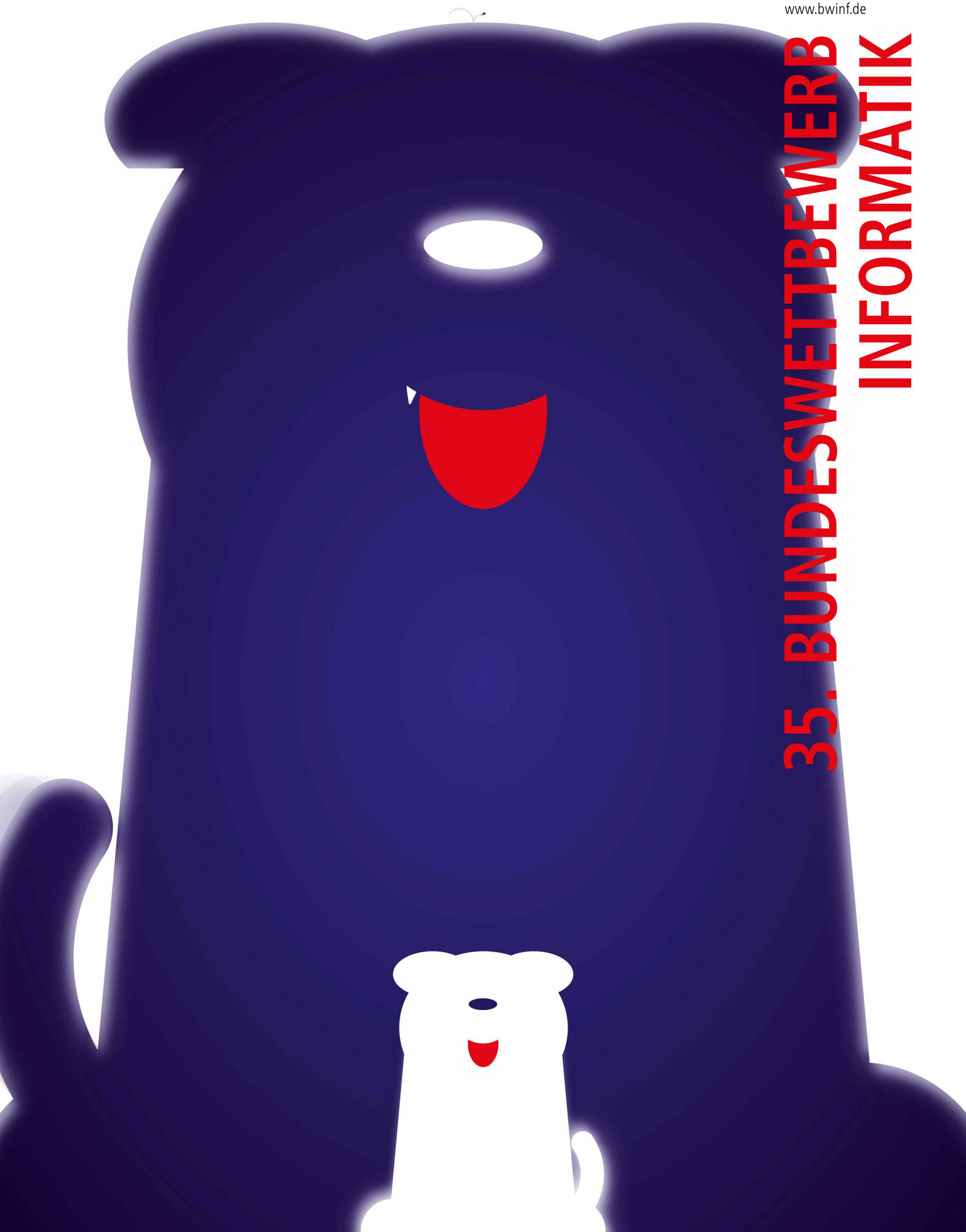
BUNDES WEIT INFORMATIK NACHWUCHS FÖRDERN

www.bwinf.de

35. BUNDESWETTBEWERB INFORMATIK



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 Industrie-kunden 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

BUNDESWEIT INFORMATIK NACHWUCHS FÖRDERN

Bundeswettbewerb Informatik

Unter der Schirmherrschaft des Bundespräsidenten

Von der Kultusministerkonferenz empfohlener Schülerwettbewerb

Gesellschaft für Informatik

Fraunhofer IUK-TECHNOLOGIE

max planck institut informatik

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.

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

max planck institut informatik

INFORMATIK RWTH AACHEN

technische universität dortmund

Technische Universität Braunschweig

INFORM

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

QAWARE SOFTWARE ENGINEERING

Google

dpunkt.verlag

O'REILLY

Die BWINF-Partner wünschen allen Teilnehmerinnen und Teilnehmern des 35. 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 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.

Biber goes BwInf

Deshalb werden Schulen, die unter ihren Teilnehmerinnen und Teilnehmer am Informatik-Biber sollen dazu angeregt werden, beim 35. Bundeswettbewerb Informatik mitzumachen. fünf ehemalige Biber-Teilnehmer aus den Jahren 2015 und früher nachweisen können, mit einem **„Biber-goes-BwInf"-Schulpreis** ausgezeichnet. Die für den Preis gewerteten Schülerinnen und Schüler erhalten als Anerkennung einen USB-Stick. Mehr dazu online: für Schüler (bwinf.de/bgb/schueler) und Lehrkräfte (bwinf.de/bgb/lehrer)

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.

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 28.11.2016 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
42376183321681306437343953352208634086 28
```

Quelltext

```
#!/usr/bin/env python3

# Für Übergabe von Argumenten in der Kommandozeile:
import sys

# 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
# Leeres Array erzeugen …
Speicher = numpy.empty([N+1, k+1],
                       dtype = numpy.object)
# … und auf allen Positionen mit -1 initialisieren:
Speicher.fill(-1)

def verteilungszahl(AnzFlaschen, AnzBehaelter, FassungsListe):
    # Ohne Flaschen gibt es eine Möglichkeit:
    # Alle Behälter bleiben leer.
    if AnzFlaschen == 0: return(1)
    # 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))
```