

Krocket

Laura findet beim Spielen im Garten einige aufgestellte Tore eines Krocketspiels. Ihr Vater ruft aus der Küche, dass er es mit nur zwei Schlägen geschafft hat, den Ball durch alle Tore in der richtigen Reihenfolge zu stoßen. Das möchte Laura verbessern, wobei sie sich allerdings erlaubt, von einem selbst gewählten Punkt aus abzuschlagen. Aber geht das überhaupt?

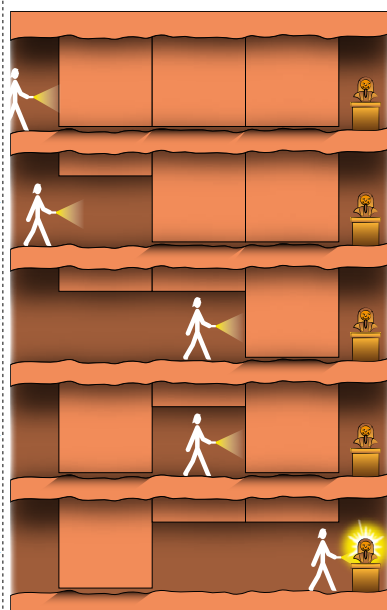
Aufgabe 4

Hilf Laura, indem du ein Programm schreibst, das überprüft, ob alle Tore in der richtigen Reihenfolge mit nur einem Schlag durchquert werden können. Falls ja, soll dein Programm ein entsprechendes Paar aus Startpunkt und Schlagrichtung ausgeben.

Der Ball ist rund und hat einen in der Eingabe angegebenen Radius r . Die Tore sehen von oben wie verschieden lange Geradensegmente aus und sind in der vorgeschriebenen Reihenfolge jeweils durch ihre zwei Endpunkte dargestellt.

Hinweis: Es ist hilfreich, den Radius des Balls in einem ersten Lösungsversuch zu vernachlässigen, das heißt $r = 0$ anzunehmen.

Wende dein Programm mindestens auf alle Beispiele an, die du auf den [BWINF-Webseiten](#) findest, und dokumentiere die Ergebnisse.



Das ägyptische Grabmal

Viele Computerspiele, Abenteuer- und Science Fiction-Filme enthalten gefährliche Fallen, durch welche die Heldin oder der Held durch geschicktes Hin- und Herlaufen hindurchkommen können.

Petra steht vor einem solchen Szenario: Ein Grabmal kann nur durch einen langen Gang erreicht werden. Dieser Gang besteht aus Abschnitten, die lückenlos aufeinander folgen. In jedem Abschnitt ist in der Decke ein Steinquader eingelassen, der ihn blockieren kann.

Am Anfang ist jeder Abschnitt durch seinen Quader blockiert.

Beim Erreichen des Ganges löst Petra einen Mechanismus aus, der die Quader bewegt. Petra beobachtet, dass sich jeder Quader periodisch bewegt mit einer ganzzahligen Anzahl von Minuten als Periodenlänge.

Ein Quader mit Periode 5 Minuten blockiert seinen Abschnitt zunächst 5 Minuten, um ihn dann für 5 Minuten freizugeben und nach 5 Minuten erneut zu blockieren.

Petra überlegt sich zunächst ein einfaches Beispiel: Drei Quader mit den Perioden 5, 8 und 12 Minuten durchlaufen fünf Konfigurationen, wie im Bild dargestellt.

Man kann den Gang passieren, indem man nach 8 Minuten in den zweiten Abschnitt läuft, dort 4 Minuten wartet und dann durch den dritten Abschnitt zum Grabmal läuft.

Petra kann sich nur in Abschnitten aufhalten, die nicht blockiert sind. Steht sie in einem Abschnitt, kann sie in einen benachbarten Abschnitt nur dann laufen, wenn beide Abschnitte gleichzeitig nicht blockiert sind.

Aufgabe 5

Schreibe ein Programm, das Instruktionen für ein schnellstmögliches Erreichen des Grabmals ausgibt. Für die Eingabe 5, 8, 12 könnten diese Instruktionen so aussehen: *Warte 8 Minuten, laufe in den Abschnitt 2, warte 4 Minuten, laufe zum Grabmal.*

Wende dein Programm mindestens auf alle Beispiele an, die du auf den [BWINF-Webseiten](#) findest, und dokumentiere die Ergebnisse.

Teilnehmen

Dieses Blatt enthält die Aufgaben der 1. Runde des 43. Bundeswettbewerbs Informatik. Die Junioraufgaben sind gleichzeitig die Aufgaben der 3. und letzten Runde des Jugendwettbewerbs Informatik 2024.

Einsendeschluss für beide Wettbewerbe: 18. November 2024.

Anmelden

online unter: login.bwinf.de

Sobald du dort registriert bist, kannst du dich dort auch zur Teilnahme anmelden: für Jugendwettbewerb (3. Runde), Bundeswettbewerb oder beides. Bei der Anmeldung zum Jugendwettbewerb musst du deine Kennung der Online-Runden (E-Mail-Adresse oder Logincode) angeben.

Bearbeiten

In der 3. Runde des Jugendwettbewerbs bearbeitest du einzeln oder im Team von bis zu vier Personen die beiden Junioraufgaben. Im Bundeswettbewerb sind die Junioraufgaben Schülerinnen und Schülern vor der Qualifikationsphase des Abiturs vorbehalten; wer in die 2. Runde kommen will, muss drei oder mehr Aufgaben bearbeiten, einzeln oder im Team beliebiger Größe.

Einsenden

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

- > deine **Lösungsidee** beschreiben;
- > die **Umsetzung** der Idee in ein Programm erläutern;
- > an genügend **Beispielen** zeigen, dass und wie deine Lösung funktioniert; und
- > die wichtigsten Teile des Quelltextes anfügen.

Achtung: eine gute Dokumentation muss nicht lang sein, aber unbedingt die **Beispiele** enthalten!

Der praktische Teil deiner Einsendung ist die **Implementierung** und umfasst den kompletten Quelltext und das ausführbare Programm (Windows, Linux, MacOS X oder Android).

Die **Einsendung** wird über login.bwinf.de als ZIP-Dateiarchiv abgegeben. Ein Team gibt gemeinsam nur eine Einsendung ab.

Weitere Informationen unter:



bwinf.de/bundeswettbewerb/teilnehmen

Doppelteilnahme: Teilnehmende am Jugendwettbewerb vor der Qualifikationsphase können ihre Bearbeitungen der Junioraufgaben auch zur 1. Runde des Bundeswettbewerbs einsenden, gemeinsam mit der Bearbeitung mindestens einer weiteren Aufgabe.

Fragen?

Wende dich an BWINF:

- > E-Mail: bundeswettbewerb@bwinf.de bzw. jugendwettbewerb@bwinf.de
- > Diskutiere mit den Mitgliedern der BWINF-Community auf Discord: <https://discord.com/invite/bwinfcommunity>

Tipps und Infos

Unter bwinf.de/bundeswettbewerb/tipps findest du

- > genauere Hinweise zur Einsendung;
- > Beispiele für Aufgabenbearbeitungen;
- > Tipps zu Informatik und Programmierung.



Entdecke die spannende Welt der Informatik auf Einstieg Informatik! Hier findest du alles, was du braucht – von Infos über Studiengänge bis hin zu nützlichen Tools. Außerdem gibt es Tipps und Hinweise zu den neuesten Events, von Hackathons bis zu Stipendienmöglichkeiten. Leg los auf einstieg-informatik.de!

Deine Chancen

Mit einer Teilnahme am Bundeswettbewerb Informatik kannst du nur gewinnen. In allen Runden gibt es **Urkunden** sowie kleine **Geschenke** für alle.

Bei erfolgreicher Teilnahme an der 1. Runde kannst du zu **Informatik-Workshops** eingeladen werden, die von vielen BWINF-Partnern wie dem Hasso-Plattner-Institut und der Frankfurt School of Finance & Management ausgerichtet werden. Google lädt Teilnehmerinnen zum **Girls@Google Day** ein und mit weiteren Partnern werden Camps im Rahmen von **girls@BWINF** veranstaltet.

Nach deiner Teilnahme an der 2. Runde winken die **Forschungstage Informatik** des Max-Planck-Instituts für Informatik und einige Buchpreise. Die Einsendung zur 2. Runde kann in einigen Bundesländern als besondere Lernleistung in die Abiturwertung eingebracht werden.

Die Besten der 2. Runde erreichen die **Endrunde**. Dort werden Bundessiege und weitere Preise vergeben; sie sind mit **Geldpreisen** dotiert. Bundesiegerinnen und -sieger werden in der Regel ohne weiteres Auswahlverfahren in die **Studienstiftung des deutschen Volkes** aufgenommen.

Siehe auch: bwinf.de/bundeswettbewerb/chancen



Tritt dem BWINF-Discord bei!

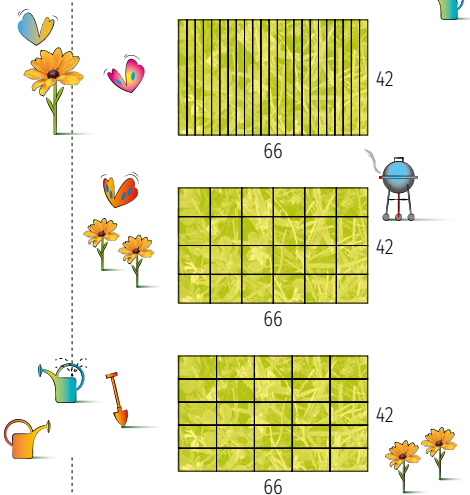
- discord.com/invite/bwinfcommunity
- [instagram.com/bwinf](https://www.instagram.com/bwinf)
- mastodon.social/@BWINF



Quadratisch, praktisch, grün

Herr Grün hat ein großes Grundstück, auf dem eine Kleingartenanlage eingerichtet werden soll. Dazu soll das Grundstück in mehrere Kleingärten gleicher Größe aufgeteilt werden. Am besten sind quadratische Kleingärten, denn sie können besonders vielfältig bepflanzt werden. Es haben sich schon einige Interessenten gemeldet.

Herr Grün fragt sich, wie er das Grundstück möglichst passend aufteilen kann. Wenn beispielsweise sein Grundstück 42 m x 66 m groß wäre und er 23 Interessenten hätte, könnte er folgende Aufteilungen machen.



Wie man sieht, ergibt die Aufteilung in 24 Kleingärten fast quadratische Kleingärten und gefällt Herrn Grün am besten.

Junioraufgabe 1

Hilf Herrn Grün und schreibe ein Programm, das die Grundstücksgröße und die Anzahl von Interessenten einliest und eine Aufteilung in Kleingärten vorschlägt. Es müssen mindestens so viele Kleingärten entstehen, wie es Interessenten gibt, aber höchstens 10 % mehr. Unter dieser Bedingung sollen die Kleingärten so quadratisch wie möglich sein.

Überlege dir zunächst, was „so quadratisch wie möglich“ bedeuten könnte und dokumentiere dies.

Wende dein Programm mindestens auf alle Beispiele an, die du auf den [BWINF-Webseiten](#) findest, und dokumentiere die Ergebnisse.

Texthopsen

Beispieltexte sind toll

Bela und Amira ist im Deutschunterricht oft langweilig. Daher haben sie sich ein neues Spiel ausgedacht: Texthopsen. Sie suchen sich einen Text im Schulbuch aus und hopsen darin um die Wette. Das geht so: Jeder Buchstabe hat eine Sprungweite zugeordnet (siehe Tabelle). Bela fängt beim ersten, Amira beim zweiten Buchstaben des Textes an. Von dieser Startposition springen sie jeweils so viele Positionen weiter, wie die Sprungweite des Buchstabens vorgibt. Alles was kein Buchstabe ist, überspringen sie dabei einfach. An der neuen Position lesen sie dann jeweils den nächsten Buchstaben mit einer neuen Sprungweite. Das Hopsen wiederholen sie abwechselnd so lange, bis sie aus dem Text herauspringen. Wer dies als Erstes schafft, hat gewonnen.

Buchstabe	a	b	c	...	z	ä	ö	ü	ß
Sprungweite	1	2	3	...	26	27	28	29	30

Junioraufgabe 2

Schreibe ein Programm, das für einen gegebenen Text bestimmt, ob Bela oder Amira beim Texthopsen gewinnt.

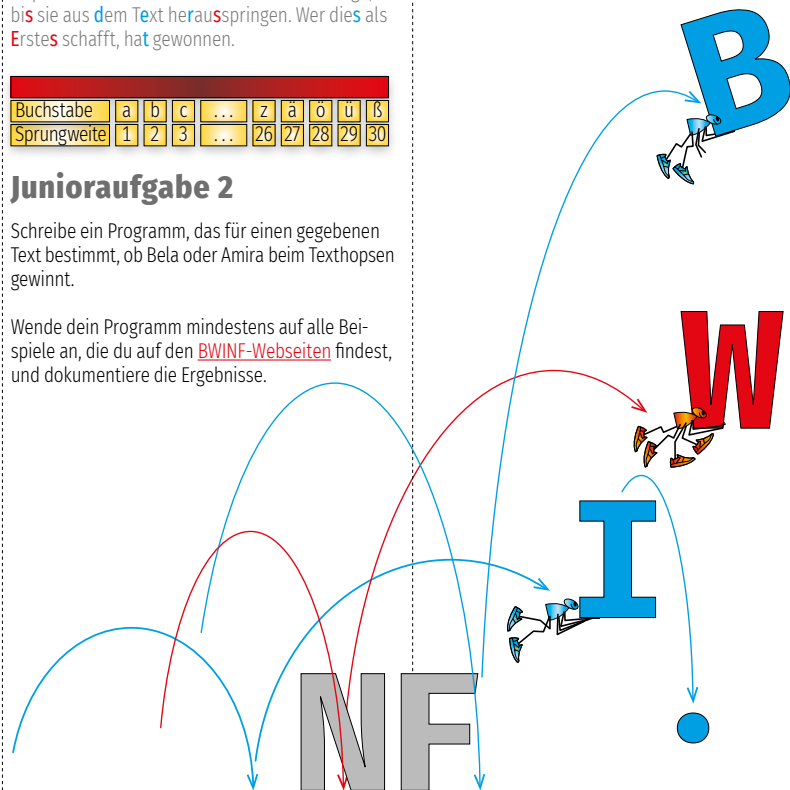
Wende dein Programm mindestens auf alle Beispiele an, die du auf den [BWINF-Webseiten](#) findest, und dokumentiere die Ergebnisse.

Hopsitexte

Zara hat das Texthopsen schon programmiert. Dabei ist ihr aufgefallen, dass die beiden Spieler erstaunlich häufig von derselben Stelle aus dem Text herauspringen. Sie versucht einen möglichst langen Text zu schreiben, der diese Eigenschaft nicht hat. Das heißt, die Sprungsequenzen, die am ersten und zweiten Buchstaben des Textes beginnen, sollen nicht zur selben Endposition führen. So einen Text nennt sie einen „Hopsitext“. Dieser Aufgabentext ist zum Beispiel ein solcher Hopsitext.

Aufgabe 1

Schreibe ein Programm, das Zara hilft, möglichst lange und sinnvolle deutsche Hopsitexte zu erstellen. Überlege dir, was das Programm bieten muss, damit Zara besonders gut damit arbeiten kann.



Schwierigkeiten

Dua L. Graph ist Lehrerin für Informatik. Im Laufe der Jahre hat sie eine Reihe von Klausuraufgaben ausgearbeitet. Damit hat sie sich so viel Mühe gemacht, dass sie ihre Aufgaben – mit zeitlichem Abstand und nur leicht verändert – immer wieder verwendet.

Dua leitet aus den Ergebnissen früherer Klausuren ab, welche Aufgaben darin leichter waren als andere. Zum Beispiel hat sie aus vier alten Klausuren die folgenden Schwierigkeitsabstufungen abgeleitet (< bedeutet ‚ist leichter als‘):

Klausur	Abstufung
1	$B < A < D < F$ (und damit auch $B < D$, $B < F$ und $A < F$)
2	$D < F < G$
3	$A < E < D < C$
4	$G < F < C$

Wenn Dua aus bisherigen Aufgaben eine neue Klausur zusammen stellt, möchte sie die Aufgaben nach Schwierigkeit anordnen, leichte Aufgaben zuerst. In der nächsten Klausur möchte Dua die Aufgaben B, C, D, E und F verwenden. Sie ordnet die Aufgaben so an: B; E; D; F; C. Geschafft! Ihre Anordnung ist eine gute Anordnung, denn sie steht zu keiner vorherigen Schwierigkeitsabstufung im Widerspruch.

Dua ist erleichtert. Als begeisterte Informatikerin will sie sich für die Zukunft das Leben einfacher machen und die Anordnung ihrer Aufgaben automatisieren. Sie stellt aber fest, dass das nicht immer leicht ist: Wenn sie zum Beispiel die Aufgaben F und G auswählt, ist laut Klausur 2 Aufgabe F einfacher als G, aber laut Klausur 4 Aufgabe G einfacher als F. Eine solche Situation nennt Dua einen Konflikt.

Aufgabe 2

Schreibe ein Programm, das Dua hilft. Es soll die Schwierigkeitsabstufungen einer Reihe alter Klausuren sowie eine Menge von Aufgaben aus diesen Klausuren einlesen und eine gute Anordnung dieser Aufgaben ausgeben.

1. Gehe zunächst davon aus, dass es in den alten Klausuren keine Konflikte gibt.
2. Überlege, wie dein Programm mit Konflikten umgehen könnte und erweitere es entsprechend.

Wende dein Programm mindestens auf alle Beispiele an, die du auf den [BWINF-Webseiten](#) findest, und dokumentiere die Ergebnisse.

Wandertag

Einmal im Jahr veranstaltet die Informatik-Gesundheitskasse (IGK) für ihre Mitglieder einen Wandertag. Leider hat sich in den letzten Jahren gezeigt, dass viele Mitglieder nicht teilnehmen wollten, weil die Strecke zu lang oder zu kurz war.

Deshalb hat die Krankenkasse jetzt erstmalig eine Befragung zu dem Wandertag durchgeführt. Jedes Mitglied hat dabei eine minimale und maximale Streckenlänge angegeben: Liegt die Länge der angebotenen Strecke in diesem Bereich, nimmt das Mitglied teil.

Es wird schnell offensichtlich, dass viel mehr Mitglieder teilnehmen, wenn es mehrere unterschiedlich lange Strecken zur Auswahl gibt.

Aufgabe 3

Unterstütze die IGK bei der Planung und schreibe ein Programm, das drei Streckenlängen so berechnet, dass möglichst viele Mitglieder teilnehmen. Es soll ausgeben, wie viele Mitglieder insgesamt teilnehmen würden und bei welcher Streckenlänge welche Mitglieder teilnehmen könnten.

Wende dein Programm mindestens auf alle Beispiele an, die du auf den [BWINF-Webseiten](#) findest, und dokumentiere die Ergebnisse.

