen vertraut machen. Auch die anderen Informatikwettbewerbe wie der Informatik-Biber wecken Neugier für dieses Fach. Über die große Resonanz freue ich mich sehr. Über 200.000 Jugendliche haben im vergangenen Jahr am Informatik-Biber teilgenommen. Alle, die durch den Informatik-Biber Lust auf die Welt der Bits und Bytes bekommen haben, können ihr Interesse und ihr Wissen beim Bundeswettbewerb Informatik vertiefen.

Die Informatikwettbewerbe tragen dazu bei, die große Bedeutung von Informatik für die Entwicklung unserer Gesellschaft zu vermitteln. Den Teilnehmerinnen und Teilnehmern wünsche ich viel Spaß und gute Ideen.

Prof. Dr. Johanna Wanka  
Bundesministerin für Bildung und Forschung

## 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.

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

  
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.

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

GEFÖRDERT VOM

**Unter der Schirmherrschaft des Bundespräsidenten**

**Von der Kultusministerkonferenz empfohlener  
Schülerwettbewerb**



Bundesministerium  
für Bildung  
und Forschung

## Die Partner

Zusätzlich zur Förderung durch das Bundesministerium für Bildung und Forschung und seine Träger erfahren die Bundesweiten Informatikwettbewerbe (BWINF) und insbesondere der Bundeswettbewerb Informatik 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 33. 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. Der Bundeswettbewerb Informatik ist der traditionsreichste unter den Bundesweiten Informatikwettbewerben (BWINF), zu denen auch Informatik-Biber und das deutsche Auswahlverfahren zur Internationalen Informatik-Olympiade gehören. BWINF wird vom Bundesministerium für Bildung und Forschung gefördert; die Träger sind GI, Fraunhofer-Verbund IuK-Technologie und Max-Planck-Institut für Informatik. Die Bundesweiten Informatikwettbewerbe gehören zu den bundesweiten Schülerwettbewerben, die von den Kultusministerien der Länder empfohlen werden. Der Bundeswettbewerb Informatik steht unter der Schirmherrschaft des Bundespräsidenten.

Die Gestaltung des Wettbewerbs und die Auswahl der Sieger obliegen dem Beirat; Vorsitzender: Prof. Dr. Till Tantau, Universität Lübeck. Die Auswahl und Entwicklung von Aufgaben und die Festlegung von Bewertungsverfahren übernimmt der Aufgabenausschuss; Vorsitzender: Prof. Dr. Peter Rossmanith, RWTH Aachen. Die BWINF-Geschäftsstelle 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 bearbeitet im Team zwei Informatik-Probleme.

### **Juniorliga**

Für Schüler, die nicht älter als 16 Jahre und nicht in der Qualifikationsphase der gymnasialen Oberstufe 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 vollständige Einsendung zusätzlich 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 2015 von der TU Darmstadt 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 2016 in Russland 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 Dortmund, der TU Braunschweig und der LMU München (gemeinsam mit der QAware GmbH), von der Firma INFORM sowie vom 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 September 2015 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. **Neu:** Mindestens eine der gewerteten Einsendungen muss in Hauptliga oder Juniorliga mit einem ersten oder zweiten Preis ausgezeichnet werden.

Schulen, die diese Bedingungen erfüllen, werden als „BwInf-Schule 2014/2015“ 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.

### Biber goes BwInf

Teilnehmerinnen und Teilnehmer am Informatik-Biber sollen dazu angeregt werden, auch beim 33. Bundeswettbewerb Informatik mitzumachen. Deshalb werden Schulen, die unter

ihren BwInf-Erstteilnehmern mindestens fünf ehemalige Biber-Teilnehmer nachweisen können, mit einem besonderen „**Biber-goes-BwInf**“-**Schulpreis** ausgezeichnet. Die für den Preis gewerteten Schülerinnen und Schüler erhalten als Anerkennung einen „Biber goes BwInf“-USB-Stick. Weitere Informationen zu dieser Aktion gibt es online: für Schüler ([bwinf.de/bgb/schueler](http://bwinf.de/bgb/schueler)) und Lehrkräfte ([bwinf.de/bgb/lehrer](http://bwinf.de/bgb/lehrer)).

## Die Regeln

### Teilnahmeberechtigt

... sind Jugendliche, die nach dem 24.11.1992 geboren wurden. Sie dürfen jedoch zum 1.9.2014 noch nicht ihre (informatikbezogene) Ausbildung abgeschlossen oder eine Berufstätigkeit begonnen haben. Personen, die im Wintersemester 2014/15 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 24.11.2014 ihren Wohnsitz in Deutschland haben oder eine staatlich anerkannte deutsche Schule im Ausland besuchen.

Junioraufgaben dürfen von Jugendlichen bearbeitet werden, die jünger als 17 Jahre alt sind (geboren nach dem 24.11.1997) und, falls sie die Schule besuchen, noch nicht in der Qualifikationsphase der Sekundarstufe 2 sind. Ein Team darf Junioraufgaben bearbeiten, wenn mindestens ein Mitglied des Teams die genannten Bedingungen erfüllt.

### Weiterkommen

An der zweiten Runde dürfen jene teilnehmen, die allein oder mit ihrem Team 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 voraussichtlich noch keine zweite Runde.

### Einsendungen

... enthalten Bearbeitungen zu mindestens einer Aufgabe und werden von Einzelpersonen oder Gruppen (Teams) abgegeben. Eine Einsendung besteht für jede bearbeitete Aufgabe aus Dokumentation und (bei Aufgaben mit Programmierauftrag) Implementierung. Die **Dokumentation** enthält eine Beschreibung der Lösungsidee und Beispiele, welche 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** umfasst das (möglichst eigenständig 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 zu jeder bearbeiteten Aufgabe auf oberster Ebene enthalten:

- > die Dokumentation: ein PDF-Dokument;
- > die Implementierung: einen Ordner mit Programm- und Quelltextdateien.

### Anmeldung

Die Anmeldung ist bis zum Einsendeschluss möglich, und zwar online über: [pms.bwinf.de](http://pms.bwinf.de) Wettbewerbsteilnehmer können sich dort eigenständig registrieren, zum Wettbewerb anmelden und ggf. Teams bilden. Die Anmeldung zum Wettbewerb und das Bilden von Teams kann auch **von Lehrkräften vorgenommen** werden.

### Einsendeschluss: 24.11.2014

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: Songwriter

**Hinweis:** Der Aufgabentext wird hier nur der Vollständigkeit halber abgedruckt. Die Dokumentation zu einer Aufgabenbearbeitung muss und soll den Aufgabentext nicht enthalten.

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

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.

### Lösungsidee

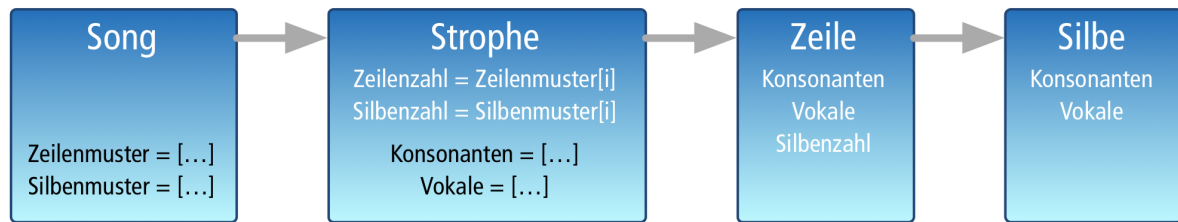
Die Lösung soll so funktionieren wie die Einheiten eines Silben-Songs beschrieben sind. Folgende, aufeinander aufbauende Einheiten gibt es: Silbe, Zeile, Strophe und Song. Für jede Song-Einheit soll ein eigener Programmteil entwickelt werden, der diese nach Vorschrift erzeugt. Dabei bauen auch die Programmteile aufeinander auf: 'Song' verwendet 'Strophe', 'Strophe' verwendet 'Zeile' und 'Zeile' verwendet 'Silbe'.

Die Programmteile sind nicht unabhängig voneinander:

- > Die Anzahl der Zeilen einer Strophe und auch die Anzahl der Silben in den Zeilen der Strophe sollen je einem Muster folgen. Wenn ein Muster eine Folge von Zahlen ist (eine Zahl pro Strophe), kann 'Song' jeder 'Strophe' die passende Zahl übergeben.
- > Für eine Strophe müssen die Mengen von Konsonanten und Vokalen bestimmt werden, aus denen die Silben der Zeilen bestehen müssen. 'Strophe' muss die beiden Buchstabenmengen an 'Zeile' übergeben (und 'Zeile' wiederum an 'Silbe').

Das folgende Bild stellt die Zusammenarbeit der verschiedenen Programmteile dar.





Für eine Strophe muss zusätzlich noch festgelegt werden, ob sie einen „Call“ hat. Wenn ja, wird einer ausgewählt und als zusätzliche Zeile hinten an die Strophe angefügt.

Schließlich muss noch beachtet werden, dass „die von deinem Programm erzeugten Texte sich voneinander unterscheiden“. Dazu kann bei der Bestimmung der Muster, der Konsonanten und Vokale einer Strophe, der Auswahl des Calls, der Auswahl von Konsonant und Vokal für eine Silbe usw. der Zufall eingesetzt werden.

## Umsetzung

Die Lösungsidee wird in ein Programm in der Sprache Python umgesetzt. Für jede Song-Einheit wird eine Funktion geschrieben, jeweils mit den nötigen Parametern:

```

Song:   def song()
Strophe: def strophe(anzahl_zeilen, anzahl_silben)
Zeile:  def zeile(anzahl_silben, konsonanten, vokale)
Silbe:  def silbe(konsonanten, vokale)
  
```

Außerdem wird eine Funktion benötigt, die die Muster (also die Zahlenfolgen) erzeugt. Die Funktion *muster*(laenge, grundmenge) liefert eine Folge von Elementen einer Grundmenge (z.B. der möglichen Zeilenanzahlen), wobei die Folge eine bestimmte Länge haben muss (z.B. die Anzahl der Strophen eines Songs). Die Folge ist zufällig von einer dieser drei Arten: (1) zwei sich abwechselnde Elemente, (2) alle Elemente gleich und (3) die Elemente der Grundmenge (nötigenfalls wiederholt, wenn die Grundmenge kürzer ist als die gewünschte Folgenlänge).

Die Zufallseffekte werden durch verschiedene Funktionen des Python-Moduls *random* erzielt:

```

random.choice(folge): wählt zufällig ein Element der Folge.
random.sample(folge, anzahl): wählt zufällig eine Anzahl von Elementen der Folge.
random.randint(int1, int2): wählt zufällig eine Integer-Zahl aus der Menge {int1, ..., int2}.
  
```

Der Song wird in der Funktion *strophe* zeilenweise ausgegeben.

## Beispiele

Durch einen Aufruf der Funktion *song* wird in der Python-Shell ein Songtext ausgegeben. Aus Platzgründen werden nur zwei Beispiele gezeigt.

```

>>> song()
da da dap di da da
bi bi bip di bi bi

bu bu bup di bu bu
do do dop di do do
BwInf rocks!
  
```

Dieser Song besteht aus zwei Strophen. Das Strophenmuster ist „immer zwei Zeilen“, das Silbenmuster ist „immer fünf Silben“. In Strophe 1 kommen die Konsonanten d und b sowie die Vokale a und i vor, in Strophe 2 b und d bzw. u und o. Strophe 2 hat einen Call.

```
>>> song()
li lip di li
li lip di li

le le lep di le le
sa sa sap di sa sa
se se sep di se se
sa sa sap di sa sa

gi gi gi gip di gi gi gi
bo bo bo bop di bo bo bo
yeah!
```

Dieser Song hat das Zeilenmuster [2,4,2] (zwei Zahlen abwechselnd) und das Silbenmuster [3,5,7] (die möglichen Silbenzahlen nacheinander). Nur die letzte Strophe hat einen Call.

### Quelltext

```
import random

vokale = ['a', 'e', 'i', 'o', 'u']
# nur geeignete Konsonanten - Geschmackssache
konsonanten = ['b', 'd', 'g', 'l', 'n', 's']
# mögliche Calls - auch 'leere' Calls
calls = ["yeah!", "fake that!", "BwInf rocks!", "", "", ""]
# mögliche Silbenanzahlen in den Zeilen - müssen ungerade sein
silbenzahlen = [3,5,7]
zeilenzahlen = [2,3,4,5] # mögliche Zeilenanzahlen in den Strophen
strophenzahlen = [2,3,4,5] # mögliche Strophenzahlen für einen Song

def silbe(konsonanten, vokale):
    """Bildet eine Silbe aus einem der Konsonanten und einem der Vokale."""
    return random.choice(konsonanten) + random.choice(vokale)

def zeile(konsonanten, vokale, anzahl_silben):
    """Bildet eine Zeile mit der gegebenen Anzahl von Silben."""
    # Die Silbe wird für die ganze Zeile einmal bestimmt.
    zeilensilbe = silbe(konsonanten, vokale)
    halbezeile = (anzahl_silben // 2) * (zeilensilbe + ' ')
    return halbezeile + zeilensilbe + 'p di ' + halbezeile

def call():
    """Sucht aus der Menge der Calls einen (eventuell leeren) aus."""
    auswahl = random.randint(0, len(calls)-1)
    return calls[auswahl]

def strophe(anzahl_zeilen, anzahl_silben):
    """Bildet eine Strophe mit den gegebenen Anzahlen für Zeilen und Silben."""
    anzahl_konsonanten = random.randint(2,3) # zwei oder drei Konsonanten
    auswahl_konsonanten = random.sample(konsonanten, anzahl_konsonanten)
    anzahl_vokale = random.randint(2,3) # zwei oder drei Vokale
    auswahl_vokale = random.sample(vokale, anzahl_vokale)
    for i in range(anzahl_zeilen):
        print(zeile(anzahl_silben, auswahl_konsonanten, auswahl_vokale))
    stropfen_call = call()
    # ein nicht-leerer Call wird an die Strophe angehängt
    if stropfen_call != "":
        print(stropfen_call)
    print("") # eine leere Zeile am Ende der Strophe
```

```
def muster(laenge, grundfolge):
    """Wählt eine Folge (der gegebenen Laenge) von Elementen
    der gegebenen Grundfolge auf drei verschiedene Arten aus."""
    folge = []
    auswahl = random.randint(1,3)
    if auswahl == 1: # Art 1: abwechselnd zwei verschiedene Elemente
        elementpaar = random.sample(grundfolge, 2) # zwei Elemente auswählen
        for i in range(laenge):
            folge.append(elementpaar[i%2]) # i%2 ergibt abwechselnd 0 und 1
    elif auswahl == 2: # Art 2: immer das gleiche Element
        element = random.choice(grundfolge) # Element auswählen
        for i in range(laenge):
            folge.append(element)
    else: # auswahl == 3; Art 3: Grundmenge der Reihe nach
        for i in range(laenge):
            folge.append(grundfolge[i%len(grundfolge)])
    return folge

def song():
    """Bildet einen vollständigen Song."""
    strophenzahl = random.choice(strophenzahlen)
    zeilenzahlmuster = muster(strophenzahl, zeilenzahlen)
    silbenzahlmuster = muster(strophenzahl, silbenzahlen)
    for i in range(strophenzahl):
        strophe(zeilenzahlmuster[i], silbenzahlmuster[i])
```

## Fähre füllen

Von Oberdellendorf verkehrt eine Autofähre über die Wesau nach Niederdellendorf. Die nächste Brücke ist weit entfernt; deshalb ist die Fähre sehr beliebt, und an der Anlegestelle ist die Warteschlange lang. Da ist es wichtig, so viele Autos wie möglich auf der Fähre unterzubringen.

Der Fährbegleiter kann die Fahrzeuge – Pkws, Pkws mit Anhängern, Kleinlaster usw. – bei der Auffahrt jeweils in eine der drei, jeweils 20 m langen Parkbahnen der Fähre dirigieren. Er weist wiederholt das Fahrzeug am Kopf der Warteschlange einer Parkbahn zu und kennt dabei die Länge dieses Fahrzeuges, aber nicht die der nachfolgenden Fahrzeuge. Zwischen zwei aufeinanderfolgenden Fahrzeugen auf einer Parkbahn wird ein Abstand von 30 cm frei gelassen. Sobald das Fahrzeug am Kopf der Warteschlange nicht mehr untergebracht werden kann, legt die Fähre ab.

### Junioraufgabe 1

1. Finde zwei einfache Strategien A und B, die der Fährbegleiter bei der Zuweisung der Fahrzeuge zu den Parkbahnen anwenden könnte. Wähle die Strategien so, dass bei manchen Folgen von wartenden Fahrzeugen die Strategie A und bei manchen anderen die Strategie B mehr Fahrzeuge auf der Fähre unterbringt.
2. Schreibe ein Programm, das eine Folge von Längen von wartenden Fahrzeugen einliest und übersichtlich ausgibt, wie die Strategien A und B mit dieser Eingabe die Fähre beladen.
3. Wende dein Programm auf die folgenden drei Beispiele an. Angegeben sind jeweils die Längen einiger Fahrzeuge (in Metern) in der Reihenfolge, in der sie in der Warteschlange stehen:

6,96; 5,06; 3,77; 3,95; 3,91; 3,54; 4,26; 4,03; 5,43; 4,04; 4,43; 4,12; 2,78  
4,14; 3,63; 3,92; 7,95; 5,23; 3,30; 4,86; 15,06  
5,23; 4,41; 3,33; 13,13; 9,12; 4,38; 6,34; 5,37; 4,11; 3,74; 10,62

Belege mit diesen oder anderen Beispielen, dass manchmal Strategie A und manchmal Strategie B besser abschneidet.

## Zahlenspiel

Ulla gibt regelmäßig Nachhilfe. Immer wieder wollen die Schüler und Schülerinnen das Kürzen von Brüchen üben. Deshalb möchte Ulla gute Übungsaufgaben zum Kürzen automatisch erzeugen lassen: Ein Programm soll Brüche  $a/b$  wie  $144/56$  erzeugen, die dann von den Schülern zu  $p/q$  möglichst weit gekürzt werden, also  $144/56 = 18/7$ .

Sie benötigt Aufgaben von drei Schwierigkeitsstufen: leicht, mittel und schwer. Die Schwierigkeit der Aufgabe „Kürze  $a/b$ “ (mit Ergebnis  $p/q$ ) hängt von der Größe der Zahlen  $a$ ,  $b$ ,  $p$  und  $q$  ab.

Ulla nennt die Gesamtanzahl der Ziffern in  $a$  und  $b$  die *Länge* des Bruchs  $a/b$ ; z.B. hat  $144/56$  die Länge 5. Sie legt die Schwierigkeitsstufen wie folgt fest:

Stufe	Länge von $a/b$	Bedingung für $p/q$
leicht	4	$p+q \leq 10$
mittel	5	$10 < p+q \leq 20$
schwer	5	$20 < p+q \leq 30$

Die für eine Stufe angegebenen Bedingungen müssen beide erfüllt sein. Außerdem müssen die Aufgaben weitere Bedingungen erfüllen:

>  $a/b$  soll tatsächlich gekürzt werden können, d.h. es soll  $q < b$  sein.

> Es soll immer  $a \neq b$  sein.

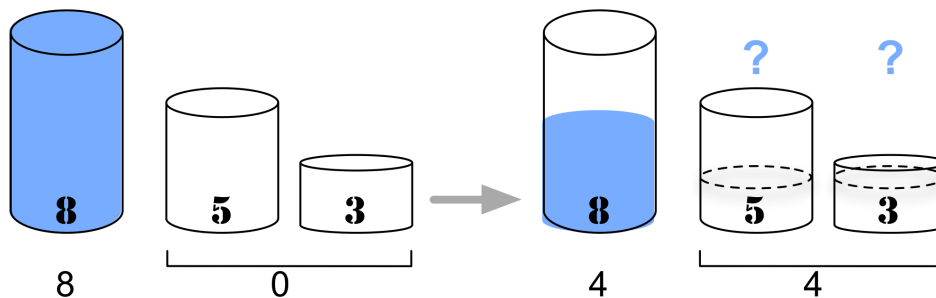
Wie man sieht, ist die Aufgabe „Kürze  $144/56$ “ schwer, und „Kürze  $22/77$ “ ist leicht.

### Junioraufgabe 2

Hilf Ulla und schreibe ein Programm, das für eine gegebene Schwierigkeitsstufe eine gewünschte Anzahl von zufällig erzeugten Aufgaben der angegebenen Stufe mit ihren Lösungen ausgibt.

## Faires Füllen

In einem Jahrhunderte alten Rätsel treffen sich zwei Personen. Die erste hat einen vollen Behälter mit acht Maßen Wein bei sich, die andere zwei leere Behälter mit einem Fassungsvermögen von drei und fünf Maßen. Können sie den Wein ohne Zuhilfenahme von anderen Behältnissen so umfüllen, dass jeder vier Maße in seinen Behältern hat? Ja, aber ganz einfach findet man die Lösung nicht.



Wir wollen dieses Rätsel hier verallgemeinern und mit Computerhilfe lösen lassen. Wieder treffen sich zwei Personen. Jede trägt eine bestimmte Menge von Behältern mit sich. Es ist bekannt, wie groß und wie voll diese Behälter zunächst sind.

### Aufgabe 1

Schreibe ein Programm, das diese Daten einliest und dann feststellt, ob ein Umfüllen so möglich ist, dass beide Personen die gleiche Gesamtmenge Weins in ihren Behältern vorfinden. Wenn ja, dann soll das Programm einen übersichtlichen Plan ausgeben, wie das Umfüllen in einer minimalen Anzahl von Schritten vollführt werden kann. Ein Schritt ist dabei das Gießen des Weins von einem Behälter A in einen Behälter B, bis entweder A leer oder B voll ist.

Wende dein Programm auf das ursprüngliche Rätsel an sowie auf die Beispiele, die du unter [www.bundeswettbewerb-informatik.de](http://www.bundeswettbewerb-informatik.de) findest.

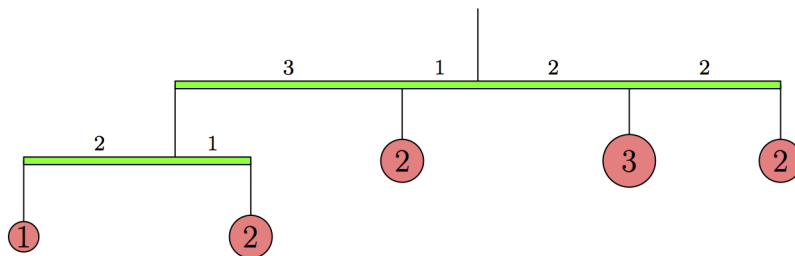
## Mobile

Ein Mobile besteht aus waagerechten Balken, senkrechten Drähten und an Drähten hängenden Figuren. Die Gewichte der Balken und Drähte können vernachlässigt werden, während die Gewichte der Figuren die möglichen Formen des Mobiles beeinflussen.

Jeder Balken hat einen Aufhängepunkt und weitere Stellen, an denen tiefer hängende Balken oder Figuren an ihm befestigt sind. Der Balken ist balanciert, wenn die Gewichte der an ihm befestigten Strukturen, multipliziert mit den jeweiligen Abständen ihrer Befestigung vom Aufhängepunkt des Balkens, die Summe Null haben, falls Abstände auf einer Seite vom Aufhängepunkt mit 1, die auf der anderen Seite mit  $-1$  multipliziert werden. Das gesamte Mobile ist balanciert, wenn alle seine Balken balanciert sind.

Zum Beispiel ist das unten abgebildete Mobile balanciert, denn:

$$1 \cdot 2 + 2 \cdot (-1) = 0 \text{ und } (1+2) \cdot 4 + 2 \cdot 1 + 3 \cdot (-2) + 2 \cdot (-4) = 0.$$



### Aufgabe 2

1. Erfinde und beschreibe eine einfache textuelle Darstellung von Mobiles, die eine Überprüfung der Balanciertheit erlaubt. Nicht für diesen Zweck relevante Aspekte, wie zum Beispiel die Länge der Drähte, sollen außer Acht gelassen werden. Zeige unter anderem das oben abgebildete Mobile in deiner Darstellung.
2. Schreibe ein Programm, das eine Folge  $w_1, \dots, w_n$  von Gewichten von  $n$  Figuren einliest und die textuelle Darstellung eines balancierten Mobiles mit den  $n$  Figuren ausgibt. Aus ästhetischen Gründen sollen an keinem Balken mehr als vier Strukturen befestigt sein.

*Freiwillige Zusatzaufgabe, ohne Einfluss auf die Bewertung:* Erweitere dein Programm so, dass es auch eine nette Zeichnung des gefundenen Mobiles produziert.

## Buffet-Lotterie

Bei der Endrunde des Bundeswettbewerbs Informatik haben die Teilnehmer es satt, Warteschlangenfutter vor dem großen Buffet im engen Korridor und Opfer der Last-Come-Longest-Hungry-Mentalität zu sein. Stattdessen soll ganz elegant und zivilisiert ausgelost werden, wer als Nächster das Buffet aufsuchen darf.

Die 28 Teilnehmer stellen sich dazu in einem großen Kreis auf und sagen den Satz

In-for-ma-tik kann uns wei-sen, wer als Nächs-ter kommt zum Spei-sen.

wiederholt laut auf. Wie beim Ene-Mene-Muh spricht jeder Teilnehmer nur eine Silbe, dann ist sein rechter Nachbar an der Reihe. Und wer die letzte Silbe des Satzes sagt, ist der Glückliche, der den Kreis verlassen und als Nächster seinen Hunger stillen darf.

Eine Teilnehmerin hat aber Geburtstag. Sie spricht natürlich die allererste Silbe, und als besondere Gunst darf sie, wann immer sie an der Reihe ist, statt einer auch zwei Silben sprechen – wenn sie das denn will. Da ihr Magen knurrt, möchte sie ihren Vorteil dazu nutzen, so schnell wie möglich zum Buffet zu kommen.

### Aufgabe 3

Schreibe ein Programm, das für eine gegebene Anzahl von Teilnehmern berechnet, wann das Geburtstagskind zwei Silben sprechen soll, um sich den bestmöglichen Platz in der Buffetreihenfolge zu verschaffen.

Dokumentiere die Wirkungsweise deines Programms für verschiedene Teilnehmerzahlen, unter anderem für die oben genannten 28 Teilnehmer.



## Alphametiken

Alphametiken sind Rechenrätsel; das wohl bekannteste Beispiel ist

$$\text{SEND} + \text{MORE} = \text{MONEY}$$

welches ein Globetrotter an seine Heimatadresse geschickt haben soll.

Die Frage ist, wie jeder Buchstabe so durch eine Ziffer ersetzt werden kann, dass die Rechnung aufgeht. Dabei sollen gleiche Buchstaben durch gleiche Ziffern und verschiedene Buchstaben durch verschiedene Ziffern ersetzt werden. Mit etwas Nachdenken lässt sich die Lösung finden: Setzt man  $O = 0$ ,  $M = 1$ ,  $Y = 2$ ,  $E = 5$ ,  $N = 6$ ,  $D = 7$ ,  $R = 8$  und  $S = 9$ , dann ergibt sich  $9567 + 1085 = 10652$ .

Auch auf Deutsch gibt es solche Rätsel:

$$\text{SUCHEN} - \text{MACHT} = \text{SPASS}$$

Kannst du die Lösung finden?

### Aufgabe 4

1. Im Allgemeinen ist eine Alphametik eine Gleichung, die aus Buchstaben und den Operatoren  $+$ ,  $-$ ,  $*$ , und  $/$  besteht. Die Operatoren dürfen nur auf der linken Seite der Gleichung auftreten. Schreibe ein Programm, das eine Alphametik einliest und eine Lösung dafür findet – oder ausgibt, dass keine Lösung existiert.
2. Nun geht es darum, solche Alphametiken selbst zu konstruieren. Wir wollen uns dabei auf „Zahl-Alphametiken“ beschränken, die nur aus deutschen Zahlwörtern bestehen und auch „gelesen“ korrekte Rechnungen beschreiben. Zum Beispiel hat die Zahl-Alphametik

$$\text{ZWEI} + \text{VIER} = \text{SECHS}$$

unter anderem die Lösungen  $4236 + 9635 = 13871$  und  $8624 + 3427 = 12051$ .

Ein weiteres Beispiel ist:

$$\text{EINS} + \text{EINS} + \text{EINS} + \text{EINS} = \text{VIER}$$

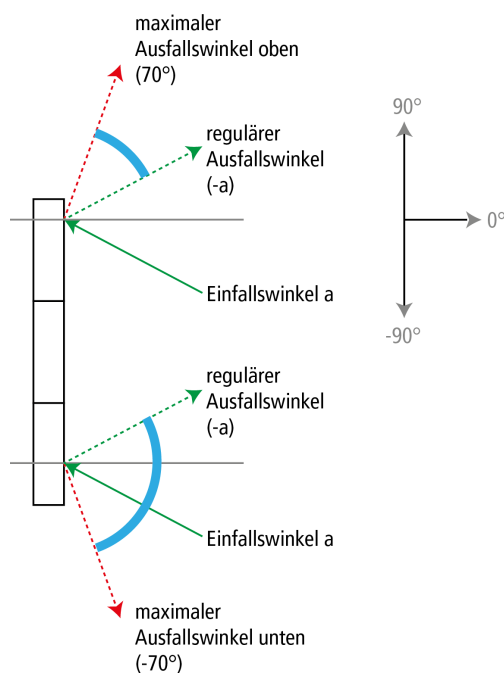
Schreibe ein Programm, das systematisch nach Zahl-Alphametiken sucht und diese mitsamt Lösungen ausgibt.

Kann dein Programm eine Zahl-Alphametik finden, die mehr als 20 Buchstaben enthält?

## Pong

In dem alten Konsolenspiel Pong steuert jeder der beiden Spieler einen Schläger. Der Ball prallt sowohl von den Schlägern als auch von den oberen und unteren Spielfeldbegrenzungen ab. Verfehlt man den Ball mit dem eigenen Schläger, bekommt der Gegner einen Punkt. Eine genauere Beschreibung findest du z. B. unter [de.wikipedia.org/wiki/Pong](http://de.wikipedia.org/wiki/Pong).

Im BWINF-Turniersystem ist eine Pong-Variante implementiert: Das Spielfeld besteht aus horizontal 65 mal vertikal 60 quadratischen Zellen. Die Schläger befinden sich in der ganz linken bzw. ganz rechten Spalte und belegen je 6 zusammenhängende Zellen. Der Ball hat die Größe einer Zelle. Oben und unten wird das Spielfeld durch Banden begrenzt, von denen der Ball nach der Regel „Einfallswinkel = Ausfallswinkel“ abprallt.



Wie der Ball von einem Schläger abprallt, hängt davon ab, welches Drittel des Schlägers er trifft:

- Trifft der Ball das mittlere Drittel des Schlägers, prallt er nach der Regel „Einfallswinkel = Ausfallswinkel“ ab.
- Trifft der Ball das obere Drittel, prallt er in eine zufällige Richtung zwischen dem regulären Ausfallswinkel (wie oben) und  $70^\circ$  ab.
- Trifft der Ball das untere Drittel, prallt er in eine zufällige Richtung zwischen  $-70^\circ$  und dem regulären Ausfallswinkel ab.

In der Skizze zeigt die blaue Farbe die Richtungen, in die der Ball von den äußeren Dritteln abprallen kann.

Man kann dem Gegner eine Überraschung bescheren, indem man sich im letzten Moment entscheidet, den Ball nicht mit der Mitte, sondern mit einem äußeren Drittel anzunehmen.

## Aufgabe 5

Programmiere einen Spieler für Pong, der den linken Schläger steuert.

Das Spiel wird in Schritten gespielt. In jedem Schritt darfst du den Schläger stehen lassen oder um eine Zelle nach oben oder unten bewegen. Dabei sieht dein Programm das komplette Spielfeld, also die Positionen der beiden Schläger sowie die Zelle, auf der der Ball dargestellt wird. Beachte, dass der Ball sich kontinuierlich bewegt und daher die sichtbare Position seine tatsächliche Position nur annähert. Zu Beginn bewegt sich der Ball mit einer Geschwindigkeit von einer Zellenseitenlänge pro Schritt. Die Geschwindigkeit wird im Laufe des Spiels immer weiter gesteigert.

Beim Einwurf stehen beide Schläger in der Mitte, und der Ball startet vom Schläger eines Spielers in einem zufälligen Winkel. Sobald ein Punkt erzielt wurde, wird der Ball neu eingeworfen. Der Einwurf passiert abwechselnd von links und rechts.

Ein Spiel besteht aus zwei Halbzeiten. Jeder Spieler bekommt in einer Halbzeit den ersten Einwurf. Eine Halbzeit besteht aus einer festgelegten Zahl von Schritten. Der Spieler mit den meisten Punkten gewinnt.

Miss dein Programm im BWINF-Turniersystem ([turnier.bundeswettbewerb-informatik.de](http://turnier.bundeswettbewerb-informatik.de)) mit anderen Programmen. 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 24. November 2014.**

### Anmelden und Einsenden

online unter  
[pms.bwinf.de](http://pms.bwinf.de)

### Fragen zu den Aufgaben?

per Telefon:  
**0228 378646**  
zu üblichen Bürozeiten

per E-Mail:  
[bwinf@bwinf.de](mailto:bwinf@bwinf.de)

Diskutiere über die Aufgaben mit anderen Teilnehmern in der **EI Community**:  
[www.einstieg-informatik.de](http://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 X oder Android).

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

- > die Dokumentation: ein PDF-Dokument;
- > die Implementierung: einen Ordner mit Programm- und Quelltextdateien.

Eine Gruppe gibt gemeinsam eine Einsendung ab.

### Tipps

Unter [www.bundeswettbewerb-informatik.de/tipps](http://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 oder der LMU München mit QAware GmbH und weitere mehr. Google lädt einige Teilnehmerinnen zum **Girls@Google Day** ein.

Nach der zweiten Runde winken die **Forschungstage Informatik** des Max-Planck-Instituts für Informatik und Buchpreise von O'Reilly für ausgewählte Gewinner eines zweiten Preises. 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, genau so wie die Mitglieder des deutschen IOI-Teams.

[www.bundeswettbewerb-informatik.de/chancen](http://www.bundeswettbewerb-informatik.de/chancen)

### Biber goes BwInf

Hast du bereits beim Informatik-Biber mitgemacht und nimmst nun zum ersten Mal am BwInf teil, erhältst du als kleine Anerkennung einen speziellen „Biber-goes-BwInf“-USB-Stick. Außerdem kannst du deiner Schule helfen, den „Biber-goes-BwInf“-Schulpreis zu gewinnen.

[www.bwinf.de/bgb/schueler](http://www.bwinf.de/bgb/schueler)