

Bebras Australia Computational Thinking Challenge

2021 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

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.

digitalcareers.csiro.au

523

Australian schools participated in Bebras R1 2021



32 311

Australian students participated in Bebras R1 2021



2.9 million

students participate worldwide



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 2021. 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 a Solutions Guide?	3
What is Computational Thinking?	5
Computational Thinking skills alignment	6
Digital Technologies curriculum key concepts	7
Digital Technologies key concepts alignment	8

Years 3+4

Teddy Bear Hunt	10
Footprints	11
Creatures Zoo	12
Bebras Tail	13
Towers of Blocks	14
Bugs on Branches	15
Train Tracks	16
Flying Bees	17
Polly's Pets	18
Dotted Blocks	20
Object Sorter	21
Gifts	22
Bowls	23
Where Do They Live?	24
Beaver Salon	25

Years 5+6

Sprinklers	27
Quiz Card	28
House Numbers	29
Treasure Island	30
Towers of Blocks	31
Theatre Performance	33
Heaviest Box	35
Connect the Dots	36
Doors and Stones	38
A Tree Structure	40
Painting the Houses	41
Museum Tour	42
Mountain Climber	44
Backward Bug	45
Bominos	46

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

2021 Round 1 Questions	Grade level	Decomposition	Pattern Recognition	Abstraction	Modelling & Simulation	Algorithms	Evaluation
Years 3+4							
Teddy Bear Hunt	Easy						
Footprints	Easy						
Creatures Zoo	Easy						
Bebras Tail	Easy						
Towers of Blocks	Easy						
Bugs on Branches	Medium						
Train Tracks	Medium						
Flying Bees	Medium						
Polly's Pets	Medium						
Dotted Blocks	Medium						
Object Sorter	Hard						
Gifts	Hard						
Bowls	Hard						
Where Do They Live?	Hard						
Beaver Salon	Hard						
Years 5+6							
Sprinklers	Easy						
Quiz Card	Easy						
House Numbers	Easy						
Treasure Island	Easy						
Towers of Blocks	Easy						
Theatre Performance	Medium						
Heaviest Box	Medium						
Connect the Dots	Medium						
Doors and Stones	Medium						
A Tree Structure	Medium						
Painting the Houses	Hard						
Museum Tour	Hard						
Mountain Climber	Hard						
Backward Bug	Hard						
Bominos	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

2021 Round 1 Questions	Abstraction	Data Collection	Data Representation	Data Interpretation	Specification	Algorithms	Implementation	Digital Systems	Interactions	Impacts
Years 3+4										
Teddy Bear Hunt										
Footprints										
Creatures Zoo										
Bebras Tail										
Towers of Blocks										
Bugs on Branches										
Train Tracks										
Flying Bees										
Polly's Pets										
Dotted Blocks										
Object Sorter										
Gifts										
Bowls										
Where Do They Live?										
Beaver Salon										
Years 5+6										
Sprinklers										
Quiz Card										
House Numbers										
Treasure Island										
Towers of Blocks B										
Theatre Performance										
Heaviest Box										
Connect the Dots										
Doors and Stones										
A Tree Structure										
Painting the Houses										
Museum Tour										
Mountain Climber										
Backward Bug										
Bomino's										

Bebras Challenge 2021 Round 1

Years 3+4



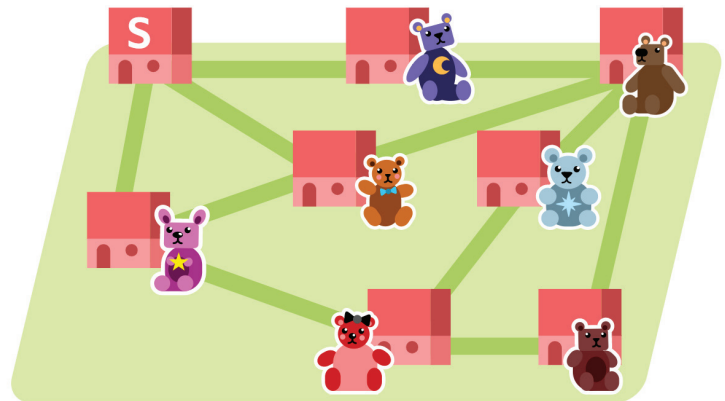
Teddy Bear Hunt

The beaver family went on a teddy bear hunt around their town (*shown on the map*).

They started from their home (*marked S*).

They walked along the roads and went back home. Along the way, they took pictures of the teddy bears that they saw.

When they got home, they saw that they only had 3 pictures!



Question

Which teddy bear did they forget to take a picture of?

Select the teddy bear that they forgot to take a picture of.

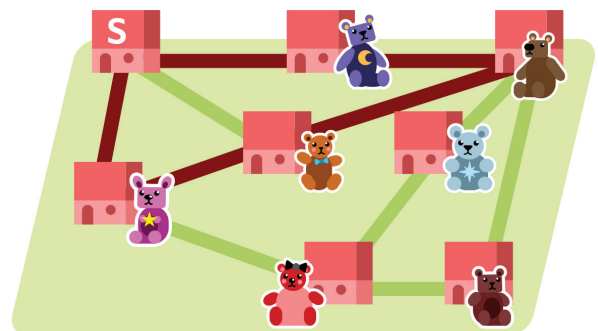


Answer

The correct answer is C.



On the picture you can see the path they took in red. It is the only path that contains the 3 bears from pictures, leads the beavers home, and contains 4 bears.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Modelling and Simulation, Algorithms

Concepts: Data Collection, Data Interpretation, Specification, Algorithms

Finding a route is a common problem in Computer Science. To solve this task you need to follow the routes and find out which route meets the conditions (4 stops). Although this task is very simple, it can become harder to solve as the map expands.

The map is a representation of a planar graph where the combination of nodes are houses and edges are the roads between houses. We chose four nodes — home and the three houses with the supplied pictures — and the task is to determine the node that must be on any route consisting of four nodes that are not home. It is a combinatorial problem on a graph. It is like drawing a map of the route from home to school, then to the swimming pool, to the shop and back home.



Footprints

There are 4 robot animals in a store.

One robot animal secretly walked around the store last night.

It left a trail of footprints on the floor.



Question

Select the robot who left the footprints.

Answer

The rabbit robot.

The rabbit robot is the only one with a paw made up of this shape:



It's Computational Thinking

Computational Thinking Skills: Decomposition, Pattern Recognition

Concepts: Data Interpretation

Solving this task firstly requires identifying the feet in each of the proposed animal robots, and secondly checking which ones match the footprints.

In doing so, one can ignore all non-foot body parts of the animals. This process is called abstraction, which is used a lot in Computer Science, and is an important part of a set of skills known as Computational Thinking. One side of abstraction is about selecting the information necessary to solve a given problem. Here, it is about seeing that only the shape of the feet is relevant.

When writing programs, programmers do this constantly: decide what to take into account from the problem they are solving. This is also part of a process called modelling, which in Computer Science is about deciding what information a computer will need to have and in what form in order to solve a given task.

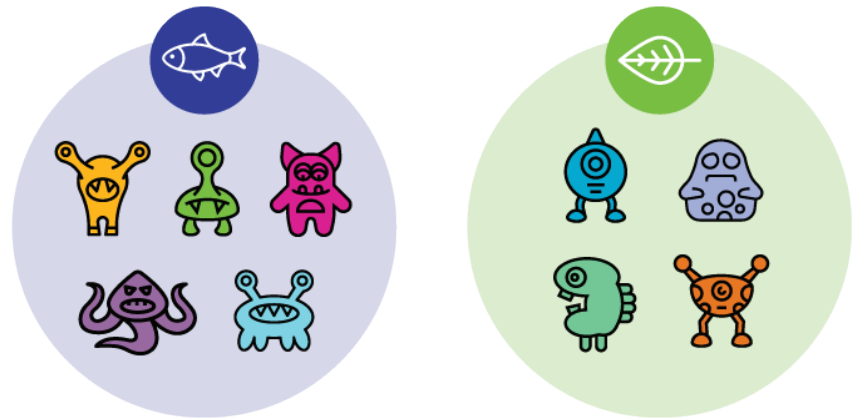
Realising that the footprint matches a given animal's foot is done by pattern recognition while comparing the images. Humans are pretty skilled at recognizing what an image is about (for instance, if there is a cat in a picture or not), but it is a rather difficult task for computers. Machine-learning techniques have been developed to make computers able to solve such tasks.



Creatures Zoo

A zookeeper wants to allocate creatures to different enclosures in groups of three different creatures.

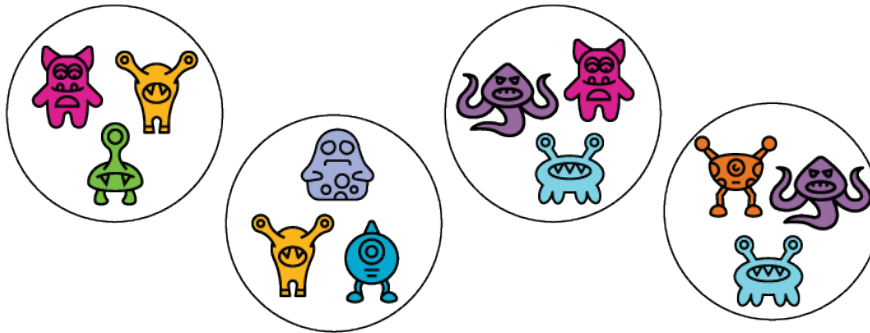
Some creatures only eat fish, others only eat plants. Because there is too much vegetation, there has to be at least one plant-eating creature in each enclosure.



Question

The zookeeper has arranged the creatures as follows.

Select the enclosures where they forgot to put a plant-eating creature:



Answer

The correct answer is that the zookeeper forgot to put a plant eater in the first and third enclosures. The two other enclosures have at least one plant-eating creature each:

- The second enclosure has a purple plant eater (and also a blue plant eater).
- The fourth enclosure has an orange plant eater.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Pattern Recognition, Algorithms

Concepts: Data Representation, Data Interpretation, Specification, Algorithms

Tasks of this type can always be solved quickly, but if it was asked in a slightly different way (pick the diets for the creatures so that each enclosure has at least one plant eater) then it would be an NP-complete problem known as SAT.

NP-complete problems are typical of the simplest problems that we don't know how to solve quickly with a computer. Some other problems are provably hard, such as computing the next best move in a game of chess. Chess is so hard that we never try to program a computer to find the exact answer quickly, only an answer that is "good enough" quickly. Other problems are relatively easy for a computer, such as sorting a list, and here we can find the exact answer quickly. However, NP-complete problems are a real headache for computer scientists, because although we suspect they are hard, we have no proof that it is impossible to solve them quickly.

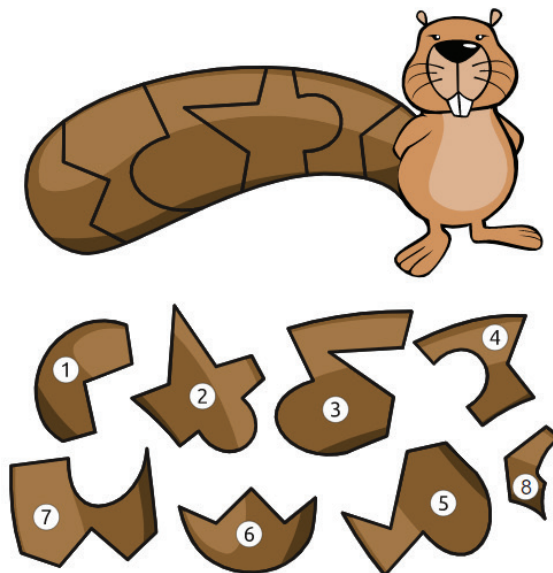
The way this problem is asked can be seen as checking the answer to a SAT problem. This is a very common thing to do in the analysis of NP-complete problems called verifying (or, "checking the answer to") an NP-complete problem. Checking the answer to NP - complete problems can always be done quickly, which adds to the mystery about whether NP - complete problems are provably hard or not.



Bebras Tail

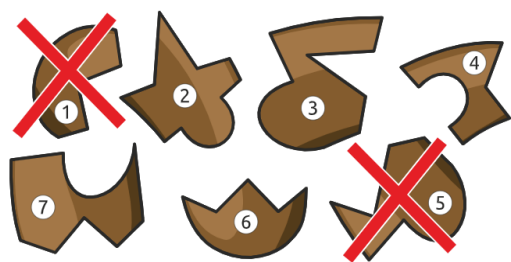
Question

Which shapes do not belong to the tail?



Answer

1 and 5 do not belong to the tail.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Pattern Recognition, Modelling and Simulation, Evaluation

Concepts: Data Interpretation, Specification

In this task we need to use pattern recognition – to analyse and recognize differences between shapes. For solving the task it is necessary to test each part to see if it will fit in the tail. A match is found based on the size and shape of the edges. Each position must be tested to be satisfied that a shape does not belong. This is an example of exhaustive search.

Pattern recognition is the automated recognition of patterns and regularities in data. It has applications in statistical data analysis, signal processing, image analysis, information retrieval, bioinformatics, data compression, computer graphics and machine learning.

In Computer Science, brute-force search or exhaustive search, is a very general problem-solving technique that consists of systematically listing all possible candidates for the solution and checking whether each candidate satisfies the problem's statement. It can sometimes be a challenge to make sure that no candidate is missing from the list.

The main strength of exhaustive search is that it is guaranteed to find the solution if there is one. If there are multiple solutions, exhaustive search will find all of them. Exhaustive search can take too long as a strategy for solving some problems.



Towers of Blocks

Sam, the little beaver, is playing with his toy blocks. He built three beautiful towers, each one made with a pile of blocks of the same size:

He noticed that one of the towers is higher than the other two. He now wants to add more blocks to the other two towers to make them as high as the tallest one, but he is not sure how many blocks he will need.



Question

Without moving any of the blocks shown in the figure, what is the smallest number of blocks Sam needs to add in order to make all towers of the same height?

1 block

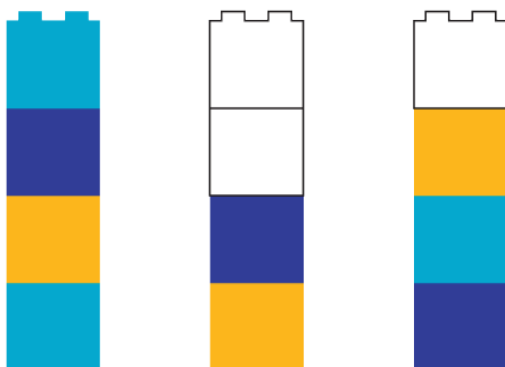
2 blocks

3 blocks

4 blocks

Answer

The correct answer is 3 blocks. The following figure indicates those 3 blocks in white.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Modelling and Simulation, Algorithms

Concepts: Abstraction, Data Representation, Data Interpretation, Specification

Each pile of blocks is what we call a stack in Computer Science, with two possible operations: push (putting new blocks on the top of the stack) and pop (removing the topmost blocks). In this problem, we are only allowed to make push operations and need to take into account the current height of each tower already built.



Bugs on Branches

While exploring in a park, some children discovered 4 creatures on 4 different tree branches.

This is what they found:

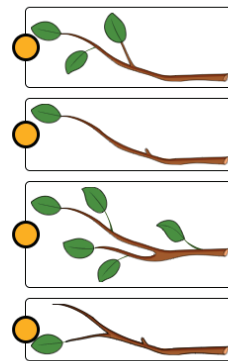
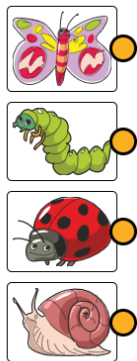
The branch with the caterpillar has the most leaves.

The branch with the butterfly has more leaves than the branch with the snail.

The branch with the ladybug has exactly 1 leaf.

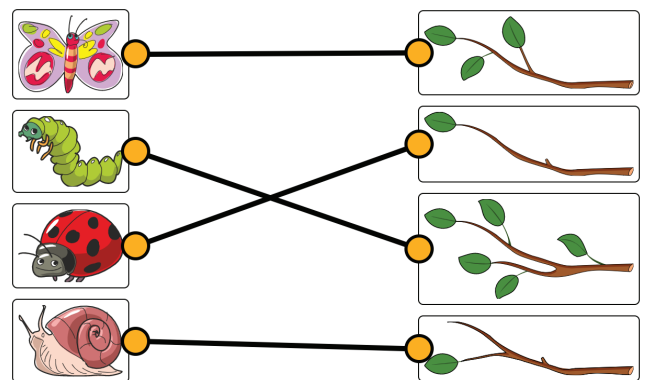
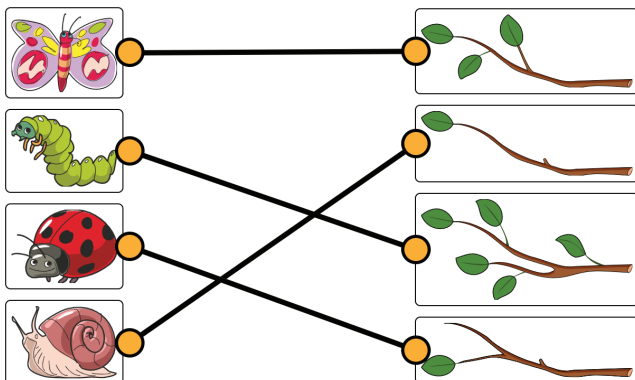
Question

Match the creatures to their branches.



Answer

There are two possible solutions.



The third branch has the most leaves (5) and so the caterpillar is on the third branch. The first branch has more leaves (3) than the other two which both have 1 leaf, so the butterfly must go on the first branch. The snail and ladybug can go on either of the remaining branches.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms, Evaluation

Concepts: Abstraction, Data Representation, Data Interpretation, Algorithms

This is a logic problem. We are given a set of statements and asked to determine an arrangement of creatures such that each statement is true. Logic is the study of statements, and determining if they are always, never, or sometimes true.

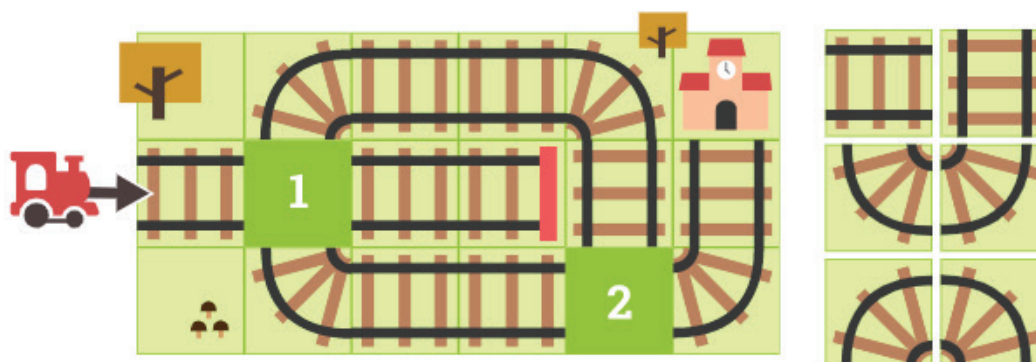


Train Tracks

Can you help the train  get to the station  safely?

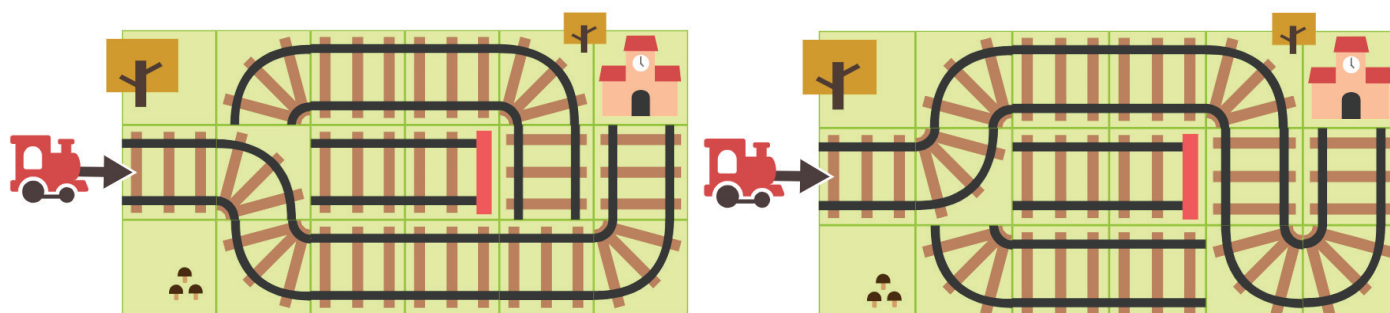
Question

Drag and drop two pieces of track onto the green numbered squares, to complete the track so the train can get to the station.



Answer

There are two correct solutions to the problem.



It's Computational Thinking

Computational Thinking Skills: Abstraction, Modelling and Simulation, Evaluation

Concepts: Abstraction, Data Interpretation, Specification

A computer program is made of many instructions, like the railroad track in the task is made of many rails. Sometimes, a computer program does not work, because instructions are missing. The programmer has to find the correct instruction to produce the desired result.

By writing a program, you learn to control an object or a character in the same way that the rails control the movement of the train. Each track section can be understood as an instruction, a command given by the programmer to the train to move in the chosen direction.

This sequence of instructions that are executed in a certain order and always produce the same result, this is what computer scientists call a “program”.

In order to solve this task and find the two missing track sections, the children have to anticipate what the train will do before it even goes through the circuit. In doing so, they analyse the system and plan the behaviour of an object that will be executed in the future. These are facets of Computational Thinking.



Flying Bees

Bees can only fly up, down, left, or right from one square to the next (not diagonally). When a bee leaves the hive in the morning, it can fly no more than 3 squares from the hive. The distance is counted by the squares.

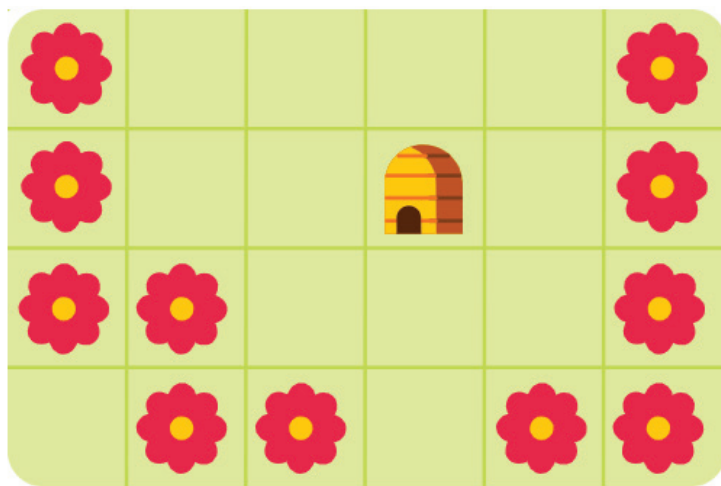
Example

The distance from the hive to the square with the 'X' on is 3 squares:



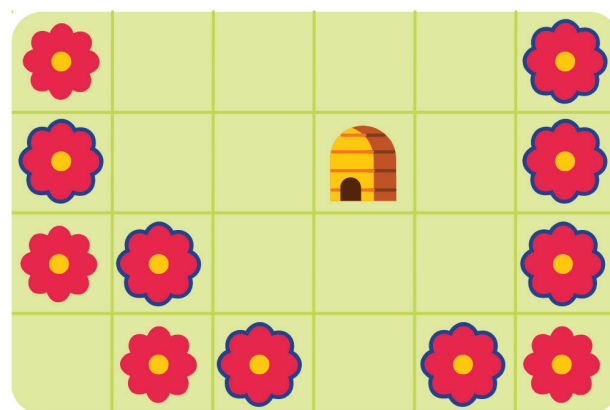
Question

Select all the flowers that can be reached by the bees.



Answer

In general, we can find all squares that are no further than three squares from the hive by at first finding the farthest squares the bee can fly. If it flies three squares to the left then it ends at point 1. It can also fly two squares to the left and one up (point 2) or down (point 3), or one square to the left and two up (point 4, which is outside the field) or down (point 5). If we apply the same method when starting to fly up, right and down and then also consider all squares that are closer to the hive than the farthest ones, we find all the squares that are not farther than three squares from the hive. These squares are painted by a darker green colour:

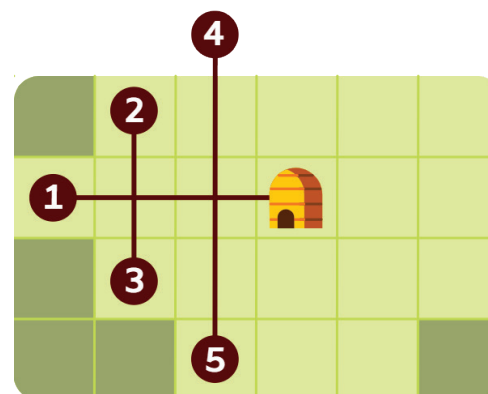


It's Computational Thinking

Computational Thinking Skills: Abstraction, Modelling and Simulation, Algorithms

Concepts: Abstraction, Data Collection, Specification

The flight of the bee is described as an algorithm. Understanding algorithms so that one can infer their properties (like how far it can fly, all the places it can fly to within a distance) is an important skill belonging to informatics.

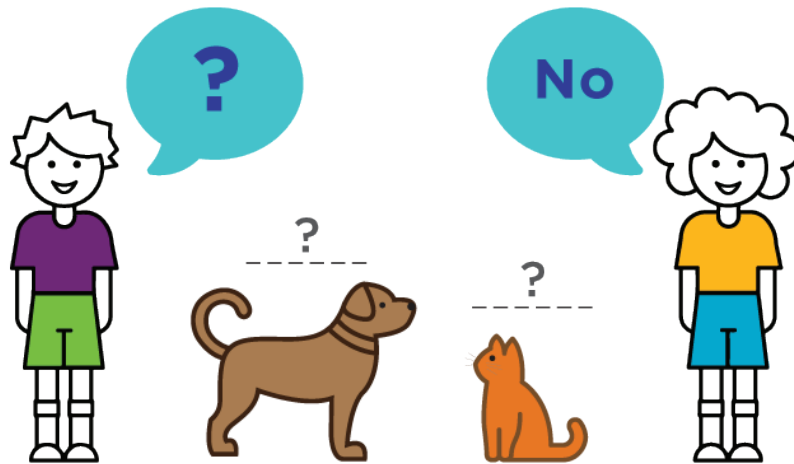


Note that the distance measured in squares, which is used in the task is not the “usual” distance between two points (also called Euclidean distance), because the bees are not allowed to travel diagonally or in any direction, but only horizontally or vertically within the grid. Such distance is called rectilinear distance (or block distance referring to a rectangular grid of streets usual in modern cities).



Polly's Pets

Kevin asks the following question to Polly, "Gummy is a cat and Puffy is a dog, right?"
Polly responds, "No, that's not true."



Question

Which of the following statements is true?

Gummy is not a cat, and Puffy is not a dog.

Either Gummy is not a cat, or Puffy is not a dog.

Gummy is not a cat, or Puffy is not a dog.

Gummy is not a cat, but Puffy is a dog.

Gummy is a cat, but Puffy is not a dog.

We cannot say anything about Gummy or Puffy.

Answer

The answer is C) "Gummy is not a cat or Puffy is not a dog".

If the combined statement "Gummy is a cat and Puffy is a dog" is not true, then at least one of the two statements that are combined with "and" must be false. That means that either Gummy is not a cat, Puffy is not a dog, or Gummy is not a cat and Puffy is not a dog. One can simplify this by simply saying "or". Therefore answer C) "Gummy is not a cat or Puffy is not a dog" is true.

Answer A) is not correct, because it could be that both Gummy and Puffy are cats. Then the combined statements "Gummy is a cat and Puffy is a dog" as well as "Gummy is not a cat and Puffy is not a dog" would both be false.

Answer B) is not correct, because it could be that Gummy is a dog and Puffy is a cat. Then the combined statement "Gummy is a cat and Puffy is a dog" would be false, but the combined statement "Either Gummy is a cat or Puffy is a dog" would also be false.

Continued on next page



Polly's Pets - continued

Answer - continued

The answer D) is not correct, because it could also be that both Gummy and Puffy are dogs. Then the combined statement "Gummy is a cat and Puffy is a dog" would be false, but the combined statement "Gummy is not a cat but Puffy is a dog" would also be false.

The answer E) is not correct, because it could also be that both Gummy and Puffy are cats. Then the combined statement "Gummy is a cat and Puffy is a dog" would be false, but the combined statement "Gummy is a cat but Puffy is not a dog" would also be false. Since we were able to say something about Gummy and Puffy, answer E) is not correct, either.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Algorithms, Evaluation

Concepts: Specification, Algorithms

This task is about the meaning of the words "and", "or", "either – or" and "not". Although we use them in our everyday language, it could be confusing when we try to combine them. In this case defining them by the use of truth tables helps to truly understand them.

The conjunction "and" is defined so that the result of combining two statements with "and" is true if and only if both combined statements are true:

<i>A</i>	<i>B</i>	<i>A and B</i>
False	False	False
False	True	False
True	False	False
True	True	True

The exclusive disjunction "either – or" is defined so that the result of combining two statements with "either – or" is true if and only if exactly one of two combined statements are true:

<i>A</i>	<i>B</i>	<i>either A or B</i>
False	False	False
False	True	True
True	False	True
True	True	False

The negation is defined so that the result is exactly the opposite of the original statement's veracity:

<i>A</i>	<i>Not A</i>
False	True
True	False

Most difficult to understand is the (non-exclusive) disjunction "or", because in everyday language people mix it with the (exclusive) disjunction "either – or". It is defined so that the result is true if and only if at least one of the original statements is true. So the only difference between the disjunction and the exclusive disjunction is that a disjunction is also true if both combined statements are true:

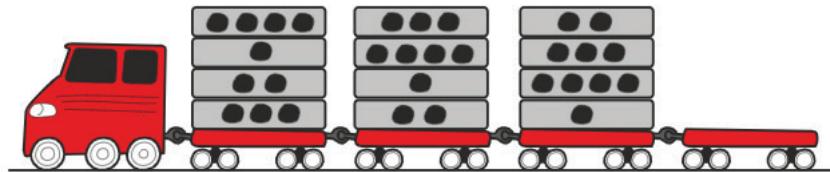
<i>A</i>	<i>B</i>	<i>A or B</i>
False	False	False
False	True	True
True	False	True
True	True	True



Dotted Blocks

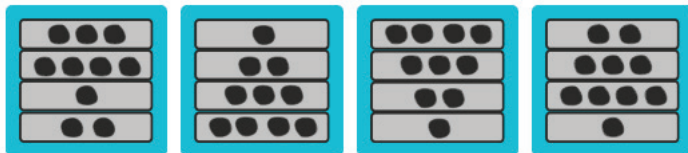
A train that has 4 trailers carries blocks that are labelled by dots.

The 1st, 2nd, and 3rd trailers are already loaded with blocks. The blocks are arranged according to a certain rule.



Question

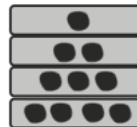
Select the block pattern you would expect to go on the fourth trailer to follow the rule.



Answer

The correct answer is B.

You can choose any block on the first trailer and check where it is on the second and third trailers. So the pattern in the fourth trailer should look like this:



Trailer 1	Trailer 2	Trailer 3	Trailer 4
4	3	2	?
1	4	3	?
2	1	4	?
3	2	1	?

By doing this, you will recognize that a block moves down by one place in the following trailer. So, for the fourth trailer, all of the blocks will be moved down by one, from the third trailer. Therefore the fourth trailer should have the blocks in this arrangement:

Trailer 1	Trailer 2	Trailer 3	Trailer 4
4	3	2	1
1	4	3	2
2	1	4	3
3	2	1	4

It's Computational Thinking

Computational Thinking Skills: Pattern Recognition, Modelling and Simulation, Algorithms, Evaluation

Concepts: Data Representation, Data Interpretation, Algorithms

In informatics it is very important to notice similar images or patterns. Pattern recognition is closely related to artificial intelligence and machine learning. Finding regularities, predicting patterns and behaviour are not easy, but can help computers perform complex tasks. In this case, a proper solution can be found by examining all of the blocks and recognising the pattern.

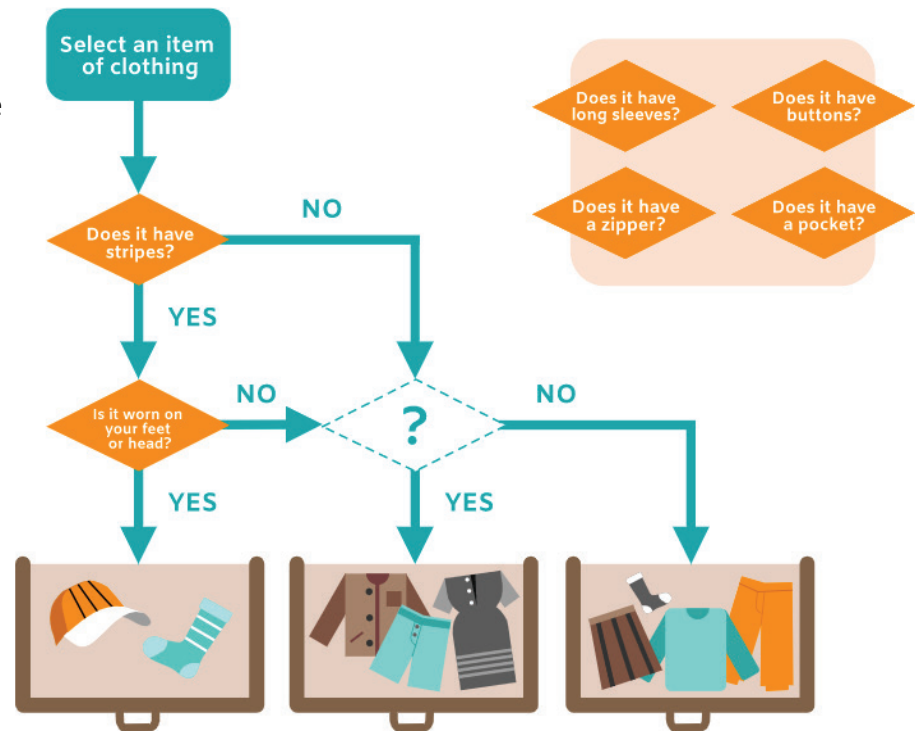


Object Sorter

Bora used a flowchart to organise her clothes into 3 different drawers. She started at the top of the flowchart by selecting an item. She then followed the arrow until it reached a question. She then decided if the answer was 'yes' or 'no' and followed the arrow that matched her answer. She continued to answer questions and follow arrows until she arrived at a drawer. She placed her item inside that drawer and started over again at the top of the flowchart with a new item until all her clothes were organised.

Question

If her clothes are organised into the drawers as shown, what question is missing from her flowchart?



Answer

The correct answer is 'Does it have buttons?'

If the answer was 'Does it have long sleeves?', then the shorts in the middle drawer would not be there. If the answer was 'Does it have a zipper?', then the shirt in the middle drawer would not be there. If the answer was 'Does it have a pocket?', then the dress in the middle drawer would not be there. The answer 'Does it have buttons?' can be verified by comparing the items in the middle and right-hand drawer. Each item in the middle drawer has buttons, and each item in the right-hand drawer does not have buttons.

It's Computational Thinking

Computational Thinking Skills: Pattern Recognition, Modelling and Simulation, Algorithms, Evaluation

Concepts: Specification, Algorithms

The flowchart used by Bora could also be called a decision tree. Decision trees can lead you to an outcome by asking relevant questions along the way. They are simple to navigate and for this reason they have many different applications.

Flowcharts are also commonly used during the planning stages of code writing. The three main building blocks of code writing, often called "sequence, selection, and repetition" can each be represented in a flowchart. The sequence actions are drawn in rounded rectangles, the selections (or decisions) are drawn in diamond shapes, and repetition could be illustrated by having an arrow return to an earlier part of the flowchart. By using a flowchart to plan out how a program should behave, programmers can agree on its behaviour and often find and fix errors in logic before coding even begins.



Gifts

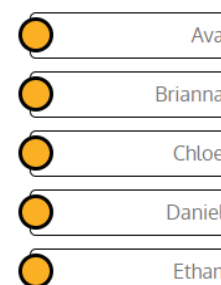
Gabriella received 5 presents but the gift tags have come off, so she doesn't know who gave her which present! She remembers the following clues about each present.

Clues:

- Ava sent a present with stars.
- Brianna sent a present with a bow.
- Chloe sent a present with wavy stripes.
- Daniel sent a long thin present.
- Ethan's present did not have a bow.

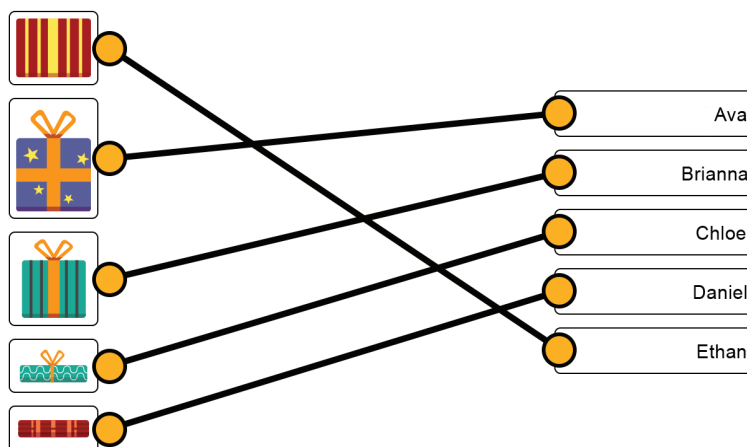
Question

Match the correct present to the friend who sent it.



Answer

The 1st clue tells us which present Ava sent so we can connect these two options. The 2nd clue does not help at this stage. The 3rd clue tells us which present Chloe sent so we can connect these two options. The 4th clue tells us Daniel sent a long thin present but there is only one of these left so we can connect Daniel to his present. The 5th clue enables us to connect Ethan to the only present left without a bow. This just leaves one present and so we can connect it to Brianna.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Pattern Recognition, Abstraction, Algorithms

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

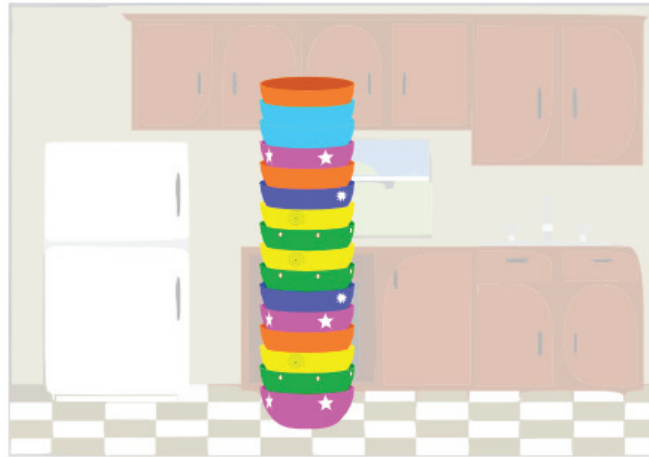
Logic is the basis of many aspects of Computer Science. This problem can be solved with logical reasoning. Students need to use abstraction to recognise which features of the present are relevant to solving the problem, and which are not. By identifying known information, the unknown information can be worked out.

This is an introductory example of a bipartite graph, where the vertices are divided into two independent sets (in this case: present and sender). Although students are very used to this kind of question, bipartite graphs are also important in Computer Science and have applications in fields as varied as social network analysis and scheduling train networks as well as in systems biology.



Bowls

3 siblings want to eat breakfast from 3 bowls of the same design. The stack includes bowls with many different designs. Bowls can only be removed from the top of the stack.



Question

What is the smallest number of bowls that need to be removed to get 3 bowls of the same design?

13

14

15

16

Answer

At least 13 bowls need to be removed to get 3 bowls of the same design (in this case, 3 orange bowls).

It's Computational Thinking

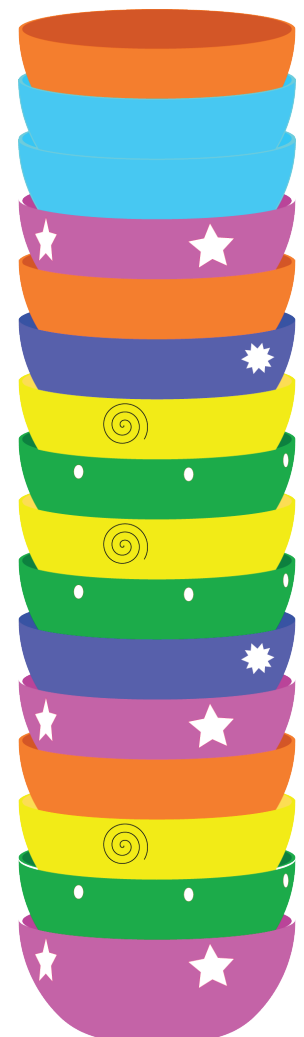
Computational Thinking Skills: Abstraction, Modelling and Simulation, Algorithms

Concepts: Abstraction, Data Collection, Data Representation, Algorithms

A stack is a very common way of storing things in Computer Science. It is very simple but also quite powerful. There are rules for how to put things on a stack, and how to take them back off a stack. In this task, we are just dealing with how to take things off a stack.

The rule is that only the item at the top of the stack can be removed at a time. If you want to get the tenth bowl in a stack of bowls, you have to remove the top ten bowls. It becomes important then to have somewhere to put the other nine bowls - this is also the case in Computer Science.

If we have a second stack, and we can make our stacks as high as we want, we can compute everything there is to compute in Computer Science! This is because a computer with two stacks can be equivalent to a Turing machine, the most studied theoretical computer. This simple little stack really is powerful!

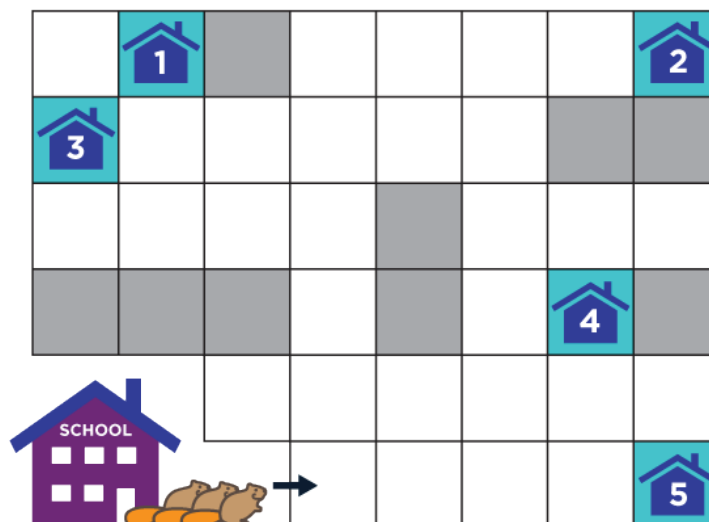




Where Do They Live?

Adam, Emil and Sam stand in front of their school and face the indicated direction. On their walk home they all make one or more turns at some point during their walk.

- Adam makes two left turns.
- Sam makes one left turn followed by one right turn.
- Emil makes one left turn.
- Each beaver lives in a different house.
- They only walk horizontally or vertically, not diagonally.
- They cannot pass through the dark grey squares.



Question

Which house does each beaver live in?

Adam in house no. 4, Sam in house no. 2, Emil in house no. 5

Adam in house no. 3, Sam in house no. 2, Emil in house no. 1

Adam in house no. 4, Sam in house no. 2, Emil in house no. 1

Adam in house no. 3, Sam in house no. 2, Emil in house no. 4

Answer

The answer is Adam lives in house no. 3, Sam lives in house no. 2 and Emil in lives house no. 4.

Adam cannot live in house no. 4. because it is necessary to turn either once to the left; once to the left and then to the right; or turn more than once to reach house no. 4. Emil cannot live in a house no. 1. because he would have to turn left twice and then right once in order to reach that house. We can arrive at the correct answer by noting that house no. 3 is the only house that can be reached by Adam who makes two left turns. Sam can reach house no.2 or no. 4 but cannot live in the same house as Emil. Emil can only reach house no. 4.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Modelling and Simulation, Algorithms, Evaluation

Concepts: Data Representation, Data Interpretation, Specification, Algorithms

Imagining the effect of some actions is normally not too difficult: if a beaver is known to have gone left, left and right on a grid, it is easy to follow its movements, provided that “left” and “right” are well defined actions, for example a motion of one square in the directions defined by the grid and the current beaver orientation. The task, however, proposes a more difficult problem, quite common in informatics: given some effects, some hints, discover what happened in the past. In many cases you have (even lots of) data, and you want to understand which kind of process could have generated them. Note that in general several processes are compatible with the same observed data: for example Sam was known to have turned left and right, and two different houses (2 and 4) are reachable by tracks which comply with this constraint.



Beaver Salon

4 stylists work at Beaver Salon: Irina, Ajani, Zuri, and Pia. Each stylist can perform 3 services, and each service takes a different amount of time to complete.

Each stylist has just started working on a beaver as shown. As soon as a stylist finishes, they begin work on the next beaver waiting in line.

Service	Time
Fur fluff	5 minutes
Teeth sharpening	8 minutes
Tail polish	15 minutes



Question

Obi is waiting for a tail polish. Which stylist will work on Obi?

Irina

Ajani

Zuri

Pia

Answer

Since Zuri will finish first, Zuri will work on Obi.

All stylists started working at the same time. Irina and Pia will both finish in 8 minutes since they are both sharpening teeth. Ajani’s tail polish will finish in 15 minutes. Zuri, who is fluffing fur, will be done in 5 minutes.

It’s Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Modelling and Simulation

Concepts: Abstraction, Data Collection, Specification

In the beaver salon there are four stylists (Irina, Ajani, Zuri, and Pia) that can work at the same time on different beavers.

The way a computer works is very similar to the situation of the beaver salon. In a modern computer there are several processors that work at the same time. Many tasks are performed at the same time, in this sense we say that the computer is multitasking. When one of the computer’s processor has finished executing a task, it receives a new one. We call this ability parallelism.

The method by which tasks are assigned to different processors in the computer is called scheduling. This ability to distribute tasks to different parts of the computer and to execute them in parallel is one of the reasons why modern computers are so fast: they are very good at managing parallelism and can perform difficult problems quicker this way.

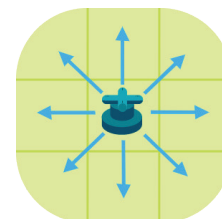
Bebras Challenge 2021 Round 1

Years 5+6



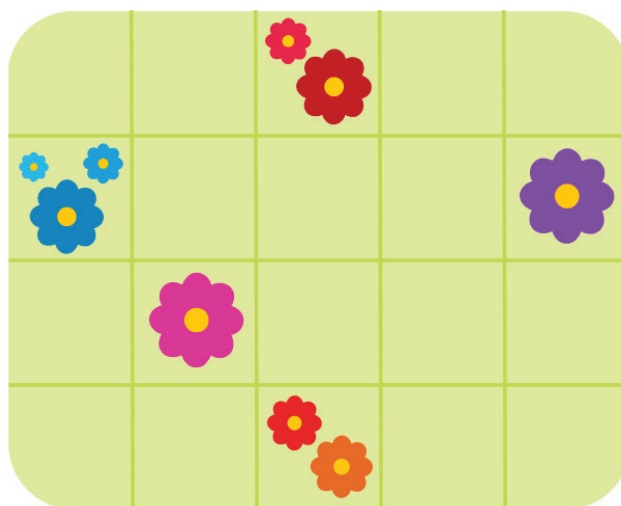
Sprinklers

Beaver Brayden planted some flower beds in his garden. Brayden wants to place some water sprinklers in the empty squares of his garden. Each sprinkler can water all the eight squares next to it, as shown in the picture on the right.



Question

Use the smallest number of sprinklers needed to water all the flowers in Brayden's garden. Click on the empty squares of the garden where you want to place the sprinklers.



Answer

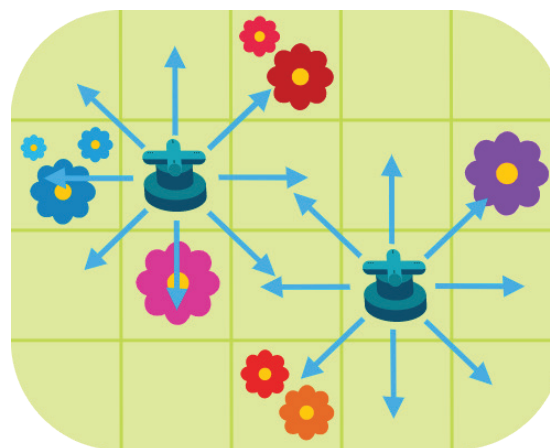
It is impossible to reach all the flower-beds with just one sprinkler.

The flower-beds in the second row cannot be watered by a single sprinkler because they are too distant. So using two sprinklers is the best we can do, as shown.

It's Computational Thinking

Computational Thinking Skills: Abstraction, Modelling and Simulation, Evaluation

Concepts: Abstraction, Data Interpretation, Specification, Digital Systems



In Computer Science we often have to deal with looking for the best possible solution of a problem. This means to decide a way to compare solutions (i.e. a criteria) and find the best solution. In our case we asked not only to water all the flower-beds but to do it using the minimum possible number of sprinklers in order to avoid water waste.

The easiest solutions to water all flower-beds could be that we use a sprinkler in every empty square or a sprinkler for each flower-bed. But it would be pricey and not so effective and ecological. It could be also dangerous to water too much some of the flower-beds (this deal to a different and more complex comparison criteria).

In this case it was easy enough to find the solution just by a try and fail approach. Due to the reduced dimension of the example it should be affordable to try all possible sprinklers' dispositions too. Increasing the size of the garden, the number of flower-beds, and possibly adding other constraints to the criteria make more difficult to find the best decision. Informatics is of big help in modelling and realising operations research techniques that arrive at optimal or near-optimal solutions to complex decision-making problems.

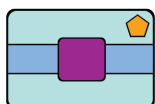


Quiz Card

Milan and Maya completed a quiz with 4 questions.
Milan's answers were:

- Question 1: Answer A
- Question 2: Answer C
- Question 3: Answer B
- Question 4: Answer A

Following the rules pictured on the right, Milan received a card representing his answers.



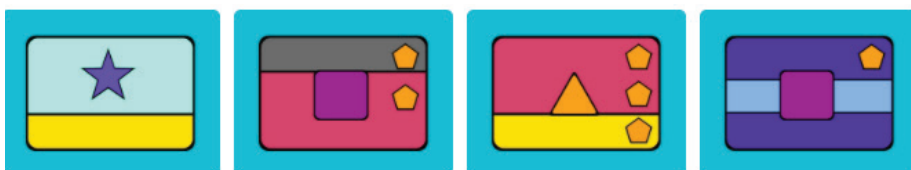
	A	B	C
1			
2			
3			
4			

Question

Maya's answers were:

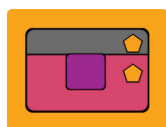
- Question 1: Answer B
- Question 2: Answer A
- Question 3: Answer B
- Question 4: Answer B

Which card represents Maya's answers?



Answer

The answer is a combination of the answers shown:



It's Computational Thinking

Computational Thinking Skills: Decomposition, Pattern Recognition, Abstraction, Modelling and Simulation

Concepts: Abstraction, Data Representation, Data Interpretation

Data visualisation is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualisation tools provide an accessible way to see and understand trends, outliers, and patterns in data.

In the world of Big Data, data visualization tools and technologies are essential to analyse massive amounts of information and make data-driven decisions.

	A	B	C
1			
2			
3			
4			



House Numbers

Beavers use symbols instead of house numbers.
They use the table on the right to translate the symbols to numbers.

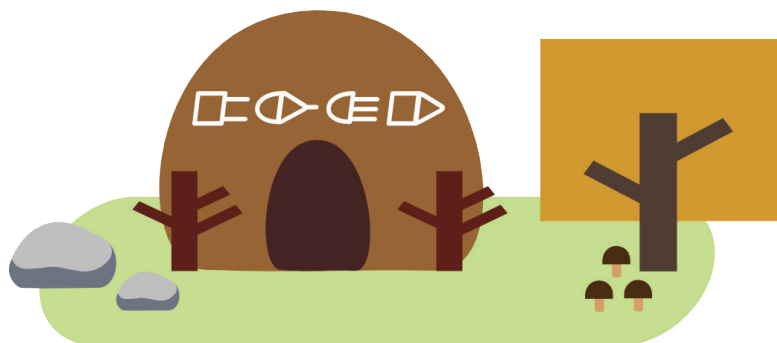
Example

5 is written as:

	-	=	≡	▷	▷
□	0	1	2	3	4
◁	5	6	7	8	9

-	=	≡	▷	▷	
□	0	1	2	3	4
◁	5	6	7	8	9

Here is a picture of one of the beavers' homes:



Question

What is this beaver's house number?

Answer

The digits on the beaver's home can be figured out by using each symbol on the home to find the correct row and column and then finding where the row and column meet.

Therefore, the digits on the beaver's home are 1973.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Pattern Recognition, Abstraction, Algorithms

Concepts: Abstraction, Data Representation, Data Interpretation, Specification

This task is about encoding. Encoding is the process of applying a code to data in order to represent the same data in a different (equivalent) way. The beavers' code in this task is the table, and they apply the table to the digits 0 through 9 in order to represent the digits in a different way using symbols. Encoding has many uses in Computer Science. Encoding data can be a way to hide messages. This idea is called cryptography. Encoding data can also be a way to reduce the size of messages. This idea is called data compression.



Treasure Island

Pirate Pablo is looking for a treasure. It is buried on one of the five islands. Luckily there is a code which describes how to get to the island with the treasure. Pablo reads one letter after another and crosses the bridge marked with that letter each time. If he stands on an island with no corresponding bridge, Pablo skips this letter and moves on to the next one.

Example

If the code is A-B-A, Pablo crosses bridge A to Island 2. There is no bridge B from Island 2, so he ignores this part of the code. Finally he moves back to Island 1 over bridge A.

The secret code to get to the treasure island is B-A-C-A-A-B.

Question

Click on the number of the island where the treasure is hidden.

Answer

Island 4 is correct.

Starting at Island 1, Pablo:

B: Crosses Bridge B to go from Island 1 to Island 3

A: He stays on Island 3 because there is no Bridge A

C: He crosses Bridge C to go from Island 3 to Island 4

A: He crosses Bridge A to go from Island 4 to Island 5

A: He crosses Bridge A to go from Island 5 to Island 4

B: He stays on Island 4 because there is no Bridge B

It's Computational Thinking

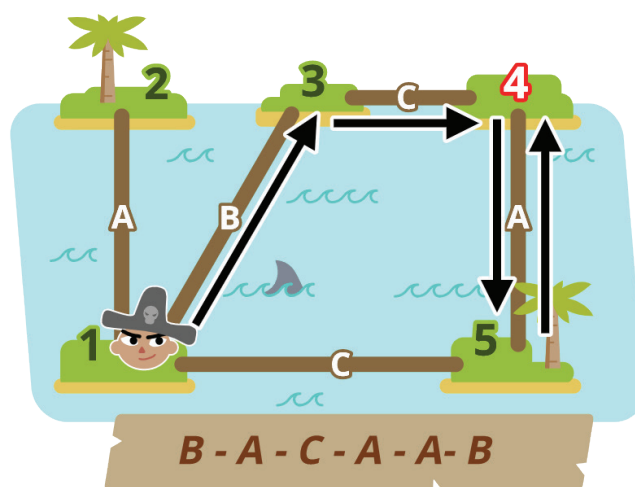
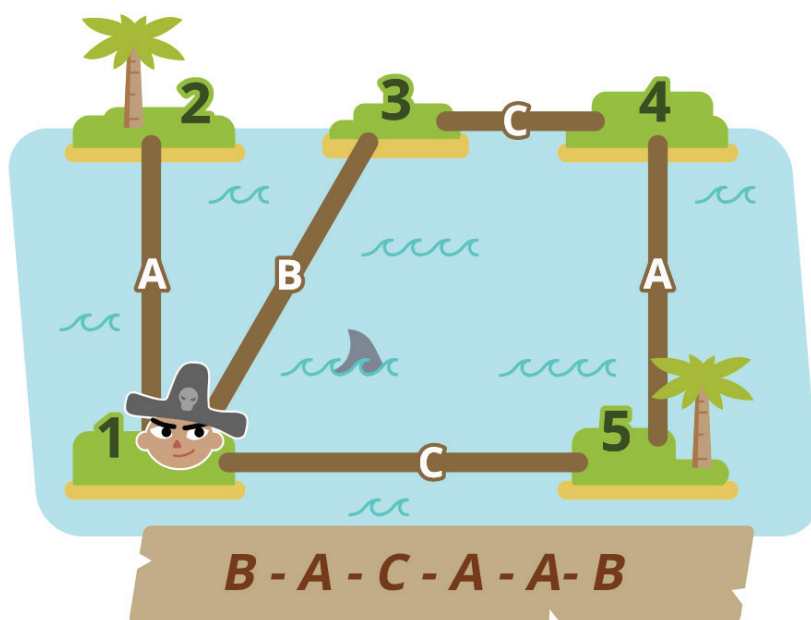
Computational Thinking Skills: Decomposition, Modelling and Simulation, Algorithms

Concepts: Data Interpretation, Specification, Algorithms

This task is about finding a route through a map. Maps are commonly handled by computers, for instance, all GPS devices store maps to compute routes between two end points. In GPS devices, maps are not stored as images but instead as structures called graphs — sets of numbers that describe locations and connections between the locations to easily allow computations on them.

Having a map stored in the computer is just the start. Computer scientists then need to write programs that can find their way through the many kinds of maps. There are algorithms for finding the fastest route, shortest route and, as in this simpler case, just following a route.

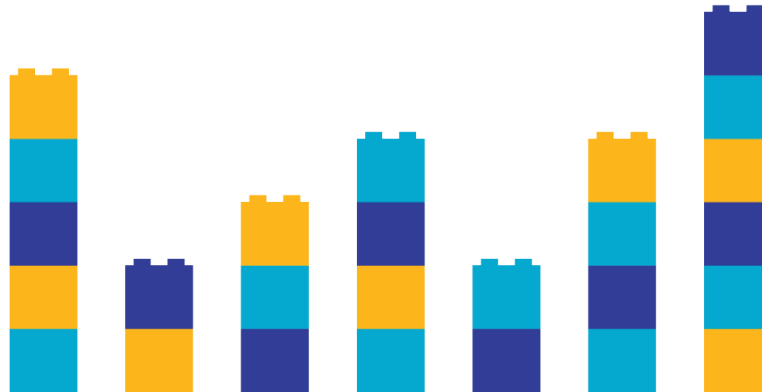
The code in this task may appear odd. Why include a bridge that is not actually available? When writing and executing programs, it is important to have a way to deal with erroneous data so that the program does not simply crash and stops working. We can usually be pretty sure that, at some point in time, our program will have to deal with erroneous data, and we must be prepared.





Towers of Blocks

Sam, the little beaver, is playing with his toy blocks. He built seven beautiful towers, each one made with blocks of the same size.



There are two ways to change the height of a tower: adding blocks to the top or removing blocks from the top. Adding or removing a block counts as a move.

For instance, if he changes the height of the leftmost tower to 2, it takes 3 moves (removing 3 blocks), and if he changes it to 7, it takes 2 moves (adding 2 blocks). Moving a block from one tower to another is considered as 2 moves.

Sam wants all towers to be the same height, and he wants to make as few moves as possible.

Question

In total, what is the minimum number of moves Sam needs in order to make all towers the same height?

6

7

8

9

10

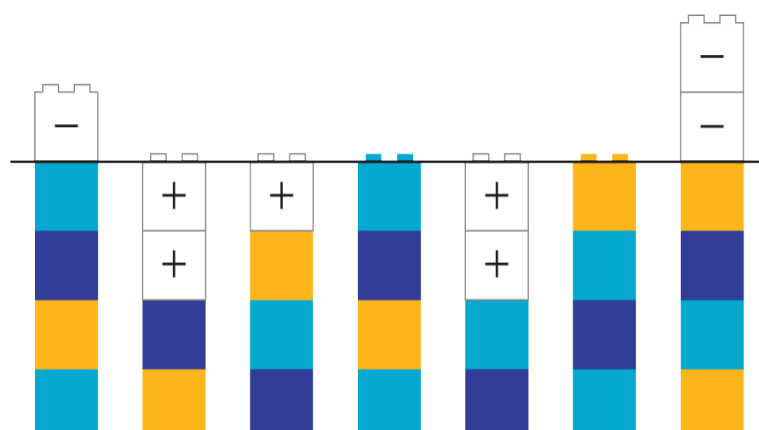
11

Answer

The correct answer is C), 8 moves.

Adding blocks is depicted as  and removing as 

The following figure depicts the correct solution with 8 movements, making all the towers 4 blocks tall:





Towers of Blocks - continued

Answer - continued

As calculated in the table below, all other possible heights would take more moves:

Moves	Original height	5	2	3	4	2	4	6	Total moves
	Height 1	4	1	2	3	1	3	5	19
	Height 2	3	0	1	2	0	2	4	12
	Height 3	2	1	0	1	1	1	3	9
	Height 4	1	2	1	0	2	0	2	8
	Height 5	0	3	2	1	3	1	1	11
	Height 6	1	4	3	2	4	2	0	16

Another way to reach the solution would be considering the sorted heights of the towers, as the best height to use here is the median: 2 2 3 4 4 5 6

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Modelling and Simulation, Algorithms, Evaluation

Concepts: Abstraction, Data Representation, Data Interpretation, Specification, Algorithms

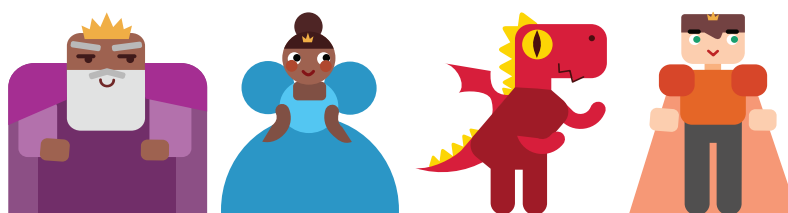
Each pile of blocks is what we call a stack in Computer Science, with two possible operations: push (putting a new block on the top of the stack) and pop (removing the topmost block). In this problem, we need to take into account the current height of each tower already built. Stacks are very useful in many algorithms more complex than basic block building. For example, they can be used to evaluate arithmetic expressions.

Searching for the best solution (such as the minimum cost) among all possibilities is a very common thing in Computer Science. This is known as an optimization problem. In this case, we could simply find and use the median, which would involve sorting the elements and selecting the middle position.



Theatre Performance

You are going to a play at the theatre. These are the characters:

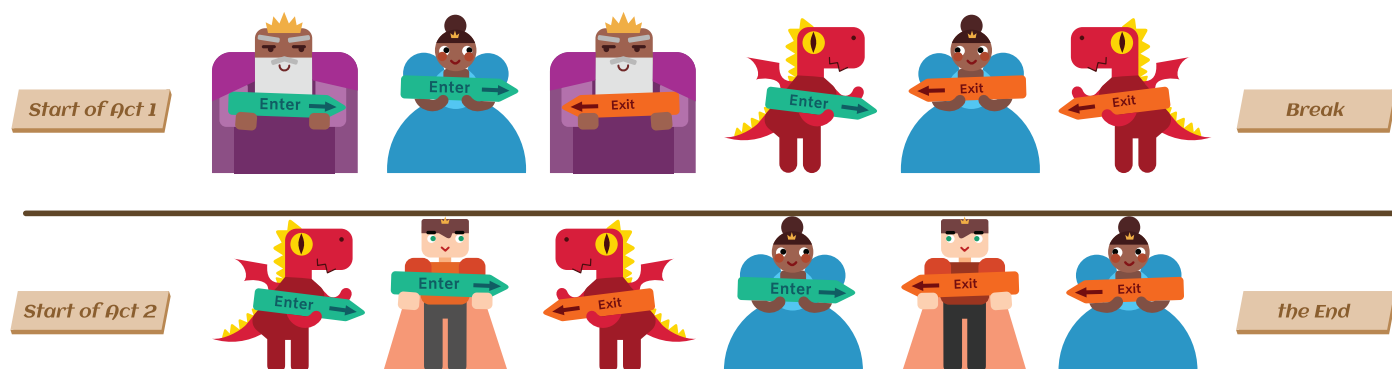


The King The Princess The Dragon The Prince

The play has two parts: Act 1 and Act 2, with a break in between.
The picture below helps the actors remember when to **enter** and **exit** the stage.

Example

In Act 1, the King enters the stage first, followed by the Princess. Then the King exits before the Dragon enters.



Question

Which statement is not true?

The Prince and the Princess are on stage together.

The King and the Dragon are on stage together.

The Prince is on stage after the break.

The Prince and the Dragon are on stage together.

Answer

'The King and the Dragon are on stage together'.

'The Prince and the Princess are on stage together' is true. After the break, the Prince entered the stage, the Dragon left the stage, and then the Princess came back onto the stage.



Continued on next page



Theatre Performance - continued

Answer - continued

'The King and the Dragon are on stage together' is not true, because the King left the stage right before the Dragon came onto the stage.

'The Prince is on stage after the break' is true.

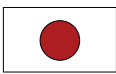
'The Prince and the Dragon are on stage together' is true. After the break, the Dragon came back onto the stage and then the Prince came onto the stage as well.

It's Computational Thinking

Computational Thinking Skills: Abstraction, Modelling and Simulation, Algorithms, Evaluation

Concepts: Abstraction, Data Representation, Data Interpretation, Specification, Algorithms

In Computer Science we often graphically depict a process (events that happen over time in a given order). Graphic visual representations of information, data, or knowledge are usually intended to present information quickly and clearly. They can improve comprehension by enhancing the human visual system's ability to see patterns and trends. Understanding visual representations (or diagrams) and drawing conclusions from them is an important skill to develop in Computer Science.



This question comes from
Japan

Years 3+4
Years 5+6 Medium

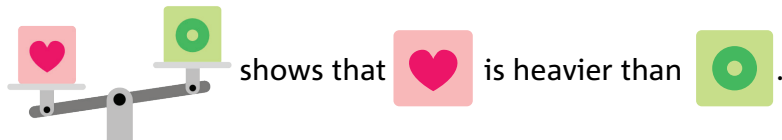
Years 7+8
Years 9+10
Years 11+12



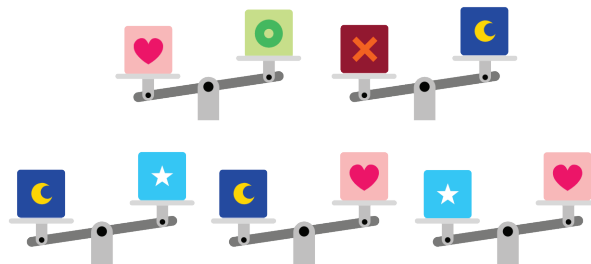
Heaviest Box

Five boxes have different shapes drawn on them.
You can compare the weights of two boxes with a see-saw.

Example



The see-saw was used five times. These were the results:



Question

Which box is the heaviest?



Answer

The correct answer is:



What can we tell about the heaviest box in advance?

It must be the heavier box (that is, it must be on the lower part of the scale) in any comparison which it is involved in. Looking into all five comparisons, the box with the X-mark is the only such box; it never is on the upper part of the scale. As each of the other boxes are the lighter box in at least one comparison, we can be certain that they are not the heaviest.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Modelling and Simulation, Algorithms, Evaluation

Concepts: Abstraction, Data Representation, Data Interpretation, Specification, Algorithms

To find the heaviest box, you had to look at five comparisons. When looking for the heaviest or best or largest object in a set, you may have to compare every object with every other. When there are many objects, this may result in many comparisons. It would have been much easier to find the heaviest box if the boxes had already been sorted.

Computers very often need to find specific information within huge data sets. This is much easier if the data are sorted - which is why sorting is important in informatics. Computer scientists have developed many sorting algorithms. As computers are sorting data often, it is important that sorting algorithms are efficient. Unfortunately, computers can compare only two values at one time, like the balance scale in this task. No matter which sorting algorithm is used, at the core of sorting is the comparison of two values. Sorting algorithms can be characterised by the number of comparisons used.



Connect the Dots

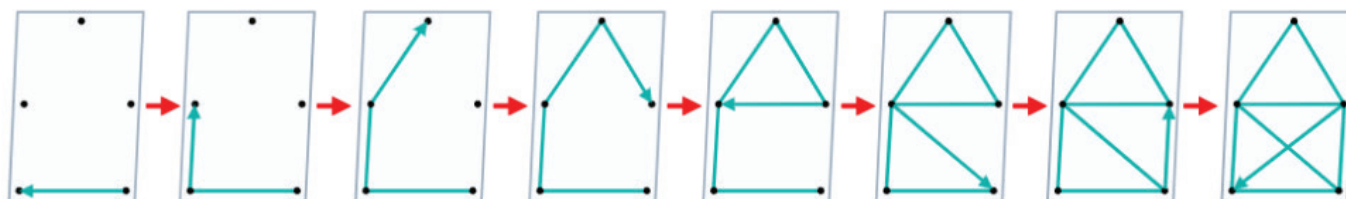
You can draw pictures without lifting your pen.

To do this, you draw lines from one dot to the next.

You may NOT draw a line over one that you already drew.

Example

You can draw a picture of a house like this:

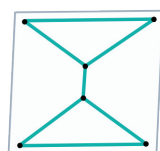


Question

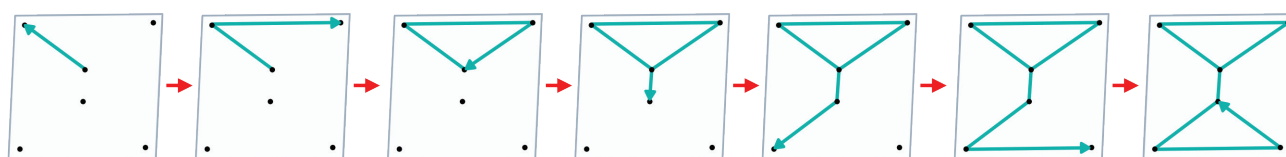
Which picture can you draw this way?



Answer



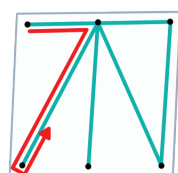
The correct answer is C).



One possible sequence of drawing steps is shown below.

Note that exactly two of the points in answer C) have an odd number of lines coming in or going out. All other points have an even number of lines coming in or going out.

For answer A), however, you have four of the points with an odd number of lines coming in or going out. They are marked in the picture below. But you can only start at one point and end at one point. To draw a picture in a way you want to draw your pictures, there must be at most two points with an odd number of lines coming in or going out. It would also be possible to draw pictures having more than two points with an odd number of lines coming in or going out if you were able to come in or go out of a point with the same line, but you can't draw the same line segment more than once.

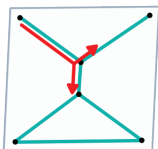




Connect the Dots - continued

Answer - continued

The situation for answer B) is very similar, as you can see by the attempt below: you must either go down or right-up, but then you'll have to go back in either case to draw the third line.



Finally, answer D) consists of a triangle drawn inside a square, but none of the points of the (inner) triangle is connected to the (outer) square, so there is no way you could draw this picture without lifting the pen.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Modelling and Simulation, Evaluation
Concepts: Abstraction, Data Interpretation, Specification

These drawing are made of points and lines connecting the points. In Computer Science this is a way of representing objects or relationships among objects: the points represent objects and the lines represent connections or relationships among them. Such a representation is called a graph. A graph consists of a set of vertices or nodes (usually depicted as points or circles) and a set of edges (usually depicted as lines, possibly curved, sometimes directed with an arrow). The edges connect vertices. A sequence of connected edges in a graph is called a path.



The graphs you draw are special: the graph has to be connected (you should be able to go from any vertex to any other vertex by following edges of the graph) and only two vertices or none at all may have an odd degree (that is that the number of edges connected to a vertex). For graphs that have these two special properties you can always draw the complete graph in one continuous motion without having to draw an edge twice. A drawing like this is called an Eulerian path named after Leonhard Euler (1707 – 1783) who first described this problem.

Euler came up with the concept by trying to solve the problem of the Seven Bridges of Königsberg. Today we can find Eulerian paths by applying the well-known Fleury's algorithm or Hierholzer's algorithm.



Doors and Stones

Duck, Beaver and Cat want to find the treasure in a secret room.

They have to pass through doors  and switch-stones .

Both doors and stones are labelled with shapes. All the doors are closed at the start.

When an animal arrives at a closed door, he must stop and wait until the door is opened.

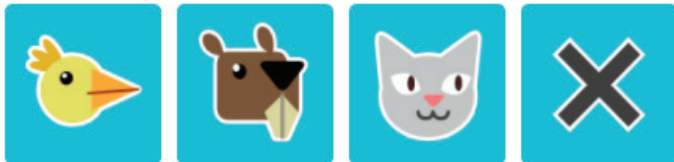
When an animal steps on a switch-stone, all the doors with the same shape on them open and stay open.

Duck, Beaver and Cat take different paths.




Question


Who will get to the secret room? Will it be Duck, Beaver, Cat, or no one?




Answer

The correct answer is Duck.

 Beaver is wrong because there is no jumping stone with a blue star. That means Beaver will be stuck at the first door on its path.

 Cat is wrong because the last door on Cat's path could only be opened by Beaver stepping on the jumping stone with the purple triangle. But Beaver will never reach that stone, as explained above.

 Duck is the correct answer. Beaver will unlock the doors with a red circle. Then Cat can unlock all doors with green crosses and white squares. At this point, all doors are open, and he can get to the secret.

 No one is wrong because Duck will get to the secret room.

It's Computational Thinking

Computational Thinking Skills: Abstraction, Modelling and Simulation, Algorithms

Concepts: Abstraction, Data Interpretation, Specification

Continued on next page



Doors and Stones - continued

It's Computational Thinking - continued

The three explorers in this task are like three computers (or processing units) working in parallel, and that are using a mechanism of locks shared between them. Each computer is able to release locks (like when an explorer steps on a jumping stone) and to stop execution until a lock is released (when an explorer encounters a closed door).

In the last few years, making a computer faster by increasing its processor's speed has become much more difficult due to the physical limits of the technology, and putting several processors (cores) in a single computer has become increasingly common.

To take advantage of these multiple processors and make their programs run faster, computer scientists have to find ways to split the computation into separate parts that can run in parallel, thus inventing parallel algorithms. Sometimes the processes need to wait for each other to complete some operations before continuing their execution, which is why they are using a mechanism of locks, such that one may not continue execution until it is unlocked by another process.



A Tree Structure

Beavers can build incredible structures from logs. Starting at their lodge S, a path to each log can be described by using the two commands:

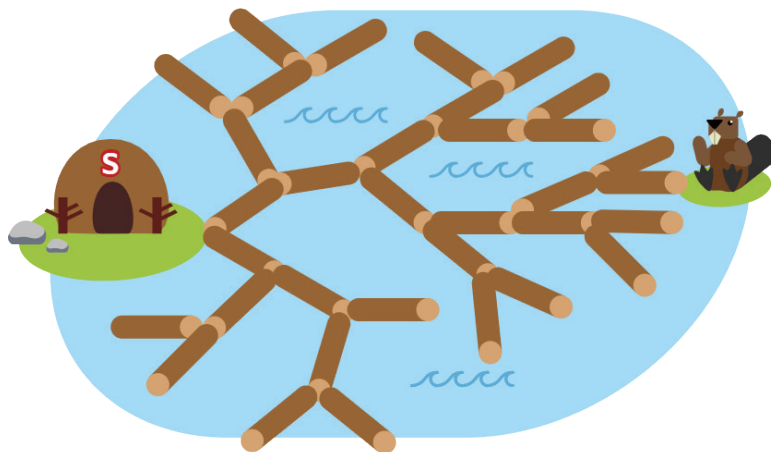
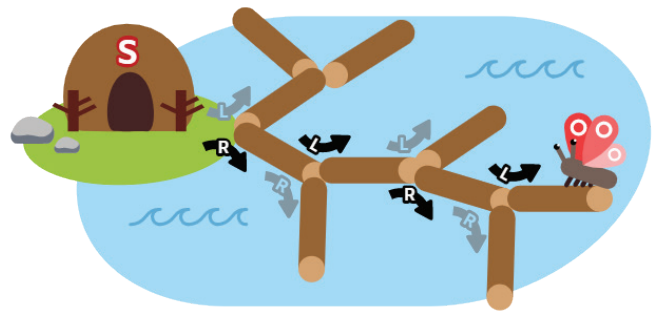
L (for left)
R (for right)

Example

The path to the butterfly is: S R L R L

Question

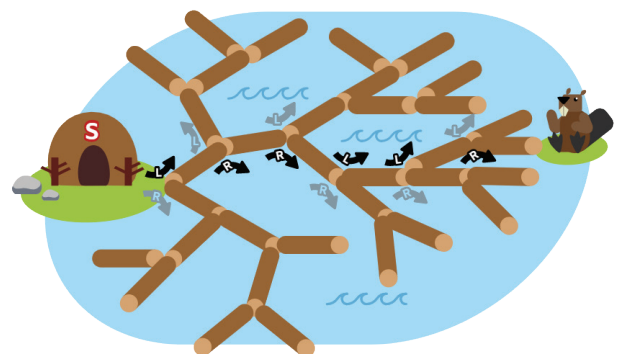
Describe the path from the lodge S to the relaxing beaver shown below.



Answer

The sequence is S L R R L L R.

Just by looking at the picture below, you can see that the first move to the right will not lead to the relaxing beaver. So it has to be left. Then you keep going until you reach the next fork and choose the direction by looking on which side the beaver is on.



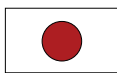
It's Computational Thinking

Computational Thinking Skills: Abstraction, Modelling and Simulation, Algorithms, Evaluation

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

Data can take many forms; it depends how we represent it. When we look at data, we are looking for a sequence of items that will assist in solving the problem. In this task logs represents a tree structure which represents the hierarchical nature of a structure in a graphical form. It is named a “tree structure” because the classic representation resembles a tree, even though the chart is generally upside down compared to an actual tree. Trees are a widely used data types that show relationships between a collection of things.

In Computer Science a tree data structure, in which each node has at most two branches (referred to as the left branch and the right branch), is called a binary tree. Binary trees are commonly used to be able to access data very fast. With only relatively few letters one can specify a big number of branches. In fact, if one uses, for example, 10 letters we can describe the position of 1024 (2 to the power of 10) different branches.



Painting the Houses

The people on Be-Taro's street decide to paint their houses according to these rules:

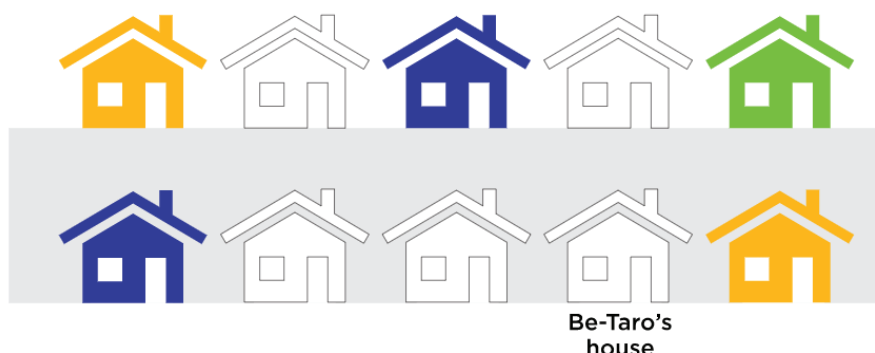
1. All houses must be painted either yellow, green or dark blue.
2. Houses next to each other must not be the same colour.
3. A house must not be the same colour as the house directly across the street.

In the picture below, you can see that some of the houses have already been painted.

Question

Choose colours for all the houses so that the above rules are followed.

Click on the white houses to switch colours. Keep clicking to cycle through the colours.



Answer

The correct colours can be worked out like this.

Start by working out the colour of the house that is second along on the top row. As this has a yellow and blue house beside it, it must be green. The fourth house on the top row has to be yellow because it has a blue and green house next to it. The second house on the bottom row of houses has a blue house next to it and a green house on the other side of the street, so it must be yellow. For a similar reason the third house on the bottom row must now be green. Finally the fourth house on the bottom row can now only be blue.

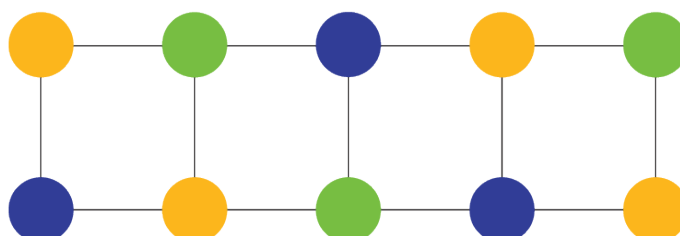


It's Computational Thinking

Computational Thinking Skills: Abstraction, Modelling and Simulation, Algorithms, Evaluation

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

The 10 houses together form a so-called graph, an object that is often used in Computer Science to represent data and how it is connected.



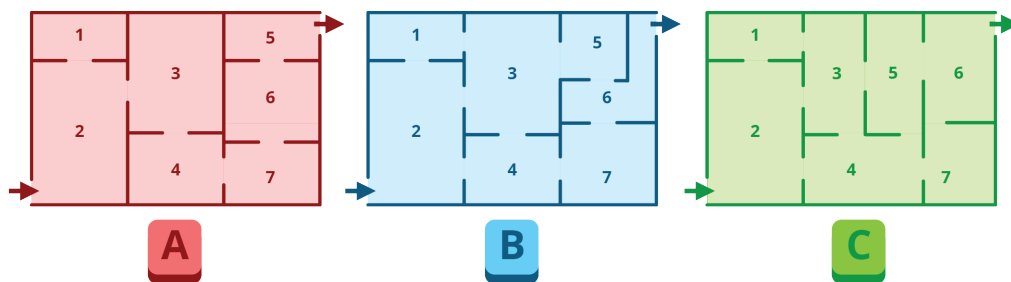


Museum Tour

Castor the architect was asked to design a museum.
Castor produced 3 designs.

He wants to choose a layout that allows visitors to go through all of the rooms exactly once, without visiting a room more than once, and without using the same door for entering and exiting. This is called a one-way tour.

The visitors must start at the door with the arrow that enters the museum and leave by way of the door with the arrow leaving the museum.



Question

Which one of the statements below is correct?

A one-way tour is possible in layout A.

A one-way tour is possible in layout B.

A one-way tour is possible in layout C.

A one-way tour is not possible in any of the layouts.

Answer

Only layout C allows a visitor to take a one-way tour.

The sequence of rooms is: 2, 1, 3, 5, 4, 7, 6. In general, if a room has only one entrance, a one-way tour is not possible: If visitors enter this room, they will have to go back to the room they came from and hence visit it twice.

In layout A, room #1 has only one entrance. In layout B, the last room #6 can be reached from room #5 and room #7. If one chooses to go via room #5, one can only visit room #7 and reach the exit afterwards only by visiting room #6 twice (and vice versa).

It's Computational Thinking

Computational Thinking Skills: Abstraction, Modelling and Simulation, Algorithms, Evaluation

Concepts: Abstraction, Data Representation, Data Interpretation, Specification, Algorithms

To be solved by a computer, we must model the problem in a way that is better understood by computers. For finding the requested tour, it is most important to know about which rooms are connected by a door (so that you can go from one to the other).

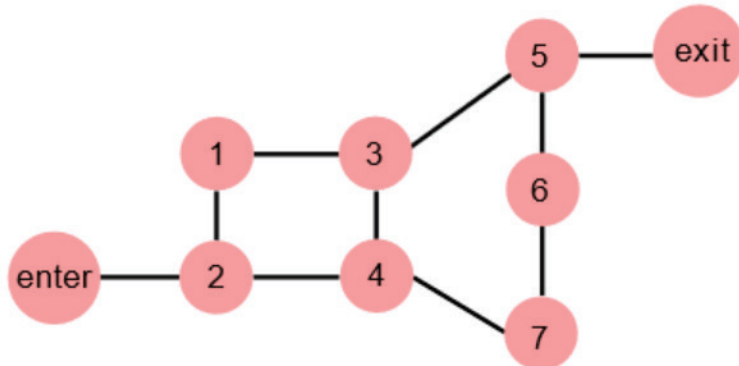
Continued on next page



Museum Tour - continued

It's Computational Thinking - continued

So, for example, layout C can be modelled as follows:



A circle indicates a room (plus the “outside rooms” before entering and after exiting), while a line indicates a door. This model is called a graph: circles are called vertices and lines are called edges. There are many problems that can be solved by computers using graphs, like finding the shortest route, determining friendships in social media, and others.

In the graph model, finding the exhibition tour corresponds to finding a path along the edges that visits all vertices. Graph theory tells us that this problem is inherently difficult. In this Bebras task, the number of rooms and doors is small, and we can solve it even without computers. However, if the number of rooms and doors gets large, even computers may fail to find a tour.



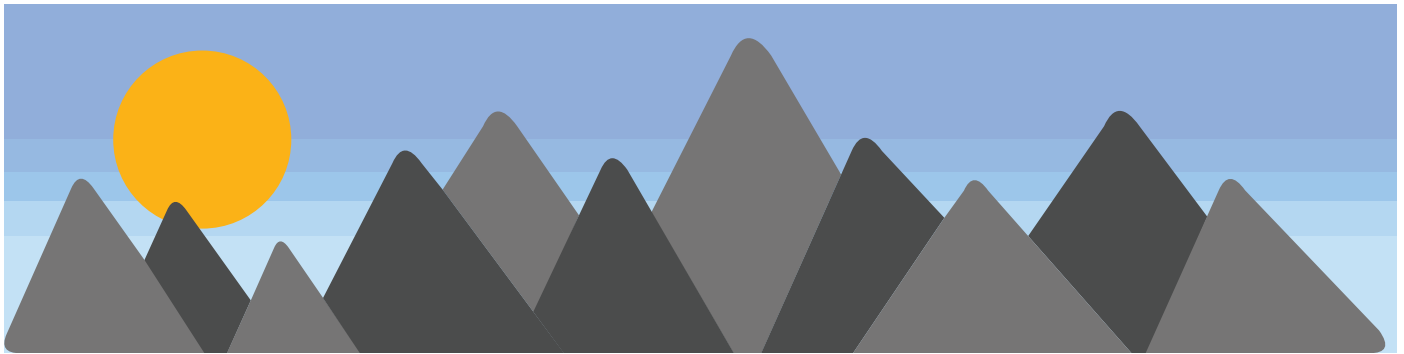
Mountain Climber

Binsa is a mountain climber. She loves the mountain range known as Eleven Peaks.

Binsa will always climb a mountain right beside her if its peak is higher than the one she is on. If there are two mountains right beside Binsa and both of these mountains have higher peaks, she will always climb to the highest one. Binsa finishes her day's climbing when both of the mountains beside her have lower peaks than the mountain she is currently on.

Question

Select all the mountains where Binsa can start her day's climb which will result in her reaching the highest peak.



Answer

If Binsa begins on either of the mountains next to the highest peak, she will immediately climb to the highest peak. So when we also include the highest peak, there are at least three mountains from which Binsa will reach the highest peak.

If Binsa begins further to the left than the mountain immediately next to the highest peak, she will never get past the 5th peak. If Binsa begins further to the right than the mountain immediately next to the highest peak, she will climb to the 10th peak and stop.

This means that there are exactly three peaks from which Binsa will reach the highest peak.



It's Computational Thinking

Computational Thinking Skills: Abstraction, Modelling and Simulation, Algorithms, Evaluation

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

The process that Binsa follows is called a greedy algorithm. The idea is that while she may be trying to reach the highest peak, she decides what to do next only by looking at the two mountains right beside her. That is, instead of trying to look for a global maximum, instead she looks for a local maximum.

Greedy algorithms do not always correctly solve problems but when they do, they usually do so very efficiently. This means computer scientists are interested in problems to which greedy algorithms apply. However, even if a greedy algorithm is incorrect, sometimes they can be used to find a solution that is "good enough". This is called an approximation algorithm.

A specific type of greedy algorithm used in artificial intelligence is called the hill-climbing algorithm. The name from this technique comes from the idea behind this problem. The story behind this task is related to an idea from geography called topographic isolation. One of the great things about studying Computer Science is how it can be applied to almost any other subject area.



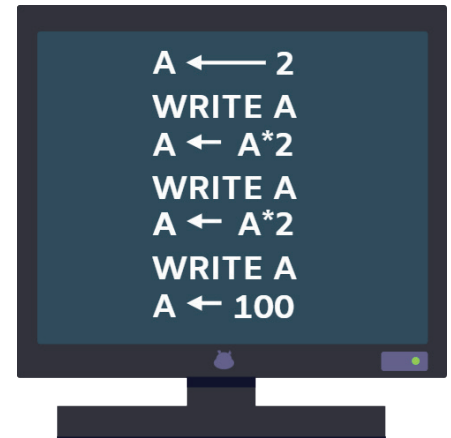
Backward Bug

My computer normally reads programs line by line, from top to bottom. When running the program shown below, it prints out:

2
4
8

The symbol \leftarrow means 'store the calculated value on the right in the variable on the left'.

E.g. In the first line, 2 will be stored as a variable called A.
The symbol * is the symbol for multiplication.



Today my computer is behaving strangely. It is reading programs line by line, from bottom to top.

Question

What will my computer print out when running this program today?

8	100	2	100
4	50	4	200
2	25	8	400
		100	

Answer

Correct answer is D.

First we reorder lines so that we could read them from the top:

A \leftarrow 100
Write A
A \leftarrow A*2
Write A
A \leftarrow A*2
Write A
A \leftarrow 2

When read from the bottom, first command to execute is A \leftarrow 100 and second command is Write A.
So the first written number must be 100.

If the program is read backwards, it does not mean that the math operations in the commands work inversely. The command A \leftarrow A*2 does not mean A \leftarrow A/2 when the program is read backwards. If the program contains only multiplication commands to multiply the number by 2, the written value must grow. So next values cannot be 50 and 25.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Modelling and Simulation, Algorithms

Concepts: Specification, Algorithms, Digital Systems, Interactions

Reading computer programs is an important thing, we need to know how a computer will execute the program and how it executes commands. Every command has to be in the correct place in the program if it is to work correctly. Programmers have to know how results change when position or order of commands in a program changes.



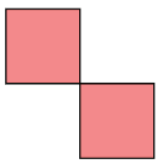
Bominos

Beavers love Bominos. Bominos are patterns made of tiles, arranged in a grid.
The tiles of a Bomino must satisfy these conditions:

1. Each tile touches at least one other tile (horizontally, vertically, or diagonally)
2. There is at least one pair of tiles where:
 - two tiles touch diagonally AND
 - there is no tile that touches both these tiles horizontally or vertically

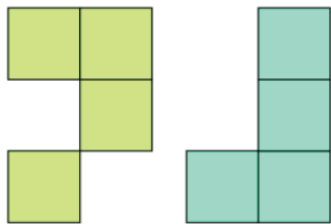
A Bomino with n tiles is called an n -bomino.

Example



This is the only possible 2-bomino.

All other arrangements of two tiles that satisfy the bomino conditions can be rotated or flipped to produce the same figure.

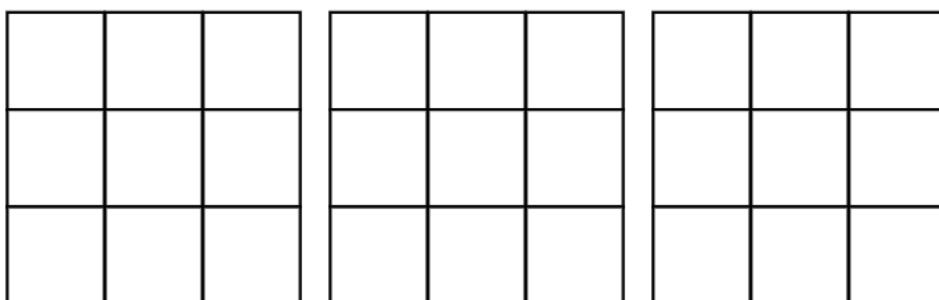


The left pattern is a 4-bomino. The right one is not.

Question

There are three possible 3-bominos (bominos made of three tiles).

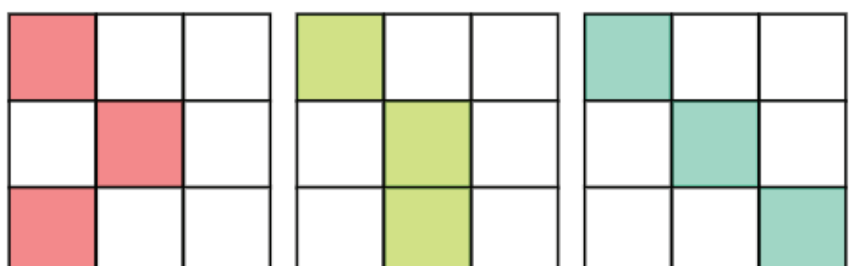
Create them below.



Please select 3 squares in each figure.

Answer

- One needs to find all the 3-bominos
- There is just one 2-bomino (this info is given in the text)
- The grid has nine tiles, 2 are taken by the 2-bomino: seven possibilities remain



Continued on next page



Bominos - continued

Answer - continued

- Two can be excluded easily because of the second rule 'there is no tile that touches both these tiles horizontally or vertically'
- The other positions are all good, but 2 have symmetries: $7 - 2 - 2 = 3$

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Modelling and Simulation, Algorithms

Concepts: Abstraction, Data Representation, Data Interpretation, Specification, Algorithms

The rules given in the task define bominos, but they do not explicitly give a way to build or draw a valid bomino. This kind of precise description (often called declarative) are very common and useful for reasoning about the general properties of the objects, but they are not suitable if your goal is to give the instructions to build one of them.

When we want to write a program for common computers, for this, an operational description is needed. Fortunately, in some cases it is possible to automatically translate the declarative definition into an operational one, and some programming languages provide constructs for both.

Curriculum and Computational Thinking skills alignment by Allira Crowe and Graeme Buckie.

This Solutions Guide was created by Hannah Piper and edited by Ruwan Devasurendra.

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

digitalcareers.csiro.au