35. Bundeswettbewerb Informatik
Aufgabenblatt 1. Runde

Der 35. Bundeswettbewerb Informatik für Jugendliche bis 21 Jahre.

Einsendeschluss ist der 28. November 2016.

Informationen und Unterlagen bitte anfordern bei:
BWINF, Reuterstr. 159, 53113 Bonn
bwinf@bwinf.de

Online: www.bundeswettbewerb-informatik.de

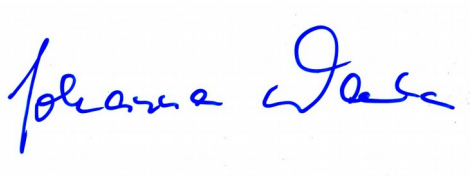
Grußwort

Selbstfahrende Autos, Roboter oder handybasierte Bezahlsysteme: Die Digitalisierung verändert und prägt unsere Zeit. Sie eröffnet neue Möglichkeiten, die unsere Gesellschaft weiterentwickeln oder uns den Alltag erleichtern können. Um diese Chancen zu nutzen, brauchen wir Menschen, die die digitale Zukunft mit Fachkompetenz, Kreativität und kritischem Verstand gestalten.

Deshalb ist es dem Bundesministerium für Bildung und Forschung (BMBF) ein wichtiges Anliegen, besonders junge Menschen für die Informatik zu begeistern. Dazu bieten die bundesweiten Informatikwettbewerbe eine gute Gelegenheit: Sie zeigen den Teilnehmerinnen und Teilnehmern, wie spannend Informatik ist und unterstützen sie, die eigenen Talente und Begabungen zu entdecken und entwickeln.

Die steigenden Teilnahmezahlen belegen das große Interesse junger Menschen an diesem spannenden Zukunftsfeld. Etwa 250 000 Kinder und Jugendliche haben im vergangenen Jahr am Informatik-Biber, einem Wettbewerb für die Klassenstufen 3 bis 13, teilgenommen. Um noch mehr jungen Menschen den Reiz der Informatik nahe zu bringen, startet in diesem Jahr der „Grundschul-Biber“. Eine neue Juniorliga wird außerdem in Zukunft weitere Möglichkeiten zur eigenen Interessensvertiefung geben.

Ich freue mich, dass sich so viele Schülerinnen und Schüler an den bundesweiten Informatikwettbewerben beteiligen. Ihnen allen wünsche ich gute Ideen und viel Erfolg.



Prof. Dr. Johanna Wanka
Bundesministerin für Bildung und Forschung

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

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 20 Instituten ermöglicht die Entwicklung übergreifender branchenspezifischer IT-Lösungen, oft zusammen mit Partnern aus der Industrie, sowie anbieterunabhängige Technologieberatung. Entwickelt werden IuK-Lösungen in den Branchenfeldern Mobilität und Transport, E-Government, Öffentliche Sicherheit, Produktion und Logistik, Medien und Kreativwirtschaft, Digital Services, Wirtschafts- und Finanzinformatik, Medizin und Gesundheit sowie Energie und Nachhaltigkeit. InnoVisions – Das Zukunftsmagazin des Fraunhofer-Verbundes IUK-Technologie informiert 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

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 über 25 Jahren auf Weltklasseniveau durch Publikationen und Software und durch seine jetzigen und ehemaligen Forscher, die Führungsrollen in Wissenschaft und Industrie übernommen haben.
www.mpi-inf.mpg.de

Unter der Schirmherrschaft des Bundespräsidenten

Von der Kultusministerkonferenz empfohlener Schülerwettbewerb

GEFÖRDERT VOM
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 35. Bundeswettbewerbs Informatik viel Erfolg!

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

BwInf.Informatik.erleben

twitter.com/_BWINF

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 Jüngere werden zwei leichtere Aufgaben gestellt, die Junioraufgaben.
Achtung: Junioraufgaben dürfen nur von Schülerinnen und Schülern bis einschließlich Jahrgangsstufe 10 (im G8: Einführungsphase) bearbeitet werden. Stammt eine Einsendung von Personen, die alle die Bedingung für Junioraufgaben erfüllen, nehmen die 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, zweite und dritte Preise unterschieden; mit einem ersten oder zweiten Preis ist die Qualifikation für die zweite Runde verbunden. Auch 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 der Verlage O'Reilly oder dpunkt.verlag. Erste Preisträger werden zur dritten Runde eingeladen, die im September 2017 am Hasso-Plattner-Institut in Potsdam ausgerichtet wird.

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

Informatik-Olympiade

Ausgewählte Teilnehmerinnen und Teilnehmer können sich in mehreren Lehrgängen für das vierköpfige deutsche Team qualifizieren, das an der Internationalen Informatik-Olympiade 2018 in Japan teilnimmt.

Informatik-Workshops etc.

Informatik-Workshops exklusiv für TeilnehmerInnen werden in Baden-Württemberg mit Unterstützung der Universität Stuttgart, 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. Die Firma Google lädt ausgewählte Teilnehmerinnen zum „Girls@Google Day" ein.

Ausgewählte Endrundenteilnehmer werden im September 2017 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 substanzielle 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. **Wichtig:** 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 2016/2017" 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 35. Bundeswettbewerb Informatik mitzumachen. Deshalb werden Schulen, die unter ihren BwInf-Erstteilnehmern mindestens fünf ehemalige Biber-Teilnehmer aus den Jahren 2015 und früher 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) und Lehrkräfte (bwinf.de/bgb/lehrer).

Die Regeln

Teilnahmeberechtigt

… sind Jugendliche, die nach dem 28.11.1994 geboren wurden. Sie dürfen jedoch zum 1.9.2016 noch nicht ihre (informatikbezogene) Ausbildung abgeschlossen oder eine Berufstätigkeit begonnen haben. Personen, die im Wintersemester 2016/17 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 30.11.2015 ihren Wohnsitz in Deutschland haben oder eine staatlich anerkannte deutsche Schule im Ausland besuchen.

Junioraufgaben dürfen von Teilnahmeberechtigten bis einschließlich Jahrgangsstufe 10 (im G8: Einführungsphase) bearbeitet werden. Ein Team darf Junioraufgaben bearbeiten, wenn mindestens ein Mitglied des Teams die genannten Bedingungen erfüllt.

Weiterkommen

Die zweite Runde erreichen alle, die eigenständig 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 leider keine zweite Runde.

Einsendungen

… enthalten Bearbeitungen zu mindestens einer Aufgabe und werden von Einzelpersonen oder 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
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: 28.11.2016

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: Flaschenzug

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

Familie Soda beginnt einen 14-tägigen Abenteuerurlaub. Ziel ist die trinkwasserlose, unbewohnte Insel Drøgø, deren Küste ringsherum sehr steil ist. Frühere Urlauber haben bereits einen Flaschenzug installiert, mit dem die vielen benötigten Getränkeflaschen nach oben gezogen werden können. Zum Glück stehen auch viele Behälter mit genügend Platz für alle Flaschen zur Verfügung, damit mehrere Flaschen auf einmal transportiert werden können.

Während die Eltern die mitgebrachten Flaschen auf Behälter verteilen, überlegen ihre Kinder Cora und Linus, wie viele Möglichkeiten es wohl insgesamt gibt, die Flaschen auf die Behälter zu verteilen.

Bei sieben Flaschen und zwei Behältern, von denen in den einen drei und in den anderen fünf Flaschen passen, gibt es genau zwei Möglichkeiten: Der kleinere Behälter ist entweder ganz voll oder enthält genau zwei Flaschen. Auf drei Behälter mit Platz für genau zwei, drei und vier Flaschen lassen sich die sieben Flaschen auf genau sechs Arten verteilen.

Aufgabe

Schreibe ein Programm, das eine Anzahl N von Flaschen, eine Anzahl k von Behältern und die k Fassungsvermögen der Behälter einliest und berechnet, auf wie viele Arten die Flaschen verteilt werden können. Die Flaschen sind nicht unterscheidbar, aber die Behälter sind es, auch wenn sie gleich groß sind.

Wende dein Programm mindestens auf die Beispiele an, die du unter bundeswettbewerb-informatik.de findest, und dokumentiere jeweils das Ergebnis.

Lösungsidee

Für bestimmte Eingabewerte – Anzahl der Flaschen, Anzahl der Behälter und Fassungsvermögen der Behälter – soll eine Zahl berechnet werden: die Anzahl der verschiedenen Möglichkeiten, die Flaschen auf die Behälter zu verteilen. Diese Zahl nennen wir *Verteilungszahl*.

Das Ganze erinnert sehr an eine Funktion in der Mathematik. Für einen Wert x berechnet eine Funktion f einen Wert y: $y = f(x)$. Die Verteilungszahl wird durch eine Funktion (genannt v) berechnet, nur mit mehreren Ausgangswerten, nämlich N (Anzahl der Flaschen), k (Anzahl der Behälter) und F (Liste mit den k Fassungsvermögen): $v(N, k, F)$

Alles aufzählen: Brute-Force

Bei der Berechnung der Funktion $v(N, k, F)$ gibt es zwei einfache Fälle:

- Wenn keine Flaschen vorhanden sind ($N = 0$), gibt es genau eine Möglichkeit, die Flaschen zu verteilen: alle Behälter bleiben leer. Es gilt also $v(0, k, F) = 1$, unabhängig von k und F.
- Wenn keine Behälter vorhanden sind ($k = 0$), gibt es keine Möglichkeit, die Flaschen zu verteilen. Es gilt also $v(N, 0, F) = 0$, unabhängig von N und F.

Was aber, wenn es Flaschen und Behälter gibt? Dann müssen systematisch alle Möglichkeiten zur Flaschenverteilung aufgezählt werden. Wir fangen mit dem ersten Behälter an und legen z. B. eine Flasche hinein. Die Anzahl der Verteilungsmöglichkeiten, bei der im ersten Behälter eine Flasche liegt, ergibt sich dann aus der Anzahl der Möglichkeiten, die anderen N–1

Flaschen auf die restlichen k–1 Behälter zu verteilen, also: $v(N–1, k–1, F[2..k])$. Dabei bedeutet F[2..k] die Liste der Fassungsvermögen ab Behälter 2. Alle Verteilungsmöglichkeiten insgesamt erhalten wir dann, wenn wir alle Möglichkeiten für den ersten Behälter durchgehen – von 0 bis zum Fassungsvermögen F[1] des ersten Behälters – und die jeweiligen Ergebnisse aufsummieren:

$$v(N, k, F) = v(N–0, k–1, F[2..k]) + v(N–1, k–1, F[2..k]) + \ldots + v(N–F[1], k–1, F[2..k])$$

Die Ausdrücke auf der rechten Seite können wir rekursiv auf die gleiche Weise berechnen, usw. Insgesamt werden so alle Möglichkeiten aufgezählt, nach dem Prinzip Brute-Force. Dabei wird das Problem rekursiv gelöst, indem auf gleiche Weise Lösungen für immer weiter „verkleinerte“ Teilprobleme gesucht werden, solange bis für einfache Fälle Lösungen direkt angegeben werden können.

Zwischenergebnisse merken

Wenden wir die beschriebene Methode einmal auf das Beispiel aus der Aufgabenstellung an:

$$\begin{aligned}
v(7,2,[3,5]) &= v(7,1,[5]) + v(6,1,[5]) + v(5,1,[5]) + v(4,1,[5]) \\
&= v(7,0,[]) + v(6,0,[]) + \ldots + v(2,0,[]) \\
&\quad + v(6,0,[]) + \ldots + v(1,0,[]) \\
&\quad + v(5,0,[]) + \ldots + v(0,0,[]) \\
&\quad + v(4,0,[]) + \ldots + v(–1,0,[]) \\
&= 0 + 0 + 0 + 0 + 0 + 0 \\
&\quad + 0 + 0 + 0 + 0 + 0 + 0 \\
&\quad + 0 + 0 + 0 + 0 + 0 + 1 \\
&\quad + 0 + 0 + 0 + 0 + 1 + 0 \\
&= 2
\end{aligned}$$

Wir erhalten das aus der Aufgabenstellung bekannte Ergebnis. Die Rechnung zeigt aber, dass sehr viele Teilprobleme mehrfach berechnet werden. Um das zu vermeiden, können die Lösungen von Teilproblemen gespeichert werden. Nur wenn dann die Lösung für ein Teilproblem noch unbekannt ist, muss sie berechnet (und gespeichert) werden; ansonsten wird sie einfach aus dem Speicher geholt.

Dabei ist wichtig, dass die Fassungsvermögen für das Speichern von Teilproblem-Lösungen keine Rolle spielen. Auch wenn in der Rekursion nicht immer alle Behälter betrachtet werden, die Liste der Fassungsvermögen bleibt insgesamt gleich. Es genügt also ein Speicher S für alle möglichen Paarungen von N und k: $S(N, k)$

Umsetzung

Die Lösungsidee wird in Python implementiert. Als **Speicher** für Teillösungen wird ein Array verwendet. Weil Python keine „richtigen“ Arrays mit mehreren Dimensionen kennt, wird das Bibliotheksmodul **numpy** verwendet, das Array-Funktionen realisiert.

Die Funktion v wird als Python-Funktion **verteilungszahl** implementiert. Sie prüft zunächst die einfachen Fälle ab und ruft sich für alle anderen Teilprobleme rekursiv auf – es sei denn, das Ergebnis für das Teilproblem ist schon gespeichert.

Außerdem muss das Einlesen der Eingabedaten aus einer Datei realisiert werden. Mit Hilfe des Moduls **sys** wird schließlich ermöglicht, dass das Programm mit der einzulesenden Datei als Parameter auf der Kommandozeile aufgerufen werden kann.

Tests zeigen, dass die Ergebnisse und damit auch die gespeicherten Teilergebnisse sehr große Zahlen sein können. Python kann mit solchen großen Zahlen ohne Weiteres umgehen. In anderen Programmiersprachen müssen dafür passende Bibliotheken verwendet oder das Rechnen mit großen Zahlen (nur die Addition wird benötigt) selbst implementiert werden.

Beispiele

Wir rufen das Python-Programm mit den verschiedenen BWINF-Eingabedateien auf. Diese Dateien liegen im gleichen Ordner wie die Programmdatei. Alle Ergebnisse werden innerhalb weniger Sekunden berechnet. Ohne den Teillösungs-Speicher wäre das nicht möglich.

```
$ ./flaschenzug.py flaschenzug0.txt
2
$ ./flaschenzug.py flaschenzug1.txt
13
$ ./flaschenzug.py flaschenzug2.txt
48
$ ./flaschenzug.py flaschenzug3.txt
6209623185136
$ ./flaschenzug.py flaschenzug4.txt
743587168174197919278525
$ ./flaschenzug.py flaschenzug5.txt
42376183321681306437343953352208634086628
```

Quelltext

```
#!/usr/bin/env python3
import sys # Für Übergabe von Argumenten in der Kommandozeile.

# Einlesen der Eingabe: Eingabedatei öffnen …
with open(sys.argv[1]) as f:
    # … und mit next(f) je eine Zeile als String einlesen.
    N = int(next(f))
    k = int(next(f))
    F = []
    for i in next(f).split(): # split() macht aus Zeile 3 eine Liste.
        F.append(int(i))

# Für den Ergebnisspeicher nehmen wir ein Array aus der Bibliothek numpy.
# Es soll beliebige Werte (Typ "object") enthalten dürfen,
# also auch beliebig große Zahlen.
import numpy
Speicher = numpy.empty([N+1, k+1], dtype = numpy.object) # Leeres Array erzeugen …
Speicher.fill(-1) # … und auf allen Positionen mit -1 initialisieren.

def verteilungszahl(AnzFlaschen, AnzBehaelter, FassungsListe):
    # Ohne Flaschen gibt es eine Möglichkeit: Alle Behälter bleiben leer.
    if AnzFlaschen == 0: return(1)
    # Zur Sicherheit: Weniger als 0 Flaschen geht gar nicht.
    if AnzFlaschen < 0: return(0)
    # Ohne Behälter gibt es keine Möglichkeit.
    if AnzBehaelter == 0: return(0)
    # Steht für die gegebenen Zahlen schon ein Wert im Speicher, nehmen wir den.
    if Speicher[AnzFlaschen, AnzBehaelter] != -1:
        return(Speicher[AnzFlaschen, AnzBehaelter])

    # Das waren die einfachen Fälle; jetzt muss rekursiv gerechnet werden:
    Speicher[AnzFlaschen, AnzBehaelter] = 0
    # Wir legen alle möglichen Anzahlen von Flaschen in den ersten Behälter …
    for AnzFlaschenB1 in range(FassungsListe[0]+1):
        if (AnzFlaschen - AnzFlaschenB1) < 0 : # (alle Flaschen verbraucht)
            break
        # … und addieren dann jeweils die Verteilungszahl
        # für die restlichen Flaschen und Behälter.
        Speicher[AnzFlaschen, AnzBehaelter] = \
```

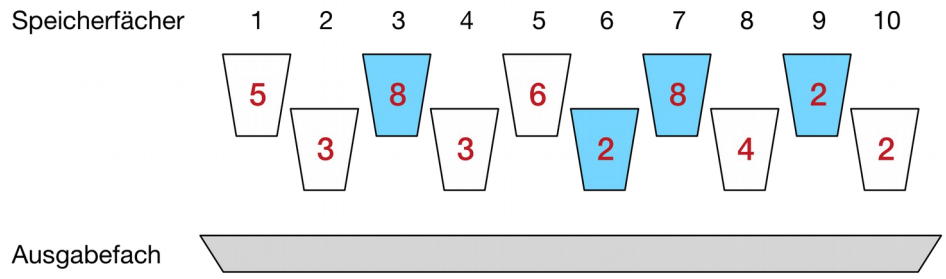
```
            Speicher[AnzFlaschen, AnzBehaelter] + \
                verteilungszahl(AnzFlaschen - AnzFlaschenB1,
                                AnzBehaelter - 1,
                                FassungsListe[1:])
    return(Speicher[AnzFlaschen, AnzBehaelter])

print(verteilungszahl(N, k, F))
```

Luftballons

Wie kommen in jede Packung immer gleich viele Luftballons? Das ist gar nicht so einfach! Gegenstand dieser Aufgabe ist eine einfache Luftballonverpackungsmaschine[1] (LVM). Sie hat 10 *Speicherfächer* sowie ein *Ausgabefach* und unterstützt folgende Operationen:

- FACH(*i*): Entleert das *i*-te Speicherfach in das Ausgabefach und füllt das Speicherfach mit einer wechselnden Anzahl von Luftballons neu.
- VERPACKEN(): Entleert das Ausgabefach und verpackt die darin enthaltenen Luftballons.

Jede Packung soll möglichst 20 Luftballons enthalten. Weniger Ballons würden zu Reklamationen führen; mehr Ballons sind kostenträchtig, aber notfalls akzeptabel.



Oben siehst du die LVM mit gefüllten Speicherfächern. Durch die Operationen

FACH(6), FACH(7), FACH(9), FACH(3), VERPACKEN()

werden genau 2 + 8 + 2 + 8 = 20 Luftballons verpackt.

Junioraufgabe 1

Schreibe ein Programm, das eine LVM simuliert und so steuert, dass jede Packung mindestens 20 und möglichst genau 20 Luftballons enthält. Gehe davon aus, dass zu Beginn alle Fächer leer sind. Simuliere die Operation FACH(*i*) mit Hilfe einer Füllfolge, das ist eine Folge von Zahlen: Die nächste noch nicht benutzte Zahl in dieser Folge gibt an, mit wie vielen Ballons das *i*-te Fach nach dem Entleeren neu gefüllt wird.

Unter bundeswettbewerb-informatik.de findest du einige Beispiel-Füllfolgen. Dein Programm soll für jedes Beispiel die durchgeführten Operationen, die Zahl der produzierten Packungen sowie die Zahl der insgesamt verbrauchten Ballons übersichtlich ausgeben.

1 In der Sendung mit der Maus kann man sehen, wie eine wirkliche Luftballonverpackungsmaschine (LVM) funktioniert: http://bit.ly/29PWZhx

LAMA

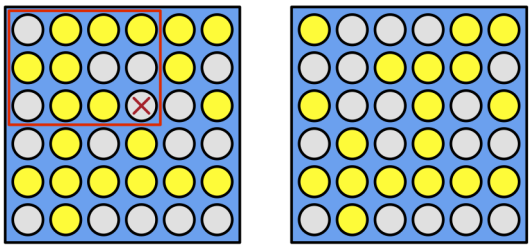
Ein neues „Gadget“ ist auf dem Markt: das LAMA (LED Advanced Matrix Arrangement). Es handelt sich um ein Schaltbrett, auf dem 25 mit je einer LED versehene Tasten in einem Raster mit 5 Reihen und 5 Spalten angeordnet sind. Die LED einer Taste kann entweder an- oder ausgeschaltet sein. Wenn eine Taste gedrückt wird, führt das LAMA eine mit der Taste verbundene Aktion aus. Die Aktionen sind programmierbar, so dass man mit dem LAMA verschiedene Spiele umsetzen kann.

Xaver wünscht sich so ein LAMA, um „Lights Out“[2] darauf programmieren und spielen zu können. Bei diesem Spiel sind die LEDs einiger Tasten zu Beginn eingeschaltet, und es geht darum, alle LEDs auszuschalten. Dazu darf man in jedem Spielzug eine Taste drücken. Solch ein Spielzug schaltet die LED der gedrückten Taste und ihre (bis zu vier) Nachbar-LEDs um. Sind alle LEDs ausgeschaltet, ist das Spiel beendet.

Leider ist es noch eine Weile bis Weihnachten. In der Zwischenzeit programmiert Xavers Schwester Yanelle mit ihm Lights Out „ganz normal“ am Computer.

Junioraufgabe 2

1. Schreibe ein Programm, mit dem man Lights Out wie auf dem LAMA spielen kann. Beim Start des Spiels muss zuerst festgelegt werden, welche LEDs zu Beginn eingeschaltet sind. Dann soll gespielt werden können; der Spieler soll also seine Züge eingeben können. Zu Beginn und nach jedem Spielzug soll der Zustand der LEDs angezeigt werden.
2. Die Hersteller denken darüber nach, LAMAs auch in anderen Größen zu produzieren. Erweitere dein Programm so, dass man auch Tasten-Raster mit anderen Anzahlen von Reihen und Spalten (bis zu einer von dir bestimmten Maximalzahl) bespielen kann.
3. Erweitere dein Programm so, dass auch die folgende Variante von Lights Out gespielt werden kann: Hier schaltet ein Spielzug die LED der gedrückten Taste und alle LEDs in dem links und oberhalb der Taste gelegenen Tasten-Rechteck um. Die Bilder zeigen ein Beispiel für ein LAMA mit 6 Reihen und 6 Spalten. Auch bei dieser Variante ist das Spiel beendet, wenn alle LEDs ausgeschaltet sind.



2 http://en.wikipedia.org/wiki/Lights_Out_(game)

Sprichwort

Zur Weihnachtszeit äußert Christian immer sehr kostspielige Wünsche. Die Standardantwort seiner Mutter ist dann die Redensart:

Das bekommst du, wenn Weihnachten und Ostern auf einen Tag fallen.

Ostern liegt immer in den Monaten März und April, aus Sicht der Mutter also in sicherem Abstand von Weihnachten. Sein Freund Igor erzählte Christian aber vor Kurzem, dass die orthodoxe Kirche ihr Weihnachtsfest erst am 7. Januar feiert. Der Grund ist, dass diese Kirche einem anderen Kalender folgt. Christian fragt sich nun, ob nicht wenigstens das orthodoxe Weihnachtsfest und das katholische und protestantische Osterfest (Ostersonntag) am gleichen Tag stattfinden können.

Aufgabe 1

Informiere dich über die Kalendersysteme (Stichworte: der gregorianische und der julianische Kalender) und erkläre, warum Christians Hoffnung berechtigt scheint.

Rechne mit einem Computerprogramm aus, wann das orthodoxe Weihnachtsfest und das katholische und protestantische Osterfest zum ersten Mal am gleichen Tag stattfinden.

Nun ging es Christian eigentlich um Weihnachtsgeschenke. Also interessiert ihn, wann das katholische und protestantische Weihnachtsfest (1. Weihnachtstag) mit dem orthodoxen Osterfest zusammenfällt. Wann wird das zum ersten Mal geschehen?

Rhinozelfant

Im Urwald von Informatien ist ein Rhinozelfant gefunden worden. Die Forscher sind begeistert: Zum ersten Mal seit Jahrhunderten wurde ein bislang unbekanntes Großtier entdeckt. Es stellt sich die Frage, warum diese Tiere solange verborgen geblieben sind. Offenbar verfügen sie über einen besonders guten Tarnmechanismus.

Nachdem die ersten Rhinozelfanten in einen Zoo gebracht wurden, stellten die Tierpfleger fest, dass diese – ähnlich wie Chamäleons – die Farbe ihrer Haut an beliebige Umgebungen anpassen können. Wenn ein Rhinozelfant merkt, dass es beobachtet wird, nimmt jede seiner Hautschuppen die Umgebungsfarbe der dem Betrachter gegenüberliegenden Körperseite an. Dadurch wird es für den Beobachter quasi „durchsichtig“.

Mit modernen Digitalkameras ist es allerdings möglich, Rhinozelfanten zu erkennen: Wenn die Auflösung nur hoch genug ist, wird jede Hautschuppe durch mehrere nebeneinander liegende Pixel dargestellt. Weil alle diese Pixel dieselbe Schuppe abbilden, haben sie genau die gleiche Farbe. Dadurch kann man feststellen, welche Pixel in einem Bild möglicherweise einen Rhinozelfanten darstellen.

Es sollen nun mehrere Fotos aus dem Rhinozelfantenwald geprüft werden, ob darauf vielleicht ein Rhinozelfant abgebildet ist.

Aufgabe 2

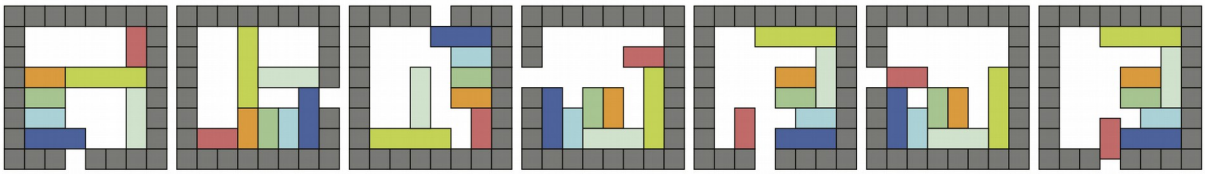
Erstelle ein Programm, das diejenigen Pixel eines Bildes weiß färbt, die zu einem Rhinozelfanten gehören könnten.

Unter bundeswettbewerb-informatik.de sind einige Bilder zur Verfügung gestellt. Auf welchen ist ein Rhinozelfant abgebildet?

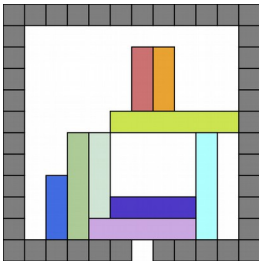
Rotation

Ein mechanisches Puzzle besteht aus einem quadratischen Rahmen, der Stäbchen verschiedener Farben enthält und an einer Stelle offen ist. Der Rahmen kann jeweils um 90° nach links oder rechts gedreht werden. Durch die Schwerkraft gleiten dann die Stäbchen nach unten, ohne ihre Ausrichtung zu verändern, bis sie auf ein Hindernis stoßen.

Ziel ist es, irgendein Stäbchen mit möglichst wenig Drehungen aus dem Rahmen herauszubekommen. Im folgenden Beispiel siehst du, wie die Stäbchen in einer Puzzleaufgabe auf sechs aufeinanderfolgende Drehungen reagieren. Am Ende gleitet ein Stäbchen aus dem Rahmen.



Auch diese Puzzleaufgabe hat eine Lösung:



Es kann aber auch vorkommen, dass es keine Möglichkeit gibt, irgendein Stäbchen herauszubekommen, da immer etwas im Weg ist.

Aufgabe 3

Eine Puzzleaufgabe ist durch die Anzahl und anfängliche Lage der Stäbchen im Rahmen, die Größe des Rahmens und die Lage der offenen Stelle im Rahmen gegeben. Schreibe ein Programm, das eine solche Aufgabe aus einer Datei einlesen kann und dann entweder löst und dazu eine kürzestmögliche Folge von Drehungen ausgibt oder meldet, dass es keine Lösungsmöglichkeit gibt.

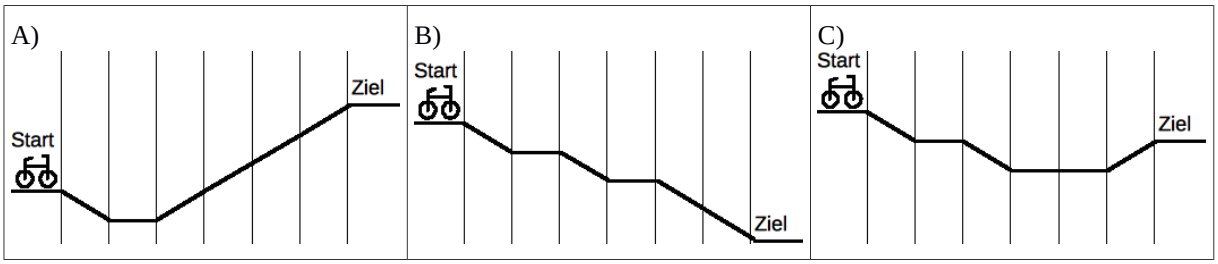
Unter bundeswettbewerb-informatik.de findest du einige Beispieleingaben und eine Beschreibung des Eingabeformats. Wende dein Programm mindestens auf diese Beispiele an und dokumentiere jeweils das Ergebnis.

Radfahrspaß

Bei den Bundesweiten Indoor-Fahrspaßwettbewerben (BwInF) kommt es darauf an, einen mit raketengetriebenen Spaßrädern zu befahrenden Parcours so genau einzuschätzen, dass man bis zum Ende kommt und dort das Rad sicher zum Stillstand bringt. Ein Parcours ist also regelkonform befahrbar, wenn es möglich ist, mit der Geschwindigkeit 0 m/s zu starten und am Ziel wieder mit der Geschwindigkeit 0 m/s zu enden; dabei ist Zurückfahren verboten.

Ein Parcours ist eine gerade Strecke mit Abschnitten, die entweder steil bergauf, steil bergab oder flach verlaufen. Auf Steilstücken kann die Geschwindigkeit des Spaßrads nicht beeinflusst werden: Pro Bergab-Abschnitt beschleunigt es um 1 m/s, pro Bergauf-Abschnitt wird es um 1 m/s verlangsamt. Auf den flachen Abschnitten muss man das Rad bewusst um exakt 1 m/s pro Abschnitt entweder beschleunigen (+) oder verlangsamen (–).

Ein Parcours wird vor einem Rennen neu aufgebaut und ist den Fahrern unbekannt. Die Fahrer dürfen zunächst einen Probelauf zu Fuß unternehmen, sich dabei aber keine Notizen machen. Danach beginnt das sogenannte Icandothat-Poker, bei dem jeder Fahrer sagt, ob er es schafft, den Parcours regelkonform zu durchfahren. Wer diese Frage mit „ja“ beantwortet, muss dann sein Können beweisen und gewinnt, wenn er/sie es auch wirklich schafft. Schafft es keiner der Ja-Sager, gewinnen alle, die mit „nein“ geantwortet haben.



Abbildungen A und B zeigen Parcours, die nicht regelkonform befahrbar sind: In A bleibt das Rad in jedem Fall mittendrin stehen, in B hat das Rad am Ziel noch mindestens 2m/s Geschwindigkeit. C kann bewältigt werden, wenn man auf den flachen Abschnitten + – – fährt.

Aufgabe 4

1. Entwickle ein möglichst einfaches Verfahren, das die BwInF-Fahrer während des Probelaufs anwenden können, um am Ende zu entscheiden, ob der Parcours regelkonform befahrbar ist. Simuliere das Verfahren mit einem Computer und verwende möglichst wenig Speicherplatz, weil auch die Fahrer sich nur wenig im Kopf merken können. Beachte, dass der Parcours beim Probelauf nur einmal durchschritten wird und sehr lang sein kann. Dein Programm darf die Eingabe also nur einmal lesen und sie dabei nicht speichern.

   Unter bundeswettbewerb-informatik.de sind verschiedene Beispiel-Parcours durch eine Folge von \ (bergab), / (bergauf) und _ (flach) vorgegeben. Die obigen Beispiele sehen in dieser Notation so aus: A) _////, B) __\\ und C) ___/.

2. Erweitere dein Programm so, dass es während eines zweiten Einlesens eines regelkonform befahrbaren Parcours für alle flachen Abschnitte je eine Anweisung „+“ (beschleunigen) oder „–“ (verlangsamen) für die Fahrer ausgibt. Wer diese Anweisungen befolgt, muss mit dem Rad am Ziel genau still stehen.

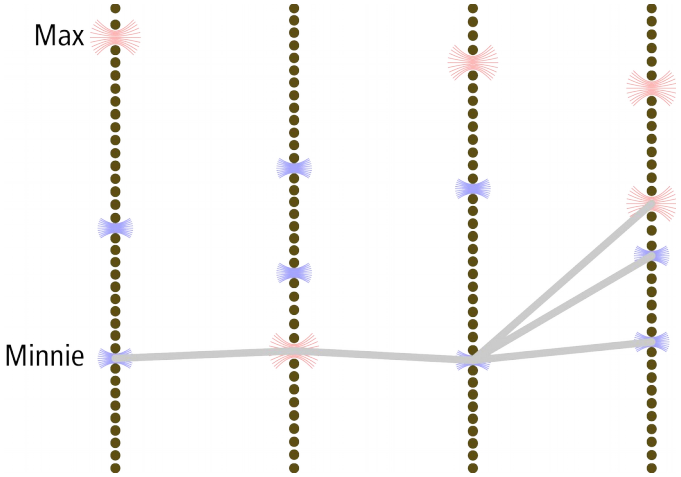
Buhnenrennen

An Nordseestränden gibt es Buhnen: Reihen von in den Sandstrand gerammten Baumstämmen, die senkrecht zur Wasserlinie stehen.

Bei Ebbe sind fast alle Baumstämme trockenen Fußes erreichbar. Durch die Lücken, die es gelegentlich zwischen den Baumstämmen gibt, kann man dann in Zickzacklinie die Buhnen passieren. Es wurde schon beobachtet, wie Hunde die unterschiedlich großen Lücken geschickt benutzen: Ein kleiner Hund ärgerte einen größeren und entwischte ihm dann durch eine kleine Lücke.

Wir betrachten einen kleinen Hund (Minnie) und einen großen (Max), der Minnie fangen will. Minnie schafft 20 km/h. Max ist schneller – 30 km/h –, aber Minnie hat den Vorteil, dass sie durch mehr Lücken laufen kann. Eine Lücke nennen wir *Maxilücke*, wenn Max (und damit auch Minnie) hindurch passt, und *Minilücke*, wenn nur Minnie hindurch passt. Die Buhnen sind jeweils 70 m voneinander entfernt.

Hier ist ein Beispiel. Die Hunde (bzw. ihre Namen) stehen an ihren Startpositionen. Maxilücken und Minilücken sind unterschiedlich groß eingezeichnet (und farblich markiert, nämlich rot bzw. blau).
Minnie kann auf drei Weisen die letzte Buhne mit Vorsprung hinter sich lassen und dann dort ihrer Besitzerin in die Arme springen. Allerdings kann Max ihr an der dritten Buhne gefährlich nahe kommen.



Aufgabe 5

Schreibe ein Programm, das Minnie hilft, einen Fluchtplan zu erstellen. Du erhältst hierzu einen Plan der Buhnen mit den Positionen aller Maxi- und Minilücken. Anfangs befinden sich Max und Minnie an Lücken in der ersten Buhne. Dann versucht Minnie alle Buhnen zu durchqueren, indem sie von Lücke zu Lücke läuft. Ein solcher Minnieweg ist sicher, wenn Max sie nicht fangen kann, egal wie er läuft.

Dein Programm soll feststellen, ob ein sicherer Minnieweg existiert, und einen solchen gegebenenfalls ausgeben. Wende dein Programm mindestens auf die Beispiele an, die du unter bundeswettbewerb-informatik.de findest, und dokumentiere jeweils das Ergebnis.

Gehe der Einfachheit halber von punktförmigen Lücken und Hunden aus. Diese Vereinfachung verfälscht das Ergebnis nicht wesentlich, wenn wir davon ausgehen, dass die Hunde nur knapp durch die Lücken passen.

Teilnehmen

Einsendeschluss ist der 28. November 2016.

Anmelden und Einsenden

online unter
pms.bwinf.de

Fragen zu den Aufgaben?

per Telefon:
0228 378646
zu üblichen Bürozeiten

per E-Mail:
kontakt@bundeswettbewerb-informatik.de

Diskutiere über die Aufgaben mit anderen Teilnehmern in der **EI Community**:
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.

Ein Team gibt gemeinsam eine Einsendung ab.

Tipps

Unter 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 vom dpunkt.verlag bzw. 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, ebenso die Mitglieder des deutschen IOI-Teams.

bundeswettbewerb-informatik.de/chancen

Biber goes BwInf

Hast du im Jahr 2015 oder früher 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.

bwinf.de/bgb/schueler