Bebras Australia
Computational Thinking Challenge

2024 Solutions Guide Round 1
Primary Divisions | **Years 3 to 6**

AUSTRALIAN MATHS TRUST

CSIRO

Bebras
COMPUTATIONAL THINKING CHALLENGE

Bebras Australia Computational Thinking Challenge

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

Bebras is a fun and engaging computational thinking challenge for students in Years 3 to 12. It is an international challenge that involves over 2.9 million students from 60 countries.

Bebras creates opportunities for students to engage in activities that use and develop their critical and creative thinking and problem-solving skills essential to further learning.

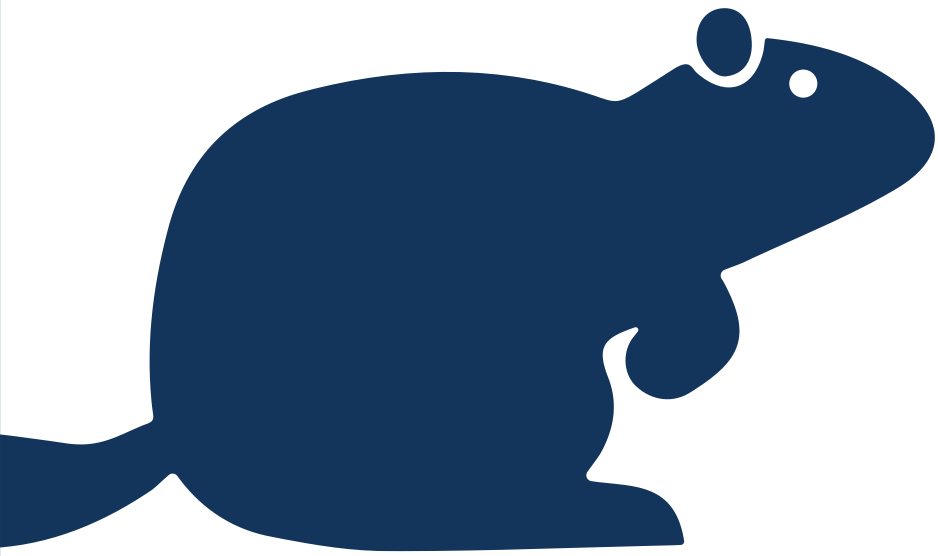
Coding skills are not required to complete the Bebras Challenge. The challenge is open twice a year for three weeks and each round has different questions. Students can participate individually or in teams of up to four.

Lithuanian for beaver, Bebras was the name chosen by the founder of the challenge, Professor Valentina Dagiene from the University of Vilnius, in honour of the animal's collaborative nature and strong work ethic.

The Challenge is coordinated by the International Bebras Committee which meets annually to assess potential questions and share resources. Bebras Australia began in 2014 and was delivered by CSIRO until 2023. From 2024, Bebras Australia is delivered by the Australian Maths Trust.

With support from CSIRO, the AMT provided the Bebras Challenge free of charge for Australian schools in 2024.

To find out more and register for the next challenge, visit www.amt.edu.au



Contents

What is a Solutions Guide?

Within this Solutions Guide you will find all of the questions and tasks from Round 1 of the Primary divisions of the 2024 Bebras Australia Computational Thinking Challenge.

On each page you will find a question, answer, an explanation, and background information explaining the skills and key curriculum concepts featured.

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.

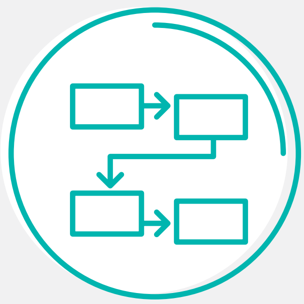
Computational Thinking uses a number of skills, including:

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.

Visit the AMT website for more Bebras resources https://www.amt.edu.au/bebras

Computational Thinking skills alignment

Question	Difficulty	Decomposition	Pattern Recognition	Abstraction	Modelling & Simulation	Algorithms	Evaluation
Middle Primary Year 3 & 4							
Zoo Walk	Easy	●			●		●
Hat Lineup	Easy	●		●		●	
Sliced Apples	Easy	●	●	●	●		●
Ball Sort	Easy	●		●		●	
Magic Tree	Easy	●		●		●	●
Riccas	Medium	●	●				
Tile Land	Medium	●		●		●	●
Umbrella	Medium	●	●			●	●
A Day at the Zoo	Medium	●		●		●	●
Flower Shop	Medium	●				●	●
Photo	Hard	●	●	●		●	●
Luggage Check-in	Hard	●		●	●	●	●
Karla's Dream House	Hard	●		●		●	●
Planting Carrots	Hard	●		●		●	●
Beaverbeard's Gold	Hard	●		●	●	●	●
Upper Primary Year 5 & 6							
A Day at the Zoo	Easy	●		●		●	●
Luggage Check-in	Easy	●		●	●	●	●
Photo	Easy	●	●	●		●	●
Flower Shop	Easy	●				●	●
Umbrella	Easy	●	●			●	●
Riccas	Medium	●	●				
Karla's Dream House	Medium	●		●		●	●
Planting Carrots	Medium	●		●		●	●
Beaverbeard's Gold	Medium	●		●	●	●	●
Thomas and His Friends	Medium	●		●	●	●	●
Closer or Further	Hard	●		●		●	●
Unloading	Hard	●		●		●	●
Sawya's Logs	Hard	●		●			●
Companion Planting	Hard	●		●		●	
Beaver Bundles	Hard	●	●	●		●	●

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

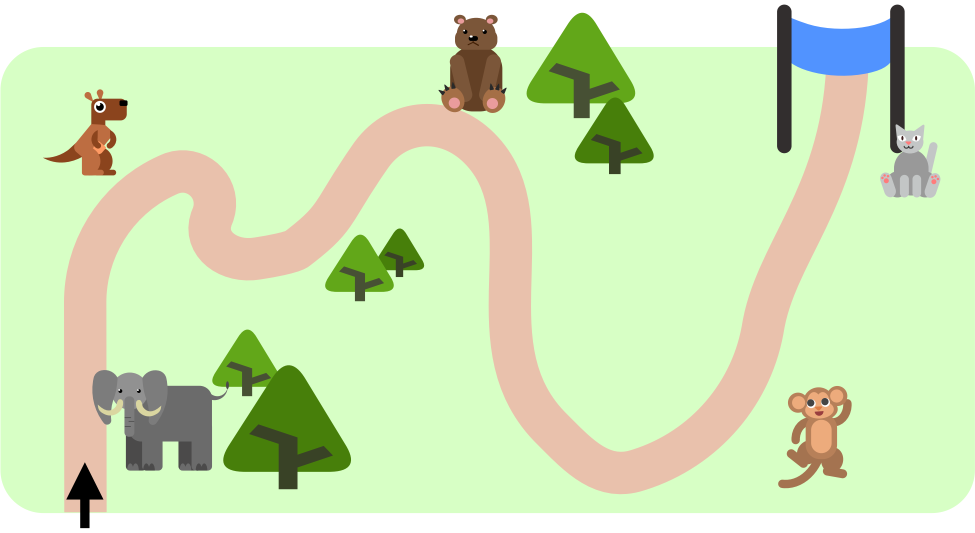
Question	Difficulty	Abstraction	Data Collection	Data Representation	Data Interpretation	Specification	Algorithms	Implementation	Digital Systems	Interactions	Impacts
Middle Primary Year 3 & 4											
Zoo Walk	Easy			●					●		
Hat Lineup	Easy	●				●	●	●			
Sliced Apples	Easy	●		●	●	●					
Ball Sort	Easy	●		●		●	●		●		
Magic Tree	Easy	●		●			●		●		
Riccas	Medium		●	●	●						
Tile Land	Medium	●		●	●	●	●				
Umbrella	Medium			●	●	●					
A Day at the Zoo	Medium	●	●	●		●	●	●			●
Flower Shop	Medium		●	●		●	●	●			
Photo	Hard	●		●	●	●					
Luggage Check-in	Hard	●		●	●		●	●	●		
Karla's Dream House	Hard	●		●		●	●				
Planting Carrots	Hard	●		●	●	●	●	●			
Beaverbeard's Gold	Hard	●		●	●	●					
Upper Primary Year 5 & 6											
A Day at the Zoo	Easy	●	●	●		●	●	●			●
Luggage Check-in	Easy	●		●	●		●	●	●		
Photo	Easy	●		●	●	●					
Flower Shop	Easy		●	●		●	●	●			
Umbrella	Easy			●	●	●					
Riccas	Medium		●	●	●						
Karla's Dream House	Medium	●		●		●	●				
Planting Carrots	Medium	●		●	●	●	●	●			
Beaverbeard's Gold	Medium	●		●	●	●					
Thomas and His Friends	Medium	●	●	●	●	●	●			●	
Closer or Further	Hard	●	●	●	●	●	●		●		
Unloading	Hard	●				●	●	●			
Sawya's Logs	Hard	●				●	●		●		
Companion Planting	Hard	●		●		●	●				
Beaver Bundles	Hard	●				●	●	●			

2024 Bebras Challenge

Round 1 | Middle Primary Year 3 & 4

Zoo Walk

Marko went for a walk in a zoo. As he walked, he drew a map of his walk.
On the map he added pictures of the animals he saw, showing where he saw them.



Question

Which of these sentences is **not** true?

- **Marko saw the cat last.**
- **Marko saw an elephant first.**
- **The second animal Marko saw was a monkey.**
- **After Marko saw the kangaroo, he saw a bear.**

Explanation

Answer:

The second animal Marko saw was a monkey.

Explanation:

Only the sentence about the monkey is false (not true). Marko saw the monkey as the fourth animal, after the bear, or as the next to last animal.

The other sentences, or the statements, describe the information in the picture correctly.

Background information

This task focuses on working with data representation and comparing the correctness of the information in the statements. The information stored in the image shows the sequence of steps taken (seeing animals during the walk). We can represent this information in various other structures as well. The questions offers 4 statements that are true or false.

A computer scientist often deals with similar situations during programming, where they must represent information as a data structure and use that structure to answer true or false statements.

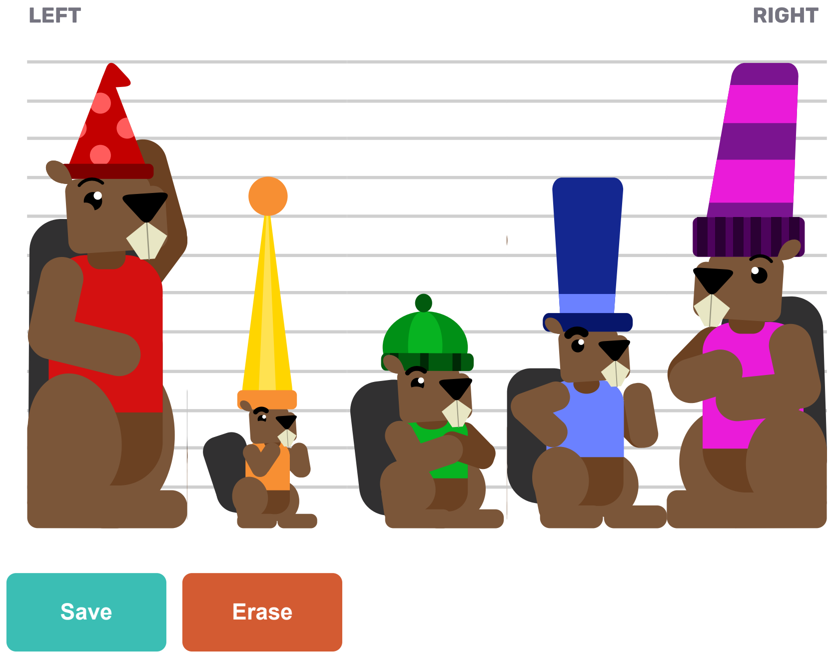
Hat Lineup

Sort the beavers in ascending order according to their hats' heights.

Question

Move the beavers by dragging them. The beavers to the right must have taller hats than those to their left.

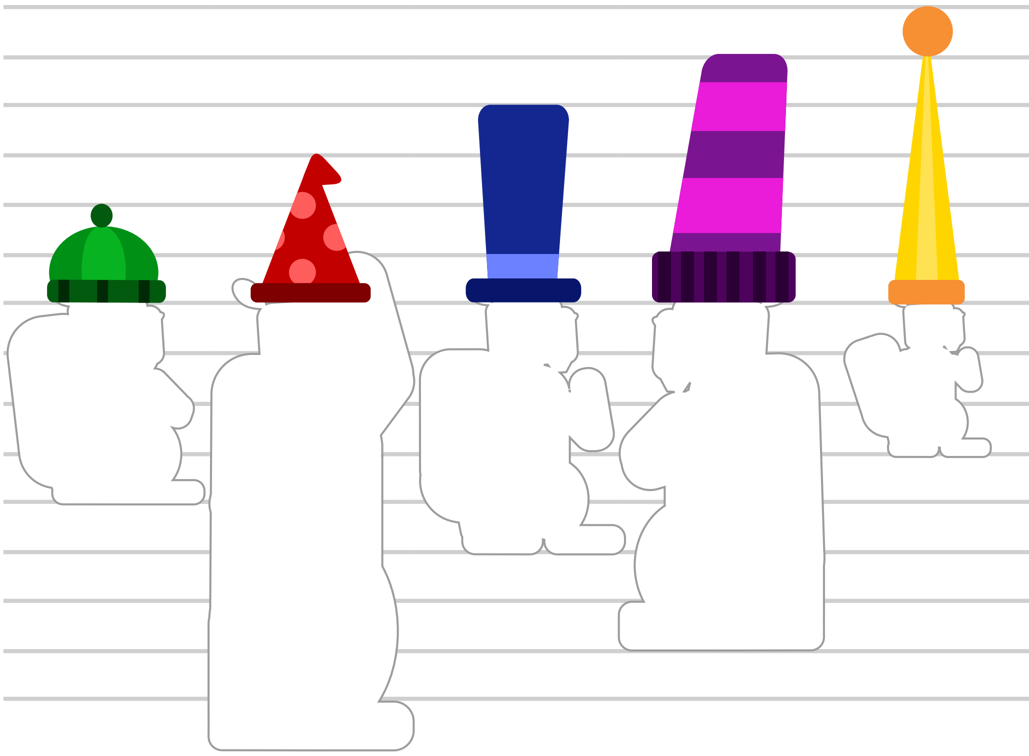
Press Save when complete.



Explanation

The correct answer is green, red, blue, purple then yellow.

We have to look at the size of the hats. When you align the hats next to each other, you get the image below.



Background information

In informatics, arranging in an ordered sequence is called sorting. It is astonishing how many things are sorted in our daily life. The lockers at a water park, the entries in dictionaries or telephone books. Without sorting the entries in a dictionary, it is extremely hard to find a required item. Even the results of any internet browser request are ordered according to their relevance. The results of nearly all sport events are ordered due to the achievements. In all these cases the sorted items are easier to handle than unsorted items. The advantage is even more evident when there is a large number of items.

First of all, it is necessary to analyse the problem and to find out that to solve this problem, one needs to separate the height of the beavers from the height of their hats. Then by analysing the data (the sequence of beaver hats), a sequence of rearrangements has to be found. An algorithm to solve this problem is based on the idea that moving taller hats to the right of hats with lower height leads closer to the desired goal. This step is repeated until all beavers are in order.

Sliced Apples

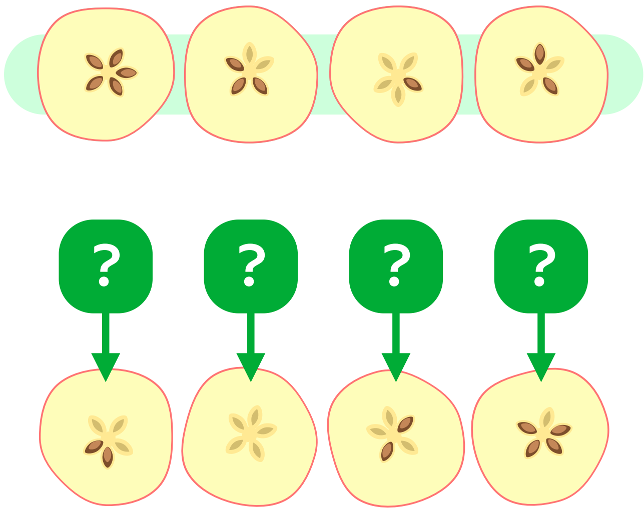
When an apple is cut in half, a star pattern is seen on both halves. This is where the seeds were. The seeds do not get cut in half. Each seed ends up in only one of the apple's two halves.

Vladislava cut 4 apples in this way and put both halves on the table. However, the apple halves in the top row got mixed up.

Question

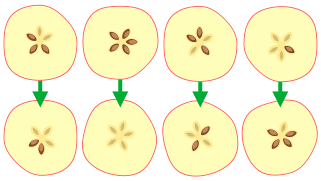
Drag the top apple halves on to the question marks, so that the two halves of each apple are back together.

Press Save when you have completed this task.



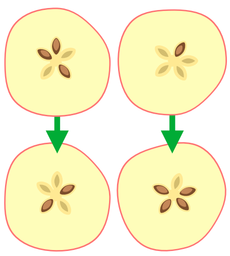
Save Erase

Explanation

The correct answer is  .

The number of all seeds in one apple must be 5. But counting them only is not enough.

Two apples with equal number of seeds in one half can be distinguished according to the pattern of full and empty seed beds placement. The halves in the right in the picture have all full seed beds placed together, without empty bed between them. The halves in the left have one full seed bed isolated, with empty beds on both sides.

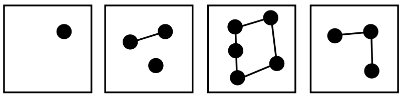


Background information

Some objects in real life must to fit together, for example, pairs of A-T and C-G of nucleobases in DNA, or locks and keys. In computer science for each password-protected file (lock) there is a password to open it (key). In programming, some functions require a set number of input parameters. It is the compiler's task to check if the provided number of parameters in a function call and their type match with the expected values by the function definition.

To solve this task, it is necessary to find how the halves of one apple fit together. Finding a supplement and topological pattern recognition are the main skills necessary to solve the task.

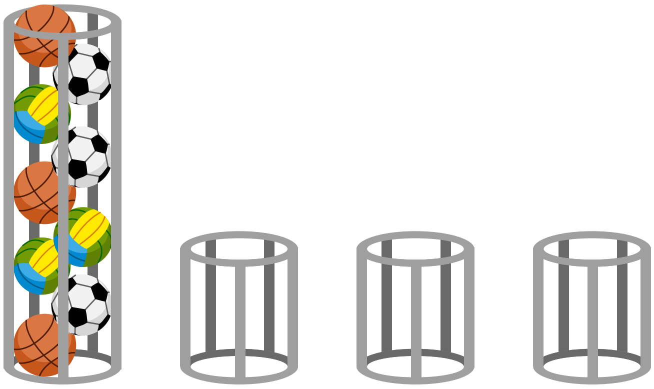
We can create a graph for each half in 1-st row of correct answer image. A node means a full seed bed and an edge means neighbourhoods of two full seed beds.



We can also create similar graph for 2nd row of halves with switched full and empty seed beds (where a node means an empty seed bed and an edge means neighbourhoods of two empty seed beds). Then we get the same graphs for halves belonging to each other.

Ball Sort

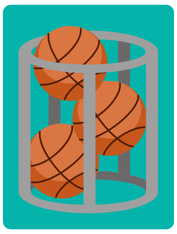
There are 9 balls in one big container. There are three different types of ball.

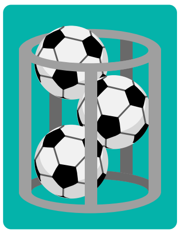


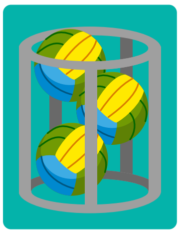
Sarah wants to sort each type of ball into separate containers. To sort them, Sarah takes the balls, one by one, from the top of the big container and puts them in the container for its type.

Question

Which ball container will Sarah fill first?

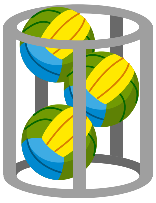






Explanation

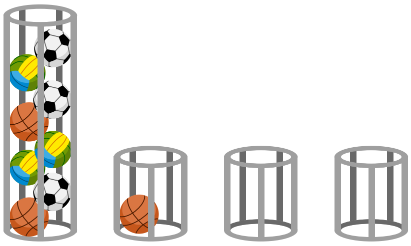
Answer:



Explanation:

Since Sarah takes the balls one by one, she takes them in order from top to bottom. She will fill the first tube of balls when she takes the third ball of that type.

The third ball that will come out first from the tube is the volleyball:

1. 

2.

3.

4.

5.

6.

Background information

The student takes out the balls from the top of the container. In computer science, this data structure is called a stack. The stack is a linear data structure used to store data by adding or removing elements from a single end, called the top of the stack.

There are many real-life examples of a stack such as a stack of plates or books, where you can take the item from the top first while the item on the bottom will be the last item taken.

Magic Tree

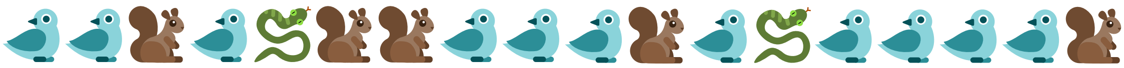
Bain the Beaver has a magical tree growing near their home.

Whenever a bird lands on it (), the tree grows 2 apples.

Whenever a squirrel climbs up it (), the tree drops 1 apple (if it has any).

Whenever a snake visits the tree (), all of the apples instantly disappear!

One morning Bain notes that the magical tree contains 25 apples.
Bain then spends the rest of the day drawing pictures of all the animals that come to the tree.
The drawings, in order, are:



Start of the day — End of the day

Question

How many apples are on the tree at the end of the day?

3 | 7 | 17 | 31

Explanation

The answer is that there are 7 apples on the tree at the end of the day.
Since all of the apples instantly disappear whenever a snake visits the tree, we can ignore everything that happens before the arrival of a snake.
After the last snake, four birds land on the tree which means the tree will sprout 4 × 2 = 8 apples.
Then one squirrel climbs up the tree which causes one apple to drop, leaving 8 – 1 = 7 apples.

Background information

This task introduces the ideas behind two fundamental programming concepts.

The first is the idea of a *variable*. A variable is used to store information that a computer program needs. The value of a variable can change depending on what the rest of the program's instructions are. In this task, the number of apples on the tree is a variable and its value can increase (), decrease (), or reset ().

In order to decide how to change the value of a variable, a computer program needs the ability to make decisions. This is the second fundamental programming concept and is referred to as *selection*. Decision-making is accomplished by using special instructions called *conditional statements* that allow you to select from different possible outcomes.
They commonly take the form "if **this**, then **that**".

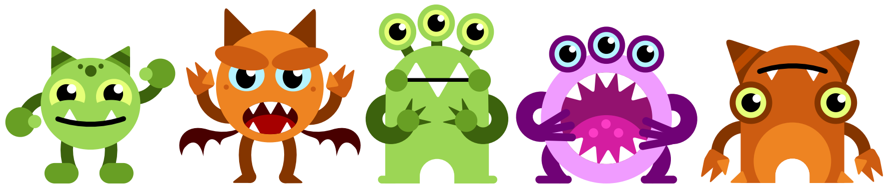
In this task, one conditional statement would be "if a bird lands on the tree, then increase the number of apples by 2".
Can you find other conditional statements in this task?

The task is in the area of abstraction, as students need to distinguish between the different cases in the task and focus only on the information they need. To solve the problem most efficiently, it is sufficient to examine the data from the last snake. The preceding markings are not relevant for us, since we always set the number of apples to 0 when the snake appears.

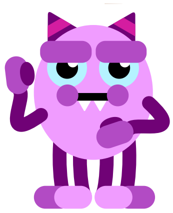
Also the solution of the task requires algorithmic thinking or algorithmic design because students need to not only solve the task step by step, but they need to choose solution depending on situations just like in case of an 'if' statement.

Riccas

Evren is trying to learn what a Ricca looks like. Evren studies the following five photos of Riccas and makes some notes that accurately describe what she sees.



Evren is then shown this sixth photo of a Ricca and realises one of her notes is definitely wrong.



Question

Which one of Evren's notes about Riccas is definitely wrong?

- **Riccas always have teeth.**
- **Riccas sometimes have wings.**
- **Riccas have either horns or three eyes, but not both.**
- **If Riccas have exactly two arms then they also have exactly two legs.**

Explanation

The answer is: If Riccas have exactly two arms then they also have exactly two legs.
It is not true that if a Ricca has exactly two arms then it also has exactly two legs. The sixth photo shows a Ricca with two arms and four legs.

The note "Riccas always have teeth" may still be true. All six photos show Riccas with teeth.

The note "Riccas sometimes have wings" is definitely true. The second photo shows a Ricca with wings.

The note "Riccas have either horns or three eyes, but not both" may still be true. Photos one, two, five, and six show Riccas with horns and "only" two eyes. The remaining photos (three and four) show Riccas with three eyes and no horns.

Background information

Evren is learning to identify a Ricca by studying photos of Riccas and looking for similar characteristics. Computers can also be taught to do this, using a process called comparison. Nowadays, we can employ machine learning to compare a vast number of items automatically and in a short time. As part of machine learning, a computer is presented with a collection of data, and then it searches for patterns within this data to draw conclusions. Just like with Evren, it is possible for a computer to arrive at the incorrect conclusion. By providing the computer with more data, and data that is more representative and inclusive, the computer can correct its mistakes and update its learning.

Detecting recurring patterns and regularities in given data is an important skill in informatics, generally referred to as pattern recognition. In programming, recognizing patterns allows us to adopt a modular approach, wherein we need to solve a recurring subtask only once. This is the foundation of efficient algorithms and software design, as it reduces code redundancy and enhances maintainability. Pattern recognition is also crucial for machine learning and artificial intelligence, as it empowers computers to learn from data and make predictions or decisions. For example, in image recognition, pattern recognition techniques assist in identifying objects in images, classifying them into different categories, and even recognizing faces. In natural language processing, pattern recognition enables computers to understand the structure of sentences, identify topics, and even generate human-like text.

If care is not taken in the selection of data for machine learning, bias may be introduced into the computer system. Consider this: what impact would a speech recognition system have on society if it were trained only on data from male native English speakers?

Pattern recognition is related to data measurement and data analysis. To solve tasks that involve these concepts, among other skills, algorithmic thinking is crucial. Students apply this when they develop a step-by-step ordered procedure to solve a problem, using their understanding of patterns, rotations, and spatial relationships. Data analysis, which includes identifying which symbols are in play and understanding their positions, allows students to make inferences and deductions based on the given information and recognized patterns. Finally, evaluation is crucial to assess the effectiveness of their solutions and identify areas for improvement.

Tile Land

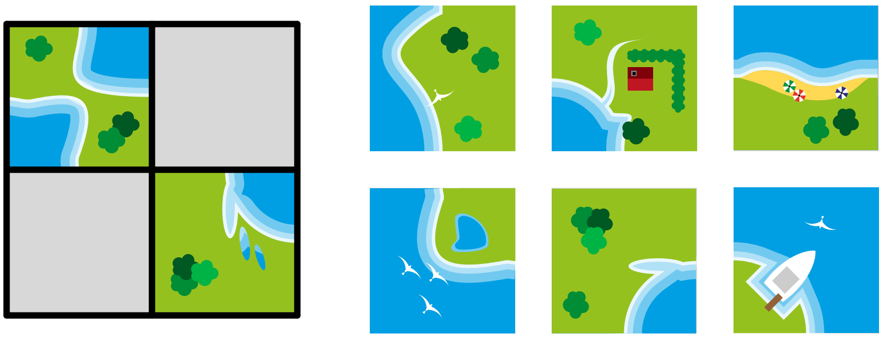
Edu has a new game called *Tile Land*. It has tiles that show areas of water and land.

Using the tiles, Edu can form landscapes.
Tiles with touching edges must match: land must touch land and water must touch water.

Examples:
The two tiles on the left have been placed correctly. The tiles on the right have not.

Now, Edu places two tiles as shown below, with gaps for two more tiles.

Question
Fill the gaps with matching tiles.
(Drag two of the tiles from the right into the two gaps on the left.)
Press Save when complete.



Save Erase

Explanation

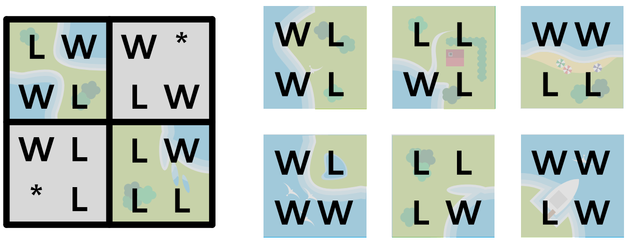
This is the correct answer:

We immediately see that this answer is correct: Each added tile matches its two neighbouring tiles. For each gap, we can look at all six tiles and see that only one tile can fill the gap, matching both neighbouring tiles. Hence, this is the only correct answer.

Background information

Let us take another, closer look at Edu's tiles. First, we realize that the tiles can be divided into sections, with the two outside borders of each section showing either both land or both water.

Hence, there are only two different types of sections. We can label them "W" (for sections with their outside borders showing water) and "L" (for sections with their outside border showing land).



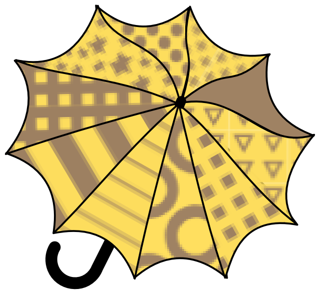
We also realize that two tiles match if their neighbouring section types are equal. Hence, for most sections of the gaps, we can fill in the required types already. For the two sections in the outer corners of the gaps, we do not care about the type and fill in a '*'. Thus, for each gap, we get a section type pattern. The tiles to fill the gaps must fit these patterns. For each pattern, only one tile on the right fits the pattern.

In summary, we discovered a special property of the tile images, and used this discovery to replace them by an arrangement of characters L and W. By this step, we reduced the information contained in the images significantly, in order to focus on what was needed to solve this Bebras task. Computer Scientists would refer to the character arrangements as *models* of the images. Modeling involves abstraction, and abstraction reduces information. Computers need to work on models of reality. In the process of creating such models, it must be taken care that important properties of reality will not get lost.

It certainly is; you may explicitly mention "abstraction" here.

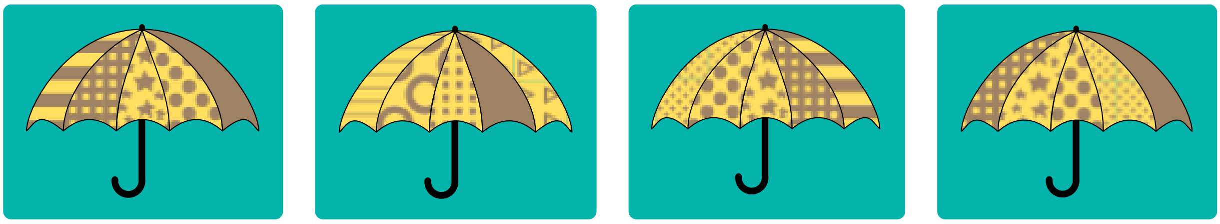
Umbrella

This is Anna's umbrella:



Question

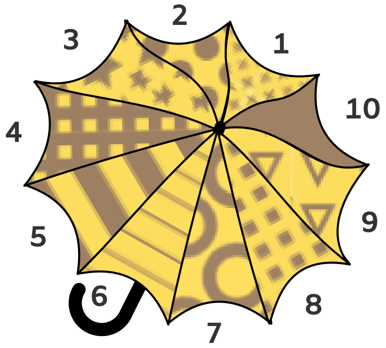
One of the four images below shows Anna's umbrella. Which one?



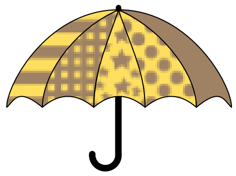
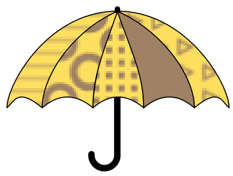
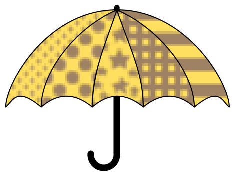
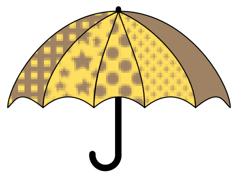
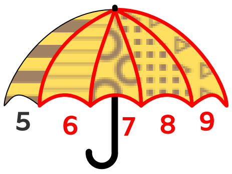
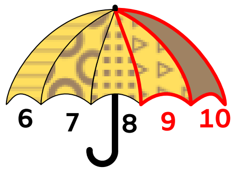
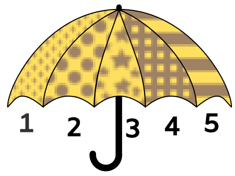
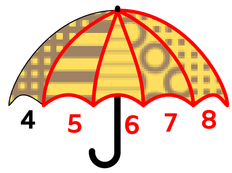
Explanation

Every pattern is used only once in Anna's umbrella. Therefore, we can compare each image with Anna's umbrella this way:

- Pick the left-most pattern from the image and find its position in Anna's umbrella.
- Check if the following patterns match Anna's umbrella.



In each image we can see a sequence of only five patterns. So we cannot decide whether the image matches the full sequence of all ten patterns of Anna's umbrella. However, only image C has a sequence of five patterns that fully matches Anna's umbrella. For this reason, only umbrella C can show Anna's umbrella. All other images show pattern sequences that only partially match that of Anna's umbrella; so they cannot show Anna's umbrella.

	A	B	C	D
answer images				
Anna's umbrella				

Background information

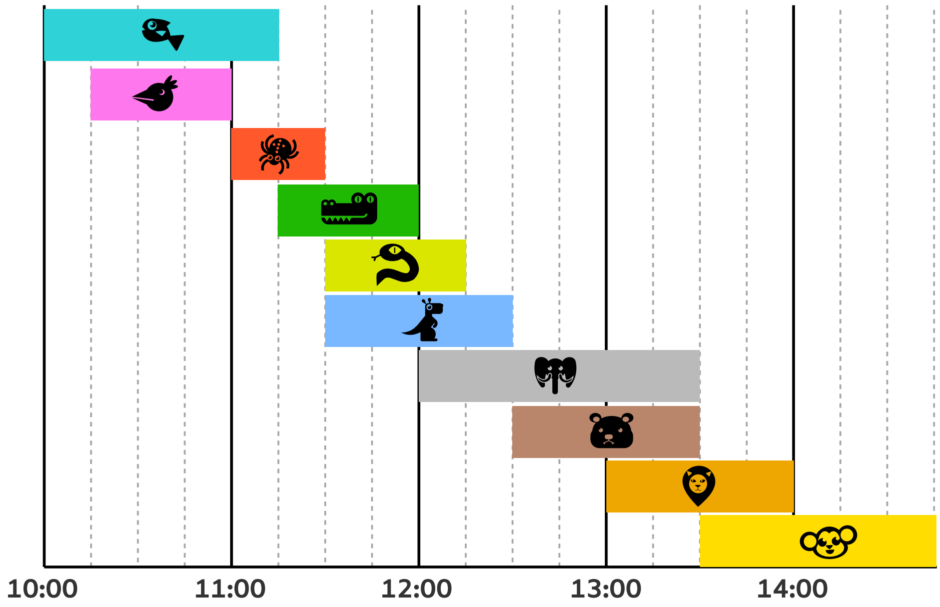
In this Bebras task, we gain only partial information about the umbrella patterns shown in the answer options. Nevertheless, we can determine which image shows Anna's umbrella: We can look at the four pattern sequences and notice that only one also occurs in Anna's umbrella.

The same principle as for the "umbrella search" is applied when searching a text document. The computer searches with the given partial information ("search sequence" of letters) for matching sequences of letters in the document. You can observe that the longer your search sequence, the less matches you will find and the higher your chance to find the spot in the document that you are looking for. The shorter your sequence, the less accurate, but the faster your search. In computer science, search procedures (search algorithms) have been designed (and are still being improved) in order to obtain the most suitable results as quickly as possible. These algorithms are capable of searching in large amounts of data, like the internet search tools you may know.

The problem can be solved by using decomposition. Decomposition means breaking down a problem into smaller problems or steps. To find a matching sequence of 5 patterns you can break it down to the steps described in the answer explanation. This task is also a very practical exercise in pattern recognition. Pattern recognition is the ability to recognize regularities, repetitions or similarities in a set of data. To find the favorite umbrella it is necessary to quickly find the matching sequences of the patterns. Later, when programming, pattern recognition allows us to use a modular approach, where have to solve a recurring sub task only once.

A Day at the Zoo

Anja spends a day at the zoo. There are many different activities.
Each activity has a start and an end time, and visitors cannot leave in the middle of an activity.



Anja wants to attend as many activities as possible, but she can only attend one activity at a time. She can't leave in the middle of an activity, so she has to attend activities from beginning to end. Since some of the activities overlap, Anja cannot attend all of the activities today.

Question

What is the maximum number of activities Anja can attend during a single day?

Save

Explanation

Answer:

5

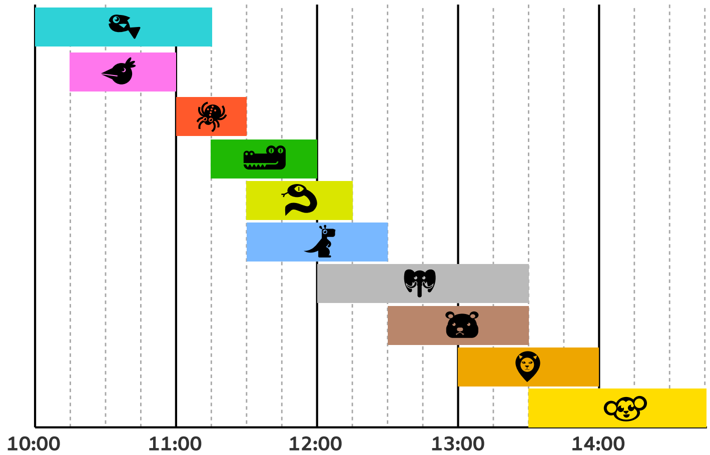
Explanation:

Anja can attend a maximum of 5 activities over the entire day.

One way to solve this problem is to try all possible combinations of non-overlapping activities. Each combination of non-overlapping activities gives us a valid schedule. Then we check which schedule has the most activities (an *optimal* schedule). This is called an *exhaustive* search.

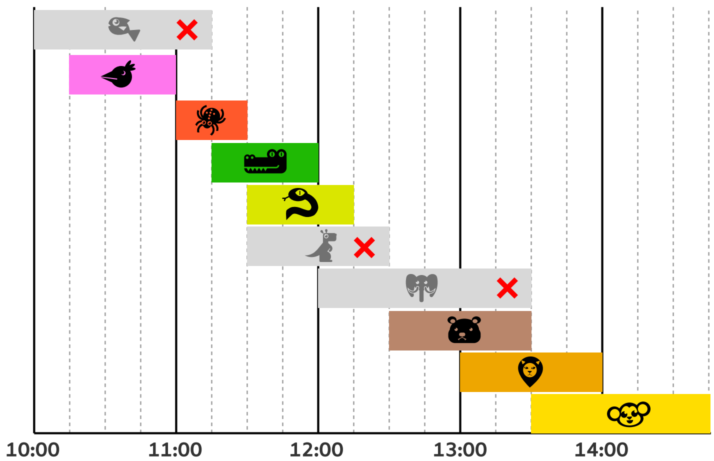
An exhaustive search can be difficult since there are too many possibilities to check. It's also hard to make sure that you've checked everything.

We can make an observation to *simplify* the problem. If two activities overlap, then only one of the two activities can be part of an optimal schedule. If one activity takes place completely during another activity, then it is always better to take the shorter activity.

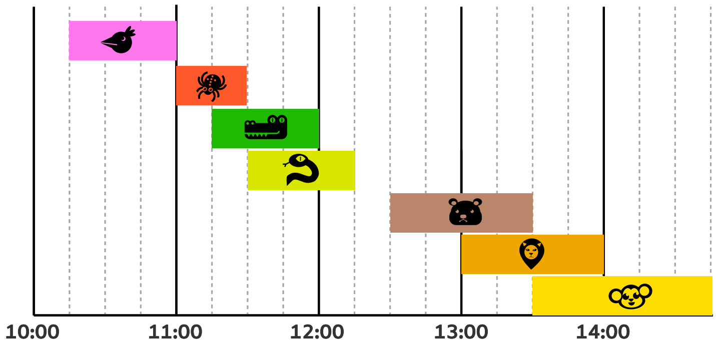


For example, the bird show  takes place completely during the aquarium light show  and it is shorter, it is better not to choose the aquarium light show, because it may overlap with a larger number of other activities.
With the same reasoning, we can do the following.

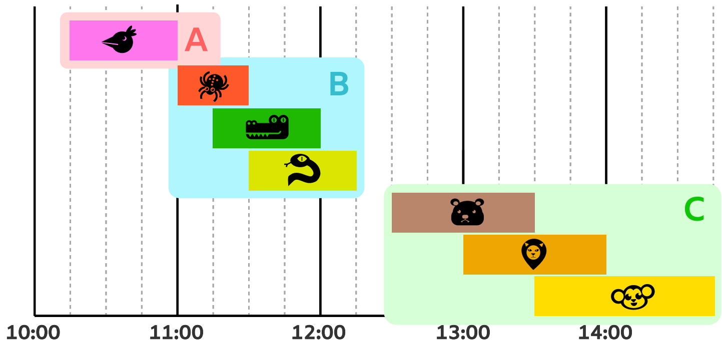
- We can exclude the kangaroo feeding , since the reptile show  takes place completely during the kangaroo feeding.
- We can exclude the elephant showering , since the bear feeding  takes place completely during the elephant showering.



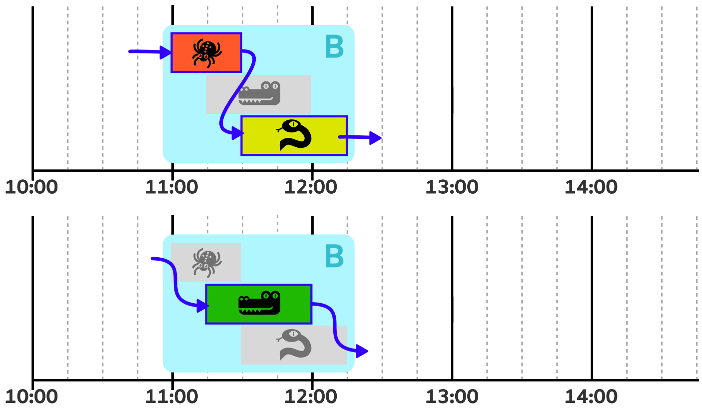
After deleting these three activities, we are left with these activities.



We can decompose (or break up) these activities into independent groups, which we call A, B and C. We can find the optimal schedule for the entire day by finding the optimal schedule for each group.

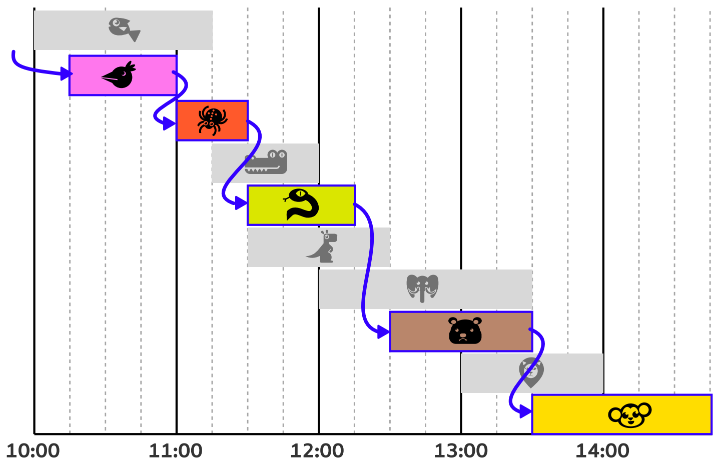


For group A, there is only one activity: the bird show . The optimal schedule is to attend the bird show.
For group B, there are three activities. Conducting an exhaustive search, there are only two valid schedules for group B.



The optimal schedule is to take the spider show  and reptile show  activities.
For group C, there are also three activities. Using a similar reasoning, the optimal schedule is to take the crafting and seal show activities.

Putting the optimal schedules together, we have the following schedule for the entire day.



There are 5 activities in this schedule, so the answer is 5.

Background information

Often in computer science, when designing an algorithm to solve a problem, the easiest algorithm to think of is an exhaustive search, as it is conceptually simple and guaranteed to give the correct answer. The main problem is that an exhaustive search can take too much time to do, either by hand or with a computer. We need to find ways to make more efficient algorithms to solve problems in a more reasonable amount of time.

In order to make an algorithm more efficient, we often have to make observations about the problem which allow us to simplify the problem. A problem where you maximize or minimize a quantity is called an optimization problem. When simplifying an optimization problem, we have to think about what decisions we can make which do not prevent us from obtaining an optimal solution. In the task, we were able to disregard the aquarium light show, koala feeding, and elephant showering activities, since we can be sure any schedule containing these activities can be potentially improved by picking a different activity instead.

Another strategy which allows us to make an algorithm more efficient is decomposing the problem into independent parts, called subproblems. Since computers are really good at doing the same task repeatedly on different inputs, it is a viable strategy to find an algorithm which can be applied to independent subproblems.
In general, the task of choosing the maximal number out of several activities in a certain period of time is called the activity selection problem or interval scheduling problem. When managing processes in a large factory or running many programs on a single computer, we might have an activity selection problem with thousands or millions of activities. In informatics, there are even more sophisticated algorithms to solve these problems without any use of exhaustive search at all.

In this task it is necessary to develop a strategy for finding a maximal solution. This can be done by analyzing the overlapping of the different activities and finding rules which activities must be excluded and which activities must be included in an optimal solution. This is a typical algorithmic thinking approach, when these rules are applied in a specific order.

If we look at the description of the solution we find also a decomposition of the problem in smaller parts that can be solved independently, a very effective strategy to solve problems.

Another solution approach may be to construct all possible combinations of visits of activities and select the ones with the maximum number of activities. This is also an algorithmic approach, but it involves many combinations and therefore much more algorithmic steps than the previously described approach.

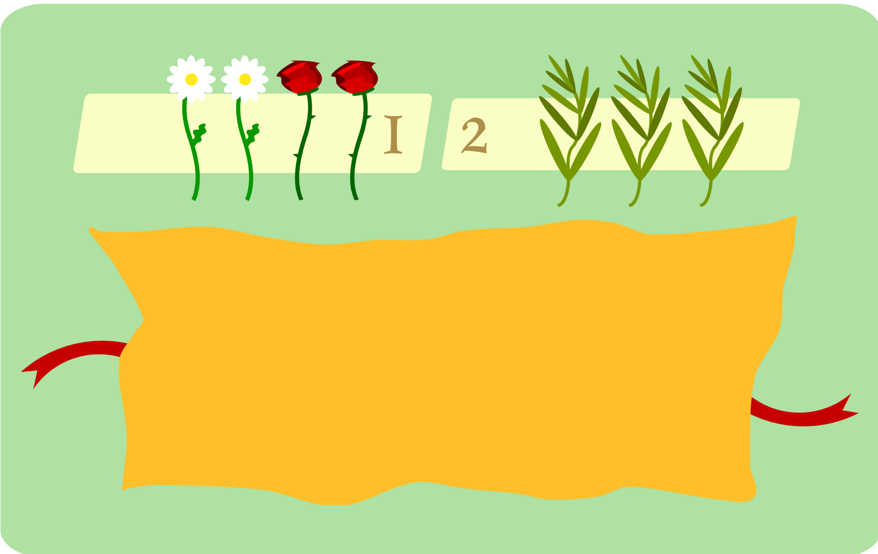
Flower Shop

A flower shop sells bunches of flowers.
The flowers are chosen by using only three steps:

1. Pick a daisy () or a rose ().
2. If the flower is a daisy, pick another daisy.
3. Pick at least one leafy twig (), to complete the bunch of flowers.

Task

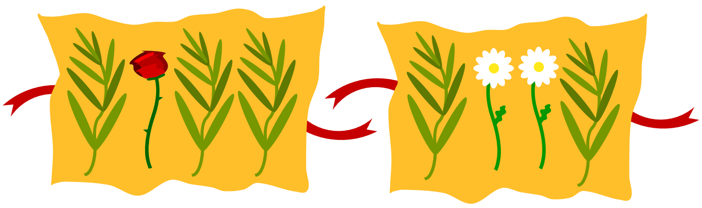
Create a bunch of flowers with exactly 4 items following the three steps.
Drag flowers or leafy twigs onto the wrapping paper below. Press Save when complete.
(There is more than one correct solution).



Save   Erase

Explanation

There are two correct solutions:



To compose the bouquet correctly, you must follow the three instructions: Following the first instruction, you can pick either a daisy or a rose.
Following the second instruction, you pick either another daisy so that you have two or no flowers, leaving you with one rose. Following the third instruction, you add three branches to the rose or two branches to the two daisies to obtain exactly four pieces.

Photo

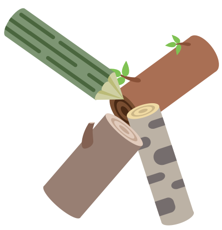
The beaver just took a photo. It is one of the four images below.

Question

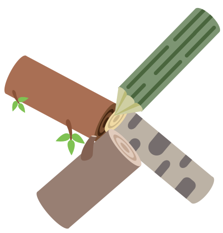
Which one?

Explanation

The correct answer is  .

To determine how the logs appear in the photo, it is important to analyze which logs are left or right to each other. Here, the log with the pointed end (looking like a pencil) is between the two brown logs, so these two options are out:

Also, you can see that the beaver, when taking the photo, sees the "pencil" log left to the big brown log (without leaves). This is the case in the correct answer only.

Background information

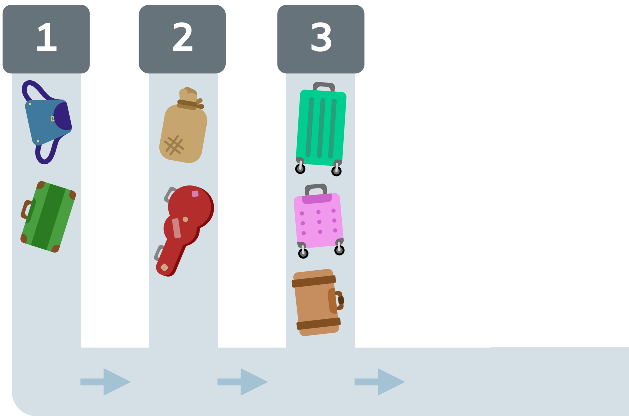
In this Bebras task, your task was to examine the order of the logs instead of their specific positions in the images. To do so, we considered special links between the logs: For each log, we can say it is linked to the log on its right and to the log on its left.

In computers, it can be useful to store data in a similar way. Each data item (like a number or a letter) will have links to two neighbors. In informatics, this structure is called a “doubly linked list”. It is very flexible: It can hold as much data as you like, and data can be added to or removed from it easily.

This task requires competence in abstraction – to identify relevant information (the arrangement of logs in a circle) and in spatial reasoning (when transforming the representation of objects, their relationships are preserved).

Luggage Check-in

At Bebrasland airport, passengers can drop off their bags at any of the 3 check-in counters shown below.

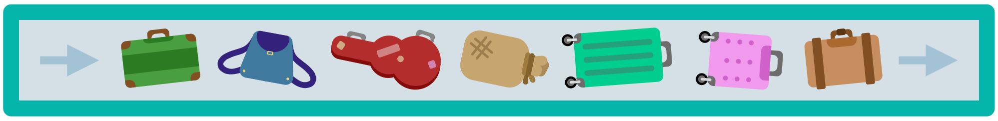


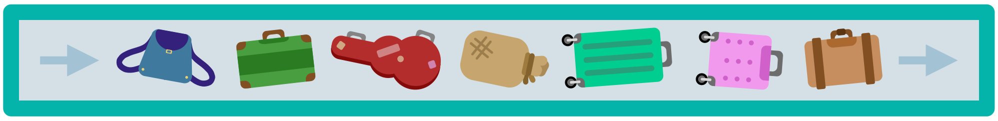
The operators at the check-in counters will place bags one at a time on the vertical lane. As each new bag is placed, the operator presses a button to move the bag forward. When the bag reaches the horizontal lane it will transfer automatically to that lane.

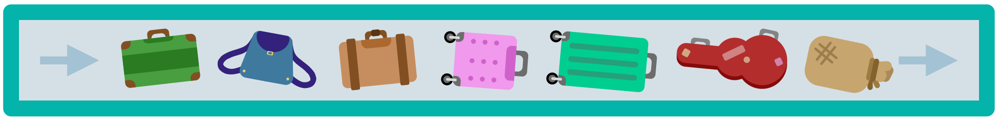
The bags already on the three vertical lanes are shown above.

Question

Which picture shows a possible order of the bags on the horizontal lane?





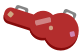
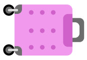
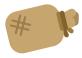


Explanation

Answer:

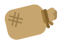
Explanation:

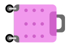
The answer above is possible in this way:

The first bag is  from lane 2, the second bag is  from lane 1, the third bag is  from lane 1, the fourth bag is  from lane 3, the fifth bag is  from lane 3, the sixth bag is  from lane 2 and the seventh bag is  from lane 3.

The other answers are not possible:

 is not possible because  will be before  from lane 1.

 is not possible because  will be before  from lane 2.

 is not possible because  will be before  and  from lane 3.

Background information

Queue is a data structure used to represent information in sequential order. It has a FIFO (first in – first out) property, because the first element into a queue is the first to leave it.

In this Task, we have different examples of queues (represented by the vertical lanes), and, for all of them, the bags are its elements. So, for all the bags on the vertical lanes, the FIFO property must be kept.

The horizontal lane can be understood as a scheduler in an operating system, that manages requests from multiple users (the vertical lanes).

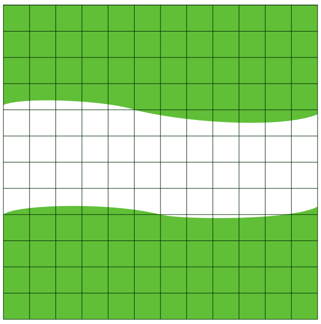
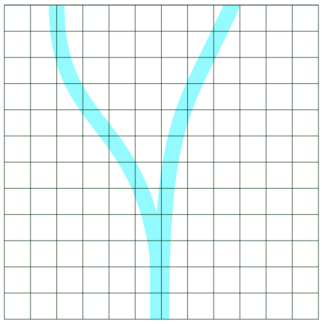
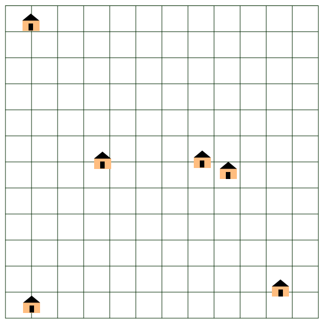
Consequently, there are many possibilities to pick up elements from queues and hence there are many possible sequences of bags on the horizontal lane.

The lanes (vertical and horizontal) and the buttons form a system by which bags are transferred. It happens according to some steps: first, a bag is put on a vertical lane, then a button at any of the counters is pressed, finally that bag moves along the corresponding vertical lane towards the horizontal lane (where it becomes part of a queue of bags). It means the airport transferring bags system follows an algorithm. Also, students have to use algorithmic thinking to analyze which possibilities of sequences of bags this algorithm allows, since a bag moves towards the horizontal lane (consequently, towards the sequence of bags) only if it is the first on a vertical lane.

Karla’s Dream House

Karla has three maps that all show exactly the same area. One map shows the forests, one shows the rivers, and one shows the houses.

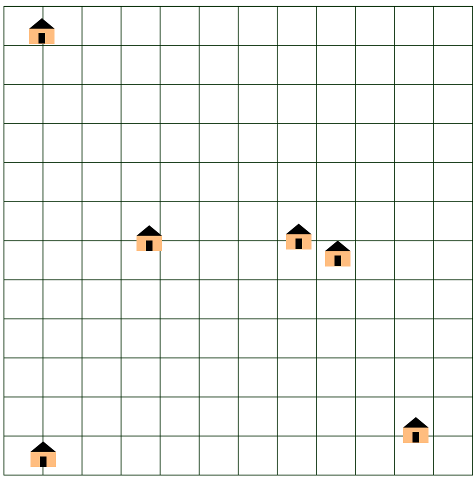
Karla's dream house is in the forest and close to the riverbank.

Forest	River	Houses
		

Question

Which is Karla's dream house?

Click on the house, then press save.

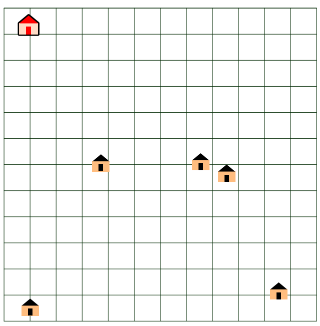


Save

Erase

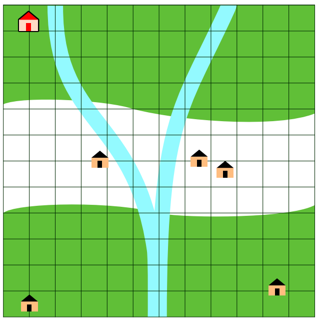
Explanation

This is the correct answer:



To identify Karla's dream house, the information from all three maps must be evaluated. The selected house must be located in a forest area and close to a river, which only applies to the house on the top left.

This is easy to see when the maps are overlayed:



Background information

If the information about the vegetation, the rivers and the buildings were shown on a single map, then it would be easy to identify the house you are looking for.

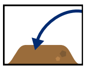
In a geographic information system (GIS), a wide variety of spatial information is brought together and displayed on a map. A GIS thus serves to visualize and analyze geodata. With the help of a GIS, it is possible for disaster control officers to compile information for evacuation plans, for example.

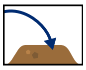
The principle of multiple layers with different image information is also known from graphics programs. An important question is always which layer or which objects are displayed in the foreground. In the example, Karla's map should be the top layer, so that the houses are not hidden by the forest areas.
The task requires students to systematically and logically combine multiple layers of maps to find a particular place. It involves geometric data structure analysis including abstraction of the three maps representation to create a coherent and informative composite map.

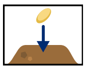
Planting Carrots

A robotic rabbit is planting carrot seeds in these four earth mounds.

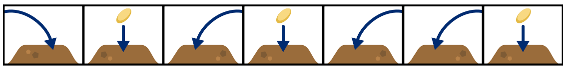
It knows these commands:

 jump left to the next mound

 jump right to the next mound

 plant a carrot seed in the mound you are on

Here are a sequence of commands for the rabbit:



We don't know which mound the rabbit started on, but we do know that when it followed this sequence, it placed each of the three carrot seeds on different mounds.

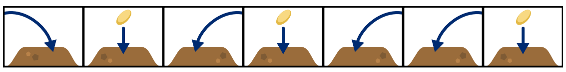
Question

Which picture shows how the carrot seeds could have been planted by the robot following the sequence of commands?

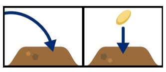
Explanation

The correct answer is 

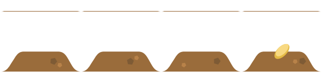
The rabbit must start on the third mound from the left, otherwise it could not jump as specified in the given sequence of commands, i.e. once to the right and then three times to the left:



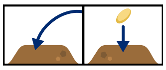
After executing the first two commands



Rabbit places the seed on the mound to the far right:

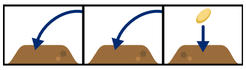


It then executes the commands



and he lays the seed:

Then he jumps twice to the left and lays the last seed

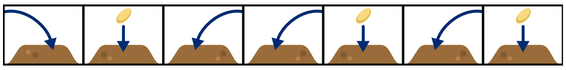


The carrot seeds will be on the hills as in correct answer:

For answer:

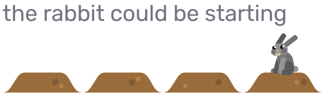
the rabbit could be starting

and the program could look like this:

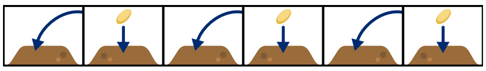


For answer:

the rabbit could be starting

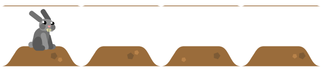


and the program could look like this:

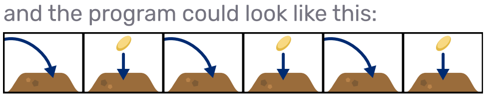


For answer:

the rabbit could be starting



and the program could look like this:



Background information

The computers are programmed similarly to how a robotic rabbit is controlled, but use more instructions. In our case, the sequence of commands for the robot is specified using picture blocks. This notation must be understood and interpreted by the pupils. They also need to realise that the instructions are given in a specific order on which the outcome depends - the placement of the carrot seeds on the hill.

In order to find the initial location of the robotic rabbit, it is necessary to trace the given sequence of instructions while observing the given conditions (the rabbit places each seed on a mound). Debugging is an important part of programming. Using this process, we not only look for bugs in the program, but we can also use it to detect appropriate inputs so that we get the desired outputs.

This related to algorithmic thinking. Thinking in therms of sequences and rules.

Beaverbeard's Gold

An island contains three treasure chests: one by the mountains, one under the palm tree, and one by the sea. At the start of the day all three treasure chests are empty .

At some point during the day, Pirate Beaverbeard fills one of the chests with gold .

Three tourists took photos of the treasure chests while exploring the island. One tourist took their photos before Beaverbeard filled a chest with gold. The other two tourists took their photos after Beaverbeard filled a chest with gold.

Alice: The chest by the sea was empty.	**Bob:** The chest by the sea was empty and the chest under the palm tree was empty.	**Carla:** The chest by the mountains was empty and the chest under the palm tree was empty.
		

Question

Luckily for Beaverbeard, no tourist found the gold. Which treasure chest was the gold in?

Click on the treasure chest (or the question mark if it cannot be determined) and press save.

Save

Erase

Explanation

The gold was in the chest by the mountains.

Since there are three possible places for the gold to be, we can explore each possibility to see which one satisfies the details of the story.

- If the gold was in the chest under the palm tree, then both Bob's and Carla's photos must have been taken before Pirate Beaverbeard arrived. The story tells us that only one tourist took photos before, so this is a contradiction. The chest under the palm tree cannot contain gold.
- If the gold was in the chest by the sea, then both Alice's and Bob's photos must have been taken before Pirate Beaverbeard arrived. This again contradicts the story. The chest by the sea cannot contain gold.
- If the gold was in the chest by the mountains, then only Carla's photos need to have been taken before Pirate Beaverbeard arrived. This satisfies the story.

Therefore, Pirate Beaverbeard filled the chest by the mountains with gold.

Background information

This task involves logical reasoning. Using the photos given and considering the three places the gold could be, it is necessary to think logically and provide reasons why something must or must not be true. Contradictions are incredibly powerful when reasoning. If logical steps lead to two statements that cannot possibly be true at the same time, then we can say with confidence that an initial starting assumption must be false.

Logic plays a key role in computer science in a huge variety of areas: databases, programming languages, artificial intelligence, hardware and software design and verification, etc.

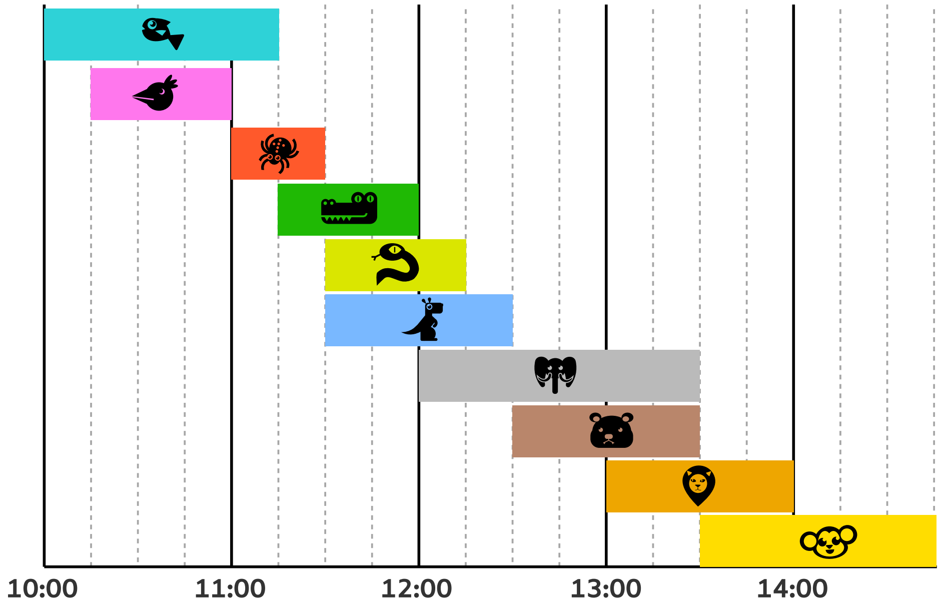
One way to solve this task is to explore each possible answer by assuming that it is true, and then following that assumption through it's logical consequences in order to determine if the answer is plausible or if it causes a conflict with other details provided in the task.

This process is called simulation and evaluation and it involves modelling a situation and then assessing it's outcomes.

2024 Bebras Challenge
Round 1 | Upper Primary Year 5 & 6

A Day at the Zoo

Anja spends a day at the zoo. There are many different activities.
Each activity has a start and an end time, and visitors cannot leave in the middle of an activity.



Anja wants to attend as many activities as possible, but she can only attend one activity at a time. She can't leave in the middle of an activity, so she has to attend activities from beginning to end. Since some of the activities overlap, Anja cannot attend all of the activities today.

Question
What is the maximum number of activities Anja can attend during a single day?

Save

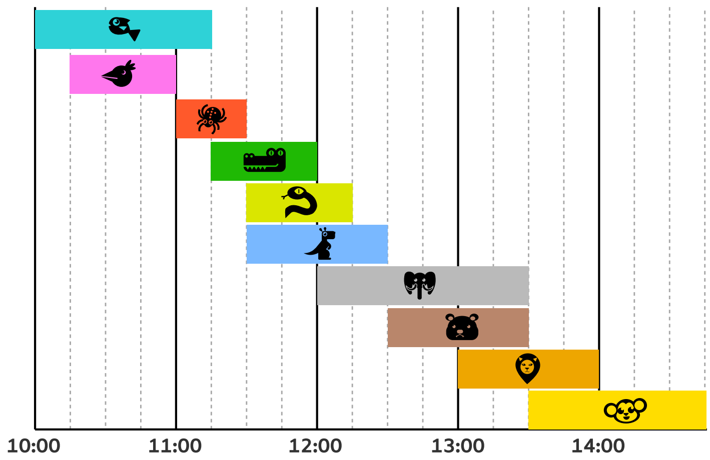
Explanation

Answer:
5
Explanation:
Anja can attend a maximum of 5 activities over the entire day.

One way to solve this problem is to try all possible combinations of non-overlapping activities. Each combination of non-overlapping activities gives us a valid schedule. Then we check which schedule has the most activities (an *optimal* schedule). This is called an *exhaustive* search.

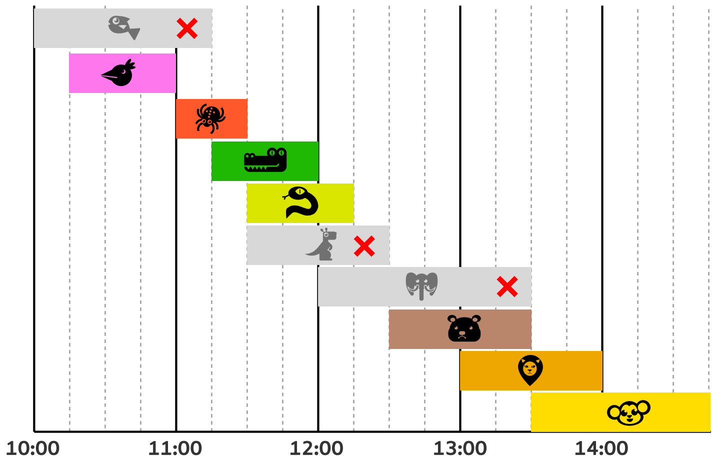
An exhaustive search can be difficult since there are too many possibilities to check. It's also hard to make sure that you've checked everything.

We can make an observation to *simplify* the problem. If two activities overlap, then only one of the two activities can be part of an optimal schedule. If one activity takes place completely during another activity, then it is always better to take the shorter activity.

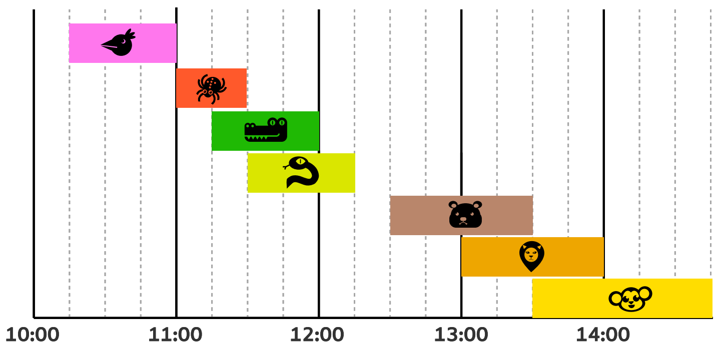


For example, the bird show  takes place completely during the aquarium light show  and it is shorter, it is better not to choose the aquarium light show, because it may overlap with a larger number of other activities.
With the same reasoning, we can do the following.

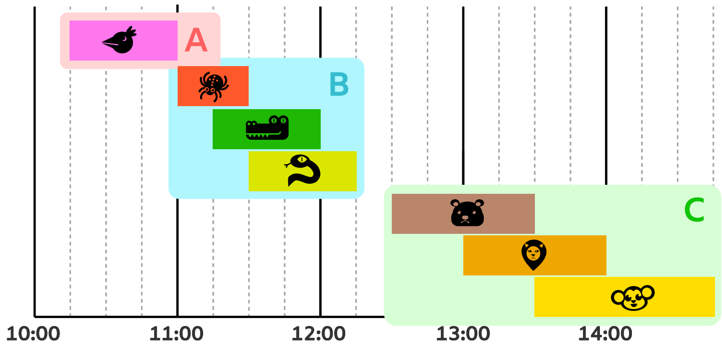
- We can exclude the kangaroo feeding  , since the reptile show  takes place completely during the kangaroo feeding.
- We can exclude the elephant showering  , since the bear feeding  takes place completely during the elephant showering.



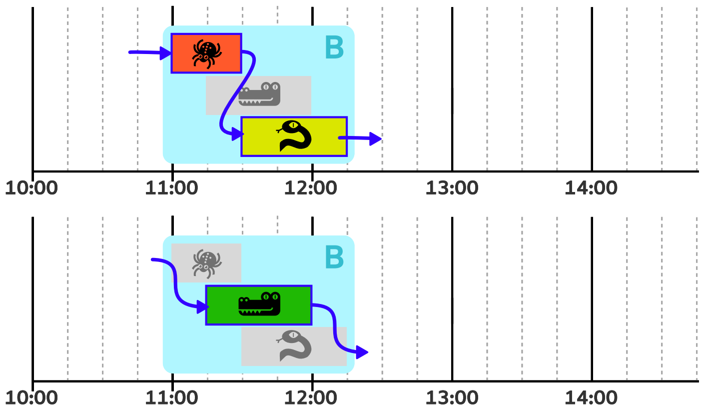
After deleting these three activities, we are left with these activities.



We can decompose (or break up) these activities into independent groups, which we call A, B and C. We can find the optimal schedule for the entire day by finding the optimal schedule for each group.

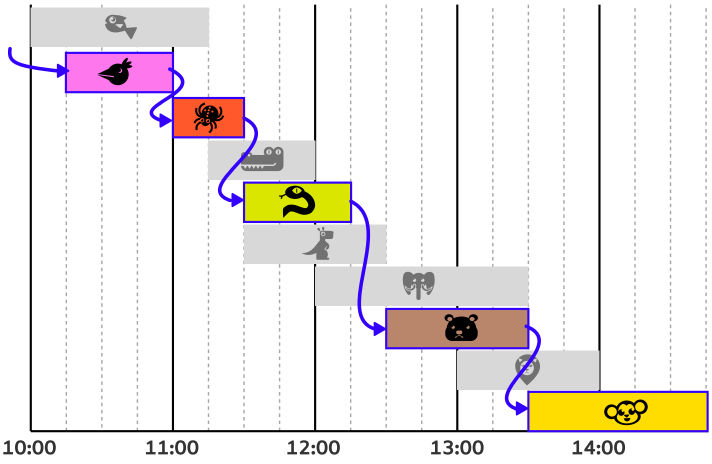


For group A, there is only one activity: the bird show . The optimal schedule is to attend the bird show.
For group B, there are three activities. Conducting an exhaustive search, there are only two valid schedules for group B.



The optimal schedule is to take the spider show  and reptile show  activities.
For group C, there are also three activities. Using a similar reasoning, the optimal schedule is to take the crafting and seal show activities.

Putting the optimal schedules together, we have the following schedule for the entire day.



There are 5 activities in this schedule, so the answer is 5.

Background information

Often in computer science, when designing an algorithm to solve a problem, the easiest algorithm to think of is an exhaustive search, as it is conceptually simple and guaranteed to give the correct answer. The main problem is that an exhaustive search can take too much time to do, either by hand or with a computer. We need to find ways to make more efficient algorithms to solve problems in a more reasonable amount of time.

In order to make an algorithm more efficient, we often have to make observations about the problem which allow us to simplify the problem. A problem where you maximize or minimize a quantity is called an optimization problem. When simplifying an optimization problem, we have to think about what decisions we can make which do not prevent us from obtaining an optimal solution. In the task, we were able to disregard the aquarium light show, koala feeding, and elephant showering activities, since we can be sure any schedule containing these activities can be potentially improved by picking a different activity instead.

Another strategy which allows us to make an algorithm more efficient is decomposing the problem into independent parts, called subproblems. Since computers are really good at doing the same task repeatedly on different inputs, it is a viable strategy to find an algorithm which can be applied to independent subproblems.
In general, the task of choosing the maximal number out of several activities in a certain period of time is called the activity selection problem or interval scheduling problem. When managing processes in a large factory or running many programs on a single computer, we might have an activity selection problem with thousands or millions of activities. In informatics, there are even more sophisticated algorithms to solve these problems without any use of exhaustive search at all.

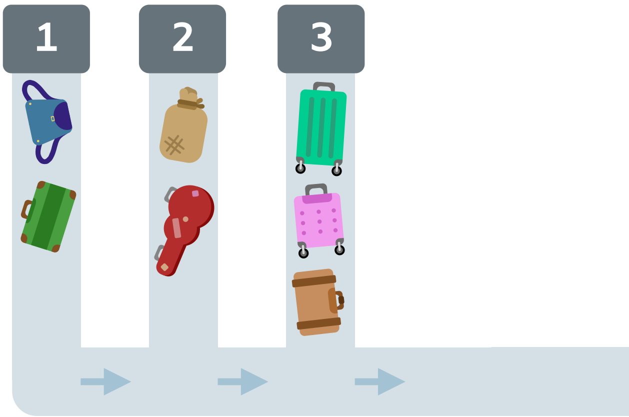
In this task it is necessary to develop a strategy for finding a maximal solution. This can be done by analyzing the overlapping of the different activities and finding rules which activities must be excluded and which activities must be included in an optimal solution. This is a typical algorithmic thinking approach, when these rules are applied in a specific order.

If we look at the description of the solution we find also a decomposition of the problem in smaller parts that can be solved independently, a very effective strategy to solve problems.

Another solution approach may be to construct all possible combinations of visits of activities and select the ones with the maximum number of activities. This is also an algorithmic approach, but it involves many combinations and therefore much more algorithmic steps than the previously described approach.

Luggage Check-in

At Bebrasland airport, passengers can drop off their bags at any of the 3 check-in counters shown below.

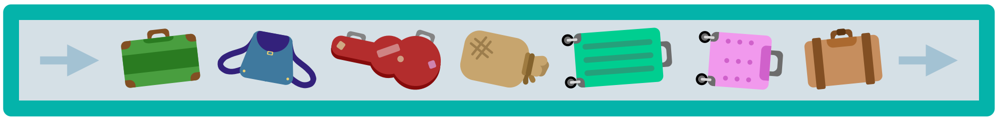


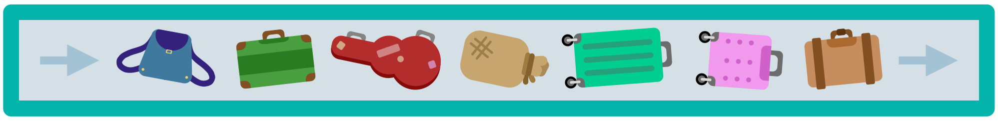
The operators at the check-in counters will place bags one at a time on the vertical lane. As each new bag is placed, the operator presses a button to move the bag forward. When the bag reaches the horizontal lane it will transfer automatically to that lane.

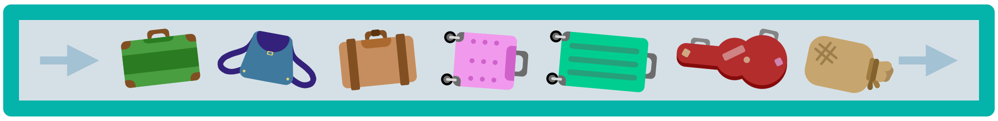
The bags already on the three vertical lanes are shown above.

Question

Which picture shows a possible order of the bags on the horizontal lane?





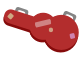
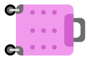
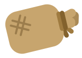


Explanation

Answer:

Explanation:

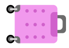
The answer above is possible in this way:

The first bag is  from lane 2, the second bag is  from lane 1, the third bag is  from lane 1, the fourth bag is  from lane 3, the fifth bag is  from lane 3, the sixth bag is  from lane 2 and the seventh bag is  from lane 3.

The other answers are not possible:

 is not possible because  will be before  from lane 1.

 is not possible because  will be before  from lane 2.

 is not possible because  will be before  and  from lane 3.

Background information

Queue is a data structure used to represent information in sequential order. It has a FIFO (first in – first out) property, because the first element into a queue is the first to leave it.

In this Task, we have different examples of queues (represented by the vertical lanes), and, for all of them, the bags are its elements. So, for all the bags on the vertical lanes, the FIFO property must be kept.

The horizontal lane can be understood as a scheduler in an operating system, that manages requests from multiple users (the vertical lanes).

Consequently, there are many possibilities to pick up elements from queues and hence there are many possible sequences of bags on the horizontal lane.

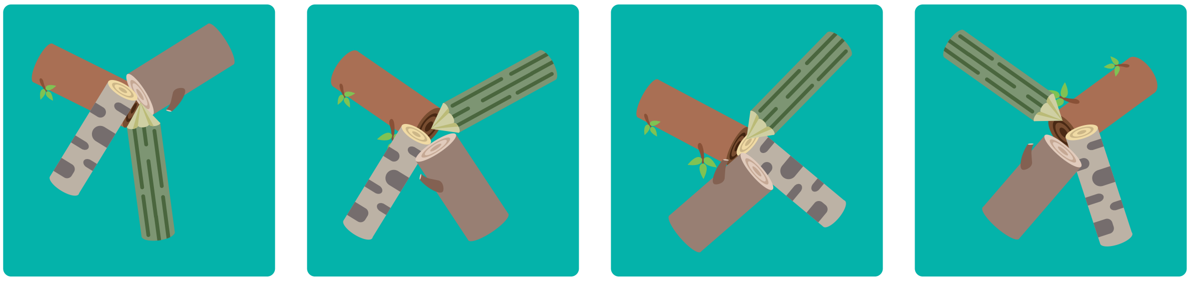
The lanes (vertical and horizontal) and the buttons form a system by which bags are transferred. It happens according to some steps: first, a bag is put on a vertical lane, then a button at any of the counters is pressed, finally that bag moves along the corresponding vertical lane towards the horizontal lane (where it becomes part of a queue of bags). It means the airport transferring bags system follows an algorithm. Also, students have to use algorithmic thinking to analyze which possibilities of sequences of bags this algorithm allows, since a bag moves towards the horizontal lane (consequently, towards the sequence of bags) only if it is the first on a vertical lane.

Photo

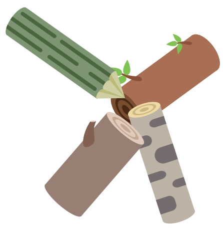
The beaver just took a photo. It is one of the four images below.

Question

Which one?



Explanation

The correct answer is  .

To determine how the logs appear in the photo, it is important to analyze which logs are left or right to each other. Here, the log with the pointed end (looking like a pencil) is between the two brown logs, so these two options are out:

Also, you can see that the beaver, when taking the photo, sees the "pencil" log left to the big brown log (without leaves). This is the case in the correct answer only.

Background information

In this Bebras task, your task was to examine the order of the logs instead of their specific positions in the images. To do so, we considered special links between the logs: For each log, we can say it is linked to the log on its right and to the log on its left.

In computers, it can be useful to store data in a similar way. Each data item (like a number or a letter) will have links to two neighbors. In informatics, this structure is called a “doubly linked list”. It is very flexible: It can hold as much data as you like, and data can be added to or removed from it easily.

This task requires competence in abstraction – to identify relevant information (the arrangement of logs in a circle) and in spatial reasoning (when transforming the representation of objects, their relationships are preserved).

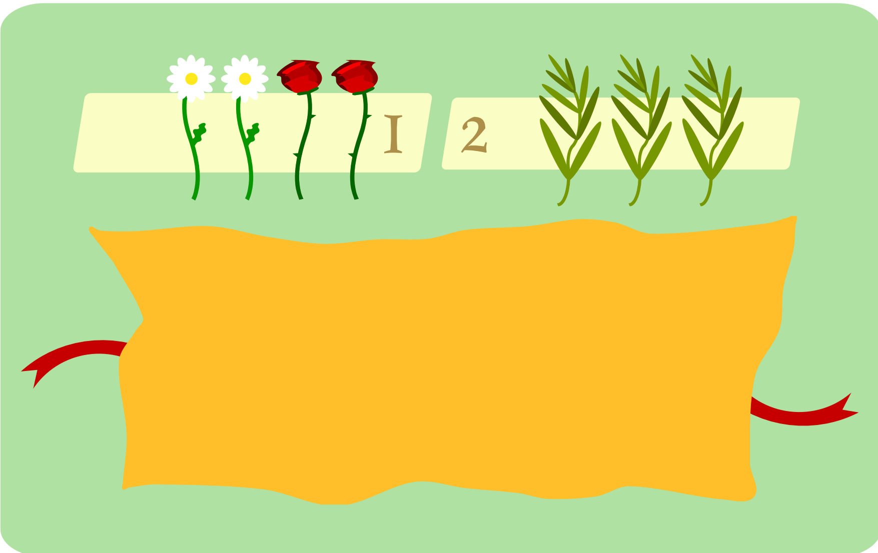
Flower Shop

A flower shop sells bunches of flowers.
The flowers are chosen by using only three steps:

1. Pick a daisy () or a rose ().
2. If the flower is a daisy, pick another daisy.
3. Pick at least one leafy twig (), to complete the bunch of flowers.

Task

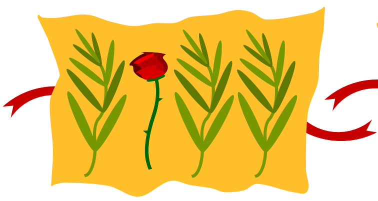
Create a bunch of flowers with exactly 4 items following the three steps.
Drag flowers or leafy twigs onto the wrapping paper below. Press Save when complete.
(There is more than one correct solution).



Save Erase

Explanation

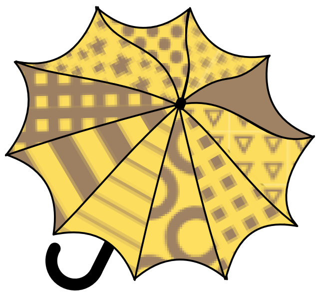
There are two correct solutions:



To compose the bouquet correctly, you must follow the three instructions: Following the first instruction, you can pick either a daisy or a rose.
Following the second instruction, you pick either another daisy so that you have two or no flowers, leaving you with one rose. Following the third instruction, you add three branches to the rose or two branches to the two daisies to obtain exactly four pieces.

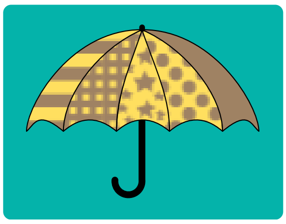
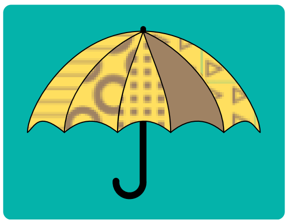
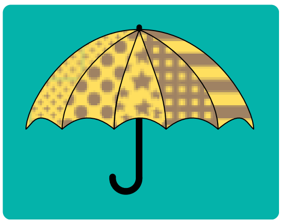
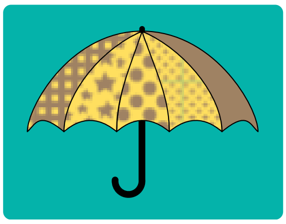
Umbrella

This is Anna's umbrella:



Question

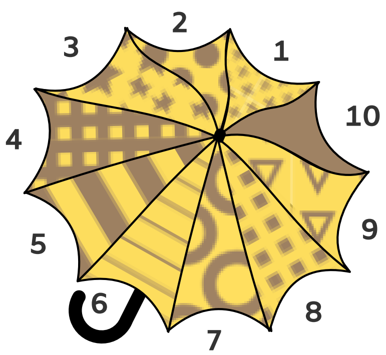
One of the four images below shows Anna's umbrella. Which one?

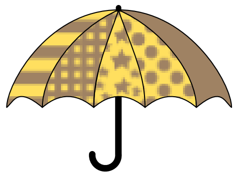
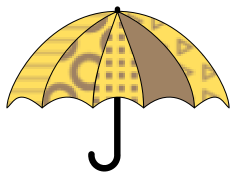
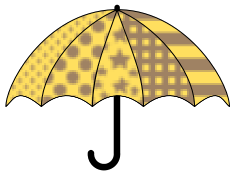
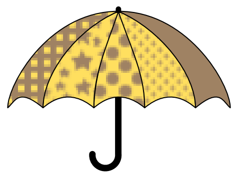
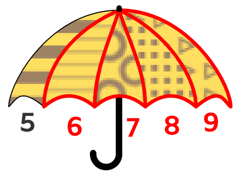
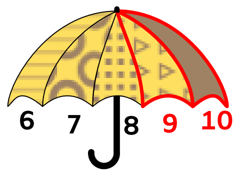
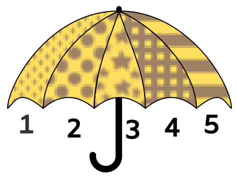
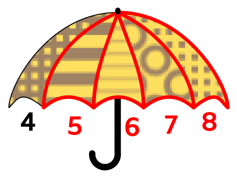
Explanation

Every pattern is used only once in Anna's umbrella. Therefore, we can compare each image with Anna's umbrella this way:

- Pick the left-most pattern from the image and find its position in Anna's umbrella.
- Check if the following patterns match Anna's umbrella.



In each image we can see a sequence of only five patterns. So we cannot decide whether the image matches the full sequence of all ten patterns of Anna's umbrella. However, only image C has a sequence of five patterns that fully matches Anna's umbrella. For this reason, only umbrella C can show Anna's umbrella. All other images show pattern sequences that only partially match that of Anna's umbrella; so they cannot show Anna's umbrella.

	A	B	C	D
answer images				
Anna's umbrella				

Background information

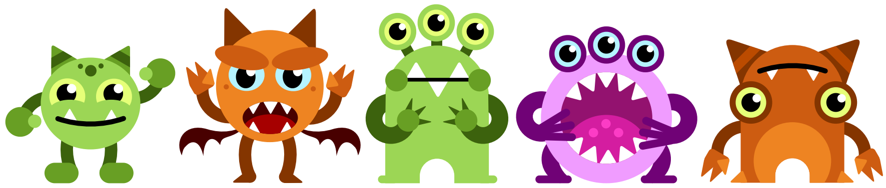
In this Bebras task, we gain only partial information about the umbrella patterns shown in the answer options. Nevertheless, we can determine which image shows Anna's umbrella: We can look at the four pattern sequences and notice that only one also occurs in Anna's umbrella.

The same principle as for the "umbrella search" is applied when searching a text document. The computer searches with the given partial information ("search sequence" of letters) for matching sequences of letters in the document. You can observe that the longer your search sequence, the less matches you will find and the higher your chance to find the spot in the document that you are looking for. The shorter your sequence, the less accurate, but the faster your search. In computer science, search procedures (search algorithms) have been designed (and are still being improved) in order to obtain the most suitable results as quickly as possible. These algorithms are capable of searching in large amounts of data, like the internet search tools you may know.

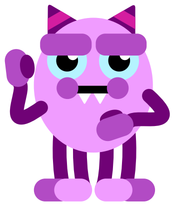
The problem can be solved by using decomposition. Decomposition means breaking down a problem into smaller problems or steps. To find a matching sequence of 5 patterns you can break it down to the steps described in the answer explanation. This task is also a very practical exercise in pattern recognition. Pattern recognition is the ability to recognize regularities, repetitions or similarities in a set of data. To find the favorite umbrella it is necessary to quickly find the matching sequences of the patterns. Later, when programming, pattern recognition allows us to use a modular approach, where have to solve a recurring sub task only once.

Riccas

Evren is trying to learn what a Ricca looks like. Evren studies the following five photos of Riccas and makes some notes that accurately describe what she sees.



Evren is then shown this sixth photo of a Ricca and realises one of her notes is definitely wrong.



Question

Which one of Evren's notes about Riccas is definitely wrong?

- **Riccas always have teeth.**
- **Riccas sometimes have wings.**
- **Riccas have either horns or three eyes, but not both.**
- **If Riccas have exactly two arms then they also have exactly two legs.**

Explanation

The answer is: If Riccas have exactly two arms then they also have exactly two legs.
It is not true that if a Ricca has exactly two arms then it also has exactly two legs. The sixth photo shows a Ricca with two arms and four legs.

The note "Riccas always have teeth" may still be true. All six photos show Riccas with teeth.

The note "Riccas sometimes have wings" is definitely true. The second photo shows a Ricca with wings.

The note "Riccas have either horns or three eyes, but not both" may still be true. Photos one, two, five, and six show Riccas with horns and "only" two eyes. The remaining photos (three and four) show Riccas with three eyes and no horns.

Background information

Evren is learning to identify a Ricca by studying photos of Riccas and looking for similar characteristics. Computers can also be taught to do this, using a process called comparison. Nowadays, we can employ machine learning to compare a vast number of items automatically and in a short time. As part of machine learning, a computer is presented with a collection of data, and then it searches for patterns within this data to draw conclusions. Just like with Evren, it is possible for a computer to arrive at the incorrect conclusion. By providing the computer with more data, and data that is more representative and inclusive, the computer can correct its mistakes and update its learning.

Detecting recurring patterns and regularities in given data is an important skill in informatics, generally referred to as pattern recognition. In programming, recognizing patterns allows us to adopt a modular approach, wherein we need to solve a recurring subtask only once. This is the foundation of efficient algorithms and software design, as it reduces code redundancy and enhances maintainability. Pattern recognition is also crucial for machine learning and artificial intelligence, as it empowers computers to learn from data and make predictions or decisions. For example, in image recognition, pattern recognition techniques assist in identifying objects in images, classifying them into different categories, and even recognizing faces. In natural language processing, pattern recognition enables computers to understand the structure of sentences, identify topics, and even generate human-like text.

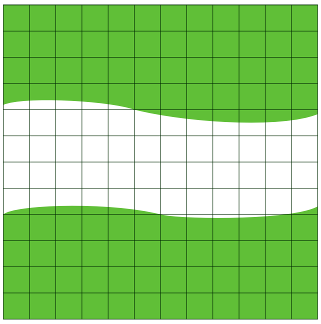
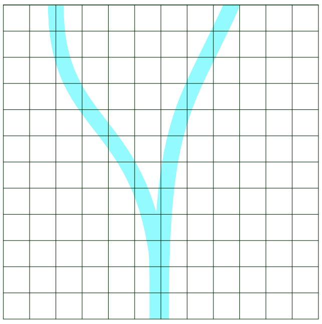
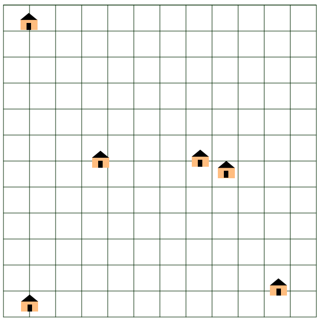
If care is not taken in the selection of data for machine learning, bias may be introduced into the computer system. Consider this: what impact would a speech recognition system have on society if it were trained only on data from male native English speakers?

Pattern recognition is related to data measurement and data analysis. To solve tasks that involve these concepts, among other skills, algorithmic thinking is crucial. Students apply this when they develop a step-by-step ordered procedure to solve a problem, using their understanding of patterns, rotations, and spatial relationships. Data analysis, which includes identifying which symbols are in play and understanding their positions, allows students to make inferences and deductions based on the given information and recognized patterns. Finally, evaluation is crucial to assess the effectiveness of their solutions and identify areas for improvement.

Karla’s Dream House

Karla has three maps that all show exactly the same area. One map shows the forests, one shows the rivers, and one shows the houses.

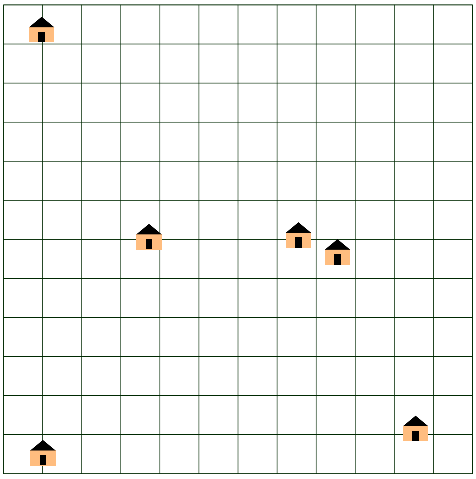
Karla's dream house is in the forest and close to the riverbank.

Forest	River	Houses
		

Question

Which is Karla's dream house?

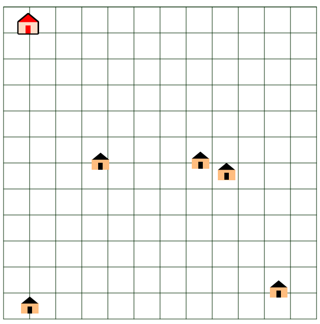
Click on the house, then press save.



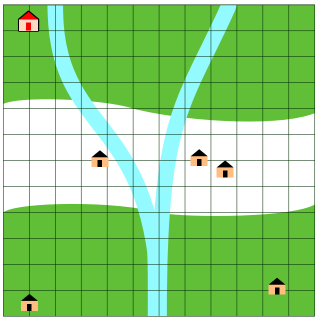
Save   Erase

Explanation

This is the correct answer:



To identify Karla's dream house, the information from all three maps must be evaluated. The selected house must be located in a forest area and close to a river, which only applies to the house on the top left.
This is easy to see when the maps are overlayed:



Background information

If the information about the vegetation, the rivers and the buildings were shown on a single map, then it would be easy to identify the house you are looking for.

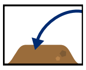
In a geographic information system (GIS), a wide variety of spatial information is brought together and displayed on a map. A GIS thus serves to visualize and analyze geodata. With the help of a GIS, it is possible for disaster control officers to compile information for evacuation plans, for example.

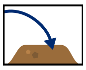
The principle of multiple layers with different image information is also known from graphics programs. An important question is always which layer or which objects are displayed in the foreground. In the example, Karla's map should be the top layer, so that the houses are not hidden by the forest areas.
The task requires students to systematically and logically combine multiple layers of maps to find a particular place. It involves geometric data structure analysis including abstraction of the three maps representation to create a coherent and informative composite map.

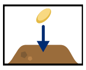
Planting Carrots

A robotic rabbit is planting carrot seeds in these four earth mounds.

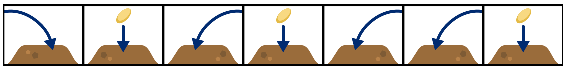
It knows these commands:

 jump left to the next mound

 jump right to the next mound

 plant a carrot seed in the mound you are on

Here are a sequence of commands for the rabbit:



We don't know which mound the rabbit started on, but we do know that when it followed this sequence, it placed each of the three carrot seeds on different mounds.

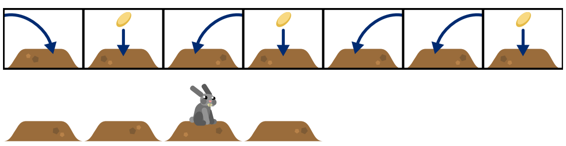
Question

Which picture shows how the carrot seeds could have been planted by the robot following the sequence of commands?

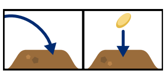
Explanation

The correct answer is 

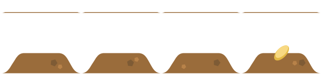
The rabbit must start on the third mound from the left, otherwise it could not jump as specified in the given sequence of commands, i.e. once to the right and then three times to the left:



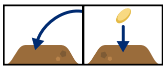
After executing the first two commands



Rabbit places the seed on the mound to the far right:

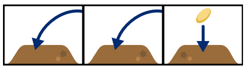


It then executes the commands



and he lays the seed:

Then he jumps twice to the left and lays the last seed

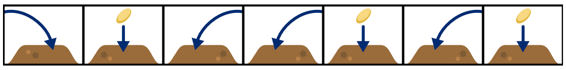


The carrot seeds will be on the hills as in correct answer:

For answer:

the rabbit could be starting

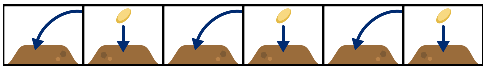
and the program could look like this:



For answer:

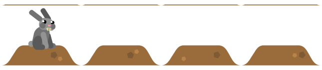
the rabbit could be starting

and the program could look like this:

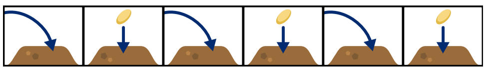


For answer:

the rabbit could be starting



and the program could look like this:



Background information

The computers are programmed similarly to how a robotic rabbit is controlled, but use more instructions. In our case, the sequence of commands for the robot is specified using picture blocks. This notation must be understood and interpreted by the pupils. They also need to realise that the instructions are given in a specific order on which the outcome depends - the placement of the carrot seeds on the hill.

In order to find the initial location of the robotic rabbit, it is necessary to trace the given sequence of instructions while observing the given conditions (the rabbit places each seed on a mound). Debugging is an important part of programming. Using this process, we not only look for bugs in the program, but we can also use it to detect appropriate inputs so that we get the desired outputs.

This related to algorithmic thinking. Thinking in therms of sequences and rules.

Beaverbeard's Gold

An island contains three treasure chests: one by the mountains, one under the palm tree, and one by the sea. At the start of the day all three treasure chests are empty .

At some point during the day, Pirate Beaverbeard fills one of the chests with gold .

Three tourists took photos of the treasure chests while exploring the island. One tourist took their photos before Beaverbeard filled a chest with gold. The other two tourists took their photos after Beaverbeard filled a chest with gold.

Alice: The chest by the sea was empty.	**Bob:** The chest by the sea was empty and the chest under the palm tree was empty.	**Carla:** The chest by the mountains was empty and the chest under the palm tree was empty.
		

Question

Luckily for Beaverbeard, no tourist found the gold. Which treasure chest was the gold in?

Click on the treasure chest (or the question mark if it cannot be determined) and press save.

Save Erase

Explanation

The gold was in the chest by the mountains.

Since there are three possible places for the gold to be, we can explore each possibility to see which one satisfies the details of the story.

- If the gold was in the chest under the palm tree, then both Bob's and Carla's photos must have been taken before Pirate Beaverbeard arrived. The story tells us that only one tourist took photos before, so this is a contradiction. The chest under the palm tree cannot contain gold.
- If the gold was in the chest by the sea, then both Alice's and Bob's photos must have been taken before Pirate Beaverbeard arrived. This again contradicts the story. The chest by the sea cannot contain gold.
- If the gold was in the chest by the mountains, then only Carla's photos need to have been taken before Pirate Beaverbeard arrived. This satisfies the story.

Therefore, Pirate Beaverbeard filled the chest by the mountains with gold.

Background information

This task involves logical reasoning. Using the photos given and considering the three places the gold could be, it is necessary to think logically and provide reasons why something must or must not be true. Contradictions are incredibly powerful when reasoning. If logical steps lead to two statements that cannot possibly be true at the same time, then we can say with confidence that an initial starting assumption must be false.

Logic plays a key role in computer science in a huge variety of areas: databases, programming languages, artificial intelligence, hardware and software design and verification, etc.

One way to solve this task is to explore each possible answer by assuming that it is true, and then following that assumption through it's logical consequences in order to determine if the answer is plausible or if it causes a conflict with other details provided in the task.

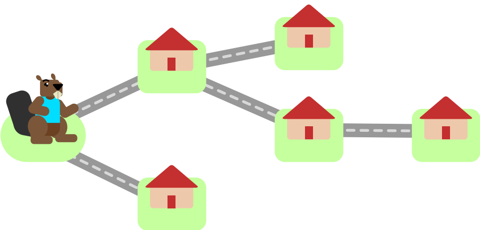
This process is called simulation and evaluation and it involves modelling a situation and then assessing it's outcomes.

Thomas and His Friends

Thomas's village has only six houses, including his own. All the roads between houses take the same amount of time to walk.

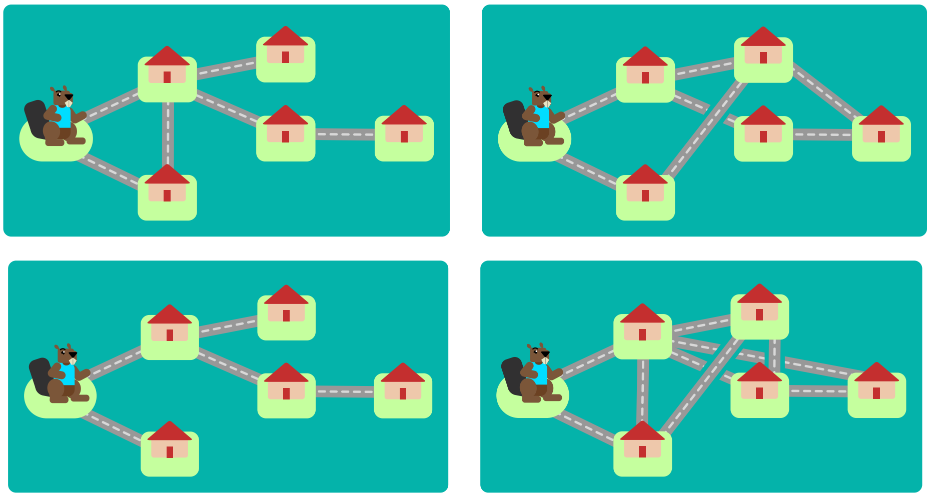
He constructed a map of roads that connect his house to other houses so that he can go from his house to any other house using the shortest path.

The map made by Thomas does not necessarily contain all the roads in the village because he included only the shortest paths.

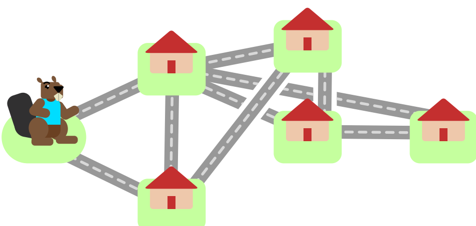


Question

There are many ways that a map of the village with all roads included could look. Which of these options **cannot** be the full map of the village, given Thomas' map?



Explanation

The correct answer is  .

In Thomas' map, the furthest house to the right requires three steps to reach.

If this is the correct map of the village, there is a faster way to reach this house, which would only take two steps. This means that either Thomas' map is incorrect, or that this cannot be a map of the village.

All other options could be a map of the village. In each of them, we can go from Thomas' house to any of his friends houses in the same number of steps shown in Thomas' map.

Background information

Graph Theory is a core concept in Computer Science. A graph is a set of vertices and edges, where an edge is an unordered pair of vertices. Graphs are useful when modelling reality, for instance, points that are joined by roads in a city.

A classic problem in Graph algorithms is to find shortest paths from one special vertex, called root, to any other vertex in the graph. This can be done by a standard algorithm called Breadth First Search (BFS). This algorithm returns a rooted tree, so that if we traverse this tree from the root to any other vertex, we will follow the shortest path to that vertex.

In this rooted tree there can be only connections from vertices whose distance from the root is at most one. Otherwise, a shorter path can be found, contradicting the optimality of the tree.

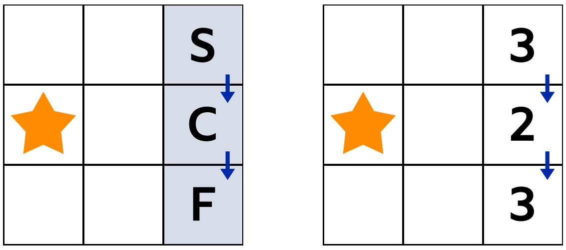
A very broad area of algorithms it is interested in optimization. We want an answer that is optimal with respect to other answers. For instance, we can say that a tree resulting from BFS is optimal if the corresponding paths from the root are shortest paths.

Understanding optimality is crucial in computational thinking. In this particular case, we can see that the optimality of the solution is lost if a particular edge joins nodes in a more optimal way than the original graph.

Closer or Further

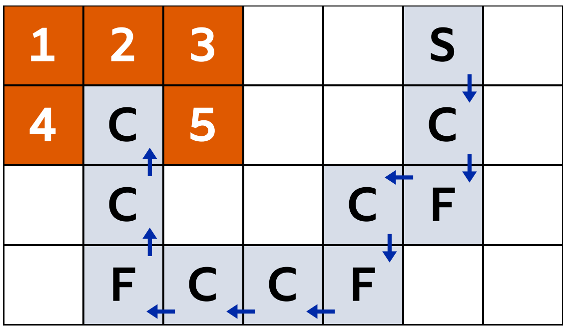
Daniel is playing a game to find out where the treasure was buried in a grid of squares.
Daniel begins from a starting square denoted as **S** and he can move **one step** at a time only horizontally or vertically to the neighbor squares.
After each step, Daniel receives a signal indicating whether he is closer (C) to or further away (F) from the treasure, where the **distance** to the treasure is the minimum number of steps required to reach it from his current square.

For example, on the 3x3 grid shown below, the treasure is buried under the square marked  .
Daniel goes forward two steps following the arrows. The distances from the two squares to the treasure are shown below on the right.
Daniel gets the signals "C" and "F", respectively, after each step.



Now, Daniel is given a 4x7 grid as shown below, where his path follows the arrows and the received signals are recorded on each square.
Then, Daniel is given a hint that the treasure is buried under one of the five numbered squares.

Question
Under which numbered square was the treasure buried?
Select the square. Press save when complete.



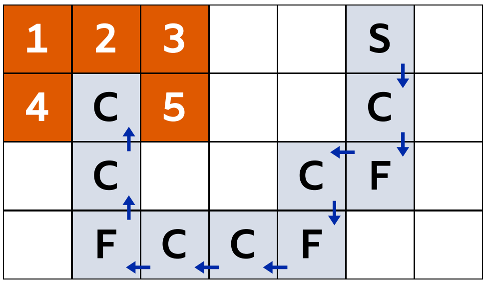
Save

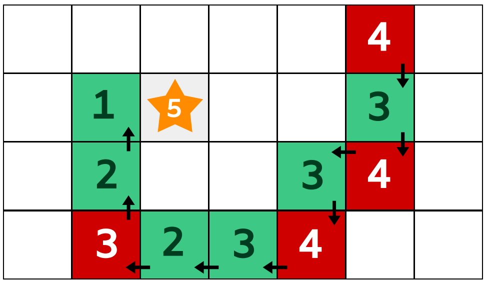
Erase

Explanation

The correct answer is square 5.

We check the accuracy of the markings "C" and "F" on the playing board by writing the distance to square 5 in each square. We can see that if the distance to square 5 increased when moving in the direction of the arrow, the letter "F" is noted, and if its distance to square 5 decreased, then "C" is noted in the square. The obtained sequence of signals is consistent with the reported result. So the treasure is placed under square 5.

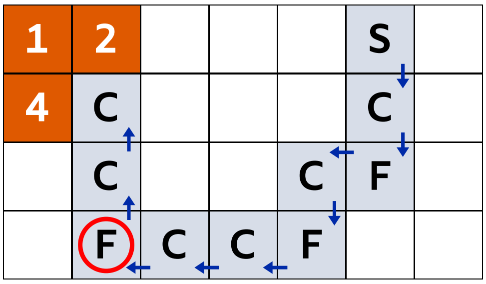




For the other possibilities of placing the treasure, we find an incorrect letter in the reported result.

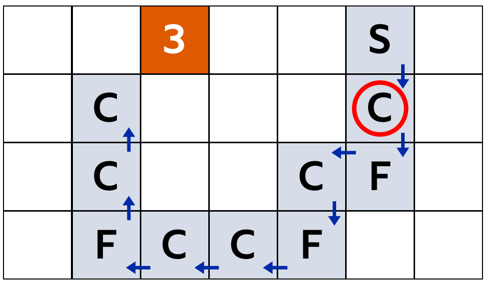
For treasure in squares 1, 2 or 4:

If the treasure was buried under one of the squares 1, 2, or 4, then the letter "F" in row 4 and column 2 would have to be a "C" because that square is closer to the treasure (to all these three locations) than was the square that it came from (i.e., row 4, column 3).



For treasure in square 3:

If the treasure was buried under square 3, then the square in row 2 and column 6 would have to be "F" because that square is further from the treasure than the square that the player started on (i.e., row 1 and column 6).



Background information

Reinforcement learning (RL) is a technique of machine learning, which concerns with how intelligent agents ought to take actions in an **environment** in order to maximize the cumulative **reward**. The fundamental components of an RL system include an agent or learner, the environment it interacts with, the policy it follows to make decisions, and the reward signal it receives after taking actions. To evaluate the reward signal, the value function measures the "goodness" of a particular state.

In this specific task, the new location in the playing board after a move represents the **environment**, while the signal obtained from the detected distance serves as a **reward signal** (with 'C' indicating a positive reward and 'F' indicating a negative reward). The player, Daniel, who collects reward signals by making optimal decisions for the next step, serves as the agent. He must consider the possibility of a square containing the treasure, which is similar to the value function.

It is worth noting that the distance between squares in this scenario is measured by the **Manhattan distance** (also called taxicab metric), with the player only able to move horizontally or vertically.

Autonomous driving is an example of an application of RL. In order to operate successfully in an unpredictable environment, an autonomous driving system must perform numerous perception and planning tasks such as vehicle path planning and motion prediction. Vehicle path planning entails the use of various low and high-level policies to make decisions that consider different temporal and spatial scales. Motion prediction, on the other hand, involves forecasting the movements of pedestrians and other vehicles, providing insight into how the situation might evolve based on the current state of the environment.

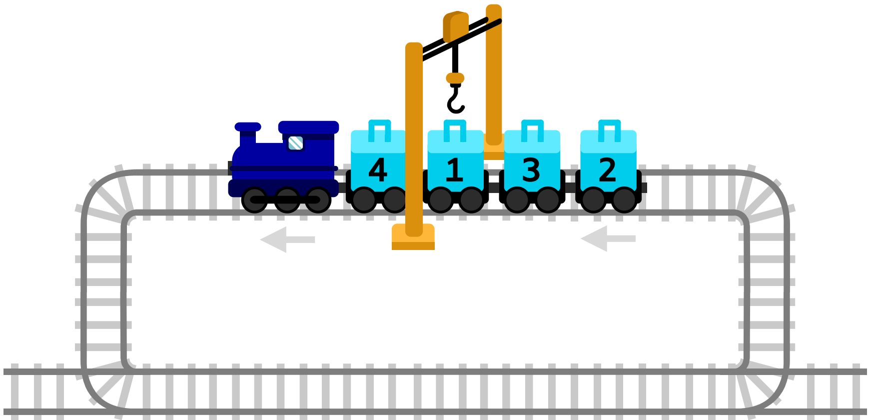
In order to tackle this particular task, when devising a strategy to move forward, it is essential to develop a systematic approach that incorporates the principles of algorithm design. This ensures a well-structured plan for navigating the grid and optimizing the treasure-seeking process, which is the ability of algorithm design.

In this task, according to the signal provided, represented by either "C" or "F," the divide-and-conquer strategy can be used to determine potential locations for the buried treasure by dividing the grid. When encountering the "F" signal while moving forward, it indicates that none of the squares ahead can be considered potential solutions. A good algorithm design strategy serves to narrow down the search space of the problem.

Unloading

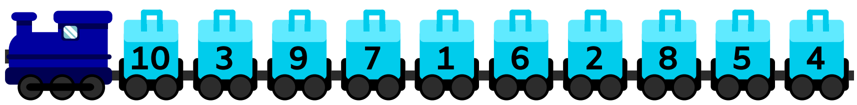
A freight train has wagons, each with a numbered box. A single crane is used for unloading. The crane is at a fixed position. To unload a box, the box has to be positioned directly below the crane.

The boxes have to be unloaded in order starting from 1. The train can move only forward. It is on a circular track, so it can go around the track and return so more boxes can be unloaded by the crane.



In the example above, the boxes have to be unloaded in the sequence of 1, 2, 3, 4. In the first round of unloading, the train skips box 4, unloads box 1, skips box 3, and unloads box 2. It the second round, it skips box 4 and unloads box 3. The train has to come back for a third round to unload the final box, number 4.

Question:
How many rounds will be needed to unload all the boxes from the following train?
(They need unloading in the order: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.)



Save

Explanation

The correct answer is 7.

The required order for unloading is 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. If we follow the procedure outlined above, then during the first round boxes 1 and 2 will be unloaded together, then 3 and 4 together, then 5, then 6, then 7 and 8 together, then 9, and finally 10. This is 7 rounds.

Alternatively, one might notice the general principal that for each number in this sequence 1, 2, ..., if the next number comes to the left of it in the train, an extra round is needed. For instance, if 3 appears to the left of 2, then 3 will be skipped to unload 2 and so an extra round is needed to bring 3 under the crane. In the given challenge, the number of such pairs that are out of order are (2,3), (4,5), (5,6), (6,7), (8,9), and (9, 10), so 6 extra rounds are needed, for a total of 7 rounds.

Background information

For any box number, if the next largest number is to the left of it on the train, we call this an "inversion" . For each such inversion, an extra round is need. If we count the number of inversions, we get the answer. Counting inversions with respect to a desired sequence has many applications. For some sorting algorithms, such as bubble sort, the number of inversions tells us how many swaps we need to sort a given sequence. If two customers rank the same set of items in order of preference, the number of inversions in their rankings tells us how aligned their tastes are. This is used by online shops to identify "similar" customers to make product recommendations.

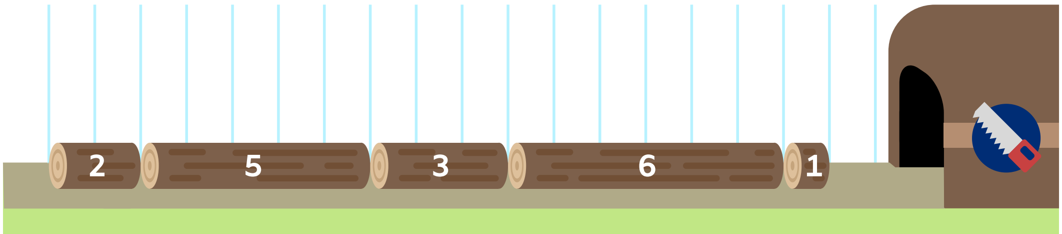
Algorithmic thinking can be used to solve this task. The sequence of steps that describes the unloading procedure is given, and we must carefully follow these steps to determine how long the unloading will take. Alternatively, decomposition combined with logic can be used to solve this task, if we understand that the number of pairs (1, 2), (2, 3), (3, 4), ..., that appear out of order in the sequence of boxes determines the number of unloading rounds needed.

Sawya's Logs

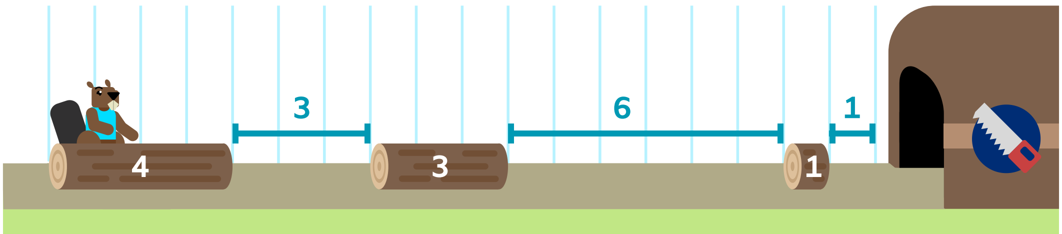
Beaver Sawya cuts wood logs of various lengths and then sells them. Whenever she has finished cutting one she puts the logs down along the 18-meter-long narrow road, one after the other, as the logs cannot fit side by side.

When Sawya puts down a log, she places it in the first available spot from the left where the log fits. When she sells a given log, it is simply removed from the spot it previously occupied.

Sawya has prepared, in this order, logs of lengths 2, 5, 3, 6, and 1 (in meters).
This leads to this arrangement along the road:

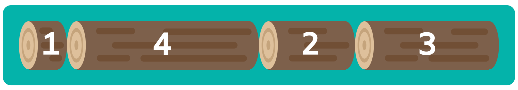


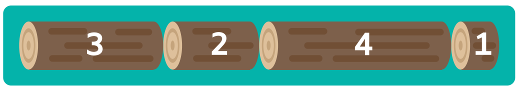
She proceeds to sell the logs with sizes 6, 2, and 5 meters. Then, she cuts a new 4-meter log.
Since she places it as far left as she can, the road now looks like this:

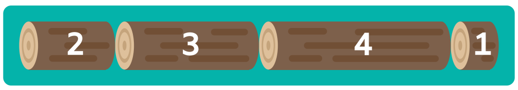


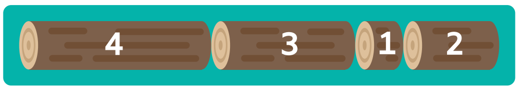
Question

Now Sawya has to cut logs of lengths 1, 2, 3 and 4 meters. Which of the following orders will allow her to store them all on the road above, with the gaps shown, provided she uses the same rule?

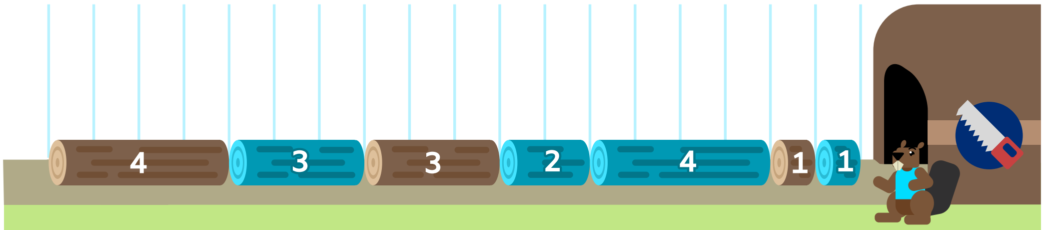








Explanation



The correct answer is 

.
First 3 m, then 2 m, then 4 m, then 1 m.

All other answers lead to Sawya being unable to put down the last log.

The state of the road before Sawya cuts the new logs, as shown in the question, has three free spots: the leftmost spot is 3 meters long, the middle spot is 6 meters long, and the last one, on the right, is 1 meter long.

totally fills the leftmost spot with the first log, then totally fills the middle spot with the next two logs, and the last log goes right into the rightmost spot: this works.

totally fills the leftmost spot with the first two logs, but then puts the 3-meter-long log in the middle spot. Now the middle spot has been reduced to just 3 meters, and the last log, which is 4 meters long, fits neither there nor in the rightmost spot.

puts the 1-meter-long log in the leftmost spot and, next, the 4-meter-log log in the middle spot. The leftmost spot can then still completely be filled with the third log, which is 2 meters long, but there is no sufficient contiguous space to put the last log.

has a similar problem, but even earlier: the third log of 4 meters already cannot be put down. The leftmost and rightmost spots are too small anyway, and the middle spot is half occupied by the second log of 3 meters.


puts the 4-meter-long log in the middle spot, the 3 meter-long log in the leftmost spot. The 1-meter-long log would go next to the 4-meter-long log in the middle, leaving no space for the last log.

Background information

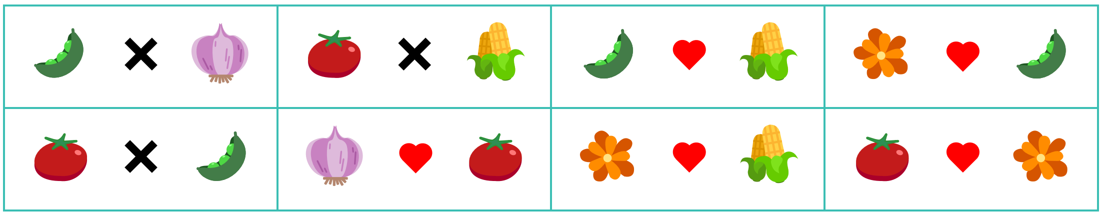
One can see Sawya's road as memory (RAM, stands for Random Access Memory) inside a computer, and the logs being put down and, later, sold, as computer processes requesting a certain amount of memory to run. In this example, although there is enough space to theoretically put down all the logs, the order (and the strategy) in which they come and go makes it sometimes impossible to fit them all. The same can happen in computers: depending on the order in which chunks of memory are allocated and freed, it may be impossible to allocate a new chunk of memory of a given size, even when the total free memory is greater than that size. This problem is known as memory fragmentation. Allocation strategies smarter than that of Sawya or memory models allowing, e.g., chunks to be moved help alleviate the fragmentation problem. Memory fragmentation is not only a problem with RAM: allocating space on a disk for writing files of various sizes leads to very similar problems.

To solve this task, one must evaluate the different proposed orders according to the described strategy, and find out which lead to a possible solution or not. While doing so, one may infer rules of thumb that can help formulate criteria for a given order (or beginning of an order) to be either promising or probably leading to a difficult situation (e.g. by creating several small memory fragments, which will later be unlikely to be used).

Companion Planting

Some plants are said to have "companion relationships". This means they help each other in some way. For example, beans help corn by providing them with nitrogen in the soil, and corn plants help beans by providing support for them to grow up.

In the picture, we can see which plants make good companions (♥), and which do not (✕).

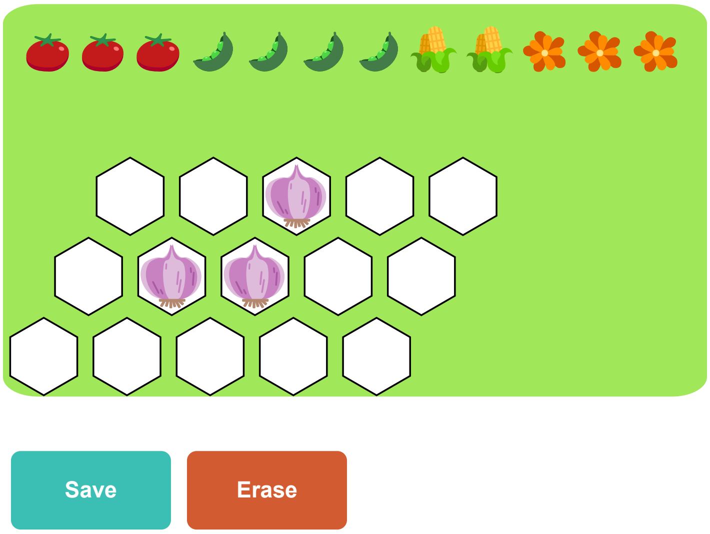


Thalia has a hexagonal garden bed shown in the diagram, with 3 garlic already planted in it. Using the companion relations from above, help Thalia to plant the rest of the plants.

Question

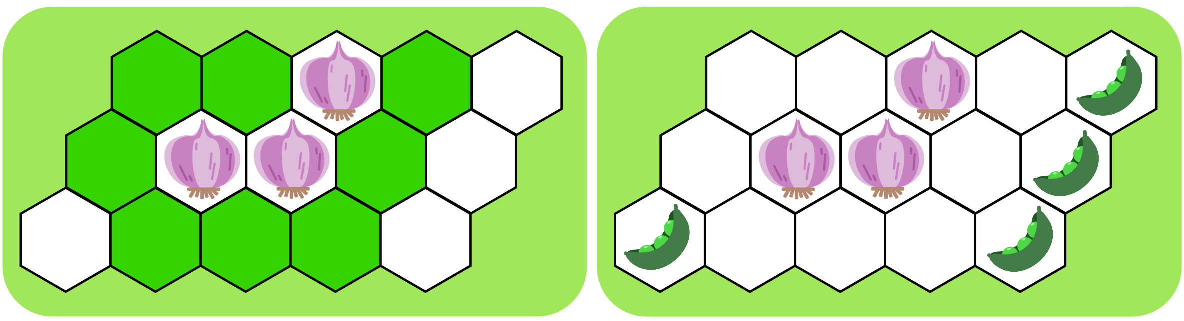
How can she make sure each plant is neighbouring a maximum of plants that will help it grow, and that no plant is touching a plant that hinders its growth?

Drag & Drop each plant into the garden bed, until there are no empty spaces. Remember to press 'Save' when you have finished.

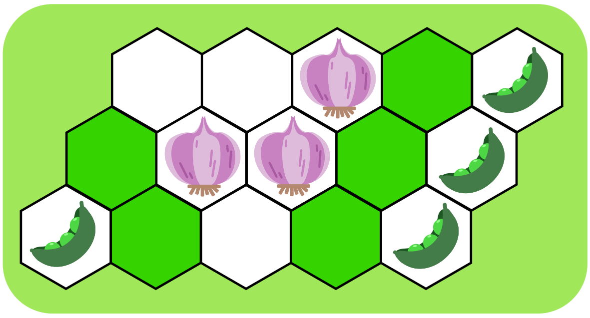


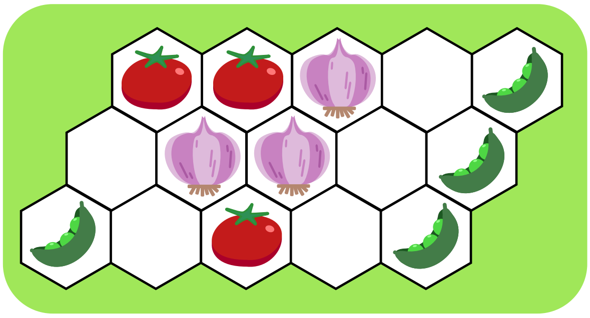
Explanation

First we observe garden with only garlic in it and color with green fields for good companion plants. On those fields we can't plant beans. So it is very clear where we have to plant beans.

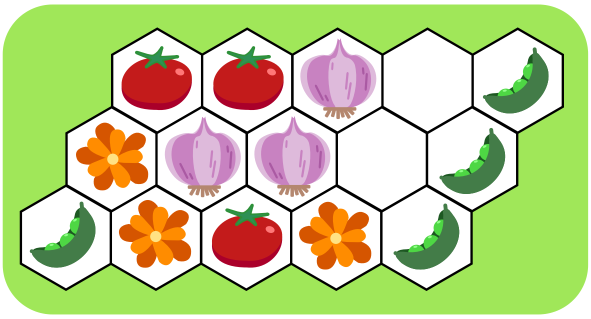


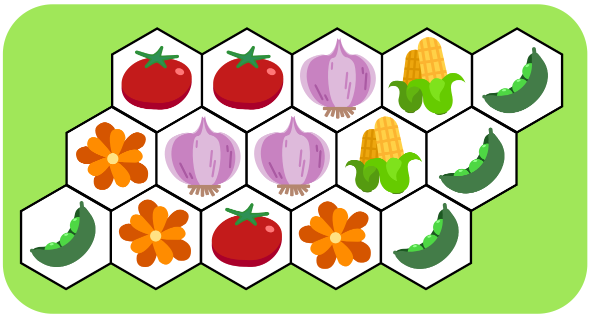
Now, we have one more constraint, and we do the same: color the fields with green for good companions for beans. On those fields we can't plant tomato. So it is very clear where we have to plant tomato:





We have left with 3 flowers and two corns. They are both good companions with beans, but the tomato is more beneficial with flower (attract bees) and it is bad companion with corn. So we will put flowers next to tomato. At last two fields we will plant corns which are good companion both for bean and the flower.





Background information

This problem can be solved in a great many ways. We could use the brute-force approach, where we try placing each plant in every square until every condition is met, but this takes a lot of time. Instead, we better analyze the problem and look for conditions that will reduce the number of options.

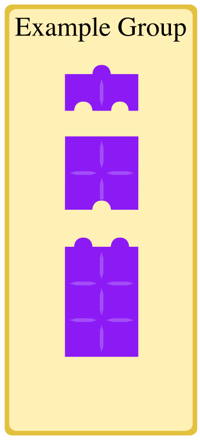
Relations between plants can be represented with an undirected constraint graph: two nodes are connected if they are good companion plants (green solid edges), and their constraints: are which plants can't be near each other (red doted edges).

This task is also combinatorial problem, characterized as a decision problem. For these, the solutions of a given instance are specified by a set of logical conditions. Many business problems can be modeled as a graph problem or an almost comparable sub-problem. They solve optimization, what-if simulation, constraint satisfaction, graph classification, graph drawing and visualization. Farmers and other gardeners use this type of conditional logic when planting out their gardens. This information helps them make informed decisions to optimize the production of their gardens. For example, beans benefit corn by providing them with nitrogen, and corn plants help beans by offering them stability. In computer science, conditions are used to help programs make decisions, based on whether information is true or false.

Beaver Bundles

Bricks in a toy construction set have four characteristics:

1. width (can be: narrow, average, or wide)
2. height (can be: short, average, or tall)
3. number of semi-circles on top (can be: zero, one or two)
4. number of semi-circles cut out of the bottom (can be: zero, one or two)



A beaver wants to organise the bricks into groups of three according to the following rule:

Each characteristic, when looked at one-by-one, has either the same value for each brick in the group, or a different value for each brick in the group.

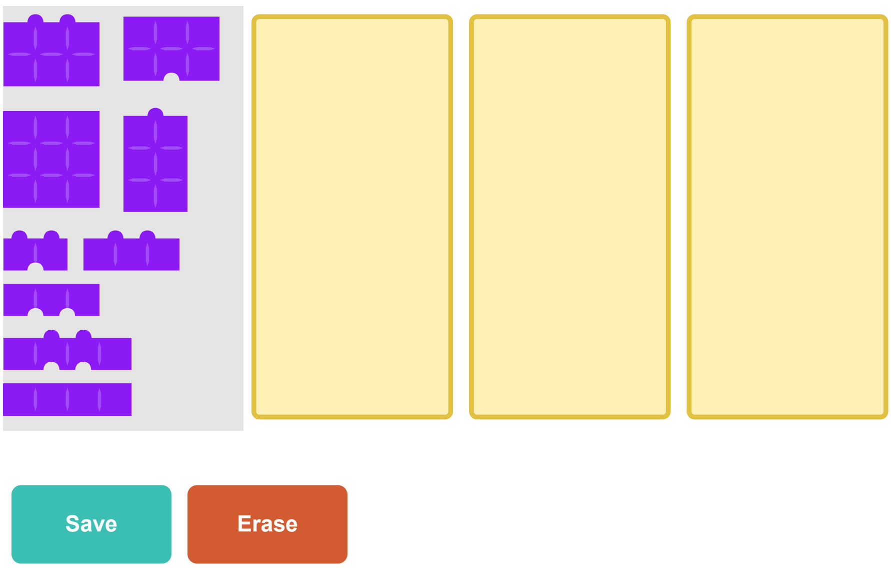
The example group on the right satisfies the rule because:

- each brick in the group has the *same* width value
- each brick in the group has a *different* height value
- each brick in the group has a *different* number of semi-circles on top
- each brick in the group has a *different* number of semi-circles cut out of the bottom

Task:

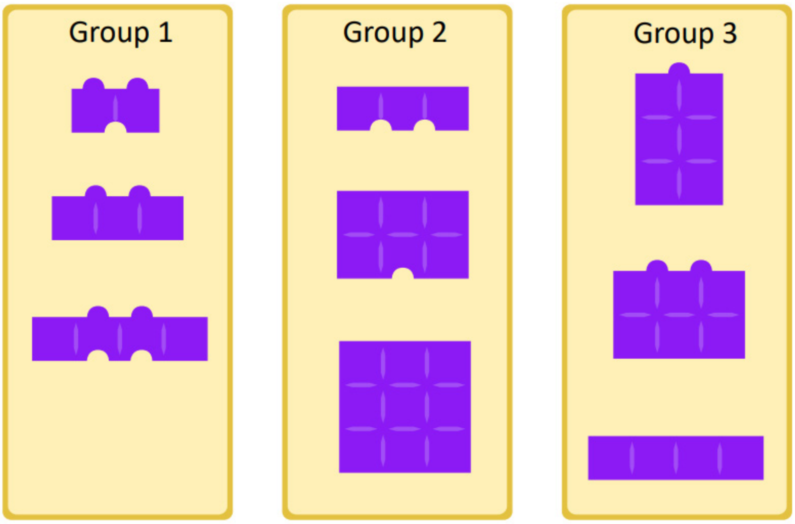
Sort the nine bricks below into groups of three so that each group satisfies the rule.

Drag and drop the bricks to form three groups. Press Save when complete.



Explanation

The correct answer is as follows:

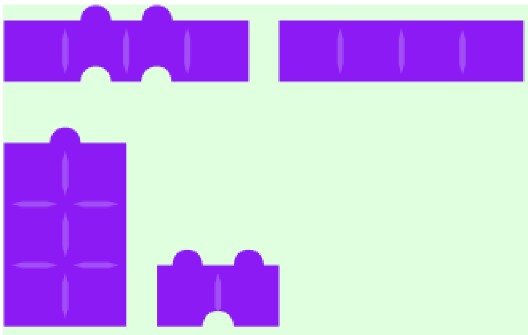


We can verify that this is correct:

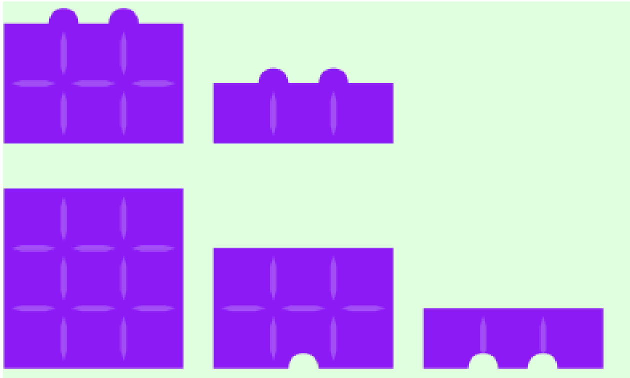
- For Group 1, the width values are different, the height values are the same, and each piece has two semi-circles on top and a different number of semi-circles cut out of the bottom.
- For Group 2, the width values are the same, the height values are different, and each piece has zero semi-circles on top and a different number of semi-circles cut out of the bottom.
- For Group 3, the width values are different, the height values are different, and each piece has a different number of semi-circles on top and zero semi-circles cut out of the bottom.

There are several ways to find this correct answer and to verify that it is the only correct answer. One approach is to test all ways of dividing these pieces into three groups. That could take a long time! A different approach using careful reasoning is described next. (You might need to manipulate these pieces to convince yourself at each step.)

There are only two pieces that have the width value "wide" and two pieces that have the width value "narrow":



This means there must be a group where all the pieces have the width value "average". Of the five pieces that have the width value "average", two have two semi-circles on top and three have zero semi-circles on top.



Therefore, these three with zero semi-circles on top must form a group which is Group 2 above.
To finish, we need to divide the remaining six pieces into two groups where all the pieces in each group have different width values. We can now focus on the characteristic height. Notice that there must be a group of pieces that all have the height value "short" and a group of pieces that all have different height values. By looking at the bottom of these pieces, we can determine that Group 3 must be formed.
The remaining three pieces must form the final group which is Group 1.

2024 Round 1 All

ID	Question	Year Level	Decomposition	Pattern Recognition
2023-SK-04	Zoo Walk	3/4 Easy	●	
2023-LT-01	Hat Lineup	3/4 Easy	●	
2023-CZ-03	Sliced Apples	3/4 Easy	●	●
2023-SA-01	Ball Sort	3/4 Easy	●	
2023-CA-01	Magic Tree	3/4 Easy	●	
2023-CA-02	Riccas	3/4 Medium, 5/6 Easy	●	●
2023-DE-06	Tile Land	3/4 Medium	●	
2023-CH-01	Umbrella	3/4 Medium, 5/6 Easy	●	●
2023-AT-01	A Day at the Zoo	3/4 Medium, 5/6 Easy	●	
2023-DE-02	Flower Shop	3/4 Medium, 5/6 Easy	●	
2023-LT-02	Photo	3/4 Hard, 5/6 Easy	●	●
2023-PK-08	Luggage Check-in	3/4 Hard, 5/6 Easy	●	
2023-DE-04	Karla's Dream House	3/4 Hard, 5/6 Medium, 7/8 Easy	●	
2023-SK-02	Planting Carrots	3/4 Hard, 5/6 Medium	●	
2023-UY-01	Beaverbeard's Gold	3/4 Hard, 5/6 Medium, 7/8 Easy	●	
2023-PE-02	Thomas and His Friends	5/6 Medium, 9/10 Easy	●	
2023-SK-07	Closer or Further	5/6 Hard, 7/8 Medium, 9/10 Easy	●	
2023-IN-03b	Unloading	5/6 Hard, 7/8 Medium, 9/10 Easy	●	
2023-CH-05	Sawya's Logs	5/6 Hard, 7/8 Medium	●	
2023-AU-01b	Companion Planting	5/6 Hard, 7/8 Medium	●	
2023-AU-05a	Beaver Bundles	5/6 Hard, 7/8 Hard, 9/10 Medium	●	●
2023-US-03	Autonomous Car	7/8 Easy	●	
2023-RO-01	Trains	7/8 Easy	●	
2023-BR-05	Juice Carts	7/8 Medium, 9/10 Medium, 11/12 Easy	●	
2023-ME-03b	New Toy	7/8 Hard, 9/10 Easy	●	●
2023-IE-02b	Ogham Code	7/8 Hard, 9/10 Medium, 11/12 Easy	●	●
2023-US-01	Walking Holiday	7/8 Hard, 9/10 Hard, 11/12 Easy	●	●
2023-CS-01uk	Bumping Robots	7/8 Hard	●	
2023-RO-02	Reverse Polish Notation	9/10 Medium, 11/12 Easy	●	
2023-HU-37	Beaver Building Company	9/10 Medium, 11/12 Easy	●	
2023-BE-01	Emma's Tasks	9/10 Hard, 11/12 Medium	●	
2023-TW-04b	Uphill	9/10 Hard, 11/12 Medium	●	●
2023-NZ-01	Bridge Building	9/10 Hard	●	●
2023-DE-08	Zerobot's Dilemma	9/10 Hard, 11/12 Medium	●	

Abstraction	Modelling & Simulation	Algorithms	Evaluation	Abstraction	Data Collection	Data Representation	Data Interpretation	Specification	Algorithms	Implementation	Digital Systems	Interactions	Impacts
	●		●			●					●		
●		●		●				●	●	●			
●	●		●	●		●	●	●					
●		●		●		●		●	●		●		
●		●	●	●		●			●		●		
					●	●	●						
●		●	●	●		●	●	●	●				
		●	●			●	●	●					
●		●	●	●	●	●		●	●	●			●
		●	●		●	●		●	●	●			
●		●	●	●		●	●	●					
●	●	●	●	●		●	●		●	●	●		
●		●	●	●		●		●	●				
●		●	●	●		●	●	●	●	●			
●	●	●	●	●		●	●	●					
●	●	●	●	●	●	●	●	●	●			●	
●		●	●	●	●	●	●	●	●		●		
●		●	●	●				●	●	●			
●			●	●				●	●		●		
●		●		●		●		●	●				
●		●	●	●				●	●	●			
●		●	●	●					●	●			●
●		●	●	●				●	●	●	●		
●	●		●	●				●	●			●	
●		●	●	●				●	●	●	●		
		●	●			●	●		●	●			
●	●	●	●	●				●	●	●			
	●	●	●					●	●	●			
			●			●		●		●	●		
●		●	●	●			●	●	●	●	●		
●	●	●	●	●				●	●	●		●	
●	●	●	●	●				●	●	●	●		
	●	●	●				●	●	●	●		●	
●	●	●	●	●				●	●				●

ID	Question	Year Level	Decomposition	Pattern Recognition
2023-IR-02	Job Assignment	11/12 Medium	●	
2023-UA-01	Electronic Lock	11/12 Medium	●	
2023-DE-01uk	Disagreement Detector	11/12 Hard	●	
2023-AT-10	Painting by Numbers	11/12 Hard	●	●
2023-DE-10	Random Gift Paper	11/12 Hard	●	
2019-KR-04	Buying Jeans	11/12 Hard	●	
2023-DE-09	Domino	11/12 Hard	●	●

Abstraction	Modelling & Simulation	Algorithms	Evaluation	Abstraction	Data Collection	Data Representation	Data Interpretation	Specification	Algorithms	Implementation	Digital Systems	Interactions	Impacts
●		●	●	●				●	●				
		●	●			●	●	●	●				
	●	●						●	●	●	●		
●		●		●					●				
●	●	●	●	●				●	●	●			
●		●	●	●					●	●	●		
	●	●	●					●	●	●	●		