Bebras Australia Computational Thinking Challenge

2022 Solutions Guide
Round 1

Primary School
Grades 3–6

bebras.edu.au

Bebras Australia Computational Thinking Challenge

Bebras is an international initiative aiming to promote Computational Thinking skills among students.

Started in 2004 by Professor Valentina Dagiene from the University of Vilnius, ‘Bebras’ is Lithuanian for beaver. This refers to their collaborative nature and strong work ethic.

The International Bebras Committee meets annually to assess potential questions and share resources. Questions are submitted by member countries and undergo a vetting process.

The Bebras international community has now grown to 60 countries with over 2.9 million students participating worldwide!

Bebras Australia began in 2014 and is now administered through CSIRO Digital Careers.

In Australia, the Bebras Challenge takes place in March and August–September each year. As of 2020, two separate challenges are offered for each round.

To find out more and register for the next challenge, visit **bebras.edu.au**

Engaging young minds for Australia’s digital future

481
Australian schools participated in Round 1 2022

27,435
Australian students participated in Round 1 2022

2.9 million
Students participate worldwide

CSIRO Digital Careers supports teachers and encourages students’ understanding of digital technologies and the foundational skills they require in an ever-changing workforce. Growing demand for digital skills isn’t just limited to the ICT sector. All jobs of the future will require them, from marketing and multimedia through to agriculture, finance and health. Digital Careers prepares students with the knowledge and skills they need to thrive in the workforce of tomorrow.

csiro.au/digital-careers

CSIRO digital **careers**

What is a Solutions Guide?

Computational Thinking skills underpin the careers of the future. Creating opportunities for students to engage in activities that utilise their critical and creative thinking along with problem solving skills is essential to further learning. The Bebras Challenge is an engaging way for students to learn and practice these skills.

Within this Solutions Guide you will find all of the questions and tasks from Round 1 of the Bebras Australia Computational Thinking Challenge 2022. On each page above the question you will find the age group, level of difficulty, country of origin and key Computational Thinking skills.

After each question you will find the answer, an explanation, the Computational Thinking skills most commonly used, and the Australian Digital Technologies curriculum key concepts featured.

Contents

What is Computational Thinking?

Computational Thinking is a set of skills that underpin learning within the Digital Technologies classroom. These skills allow students to engage with processes, techniques and digital systems to create improved solutions to address specific problems, opportunities or needs. Computational Thinking uses a number of skills, including:

DECOMPOSITION
Breaking down problems into smaller, easier parts.

PATTERN RECOGNITION
Using patterns in information to solve problems.

ABSTRACTION
Finding information that is useful and taking away any information that is unhelpful.

MODELLING AND SIMULATION
Trying out different solutions or tracing the path of information to solve problems.

ALGORITHMS
Creating a set of instructions for solving a problem or completing a task

EVALUATION
Assessing a solution to a problem and using that information again on new problems.

More Computational Thinking resources

Visit digitalcareers.csiro.au/CTIA to download the Computational Thinking in Action worksheets. These can be used as discussion prompts, extension activities or a framework to build a class project.

Each resource was designed to develop teamwork; critical and creative thinking; problem solving; and Computational Thinking skills.

Computational Thinking skills alignment

2022 Round 1 Questions	Grade level	Decomposition	Pattern Recognition	Abstraction	Modelling & Simulation	Algorithms	Evaluation
Years 3+4							
Football Uniform	**Easy**	✓	✓				✓
Coin Triangle	**Easy**	✓	✓	✓		✓	✓
Dancing Doll A	**Easy**	✓	✓	✓		✓	✓
Stamping C	**Easy**	✓	✓	✓	✓	✓	✓
Bridge Builder	**Easy**	✓		✓	✓	✓	✓
Go to the Market	**Medium**	✓		✓	✓	✓	✓
Orange Juice	**Medium**	✓	✓	✓	✓	✓	✓
Dentist	**Medium**	✓		✓	✓	✓	✓
Coin Bag	**Medium**	✓	✓	✓	✓	✓	✓
Necklaces	**Medium**	✓	✓	✓	✓	✓	✓
Moving Balls B	**Hard**	✓	✓	✓	✓	✓	✓
Do they Meet?	**Hard**	✓	✓	✓	✓	✓	✓
Tortoise Path	**Hard**	✓	✓	✓	✓	✓	✓
Preferences A	**Hard**	✓		✓		✓	✓
Farmer Beaver	**Hard**	✓		✓	✓	✓	✓
Years 5+6							
Strange Sorting	**Easy**	✓		✓		✓	✓
The Lost Gold Bar	**Easy**	✓		✓		✓	✓
Picking up Carrots A	**Easy**	✓		✓	✓	✓	✓
Counting Out A	**Easy**		✓	✓	✓	✓	✓
Looking in a Mirror	**Easy**		✓	✓			✓
Science Fair	**Medium**	✓		✓	✓	✓	
Forest Watch	**Medium**	✓		✓		✓	✓
Mission Possible	**Medium**	✓	✓			✓	✓
Choices B	**Medium**	✓		✓		✓	✓
Maze	**Medium**	✓	✓			✓	✓
Crested Birds	**Hard**		✓	✓		✓	✓
Strawberry Thief C	**Hard**		✓	✓	✓	✓	✓
The Present	**Hard**	✓				✓	✓
Tree Pruning	**Hard**	✓		✓	✓	✓	✓
Elephants in the Fridge	**Hard**	✓		✓	✓	✓	✓

Australian Digital Technologies curriculum key concepts

Abstraction

Hiding details of an idea, problem or solution that are not relevant, to focus on a manageable number of aspects.

Data Collection

Numerical, categorical, or structured values collected or calculated to create information, e.g. the Census.

Data Representation

How data is represented and structured symbolically for storage and communication, by people and in digital systems.

Data Interpretation

The process of extracting meaning from data. Methods include modelling, statistical analysis, and visualisation.

Specification

Defining a problem precisely and clearly, identifying the requirements, and breaking it down into manageable pieces.

Algorithms

The precise sequence of steps and decisions needed to solve a problem. They often involve iterative (repeated) processes.

Implementation

The automation of an algorithm, typically by writing a computer program (coding) or using appropriate software.

Digital Systems

A system that processes data in binary, made up of hardware, controlled by software, and connected to form networks.

Interactions

Human-Human Interactions: How users use digital systems to communicate and collaborate.
Human-Computer Interactions: How users experience and interface with digital systems.

Impact

Analysing and predicting how existing and created systems meet needs, affect people, and change society and the world.

For more information on the Digital Technologies curriculum, please visit the Australian Curriculum, Assessment and Reporting Authority (ACARA) website:
australiancurriculum.edu.au/f-10-curriculum/technologies/digital-technologies

Digital Technologies key concepts alignment

2022 Round 1 Questions	Grade level	Abstraction	Data Collection	Data Representation	Data Interpretation	Specification	Algorithms	Implementation	Digital Systems	Interactions	Impacts
Years 3+4											
Football Uniform	**Easy**			🟦	🟦	🟦	🟦				
Coin Triangle	**Easy**	🟦		🟦	🟦	🟦	🟦				
Dancing Doll A	**Easy**	🟦		🟦	🟦	🟦	🟦				
Stamping C	**Easy**	🟦		🟦	🟦	🟦	🟦				
Bridge Builder	**Easy**	🟦		🟦	🟦	🟦	🟦				
Go to the Market	**Medium**	🟦	🟦	🟦	🟦	🟦	🟦				
Orange Juice	**Medium**	🟦		🟦	🟦	🟦	🟦				
Dentist	**Medium**	🟦		🟦	🟦	🟦	🟦		🟦		
Coin Bag	**Medium**	🟦		🟦	🟦	🟦	🟦				
Necklaces	**Medium**	🟦		🟦	🟦	🟦	🟦				
Moving Balls B	**Hard**	🟦		🟦	🟦	🟦	🟦				
Do they Meet?	**Hard**	🟦		🟦	🟦	🟦	🟦				
Tortoise Path	**Hard**	🟦		🟦	🟦	🟦	🟦		🟦		
Preferences A	**Hard**	🟦		🟦	🟦	🟦	🟦				
Farmer Beaver	**Hard**	🟦		🟦	🟦	🟦	🟦				
Years 5+6											
Strange Sorting	**Easy**	🟦		🟦	🟦	🟦	🟦				
The Lost Gold Bar	**Easy**	🟦	🟦	🟦	🟦	🟦	🟦		🟦		
Picking up Carrots A	**Easy**				🟦	🟦	🟦		🟦		
Counting Out A	**Easy**	🟦	🟦			🟦	🟦				
Looking in a Mirror	**Easy**			🟦	🟦				🟦		
Science Fair	**Medium**	🟦			🟦		🟦	🟦			
Forest Watch	**Medium**	🟦		🟦	🟦	🟦	🟦	🟦			
Mission Possible	**Medium**	🟦		🟦		🟦	🟦		🟦		
Choices B	**Medium**	🟦				🟦	🟦	🟦			
Maze	**Medium**				🟦	🟦	🟦		🟦		
Crested Birds	**Hard**	🟦					🟦				
Strawberry Thief C	**Hard**	🟦					🟦				
The Present	**Hard**				🟦	🟦	🟦				
Tree Pruning	**Hard**	🟦			🟦	🟦	🟦				
Elephants in the Fridge	**Hard**	🟦		🟦	🟦	🟦	🟦				

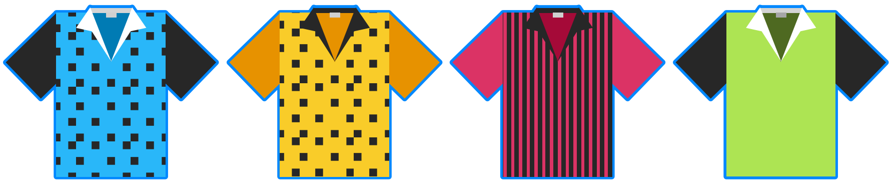
Bebras Challenge 2022 Round 1

Years 3+4

This question comes from
Ireland

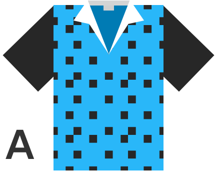
Football Uniform

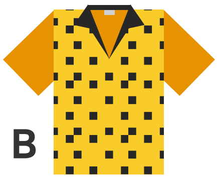
Anne is packing her bag for a football match. She needs to pack the shirt which does not have stripes and does not have black sleeves. It must have a black collar.

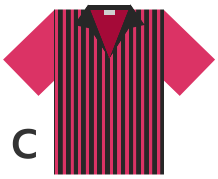


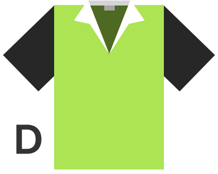
Question

Which shirt should Anne pack?


A


B


C


D

Answer

The correct answer is shirt B.

Explanation

Shirt A is not correct because it has black sleeves.

Shirt B is correct because it does not have black sleeves, it does have a black collar, and does not have stripes.

Shirt C is not correct because it has stripes.

Shirt D is not correct because it has black sleeves.

BACKGROUND INFORMATION

In this task, one condition has to be true (black collar) and two conditions have to be false (stripes and black sleeves). Understanding conditions is very important in computer programming. All computer programming languages have conditions. Conditions can be used to tell which parts of a computer program should be run next (an “if” statement) and some conditions can be used to tell which objects should be included in, or left out of, lists of objects (a list comprehension).

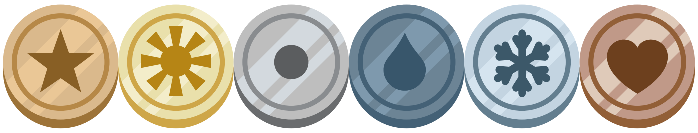
This task can be used to introduce the logical Boolean operators AND and NOT.

In the field of machine learning, *classification* is the concept of a computer program learning to group objects together based on their features. For example, the machine learning computer program might be given lots of examples of football shirts and it would figure out what combination of conditions (collar, sleeves, stripes, stars, colour, length, crest) are best to separate two shirt types.

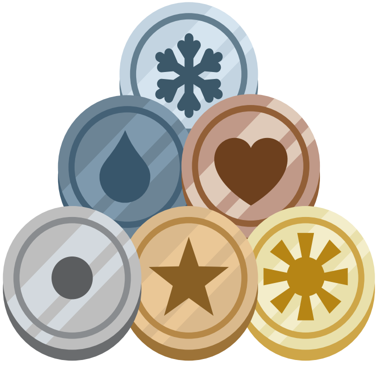
This question comes from
Czechia

Coin Triangle

Emily has six coins:



She put the coins on top of each other to make a triangle shape.

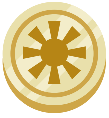


Question

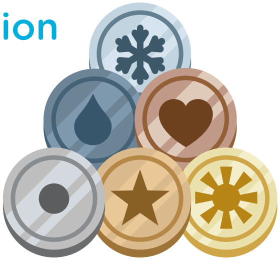
Which was the fourth coin Emily put down?

EXPLANATION

Answer



Explanation



Each coin overlaps at least one other, so you can start by finding the coin that is not on top of any others. This is the top one in the triangle, and must have been put down first. The next coin to be placed can only be above the starting coin. By working our way around in this way, we can find the fourth coin to be placed.

BACKGROUND INFORMATION

The coins in the picture are laid in a *sequence*.

You can see the same effect if you're drawing a picture on a computer. If you draw a yellow circle, then two dots, and then a curved line, you get a smiley face. If you had drawn the yellow circle last, the two dots and curved line would have been hidden behind the circle.

Computers usually work sequentially. Most computer programs are written so that first action and then another action happens. So a computer program for drawing a smiley face could look like this:

```
draw circle at (5,5) with radius 5
draw dot at (2,7)
draw dot at (7,7)
draw left curved line from (2,2) to (7,2)
```

Of course, sequences are not all that computers can do. In order to program more complex programs, a computer needs to be able to make decisions and perform *repetitions*.

This question comes from India

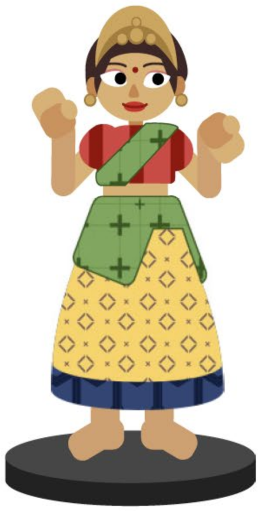
Years 3+4 Easy
Years 5+6
Years 7+8
Years 9+10
Years 11+12

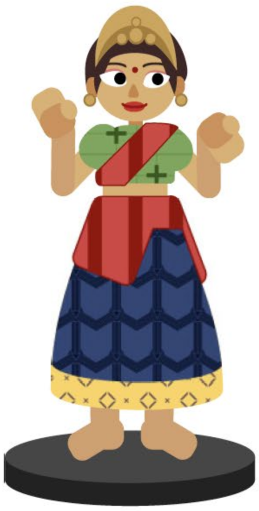
Dancing Doll

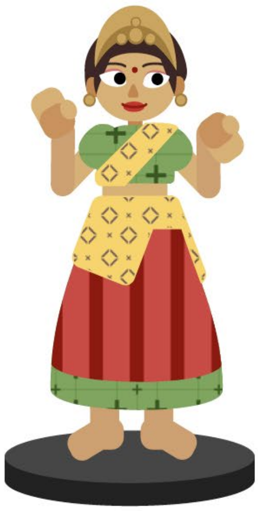
A dressmaker was asked to make dresses for four dolls. Each dress should be made from four different patterned materials.

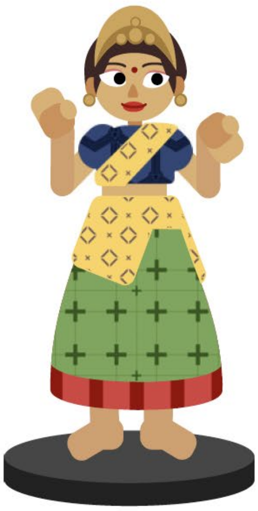
Question

Which doll is wearing a dress that is NOT made from four different patterns?



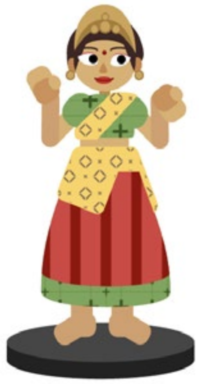






Answer

Which doll is wearing a dress that is NOT made from four different patterns?



Explanation

The dress being worn by this doll is made from only three different patterned fabrics. The others are made from four.

BACKGROUND INFORMATION

Data can take the form of images, numbers or text.

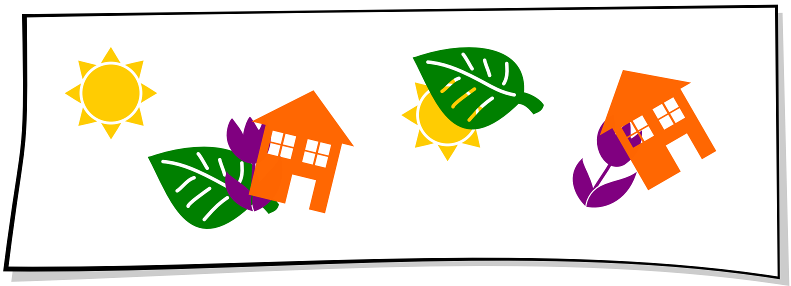
In this task, the four materials form the data structure. The students are asked to do a *data verification* process to check that the required rules, about how many different coloured patterns have to be used, are correct in each case.

This question comes from Hungary

Stamping

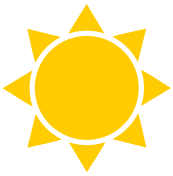
Casey has four stamps which make four different pictures.

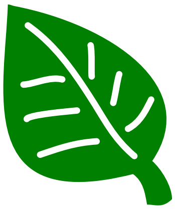
Casey picks up each stamp just *once* and uses it to stamp *twice*. This is what Casey has made:

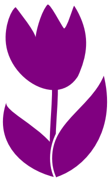


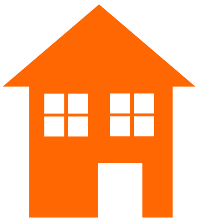
Question

Which stamp did Casey use first?



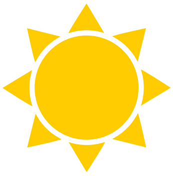






EXPLANATION

Answer



Explanation

The sun is “under” the leaf so it must have been stamped before the leaf. The leaf is *underneath* the flower and the house, so the sun must have been stamped first.

BACKGROUND INFORMATION

There are a lot of computer programs and apps where you can draw or manipulate pictures. Most of them use *layers*. With layers you can define the order of image-parts. The pictures on the bottom layer will be overlapped by the pictures on other layers. Of course, the layers can be different sizes and they can also have transparent backgrounds.

In an animation, where you use frames (pictures) after each other, the bottom layer can contain the background – and it can be copied to the next frame. You can manipulate the layers separately so it is easier to modify, copy or delete a smaller part of the picture.

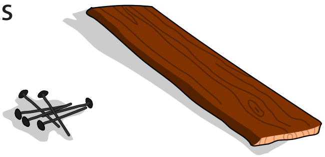
This question comes from Czechia

Bridge Builder

Bella has a new building project! She needs: a hammer, nails, wooden boards and a rope.

She already has a hammer and a rope:

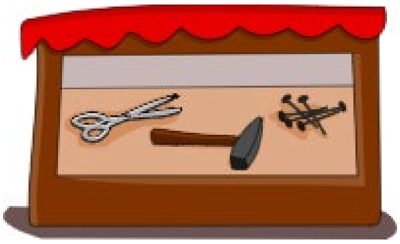
So she needs to buy some nails and wooden boards:

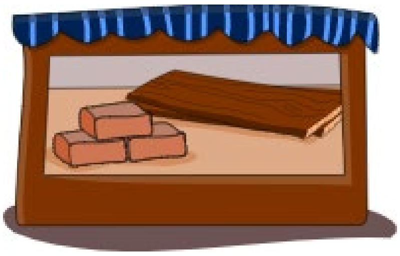


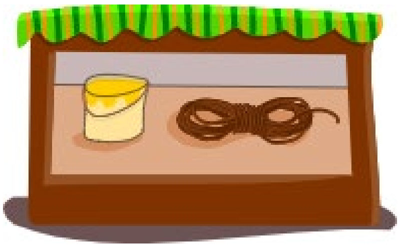
There are three shops in the village where Bella can get these things, but she wants to visit as *few* shops as possible.

Question

Which of the shops does Bella need to visit?

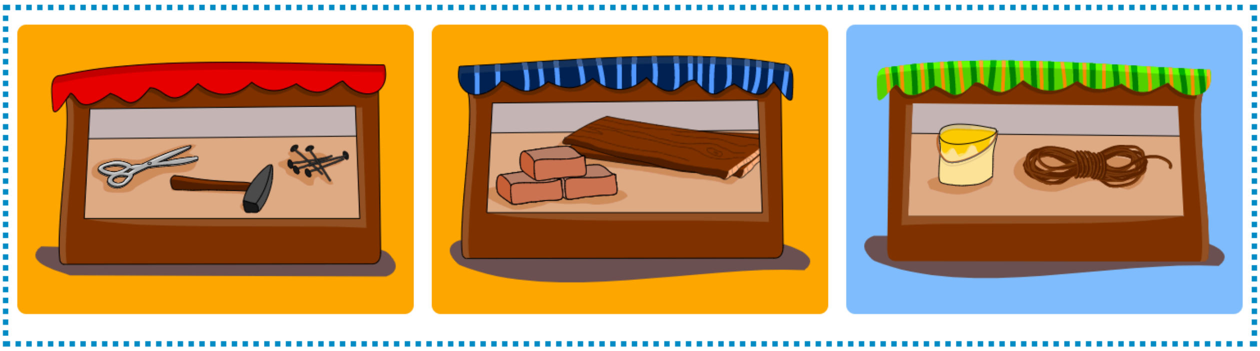






EXPLANATION

Answer

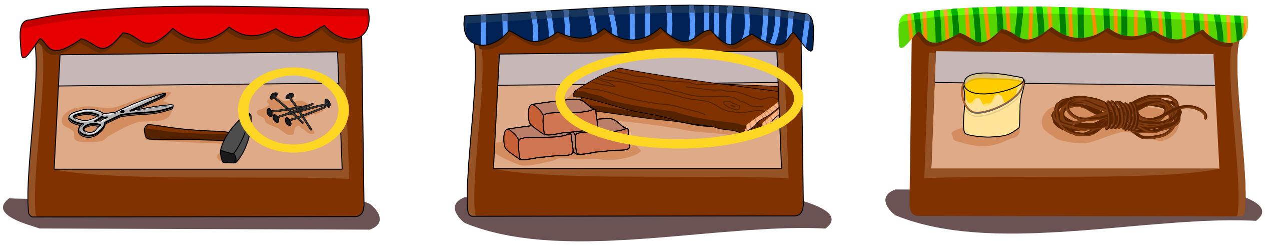


Continued on next page

Bridge Builder– continued

Explanation:

The missing items are shown below



BACKGROUND INFORMATION

This simple problem shows how a big *application* (the village) may offer a lot of resources (procedures), but these resources are organised into smaller groups called *microservices*. When a *process* (Bella) needs some procedures in its logic, it doesn't have to implement them itself (Bella doesn't need to craft her own items), but it requests them from other services (in this case, Bella buys the resources from other stores).

This problem asks for the stores on which Bella depends. When writing more complex programs, it is good practice to check which services a process depends on so that when something doesn't work all affected processes can be traced to their root cause.

This question comes from
Indonesia

Go to the Market

Anushka, Barnie and Chloe are planning a party for their family. They need to buy *all* the items shown below.

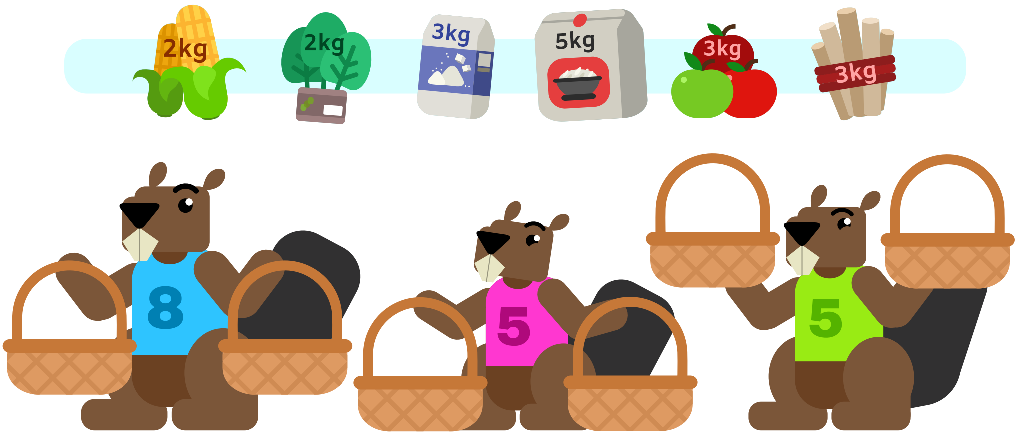
Each beaver can only carry two baskets:

- Anushka can carry *8kg* in total.
- Barnie and Chloe can each carry *5kg* in total.

All the food must be carried home in a single trip.

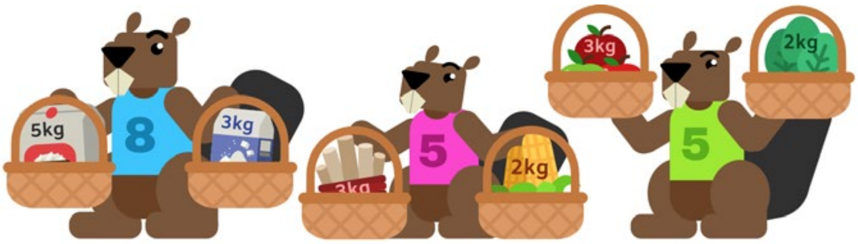
Question

Which items should each beaver carry? Drag the items to their baskets.



EXPLANATION

One possible answer



Explanation

There are actually quite a few possible answers, but the beaver that can carry 8kg *must* carry the 5kg package and one of the 3kg foods.

This leaves the other two beavers that can carry 5kg. They must both carry one 2kg and one 3kg item.

BACKGROUND INFORMATION

This task requires students to satisfy some *constraints* to determine which beavers can carry which food items. While there are a number of options, there are some important constraints that must be satisfied.

In a computer system, there are often processes that can only be performed if a given set of constraints are satisfied.

This question comes from
Uruguay

Years 3+4 Medium
Years 5+6
Years 7+8
Years 9+10
Years 11+12

Orange Juice

The beavers are playing a logic game.

Beaver Kamal can drink from a bottle when:

A) there is a bottle with *less* juice immediately to the *left* of this bottle, *and*

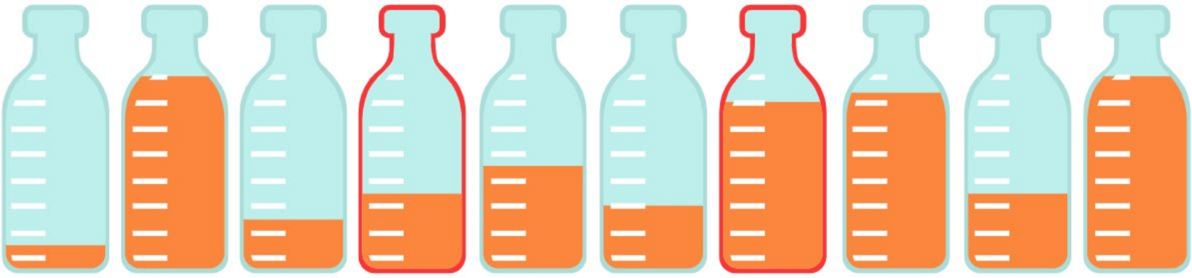
B) there is a bottle with *more* juice immediately to the *right* of this bottle.

Question

Click on all the bottles that Kamal can drink from.

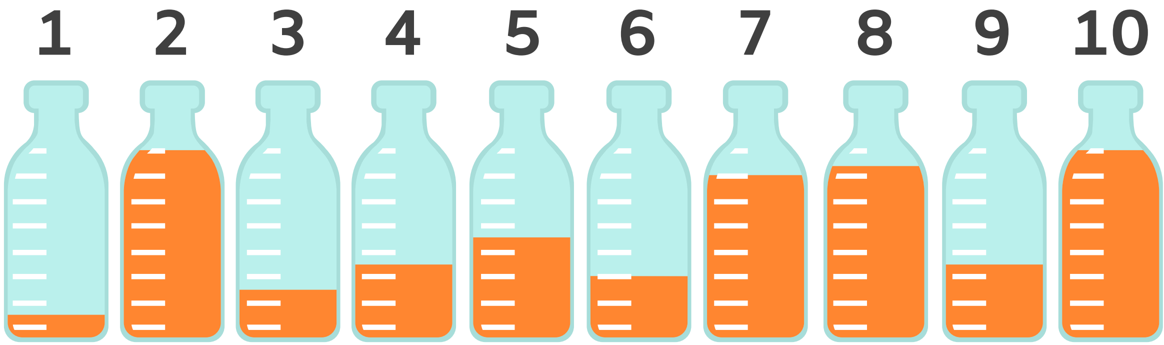
EXPLANATION

Answer



Explanation

Only the bottles numbered 4 and 7 meet the given conditions, that is, *less* juice on the left AND *more* juice on the right.



BACKGROUND INFORMATION

Computer Science often involves solving problems that are specified by a set of logical *constraints*. The task is to find a solution that satisfies all of these restrictions.

More complex tasks can be considered where the constraints are combined by logical operators, for example:

- A **AND** B means that both constraints A and B must be satisfied, as in this task;

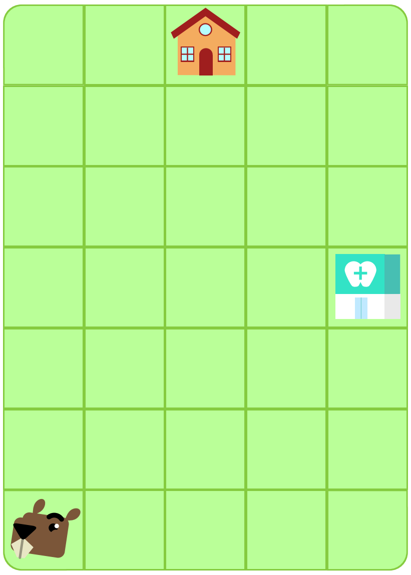
- A **OR** B means satisfying only one of them is sufficient.

The process that the beavers are running here has close ties to *scheduling*, that is, a method of deciding how the bottles are chosen.

Dentist

Beaver Ruth is heading home from school but first she must go to the dentist.

To make the journey more fun, she plays a game.



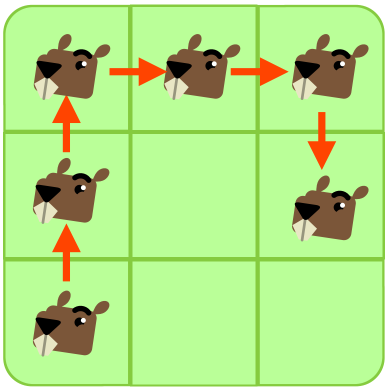
Game rules:

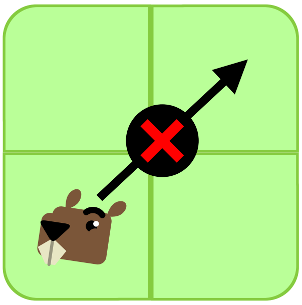
Ruth can only move *straight* ahead, from one square to another.

Instead of moving forward she can turn *right*.

She can follow rules 1 and 2 as many times as she wants, and in *any* order.

She must NOT go diagonally.





Question

Is it possible to go to the dentist *first* and then home if Ruth follows her rules?

It is not possible to reach both places in this way

It is possible, if she turns right exactly 2 times

It is possible, if she turns right exactly 4 times

It is possible, if she turns right exactly 6 times

Continued on next page

Dentist – continued

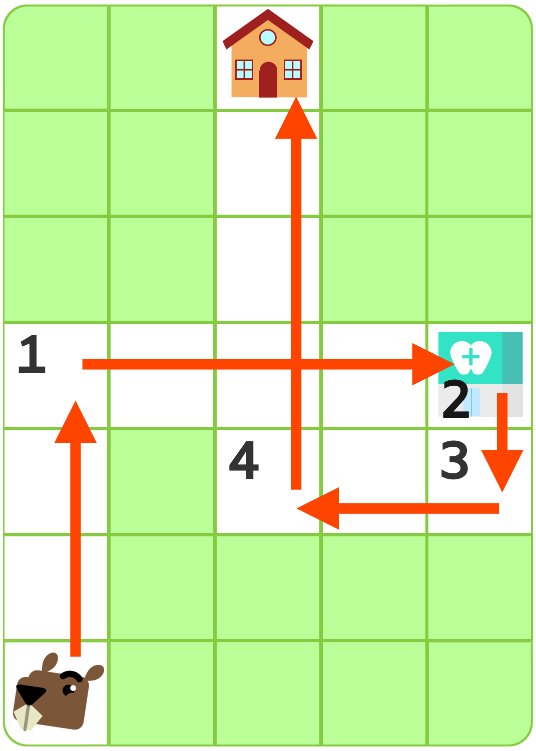
EXPLANATION

Answer

It is possible, if she turns right exactly 4 times.

Explanation

The beaver can get to her home after visiting the dentist if she does a combination of *movement conditions* (moving straight and right) according to the following picture:



BACKGROUND INFORMATION

One of the main tasks in *computer science* is to search for possible solutions. These solutions have to follow certain conditions. The question that is often asked is whether there is at least one possible solution.

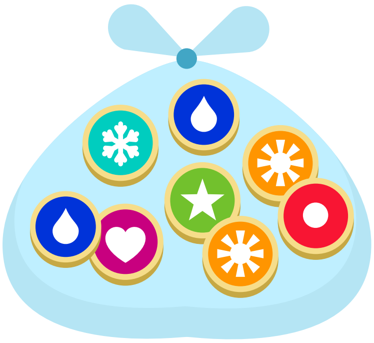
In this task it is necessary to write a program for the movement of the beaver according to the given conditions, and with certain restrictions. By doing so, it can be confirmed that a solution is indeed possible.

This question comes from
Ireland

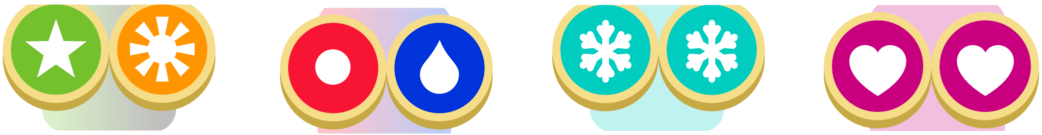
Years 3+4 Medium
Years 5+6
Years 7+8
Years 9+10
Years 11+12

Coin Bag

This is Sam's bag of coins:



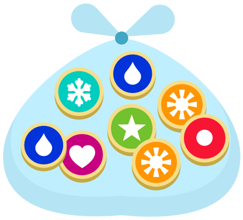
In Sam's country there are only four types of coins. The image below shows *both* sides of each type of coin:

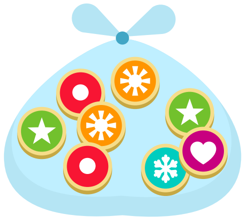


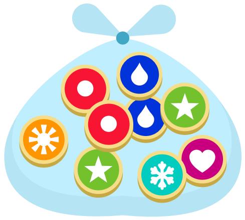
Sam's bag of coins has been shaken and placed next to three other bags.

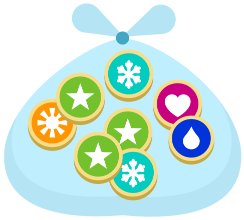
Question

Which is Sam's bag of coins?



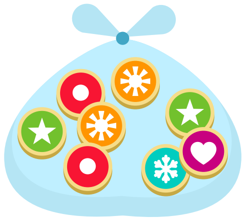






EXPLANATION

Answer



Explanation

Sam's bag has:

- four coins of type *one* (green/yellow),

- two coins of type *two* (red/blue),

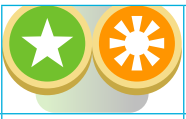
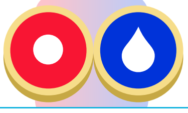
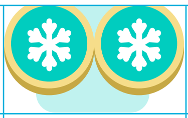
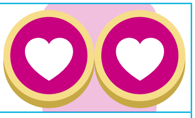
- one coin of type *three* (orange), and

- one coin of type *four* (purple)

Continued on next page

Coin Bag – continued

This is illustrated in the table below.

				
Sam's bag	4	2	1	1
Bag A	3	3	1	1
Bag B	4	1	2	1
Bag C	4	2	1	1
Bag D	2	4	1	1

is not correct because it has three star/sun coins but Sam's bag has four star/sun coins.

is not correct because it has two snow coins but Sam's bag has only one snow coin.

is correct because it has the correct number of each type of coin.

is not correct because it has three star/sun coins but Sam's bag has four star/sun coins.

BACKGROUND INFORMATION

Some of the world's information (stories, conversations, messages, shopping lists) can have different lengths and unstructured information. Computer scientists often have to invent a structure, or way of organising information so that the information can be processed by computer programs. Sometimes, certain features have to be ignored, and things that look different have to be treated as being the same. When we organise information in this way we are using the computational thinking skill called *abstraction*.

In this task, a bag is used as an example of unstructured data. (The bag has no particular ordering for its coins and can have multiple coins of the same type.) There are more complicated real-world examples too. Extracting meaning from human language is one very difficult but important task for computers that deals with unstructured data. For example, imagine if people were asked "What did you like best about this movie?" and various people responded with these answers:

"I loved the score"

"The music in this movie"

"The complete audio experience"

"I recognised my favourite song"

A computer program analysing these peoples' responses would have to recognise that, in this context, all of these statements should be represented as being the same, even though they all use different words.

This question comes from
Slovakia

Years 3+4 Medium
Years 5+6
Years 7+8
Years 9+10
Years 11+12

Necklaces

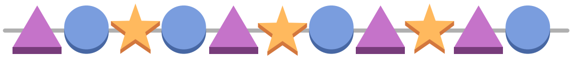
Beavers Anna, Bella and Lena made bracelets to spell out their names.

They used different patterns of just *two* types of bead for each letter: ▲ and 🔵 .

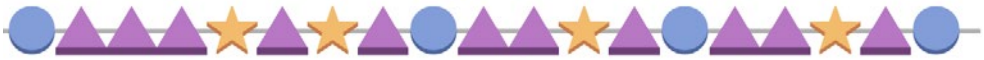
To separate the letters in the bracelets they used ★ beads.

The finished bracelets:

Anna

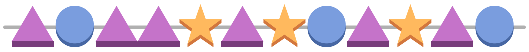


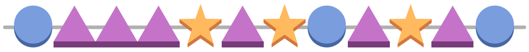
Bella

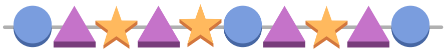


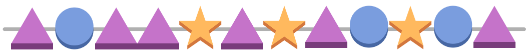
Question

Which bracelet did Lena make?



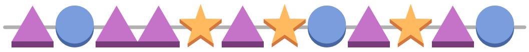






EXPLANATION

Answer



Explanation

The letter L is in the name BELLA and is the third letter in the bracelet, so it is made of beads ▲🔵▲▲

From Bella's necklace we can also see that E is ▲.

From Anna's necklace we see that N is 🔵▲ and that A is ▲🔵.

BACKGROUND INFORMATION

Computer scientists often have to *encode* information, mainly to make communication easier, to write information, or to store it more economically. We can also make secret codes. Here the encoding of letters is based on the Morse Code, where the dot (●) of Morse Code is replaced by ▲ and the dash (▁) is replaced by 🔵 .

Therefore, 'A' in ANNA's name is encoded as ▲🔵 (● ▁).

If we don't know which encoding is used, then we can't create necklaces for names with other letters. We would have to agree on how to encode each letter from the alphabet. Computer scientists can also encode information for pictures, sounds or videos.

This question comes from
South Korea

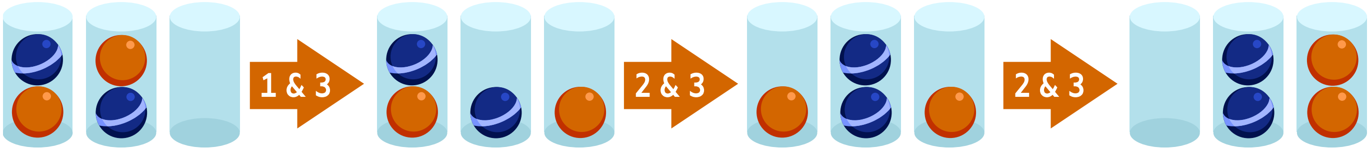
Moving Balls

Beavers are playing a game. The aim of the game is to move the balls so that the same coloured balls are in the same tubes.

Game Rules:

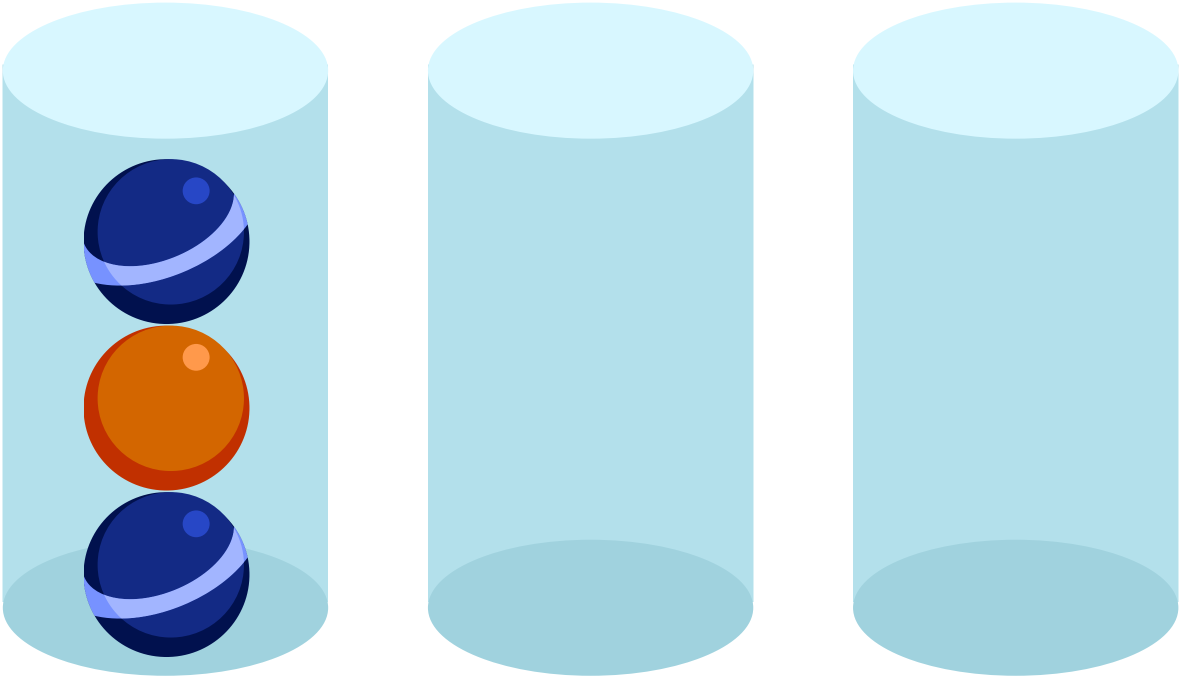
- Rule 1: Only one ball can be moved at a time.
- Rule 2: Only balls at the top of a tube can be moved.
- Rule 3: A ball can be moved into an empty tube.
- Rule 4: When there is a space in a tube, a ball can only be moved on a ball of the same colour.

Example:



Question

Drag the balls from tube to tube, obeying the rules of the game, until the same balls are in the same tubes.

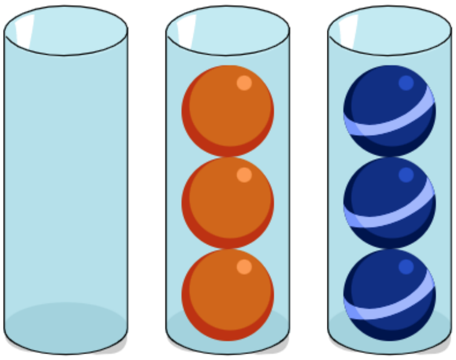


Moving Balls – continued

EXPLANATION

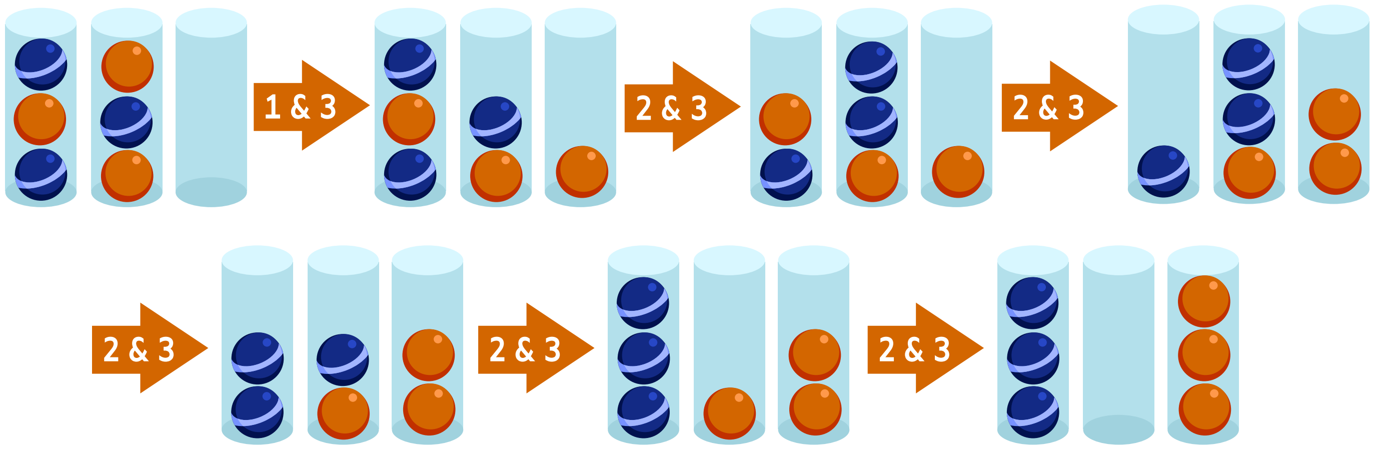
Answer

There are various solutions to this task as long as the balls all end up sorted into their own columns, for example, like this:



Explanation

The desired outcome has to be achieved by making a series of allowed moves. A possible solution and the moves required to achieve it are shown below:



BACKGROUND INFORMATION

In Computer Science data is often stored in computer memory in *stacks*. The special characteristic of stacks is that you can only access and remove the last added data.

The cylinders in our example behave like stacks: You can only add a ball on the top of a cylinder. And you can only remove the ball from the top. The other balls cannot be accessed.

This question comes from Lithuania

Do They Meet?

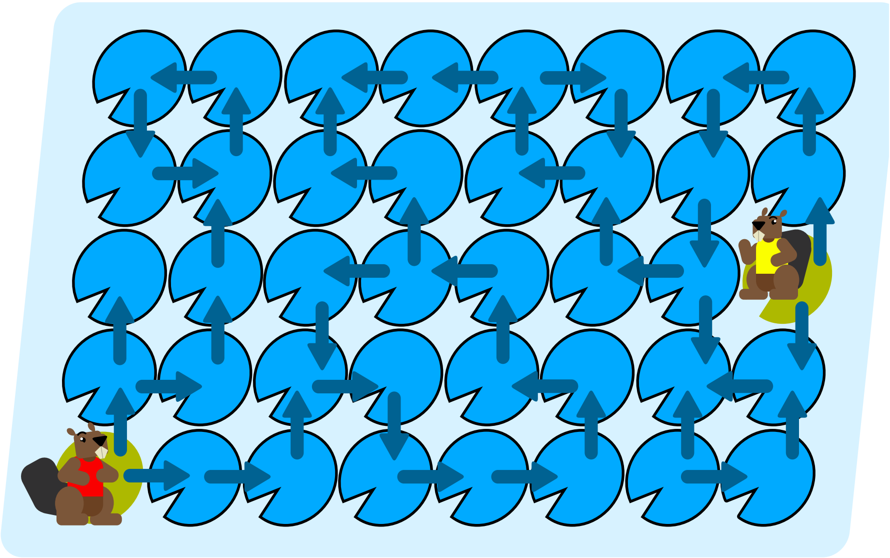
Beavers Liem and Anika would like to meet on one of the lily pads.

They start on *different* lily pads as you can see below.

The two beavers can only move from one lily pad to the next by following the arrows.

Question

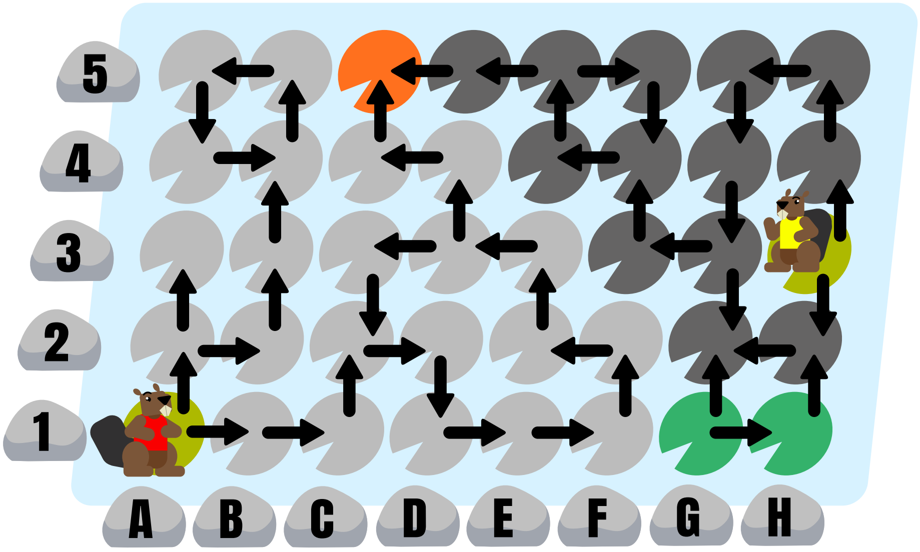
Click on the lily pad where they can meet if they follow the arrows.



EXPLANATION

Answer

The beavers may meet on pad C5.



Continued on next page

Do They Meet? – continued

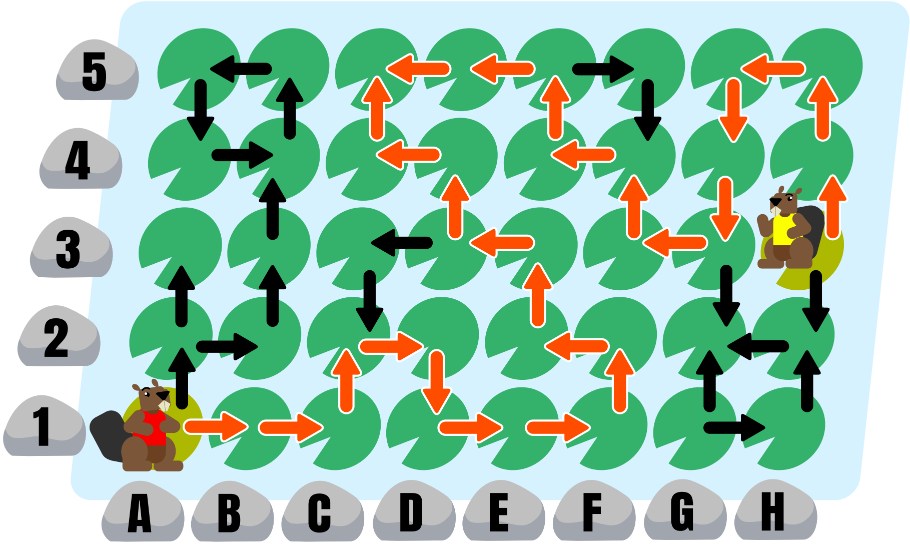
Explanation

One way to solve this problem is to consider different paths each beaver can take.

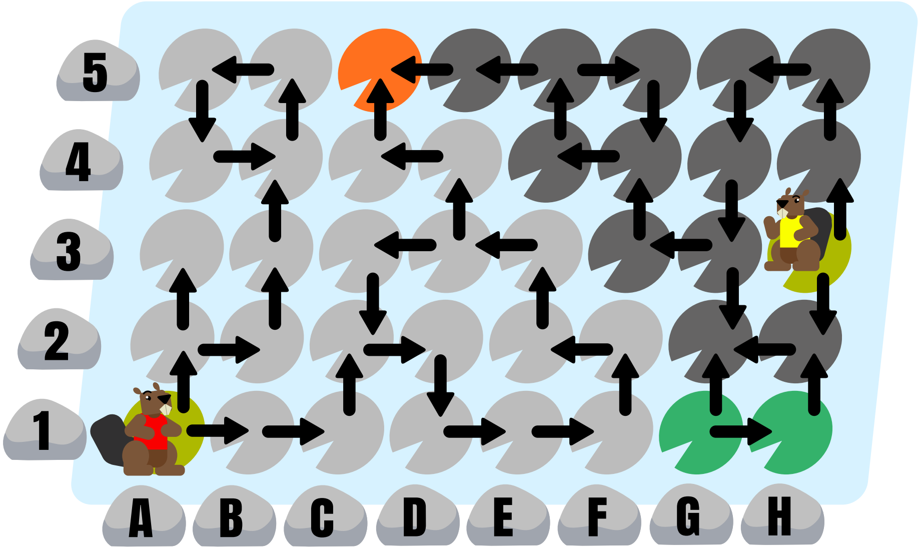
At his starting position, Liem has two options: If Liem goes “up”, then he may either run into the dead end at A3 or get stuck in the loop that begins at B4. If he goes “right” (to B1), he can continue to D3. At D3 he may either go “left” into a loop that will take him back to D3 eventually, or go “up” which makes him end up at C5 which is another dead end.

Anushka also has two options at the start. If she goes “down”, she will run into the dead end at G2. If she goes “up”, she will reach G3. From there she may either run into the G2 dead end again or go “left” and reach E5 eventually. There she may either go into a loop that will take her back to E5 again or reach another dead end at C5.

As we already know, Liem may reach C5 as well, so we can see that the beavers may meet at C5. The picture shows the ways along which both beavers can reach C5.



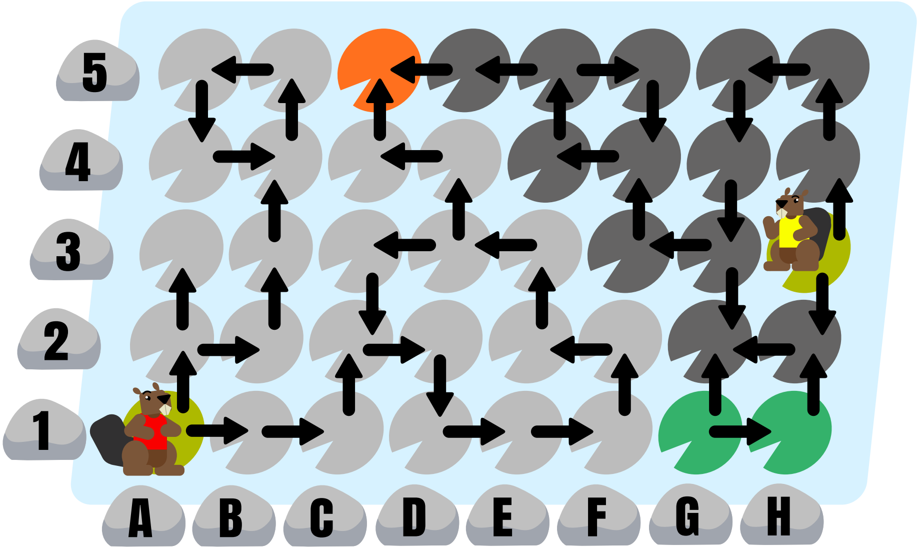
But this does not yet fully guarantee that the beavers cannot also meet at F4. Another way to think about this problem is to define all the possible pads that each beaver can get to. The next picture shows the set of pads that Liem (white) and Anushka (dark grey) may reach by following the arrows in any possible way. We can see that C5 is the only pad common to both sets.



Do They Meet? – continued

BACKGROUND INFORMATION

Let us take a closer look at how we created this picture:



For each beaver, we followed the arrows. If there was a dead end, or a loop was detected, we went back to the most recent fork in order to choose another option to proceed. Thus, we followed all possible ways.

A very similar procedure is often applied by computer scientists when solving difficult problems. The procedure tries to construct a solution step by step. Often there are several options to choose the next step. In this case, the procedure will choose one option but keep track of the other options as well. When it runs into a dead end (a loop in this case is also considered a dead end if it is detected), it goes back to the most recent choice and tries another option. This algorithmical approach to problem solving is called *backtracking*; this name gives a pretty good indication of how the approach works.

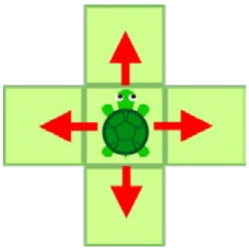
This question comes from Germany

Years 3+4 Hard
Years 5+6
Years 7+8
Years 9+10
Years 11+12

Tortoise Path

Four tortoises live in four different small gardens.

Each garden is divided into grass squares, surrounded by a stoney area. The tortoises *cannot* cross the stoney areas but they can move from one grass square to the next, as shown here:

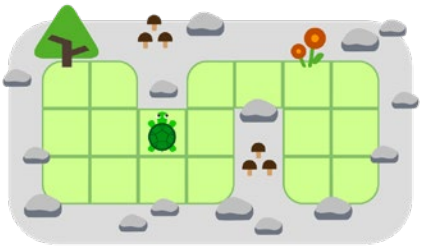


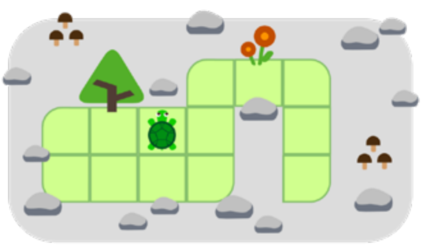
Each tortoise would like to take a perfect feeding path in its garden.

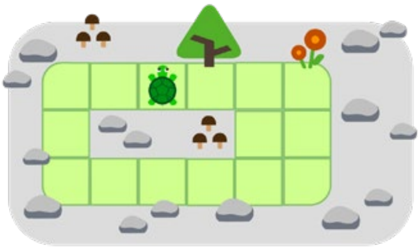
A perfect feeding path is one where the tortoise can move to all the grass squares, while visiting each grass square only once.

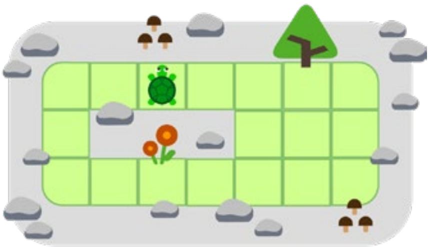
Question

Which one of the four tortoises shown below cannot take a *perfect feeding path*?





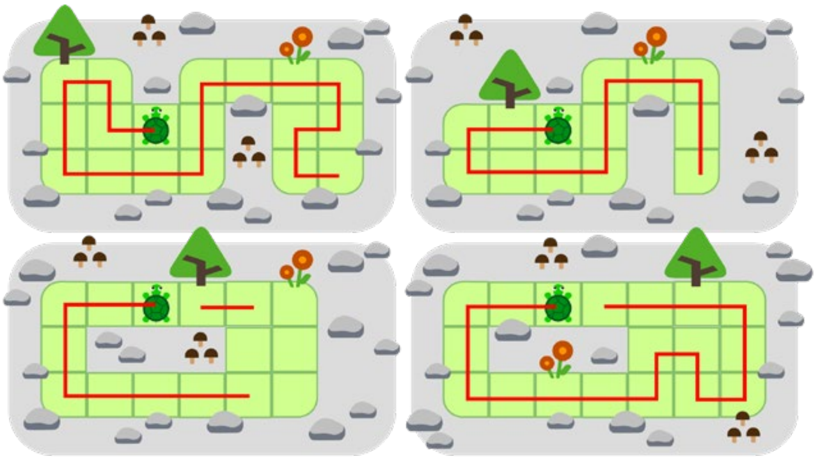




EXPLANATION

Answer

The bottom left garden should be chosen.



Tortoise Path – continued

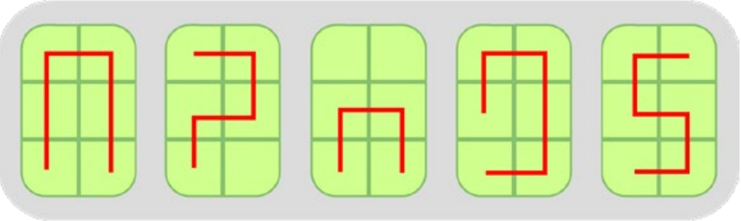
Explanation

For each of the gardens a feeding path is shown.

There is no *perfect feeding* path in the bottom left garden, though. Such a path would have to include the parts shown above. The tortoise still needs to find a path across the six remaining grass squares, starting at one end of the shown path and ending at the other, while visiting all six squares exactly once.

However, there is no such path:

Let the tortoise start on the lower left of the six squares; it is sufficient to consider this case, due to symmetry. See below all paths the tortoise can take on the six squares, visiting as many squares as possible, but each square at most once. None of these paths ends at the upper left square.



BACKGROUND INFORMATION

The tortoises want to find a path through their gardens, visiting each grass square exactly once. This problem is quite well known in computer science as the *Hamiltonian path problem*. In the 19th century, Sir William Rowan Hamilton studied a lot of these type of problems and found that some were not possible for computers to solve!

This question comes from Germany

Preferences

Three beavers each want a log.

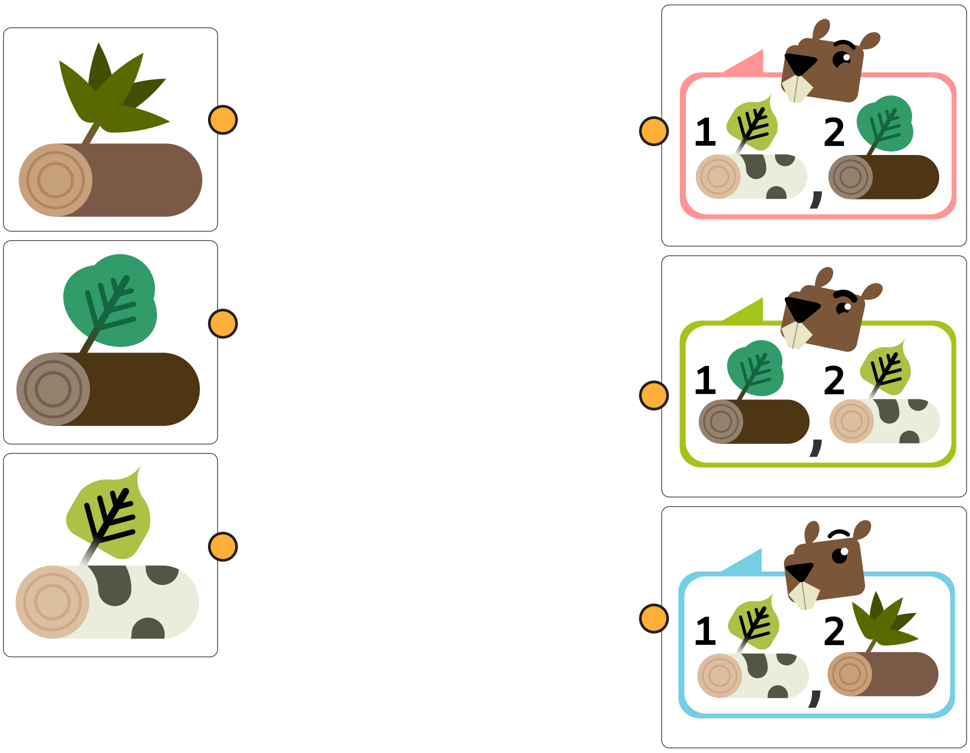
They write down what their first and second choice would be (shown below on the right).

They decide on these rules:

1. We will make sure that we get the *highest* number of *first* choices possible.
2. No one should get a stick that isn't either their first or second choice.

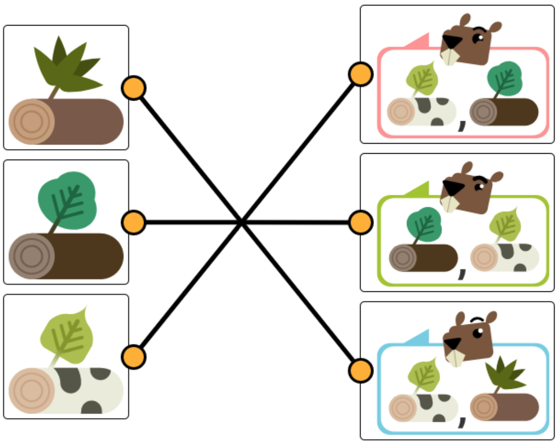
Question

Show which beaver gets which log by drawing a line between the sticks and beavers.



EXPLANATION

Answer



Explanation

We cannot satisfy all first-place preferences, because two young beavers have the same first-place preference. The assignment shown above satisfies two first-place preferences, and one second-place preference. Given that not all first-place preferences can be given, this is the best assignment possible.

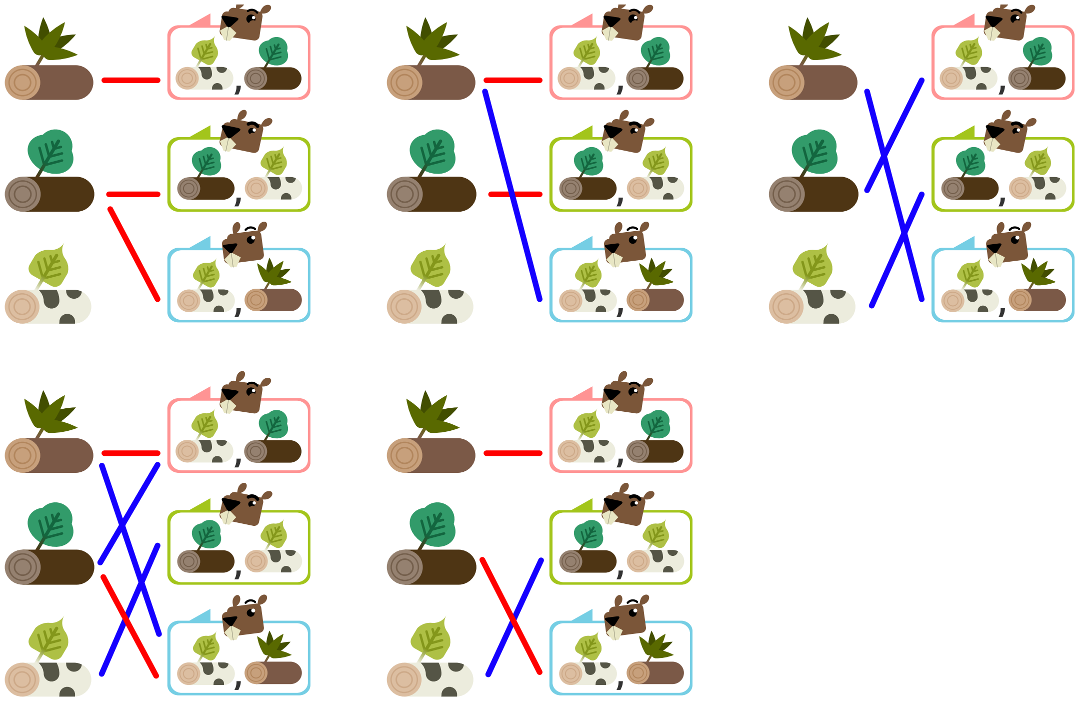
Continued on next page

Preferences – continued

BACKGROUND INFORMATION

One *algorithm* that can be used to solve this problem is to try all methods of distributing gifts and choose the best one. To do this, six options are considered: some of them will leave some beavers without gifts and from the remaining options the best one can be chosen. But if there were 10 beavers and 10 gifts, then there would be more than three million options that would have to be considered - the above *algorithm* would no longer be practical for a person to use!

Computer scientists are often faced with the task of correctly organising such options. For example, in our problem we can first consider only those gifts that the beavers want to receive in the first place. In the picture below they are marked with red links. It is immediately obvious that the second and third beavers want to receive the same gift.



Since we can satisfy the preference of only one of these beavers, two options need to be considered:

1) the second beaver receives their second preference, and

2) the second beaver does not receive their second preference.

For each of the cases a picture is shown, in which the gifts that the beavers want to receive in the second turn are indicated in blue. In the first case, the blue links can be removed from the second beaver and the second gift since the gift has already been assigned to the second beaver. In the second case there are still many options but we can do the same, breaking all the options into those in which the third beaver receives the first gift and those where they do not receive it. By splitting the options into two parts we will quickly find all possible distributions of gifts so that later we can choose the best one. This algorithm in computer science is called the *divide-and-conquer algorithm*.

This question comes from
Turkey

Years 3+4 Hard
Years 5+6
Years 7+8
Years 9+10
Years 11+12

Beaver Farmer

Farmer Mert grows wheat in fields that contain a wheat symbol in the map below.

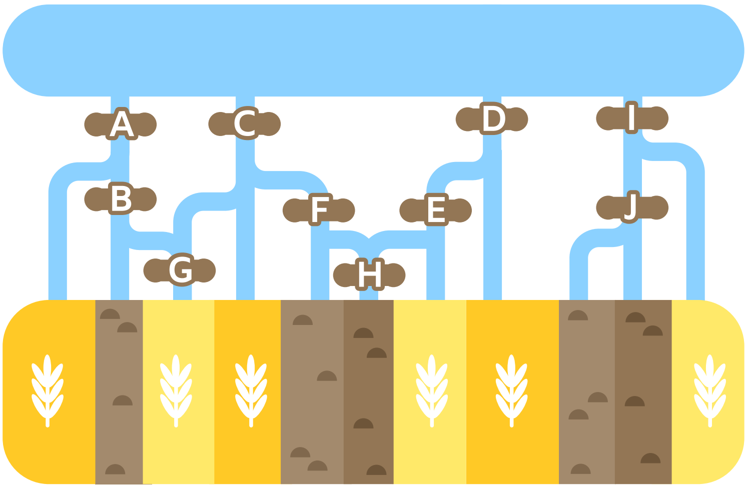
He also has stony fields where nothing grows, shown by the stones symbol.

To save water, Mert only wants to water the *wheat* fields. He can block the water channels coming from the lake at the spots marked with the letters A to J.

The water will only flow *downwards* towards the fields and will never flow back towards the lake.

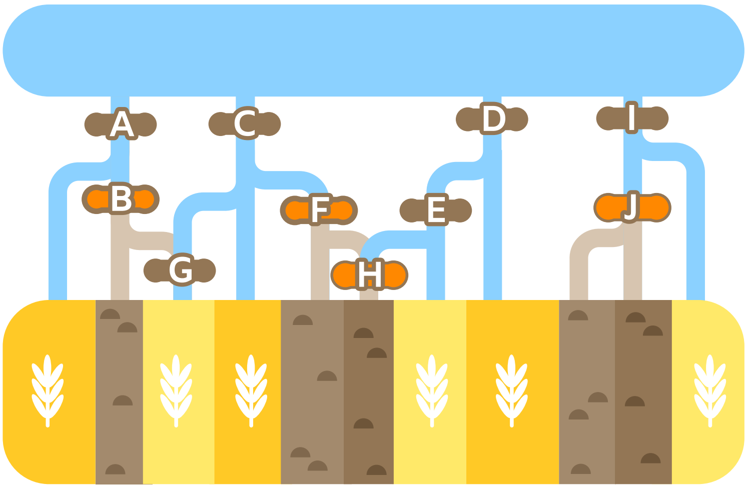
Question

Select the letters to block the water from flowing to the empty fields while still letting it flow to the wheat fields.



EXPLANATION

Answer



Beaver Farmer – continued

Explanation

If none of the gates are closed, the water will reach the fields with stones. If more gates than shown above are closed, then some wheat fields will get no water. If we review all spots we can determine the optimal choices:

- A must be open to irrigate field 1
- B has to be closed to avoid irrigating field 2. It also brings water to field 3—but this field can also get water through C
- C must be open both for field 4 and for field 3, which cannot be watered though A since B is closed
- D must be open for fields 7 and 8
- E must also be open field 7
- F must be closed to prevent the irrigation of field 5
- G, if closed, would only prevent field 3 from being watered and thus must be open as well
- H must also be closed even if F is also closed, as water can flow from the open D and E
- I must be open for field 11
- J, finally, must be closed to prevent fields 9 and 10 from getting irrigated

BACKGROUND INFORMATION

In this task, water flows to the fields based on a number of *conditions*. For instance, water flows to field 7 if both D and E are open. Water flows to field 3 if G is open and either of these conditions hold: (1) C is open; or (2) both A and B are open.

These types of compound conditions are formed with the *boolean operators*:

- AND if two gates are one after the other on the same channel;

- OR if water can flow to the same destination from two separate channel segments.

Boolean conditions are always either true or false.

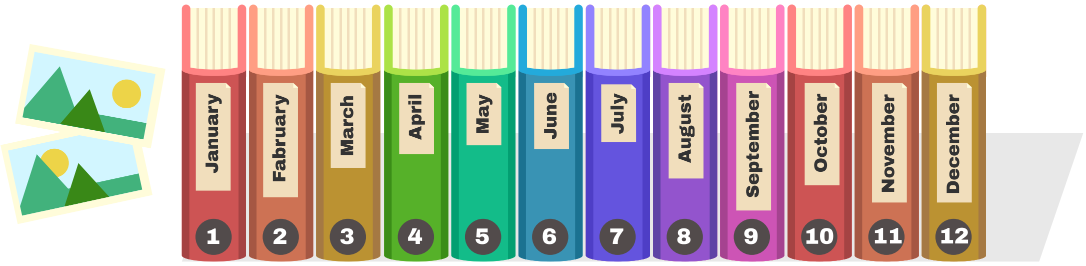
In programming, booleans commonly occur in *if statements* which are found in virtually all programming languages. If *statements* are used to check that a certain condition is true before executing a series of instructions.

Bebras Challenge 2022 Round 1

Years 5+6

Strange Sorting

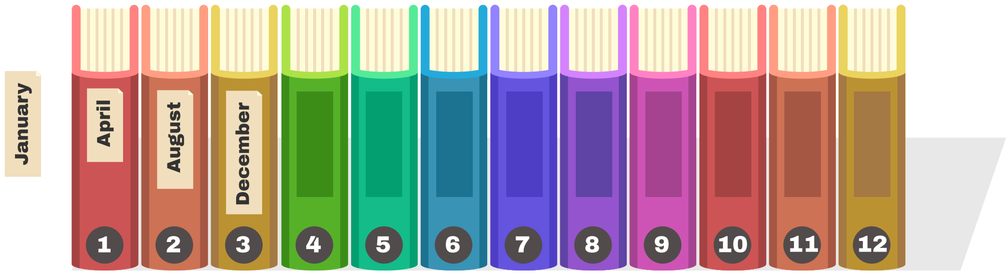
Beaver Tsuki has a pile of photos taken in 2020, each with the exact date on it. She asks her friend Luna put the pictures into 12 albums, one for each month. She also asks Luna to sort the albums when she is done. Tsuki is of course expecting the usual order:



But it is April Fools' Day, so Luna plays a prank on Tsuki and sorts the albums alphabetically by the name of the month instead and removes the labels from all but the first three.

Question

Which album contains Tsuki's pictures from January? Drag the label to the correct album.



EXPLANATION

The pictures from January will be in album #5.

When we sort the months of the year alphabetically, we get the following order:

1. April
2. August
3. December
4. February
5. January
6. July
7. June
8. March
9. May
10. November
11. October
12. September

Therefore, we will be able to find the photos from January in the fifth album.

BACKGROUND INFORMATION

We often want to sort things to help keep them organised and make them easier to find. Computer Science helps us to do this quickly. But before starting the sorting process, it is important to consider if different orders are possible and to decide which order we want.

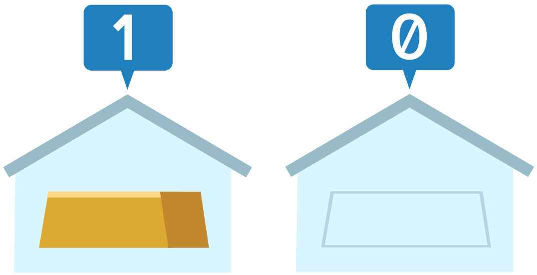
In the example, Luna knew what order Tsuki expected, and was playing a prank on her. But a computer would not have known what order Tsuki expected. When we use a computer to sort something, we must make sure to tell the computer how to perform the sort.

This question comes from
China

Years 3+4
Years 5+6 Easy
Years 7+8
Years 9+10
Years 11+12

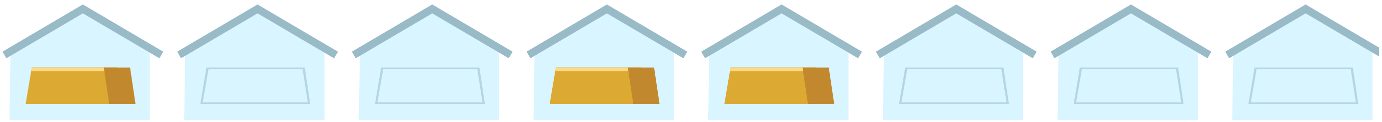
The Lost Gold Bar

One day, a bank vault was stolen, and many gold bars were lost. Beaver policemen immediately carried out a survey. They discovered that the thieves hid the treasure in three of the eight neighbouring houses. They marked the houses one by one with a 1 or 0 in order to find the lost gold bars.



If the number is marked as 1 it means there is a gold bar in the house, if the number is marked as 0 it means that there is no gold bar in the house.

The houses are arranged in the following way:



Question

How can the information around the above houses be edited into 1’s and 0’s?

10001001 | 10011000 | 01100001 | 10010000

EXPLANATION

Answer

The correct answer is B - 10011000.

Explanation:

According to the task’s description, the digit 1 represents the presence of gold bars in a house whereas 0 represents the absence of gold bars in a house. When you mark the houses in the sequence as they are in the picture according to the presence (1) or absence (0) of gold bars, you get a sequence of digits consisting of 1 and 0. Only the first, fourth and fifth house have gold barss inside, so the first, fourth and fifth digits in the sequence are 1, and the other digits are 0. The answer is 10011000.

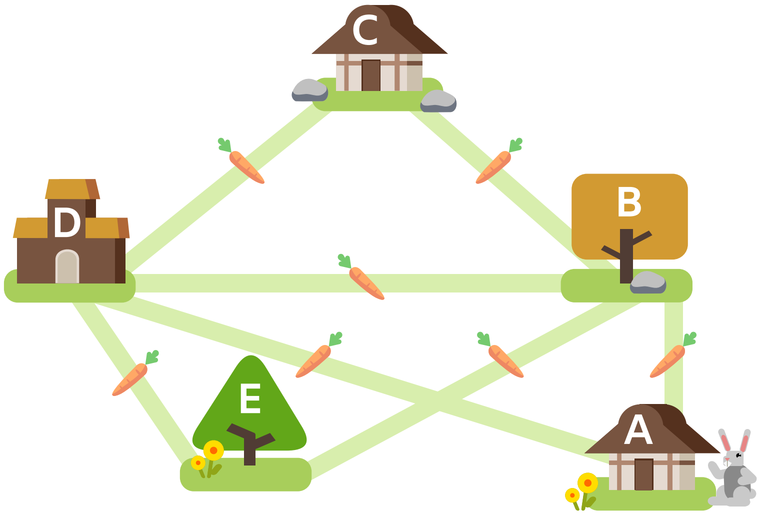
BACKGROUND INFORMATION

This task introduces the idea of the *binary system* to students. The data in human life can be represented in many ways and by many different counting systems; decimal, octal, etc. The computer uses the binary system to store data. Each digit is binary, represented by a digit 1 or 0. When the data from human life is transferred into the computer, it needs to be converted into the binary system.

Picking Up Carrots

The little rabbit's house is at A.

Starting at her house, she is going to pick up carrots. The carrots are shown in the picture below.



Question

The little rabbit wants to pick up all the carrots at the *fastest* speed and return to her house. Can you help her find an optimal route?

ABCDEBDA

ADEBADCBA

ABDABCDEBA

ADEBCDABA

EXPLANATION

Answer

The correct answer is A - ABCDEBDA.

Explanation

A There is a carrot on each road. The rabbit should pass each road exactly once if she wants to pick up all the carrots as quickly as possible.
B This route is longer than the A route, it contains one path more.
C The DE path was travelled twice and the rabbit could not pick up all the carrots.
D The DB path was not traveled so one carrot left.

BACKGROUND INFORMATION

In a *connected undirected* graph, a path from a point that passes through each route of the graph exactly once and then returns to the starting point is called an *Euler circuit*. When the number of nodes in an undirected connected graph is even, then there is an Euler circuit in the undirected graph.

How to find a Euler circuit?

Starting from any node (houses in this task), we have to find a circuit that leads back to this point. This method does not guarantee that each side (paths in this task) is travelled. If there is a node and an edge that is not travelled, then this point should be the starting point and this edge should be the starting edge, which is connected to the current circuit. This is done until all edges have been travelled. In this way, the whole graph is joined together.

Counting Out

Six Beavers play counting-out. They use a 16-part rhyme.

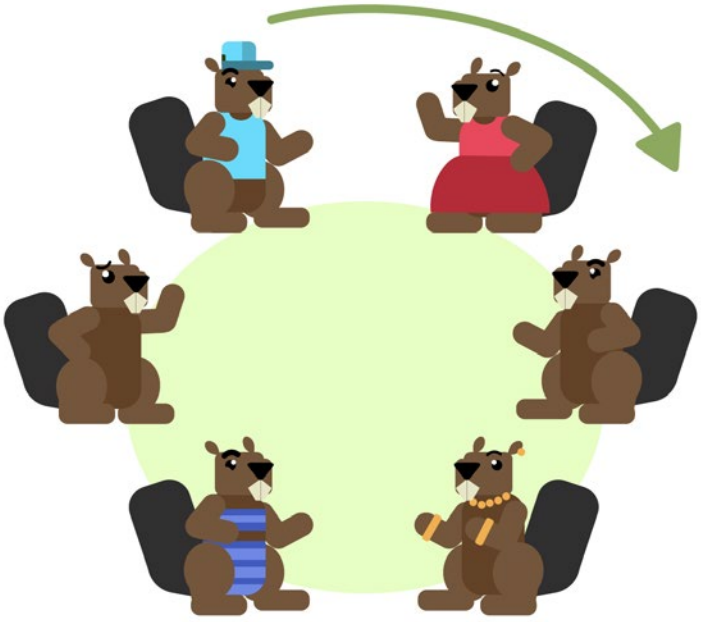
Eeny - meeny - minie - moe.
Catch a - tiger - by the - toe.
If he - hollers - let him - go.
Eeny - meeny - miny - moe.

They start at the beaver with a hat and count along the arrow. So they finish singing the 16 parts at the 4th beaver (with the jewellery) and she steps out.

Now there are only five beavers left and they start over, exactly the same way as before except this time starting the count from the next beaver after the one that got out.

Question

Which beaver will step out of the circle next?



EXPLANATION

Answer

The correct answer is "the beaver in the bathing suit".

Explanation

The simplest way to check it is to actually count out the 16-part chant. This shows the correct answer, as well as those that are wrong.

However, this is not the fastest way. As you count, you may notice that you regularly come back to the same beaver. When you count the same beaver as before, the number of remaining parts of the chant

Continued on next page

Counting Out – continued

is lower exactly by the total number of beavers in the circle. In other words, you could just subtract the number of beavers (5) from the number of parts (16) and save some counting! Now you only need to count to 11.

But why stop there? You can skip another round and subtract the number of beavers again, 11 – 5 = 6. Do this again for another round, 6 – 5 = 1. Now you can be sure that the counting stops at the beaver you first started the chant on – the beaver in the bathing suit. In summary, you just subtract the number of beavers from the chant length as long as you can.

If you want, you can be even faster: this repeated subtraction is just like calculating the remainder after division. That is, 16 divided by 5 leaves 1 as the remainder, which tells us immediately that the counting stops at the beaver you first started the chant on – the beaver in the bathing suit.

BACKGROUND INFORMATION

Computer Scientists like solving tasks of this kind: seeing a process that is slow and finding ways to make it work faster (to optimise it). If you can find a solution quickly in this case, you can even choose a place in the circle at the start of the game so that you won’t be eliminated.

Shortcuts with remainders are extremely useful. Their applications are found in cryptography (ciphering), where this technique helps prevent other people from looking at your online activity or data. This technique is also used in data checking, where the computer looks for potential corruption in received data.

This question is an example of *modular arithmetic*, using the *modulo* operation (*mod*), which returns the remainder when one number is divided by another. In this example, this is arithmetic modulo 5, and the operation 16 *mod* 5 = 1.

Another example of *modular* arithmetic is 24-hour time, using arithmetic *modulo* 12. We know that 1900 hours is 7pm because 19 *mod* 12 = 7, i.e. the remainder 7 tells us the hour, and since 19 is greater than 12 we know the time must be pm rather than am.

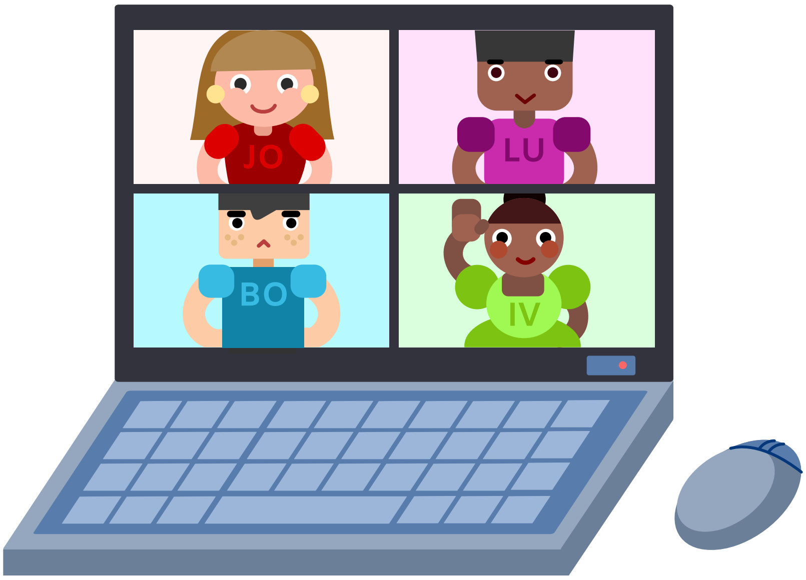
This question comes from Germany

Years 3+4
Years 5+6 Easy
Years 7+8
Years 9+10
Years 11+12

Looking in a Mirror

In online meetings you see your own image mirrored on your screen. Normally you do not notice this, because you are used to seeing yourself in the mirror.

Bo, Jo, Lu, and Vi meet in a video chat. You can recognise them by the names on their shirts.



Question

Whose screen are you looking at?

Bo | Jo | Lu | Vi

EXPLANATION

Answer

Answer D - Vi is correct. You are looking at Vi’s screen. Vi is the only name that is mirrored.

Explanation

An image of an object is inverted with the same size as the original body, except that it differs from it by being inverted. That is, the right is to the left and the left in the reverse image is the right of the body in front of the screen.

BACKGROUND INFORMATION

Computer Scientists deal with how information has to be represented in order to be processed by machines. But they also deal with human-machine interaction. Good software should function effectively and efficiently. And it should have a user interface with a clear information architecture and a well thought-out interaction design. The human user should find his way around easily. In online meetings, it just feels more natural to see yourself as if in a mirror.

This question comes from Turkey

Years 3+4
Years 5+6 Medium
Years 7+8
Years 9+10
Years 11+12

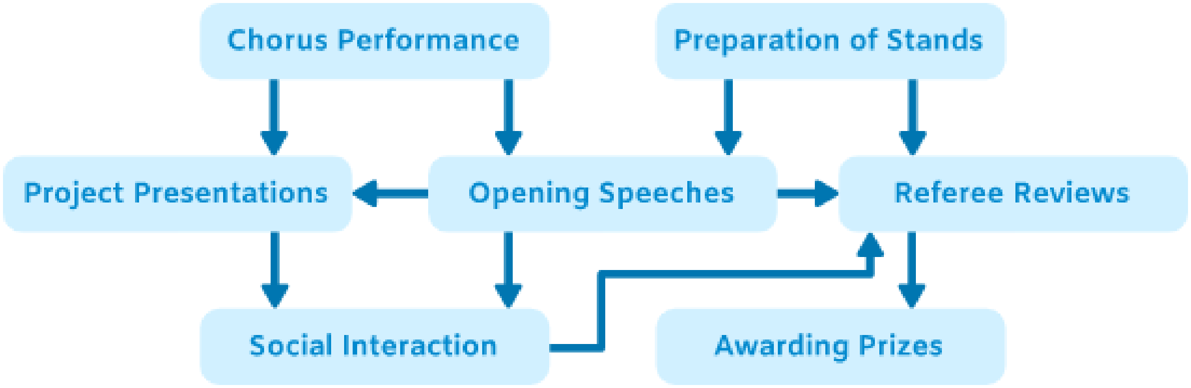
Science Fair

Bebras High School is having a science fair.

All the events should follow a specific order, and only one event can be held at a time.

The diagram shows all the events that must be included in the flow of a science fair. The arrows between events indicate that the event the arrow is drawn *from* has to occur *before* the event the arrow points *to*.

For example, the “Social Interaction” can only happen after both the “Opening Speeches” and the “Project Presentations” have finished.



Question

What is the correct order of events for the science fair? Drag the event into the correct order.

Awarding Prizes

Preparation of Stands

Social Interaction

Chorus Performance

Referee Reviews

Opening Speeches

Project Presentations

Continued on next page

Science Fair – continued

EXPLANATION

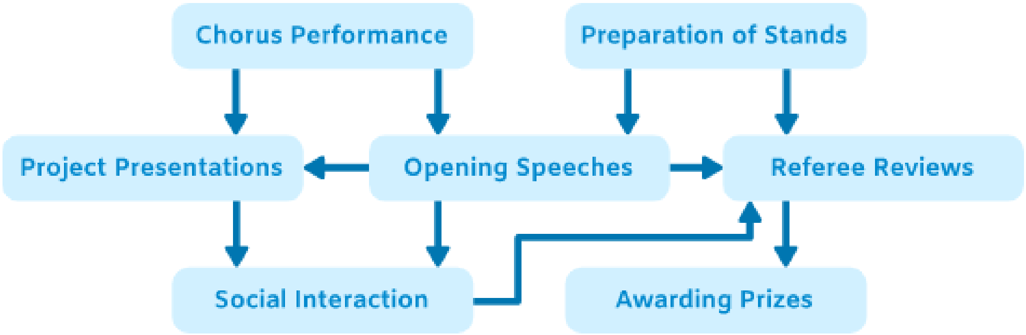
Answer

There are actually two possible correct answers in this task:

Option 1	Option 2
1. Chorus Performance 2. Preparation of Stands 3. Opening Speeches 4. Project Presentations 5. Social Interaction 6. Referee Reviews 7. Awarding Prizes	1. Preparation of Stands 2. Chorus Performance 3. Opening Speeches 4. Project Presentations 5. Social Interaction 6. Referee Reviews 7. Awarding Prizes

Explanation

One way to realise flowpattern is to start with an event that has not yet happened, and all the events pointing at it have already occurred. For example, at the beginning, “Chorus Performance” and “Preparation of Stands“ are the only events that have no other event pointing at them, so both of them can proceed in any order. Then, since “Opening Speeches” only had “Choir Performance” and “Preparation of Stands“ pointing at them, and since both events have already happened, “Opening Speeches” can now take place.



Thus, all the boxes and pointing arrows should be examined in this manner until we have a valid sequence of steps according to our diagram. Of course, there can be more than one right answer depending on the options.

BACKGROUND INFORMATION

Solving this task means performing a *topological sort* of the graph. In Computer Science, a topological sort or topological ordering of a directed 91 is a linear ordering. For instance, the vertices of the graph may represent tasks to be performed, and the edges may represent constraints that one must be performed before another. *Algorithms* are known for constructing a topological ordering of any directed acyclic graph in linear time.

This question comes from Austria

Forest Watch

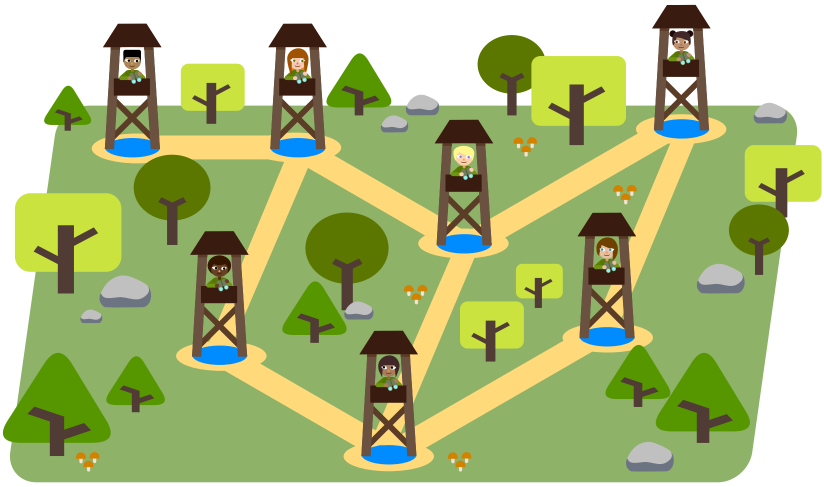
The forest rangers need to keep an eye on the animals that wander onto the paths.

They watch the paths from tall towers.

A ranger can only see the paths that connect directly to their tower.

Question

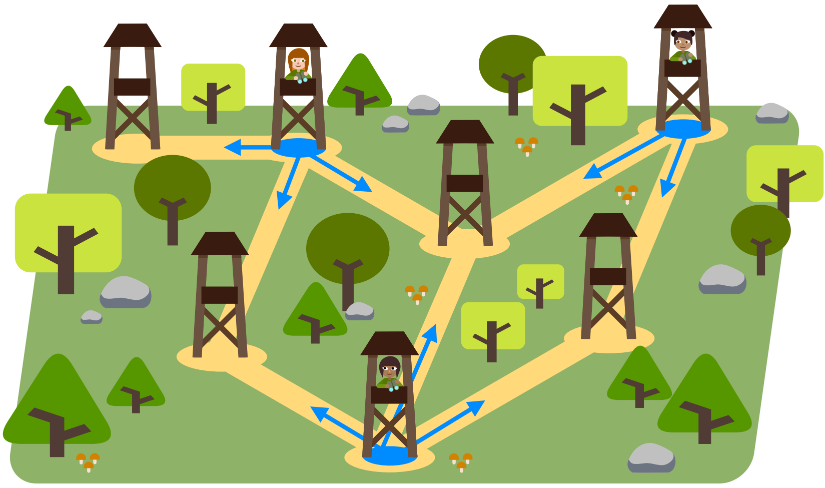
Click on the towers that must have a forest ranger in them so that all the paths are watched, but that requires as few rangers as possible.



EXPLANATION

Answer

The three occupied observation towers and their overseen paths are shown in the image below.



Explanation

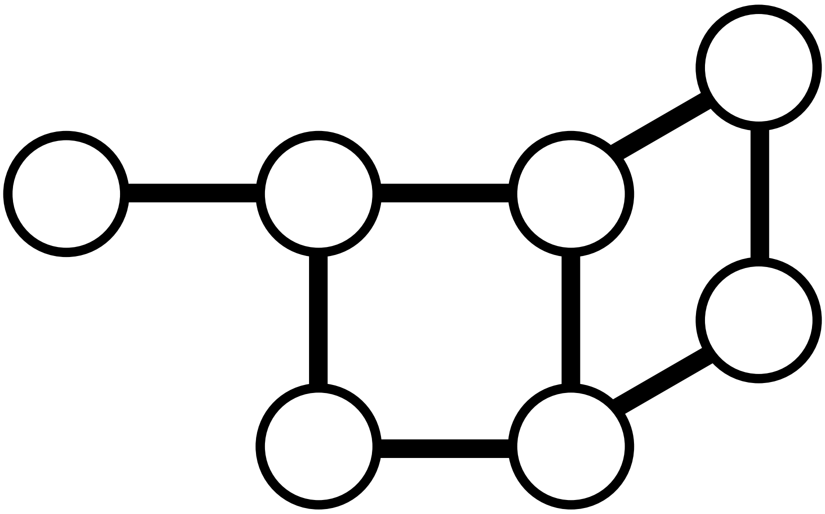
There are eight paths. If there were just two observation towers occupied, one would have to observe at least four paths. This is not possible, because no observation tower is next to four paths.

Continued on next page

Forest Watch – continued

BACKGROUND INFORMATION

In Computer Science many things can be represented with graphs. Graphs consist of *nodes* (= circles) and *edges* (= lines), which connect the nodes. For our example, the graph looks like this:



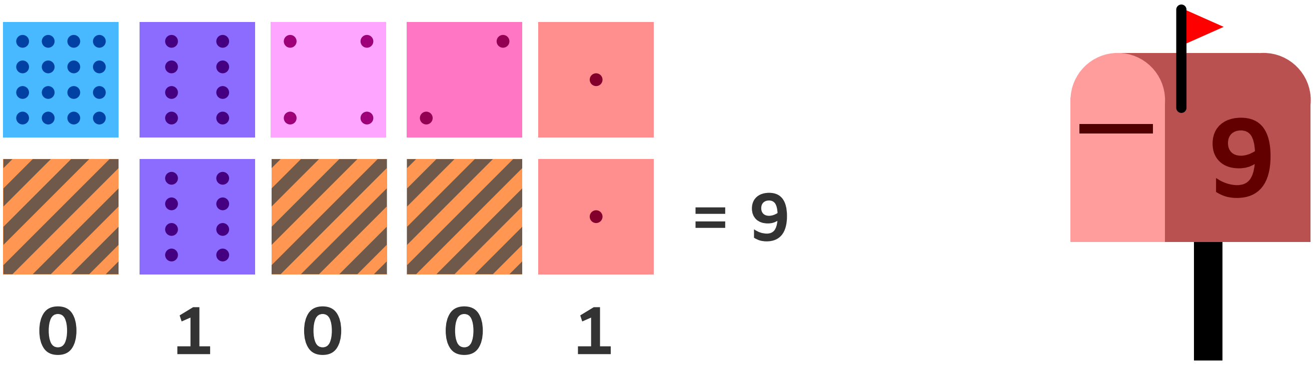
You can ask: “Which nodes (observation towers) do you have to choose at least such that every edge (forest path) is next to a chosen node (observation tower)?” This question is also known as *minimal vertex cover*. It can, for example, be applied when putting up street lights, which should illuminate all streets. Another example is placing cameras, which should cover all hallways.

This question comes from
Australia

Years 3+4
Years 5+6 Medium
Years 7+8
Years 9+10
Years 11+12

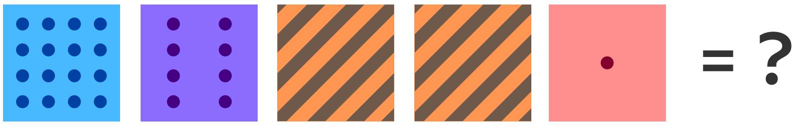
Mission Possible

Alice has been given a mission. She needs to deliver a top-secret package to a house, but first she must decipher the code to find the correct house number.



Question

Using the example above, what is the correct house number? Type your answer in the text box.



EXPLANATION

Answer

25.

Explanation

In the example, you can see we get to house number 9 by adding all dots together: 8 + 1 = 9.

Therefore, in the question we get to 25 by:

16 + 8 + 1 = 25.

BACKGROUND INFORMATION

Data can take many forms, for example, pictures, text or numbers. When we look at data in this question, we are looking for a sequence of images that will assist in solving the problem. By identifying these images we can make predictions, create rules and solve more general problems.

The binary number system plays an important role in how information is stored on computers. This questions helps students understand how binary can be used to represent information, such as digital images and numbers.

This question comes from
Germany

Years 3+4
Years 5+6 Medium
Years 7+8
Years 9+10
Years 11+12

Choices

The beaver family have five gifts for their young beavers, one gift for each.

Each young beaver tells the family, which gift would be its first choice and which would be its second choice.

The family wants to:

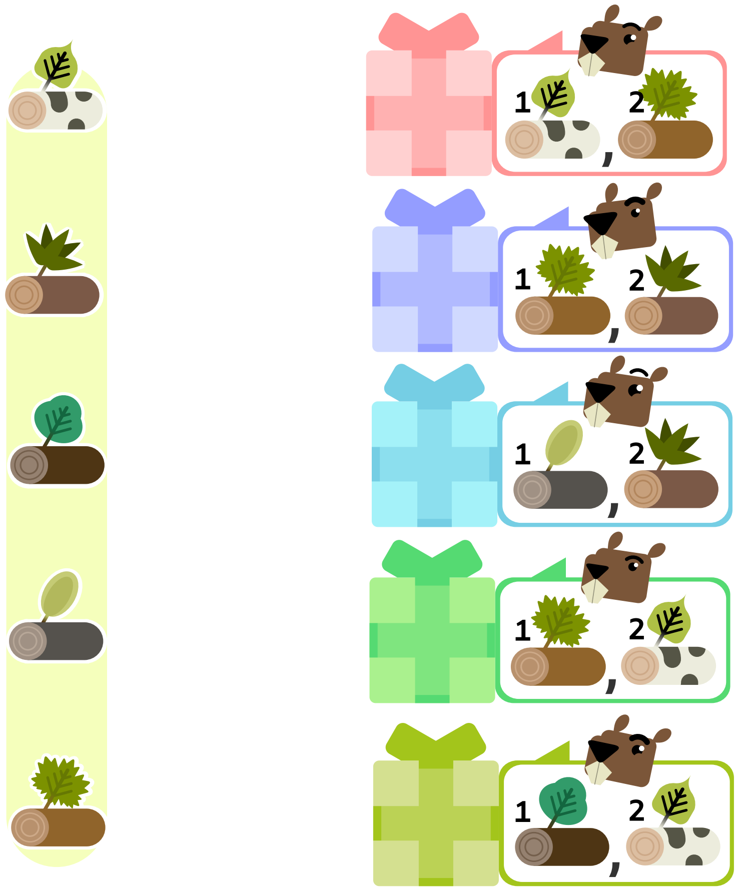
1) Give as many first choices as possible,

2) and then as many second choices as possible.

In the diagram below the gifts are shown on the left and the beavers with their choices on the right.

Question

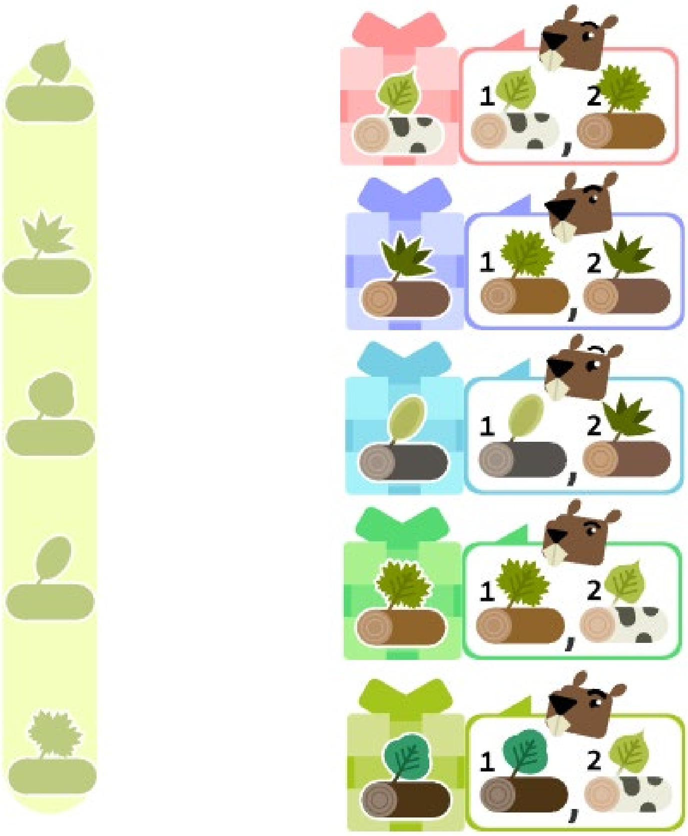
Assign the gifts to the beaver's present box.



Continued on next page

Choices – continued

Answer



Explanation

We cannot satisfy all first-place preferences, because two young beavers have the same first-place preference. The assignment shown above satisfies four first-place preferences, and one second-place preference. For this case, a better assignment is impossible. Note that if, going from left to right, top to bottom, you would assign the second gift to the second beaver (just like the first gift to the first beaver, thus satisfying the second beaver's first preference), you could not assign a preferred gift to the fourth beaver any more. This is why there is no other best assignment than the one shown above. That is, in this problem, it is not sufficient to be "greedy" and take the next best thing one by one.

BACKGROUND INFORMATION

The young beavers ranked the gifts according to their preferences. For each beaver you can consider all the gifts not mentioned to share the third place in its ranking. The beaver family wants to match gifts with young beavers in such a way that as many first choices are achieved as possible, then the most second choices possible.

A matching set is considered *optimal* if there is no other set with more first-place preferences satisfied or (if the same number of first-place preferences are satisfied) no more second-place preferences are satisfied – and so on. In Computer Science this is called *rank-maximal matching*. There are a wide variety of matching problems that have been studied by computer scientists.

Maze

The little witch Luna accidentally discovers a hidden treasure cave.

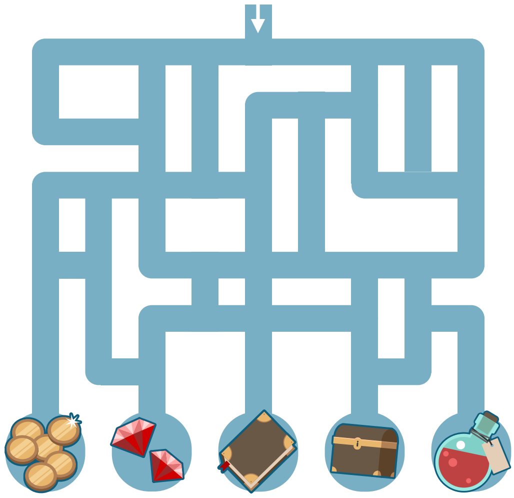
To get to the treasure, she needs to pass through a maze. Luna discovers that five exits correspond to five treasures: coins, rubies, a magic book, a treasure box, and a magic potion.

Luna did not know which treasure to choose, so she decided to follow the following rules to get through the maze:

- Go south (↓) as a priority;
- When there is no way to the south, go east (→);
- When there is no way to the south or east, go west (←);
- Never turn back and do not go north (↑).

Question

What treasure will Luna find at the end?



EXPLANATION

Answer

The treasure she would find is the chest.

Explanation

When little witch Luna enters the cave she has to fly down according to the first rule. At the first fork, she can choose to go left or right. According to the second rule, she needs to right. At the second fork she has to fly down again because this is her priority. For forks three to six the first two rules apply. At fork seven there is not way down or to the right. Therefore she has to go to the left. By applying all four rules she finally comes to the treasure chest.

BACKGROUND INFORMATION

In Computer Science *conditional statements* are used to executes different codes/actions according to certain conditions or rules. IF the conditions apply, then a code/an action is executed, ELSE other codes/actions are executed. In this task we have three rules. IF you reach a junction and there is a fork down go down. Otherwise, IF there is a fork to the right, go right. ELSE got to the left. Little witch Luna never has to fly up.

This question comes from Canada

Years 3+4
Years 5+6 Hard
Years 7+8
Years 9+10
Years 11+12

Crested Birds

Crested birds don't build nests. Instead, they move into empty nests.

When a crested bird finds a tree with empty nests, it moves into a nest as follows. It starts at the bottom of the tree and repeats the following steps until it finds an empty nest:

1. It goes up until it finds a nest.
2. If the nest is *empty*, it moves into the nest. Done.
3. If the nest is *occupied*, the bird looks at the other crested bird in the nest:

   - If the other bird has **more circular crests** on its head than it does, the bird continues **up and to the left**.

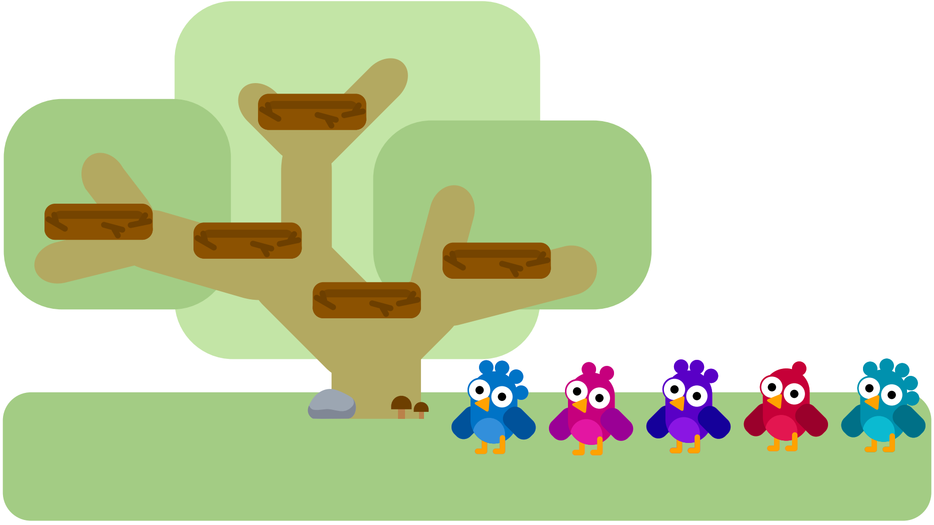
   - If the other bird has the **same number or fewer** circular crests, the bird continues **up and to the right**.

There is a tree with five empty nests, and there are five crested birds.

The birds move into the empty nests in the order they are standing, from left to right; the bird with four circular crests is the first.

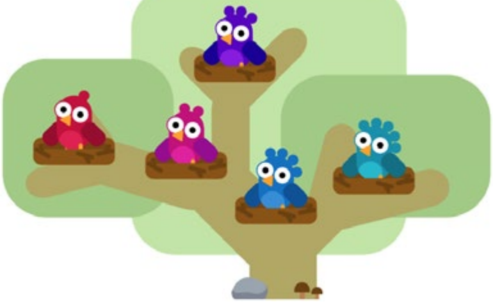
Question

Drag each bird into the correct nest following the steps above.



EXPLANATION

Answer

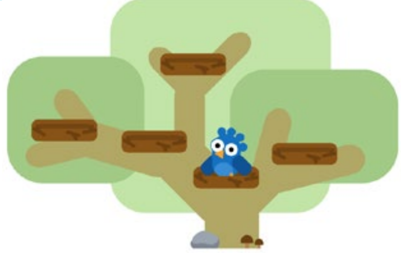
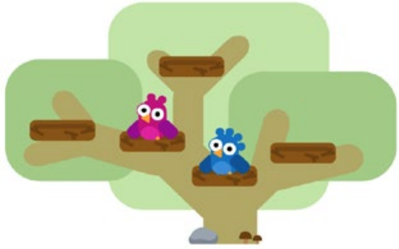
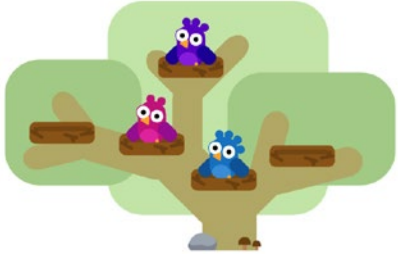
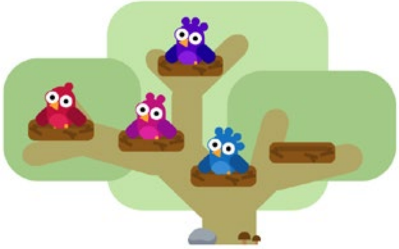
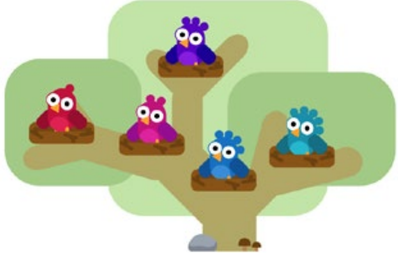


Continued on next page

Crested Birds – continued

Explanation

This is the way to find the solution:

The first bird, with four circular crests, moves into the lowest nest.	
The second bird has two circular crests. The lowest nest is occupied by the first bird, with four circular crests. Since four is more than two, the second bird continues to the left and moves into the next empty nest.	
The third bird has three circular crests. Since four is more than three, at the lowest nest the bird continues to the left. The next nest is occupied by the second bird, with twocircular crests. Since two is less than three, the bird continues to the right, and moves into the next empty nest – which is the highest nest.	
The fourth bird has four circular crest. Like the birds before, at the lowest nest, the bird continues to the left. At the next nest it must continue to the left, and moves into the next empty nest, at the very left.	
The last bird has five circular crests. It has to go right at the lowest nest and finds the next nest (at the very right) empty, where it moves in.	

BACKGROUND INFORMATION

Assigning birds to nests in this way has an interesting advantage. It makes finding particular birds quite efficient. If the bird you are looking for has fewer feathers than the current bird being observed, look in the left portion of the tree. Otherwise, check the portion of the tree to the right. By repeatedly splitting the tree in half, the bird you are looking for can be quickly found.

This structured way of organising data is called a binary search tree. It is often used in computer applications when it is necessary to be able to quickly retrieve data.

This question comes from Switzerland

Years 3+4
Years 5+6 Hard
Years 7+8
Years 9+10
Years 11+12

Strawberry Thief

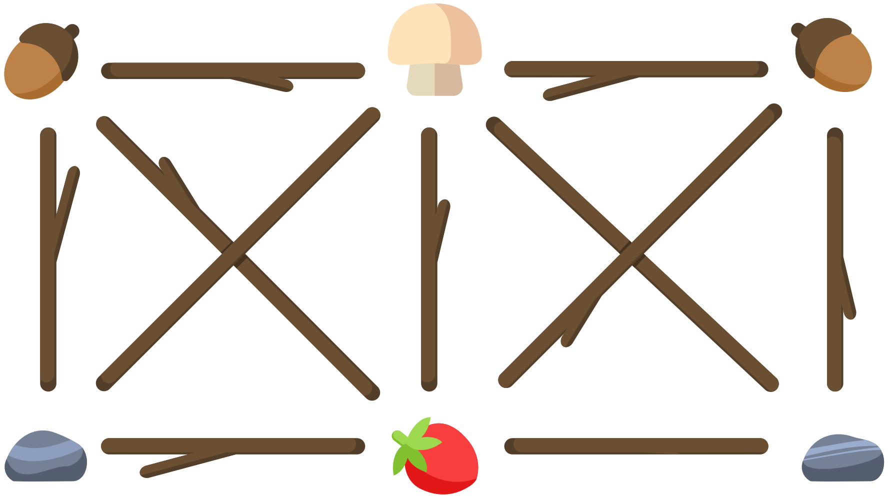
Anja makes a design using four types of objects: acorns, mushrooms, pebbles, and strawberries.

She then adds sticks to her design according to her 'Very Important Rule'.

Very Important Rule:

A stick can only go between two objects if the two objects are different types.

Here is Anja's completed design:



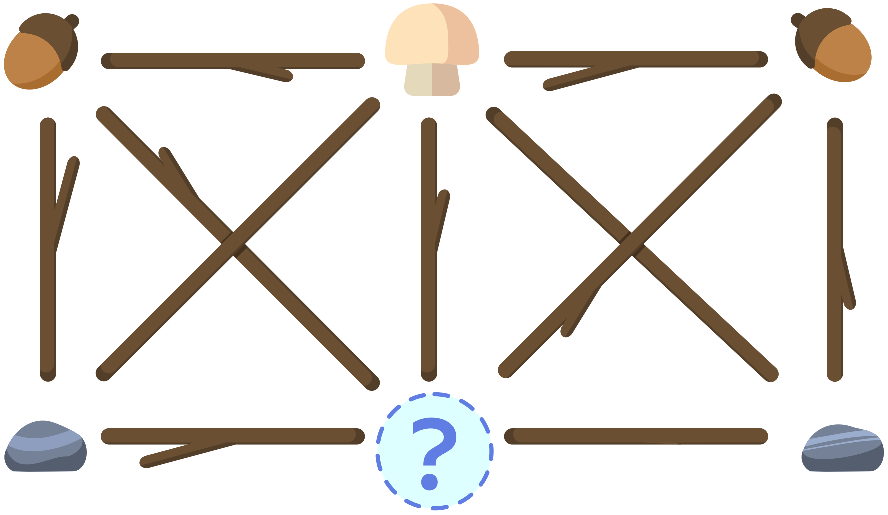
Anja's sister Zoë sees the design and eats the strawberry!

To hide what she has done she replaces the strawberry with a different type of object.

She also removes **exactly one stick** so that the 'Very Important Rule' will not be broken.

Question

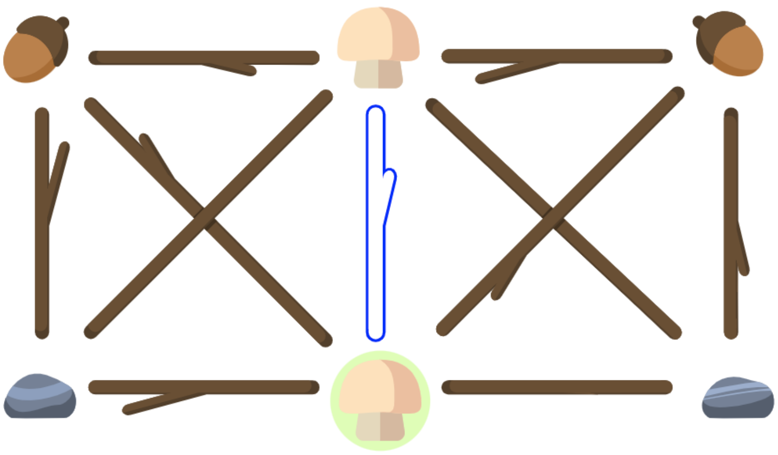
Click on the stick Zoë removes and click on the question mark until you see the object she replaces the strawberry with.



Continued on next page

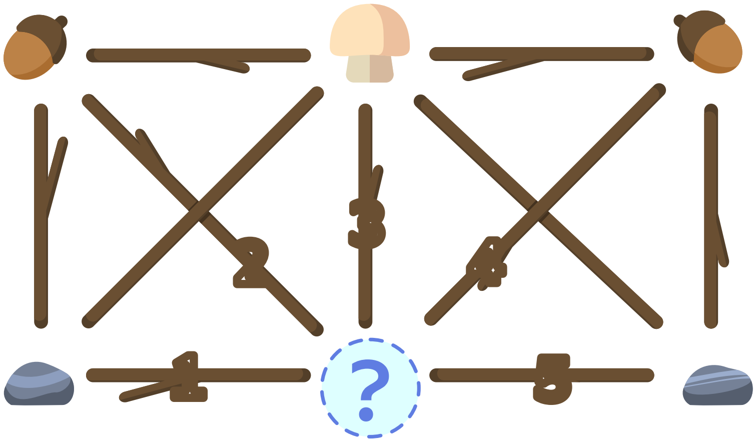
Strawberry Thief – continued

Answer



Explanation

Zoë replaced the strawberry with a mushroom and removed the stick labelled 3 since it violates the 'Very Important Rule' by connecting two objects of the same type.



All other strawberry replacements would require Zoë to remove more than one stick.
If Zoë replaced the strawberry with an acorn she would have had to remove sticks 2 and 4.
If Zoë replaced the strawberry with a stone she would have had to remove sticks 1 and 5.

BACKGROUND INFORMATION

Anja's design can be called a *graph*. The objects can be called nodes and the sticks can be called edges. In a graphs, edges connect nodes. Two nodes that share an edge are called *neighbours*.

A subset of nodes where each node is a neighbour of every other node in the subset is called a *clique*. Anja's design contains two cliques of size four: the left half and the right half of the design.

Now suppose you wanted to assign the nodes of a graph a colour so that no edges connect two nodes of the same colour. It turns out that the number of colours needed to do this is at least the size of the largest clique.

Removing the strawberry is like trying to colour Anja's graph using at most three colours. This is not possible with cliques of size four, which is one reason why Zoë needed to remove a stick as well.

The problem of how to colour a graph using the minimal number of colours has many applications. Some examples include scheduling sports competitions, designing a seating plan, and even solving a Sudoku puzzle.

This question comes from Iceland

Years 3+4
Years 5+6 Hard
Years 7+8
Years 9+10
Years 11+12

The Present

Bella's mum bought a present and locked it in a safe.

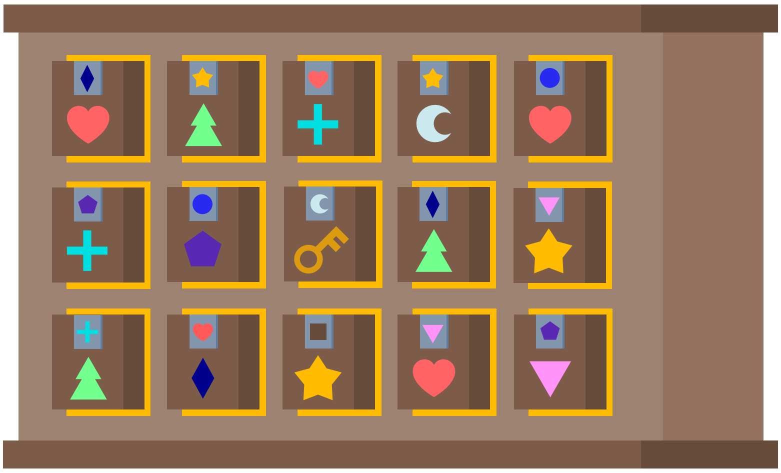
She gave Bella a blue disc and said: "You can have the present if you can solve the puzzle and get the key that is in the middlemost drawer."

To open a drawer, Bella must put an object of the correct shape in the key hole on the drawer. Then the drawer opens and she can get the object inside which is shown on the front of the drawer.

Question

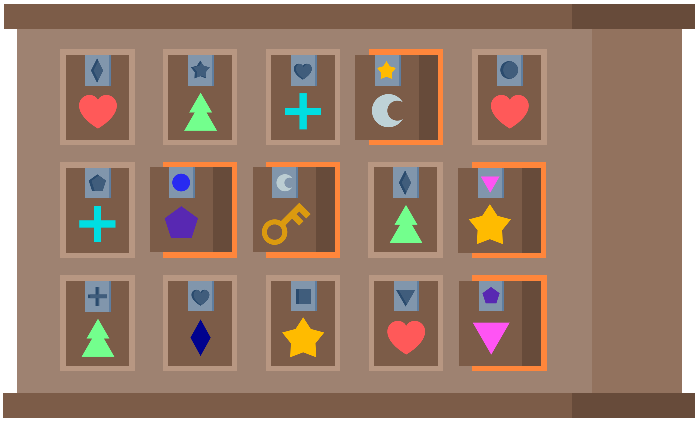
Help Bella get to the key.

Click on the drawers that Bella must open to get the key.



EXPLANATION

Answer

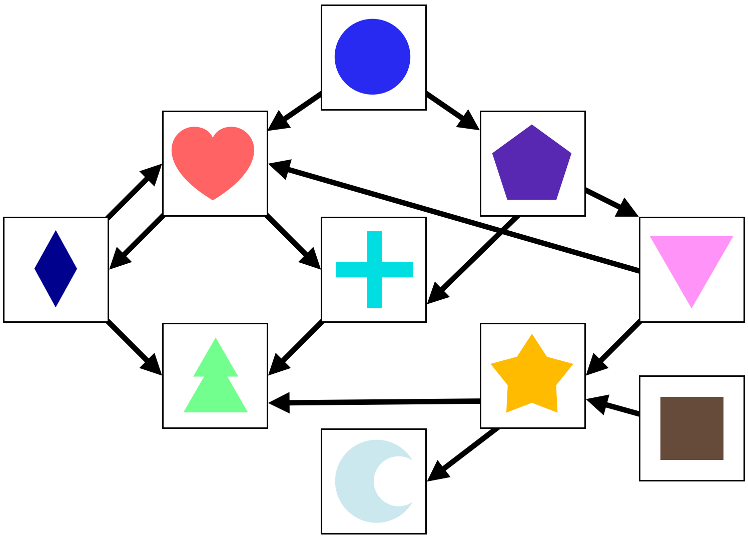


Continued on next page

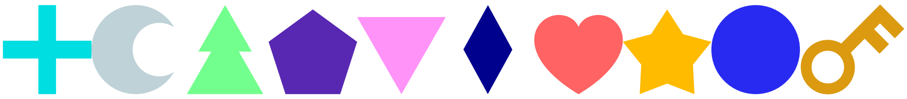
The Present – continued

Explanation

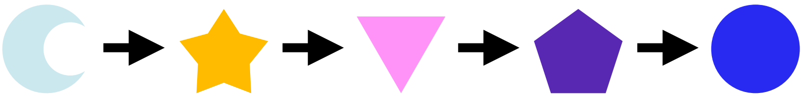
The task is to find a path from the blue disc to the moon, as the keyhole on the drawer that contains the key is in the shape of a moon.



To find the solution it helps to draw the objects and arrows (*directed graph*) to represent what object is needed to get the object from a drawer.



You can use *backtracking* from the moon to the circle to help you to find the correct path.



BACKGROUND INFORMATION

A *graph* is a diagram or pictorial representation of connections between two or more things.

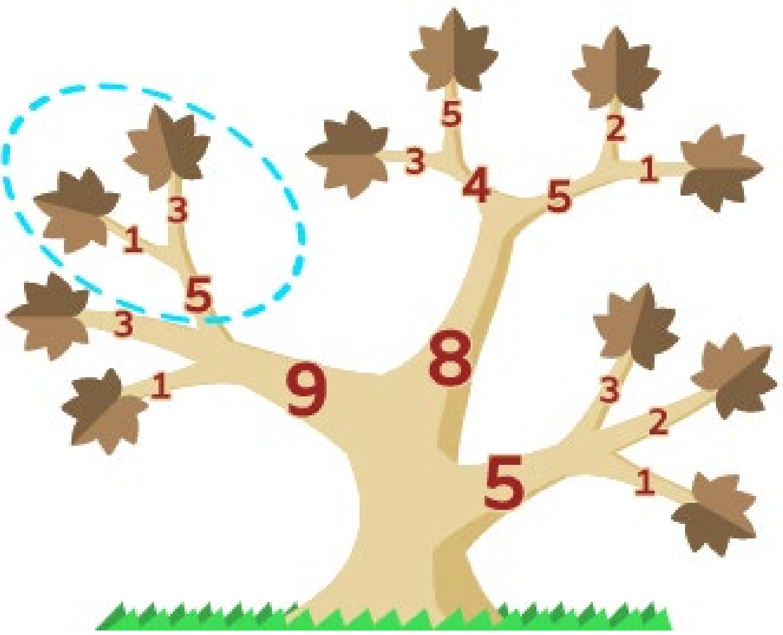
The objects and drawers describe a *directed graph*: Each object is a *vertex* connected by *edges* to represent relationships between objects and what object gives another object from the drawers.

Representing information using a graph helps to see the structure of a task.

In graphs it is also possible to use *backtracking* to find the solution. It is also important to be able to recognise patterns and shapes.

Tree Pruning

Beaver Bruno has a tree in his garden. Unfortunately, the tree has a disease and all its leaves died and turned brown. Now Bruno needs to cut off all the branches with the dead leaves. Then the tree can grow new branches with healthy leaves.



In the picture, the numbers show the time needed to cut each branch.

When Bruno cuts a larger branch (for example, the branch marked with 5 inside the blue ellipse), all the leaves attached to it fall down and then Bruno does not need to separately cut all the smaller branches (the ones marked with 1 and 3 inside the ellipse).

Question

What is the *shortest* time Bruno needs to cut off all branches with dead leaves?

19

20

22

25

EXPLANATION

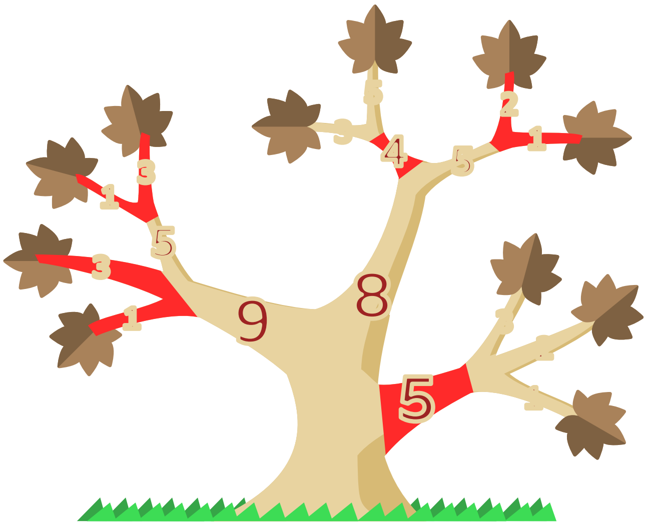
Answer

20

Continued on next page

True Pruning – continued

Explanation



We can view the tree (the plant) as a data structure that is also called tree. A tree data structure is a type of another data structure called a *graph*. With this interpretation, the task is to find the minimal weight of edges separating all the leaves from the root. This is a standard problem in graph theory known as the *minimal cut* (or min-cut for short) and can be solved by various standard algorithms (Ford-Fulkesron algorithm, for example).

However, there is a much simpler special algorithm for finding min-cut between the leaves and the root of a tree. We can start moving from leaves towards the root and recalculate for each edge whether it's optimal to cut this edge or not.

Let's illustrate the process:

Initially we start from leaves. For each leaf we cut the branch it is on, thus the initial estimate is:

1 + 3 + 1 + 3 + 3 + 5 + 2 + 1 + 3 + 2 + 1.

Now we move towards the root. In each move we can either leave the "old" cuts, or replace them by cutting a single larger branch.

The second iteration:

1 + 3 + min(5, 1 + 3) + min(4, 3 + 5) + min(5, 2 + 1) + min(5, 3 + 2 + 1) =

1 + 3 + 4 + 4 + 3 + 5.

The third iteration:

min(9, 1 + 3 + 4) + min(8, 4 + 3) + 5 = 8 + 7 + 5.

Now we reached the root of the tree, hence the final answer is 8 + 7 + 5 = 20.

BACKGROUND INFORMATION

Trees are important *data structures* in Computer Science. Moreover, there are some contemporary algorithms that use trees, like decision trees in random forest.

Finding the *minimal cut* of a directed graph allows us to also find the *maximum flow* in this graph, which is widely used in logistics. It helps to determine the maximum weight of goods that can be transported from the factories to other countries, taking into account all means of transportation between major cities and their capabilities.

This question comes from Ireland

Years 3+4
Years 5+6 Hard
Years 7+8
Years 9+10
Years 11+12

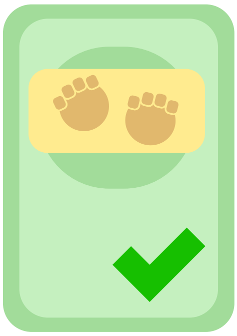
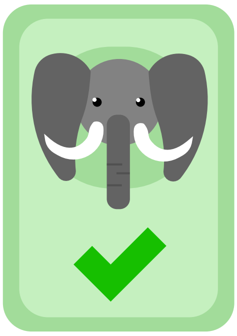
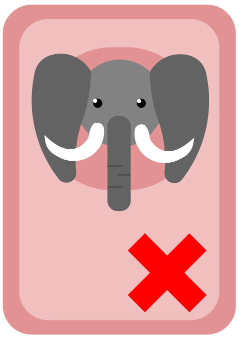
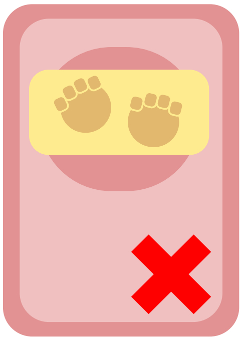
Elephants in the Fridge

Each day Joanna’s dad gives her a double sided card to tell her what happened in the fridge over night.

One side tells Joanna whether there was an elephant in the fridge.

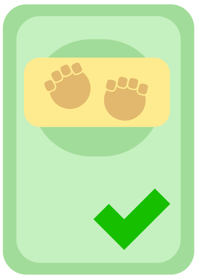
The other side tells Joanna if there were footsteps in the butter.

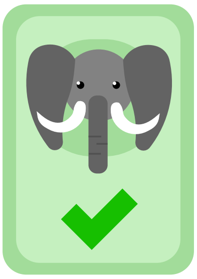
The cards from the last four days are laid out on the table as shown below:

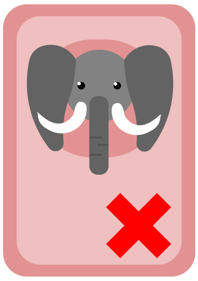
			
Day 1 **There are footprints in the butter.**	**Day 2** **An elephant has visited the fridge.**	**Day 3** **No elephant has visited the fridge.**	**Day 4** **There’s no footprints in the butter.**

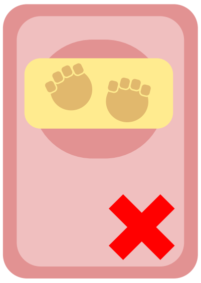
Question

Select all the cards that Joanna can turn over to give her a chance of proving her dad wrong.





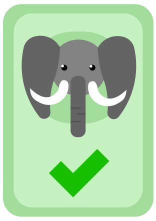




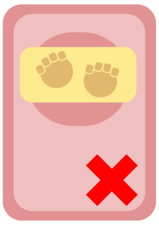
EXPLANATION

Answer

The correct cards to turn over are:



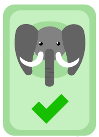
and

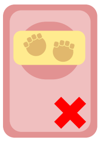


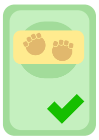
Continued on next page

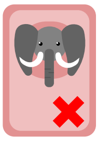
Elephants in the Fridge – cont’d

Explanation

If Joanna flips , and the other side says there are no footprints in the butter, then she has successfully proven her dad wrong. Of course, the other side might say there are footprints in the butter, but at least turning this card gives Joanna a chance to disprove her dad’s claim.

Similar reasoning can be made for the card with  showing. If Joanne flips this one over and it shows an elephant was in the fridge then this again proves her dad wrong, so it is also worth turning this card over.

If Joanna flips , it doesn’t matter what the other side says. If the other side says an elephant has visited, it reaffirms her dad’s claim. If the other side says no elephant has visited, then it doesn’t contradict her dad’s claim since he said nothing about what would happen if no elephant has visited the refrigerator.

Similar reasoning can be made for the card with  showing. Her dad did not say what would happen if an elephant did not go in the fridge, so turning this card over is not going to prove him wrong.

BACKGROUND INFORMATION

Logic plays a critical part in the theory and application of Computer Science, from simple Boolean logic to the forms of logic used in modern Artificial Intelligence systems.

The statement of this task, made by Joanna’s dad is called an *Implication* (“If A then B”, often written as “A B”).

Implications, represented with the symbol ▯, are often used in Expert Systems such as Prolog. These systems were some of the first generation of successful Artificial Intelligence systems.

Computers can represent such statements internally and combine them with other statements that are supplied – essentially performing mathematics on information – to calculate new or surprising results. Implication can be represented as a binary operation and, as such, can be represented in a truth table much like AND, NOT, and OR:

A	B	A ▯B
False	False	True
False	True	True
True	False	False
True	True	True

This truth table is somewhat surprising to many people because it uses “if A then B” in the (different) meaning of “if and only if A, then B” which means “exactly when A is true then and only then, B is also true”. Computer Scientists sometimes write “iff” instead of “if and only if” if they want to use this meaning. But the original statement “if A then B” is always true except when A is true and B is false.

We would like to thank the International Bebras Committee and community for their ongoing assistance, resources and collaborative efforts. Special thanks to Eljakim Schrijvers, Alieke Stijf and Dave Oostendorp for their support and technical expertise.

If you would like to contribute a question to the International Bebras community, please contact us via the details below.

Contact us

CSIRO Digital Careers
digitalcareers@csiro.au
csiro.au/Digital-Careers

Australia's National Science Agency

CSIRO digital careers