Bebras Australia Computational Thinking Challenge

2018 Solutions Guide

Adapted by: Allira Storey, Edited by: Sarah Hobson

www.bebras.edu.au

Thank You!

CSIROs Education & Outreach Program, Digital Careers, is privileged run the Bebras Challenge in Australia. We would like to thank the International Bebras Community for their support of the Australian Challenge as well, of course, developing the challenge questions. This year, Bebras Australia runs with a similar set of questions as the UK and we would like to thank Bebras UK for their generous support.

We would like to thank the Australian Government for funding CSIRO's Digital Careers program which manages Bebras Australia, and the Australian Curriculum, Assessment and Reporting Authority (ACARA) for providing advice on Bebras questions. Special thanks go to Eljakim Schrijvers and Daphne Blochuis for their ongoing support for the Australian Challenge. In addition, we would also like to thank the following individuals for their help and support: Peter Millican, Chris Roffey and Sue Sentence.

International Community

Bebras is an international initiative whose goal is to promote computational thinking to students in years 3-12. Questions are developed as an international community which means you can see what country created a question by looking at the flag in the top left corner of each question page.

UK Beaver

South Korea Beaver

What is a Solutions Guide?

Computational Thinking skills underpin many careers of the future. Creating opportunities for students to engage in activities that utilise their problem solving skills is the foundation for further learning. The Bebras Challenge is an engaging way for students to learn and practice these skills in the classroom environment.

On the following pages you will find the 43 tasks used in the Australian Bebras 2018 Challenge. Above each question is noted the age groups and level. Level A is easy, level B is medium and level C is hard.

After each question there is an answer, an explanation of how the answer is obtained, plus a section on how the questions are related to Computational Thinking. We have also mapped the questions to the Australian Digital Technologies Curriculum Concepts.

Computational Thinking Skills	**Concepts**
Decomposition - DE Break problems into parts **Pattern recognition - PR** Analyse the data, look for patterns to make sense of the data **Abstraction - AB** Remove unnecessary details and focus on the important data **Modelling and simulation - MS** Create models or simulations to represent processes **Algorithms - AL** Create a series of ordered steps taken to solve a problem **Evaluation - EV** Determine effectiveness of a solution, generalise and apply to new problems	•Abstraction - underpins all content •Data collection (properties, sources, collection of data) •Data representation (symbolism, separation – how digital systems represent data i.e. Binary)) •Data interpretation (patterns and contexts) •Specification (descriptions and techniques) •Algorithms (following and describing) •Implementation (translating and programming) •Digital systems (hardware, software, networks, Internet) •Interactions (people, digital systems, data and processes) •Impacts (sustainability and empowerment)

Question Contents

How many routes	page 8
Bird house	page 9
Sorting branches	page 10
The way home	page 11
Small program	page 12
Dancing man	page 13
Sudoku	page 15
Worm	page 16
Strawberry hunt	page 17
Colour Paths	page 19
Parking Lot	page 21
Message service	page 22
Beaver tournament	page 24
Grandmother's jam	page 26
Five sticks	page 28
Sticks and shields	page 30
Erase walls	page 32
News editing	page 34
Roundabout city	page 36
Brackets	page 38
Balls	page 40
Cipher wheel	page 42
Painting wallpaper	page 44
Railroad	page 45
Commuting	page 47
Candy maze	page 49
Book-sharing club	page 51
One too many	page 53
Sports	page 54
Stream	page 56
Super hero	page 58
Digit recognition	page 60
Arabot's walk	page 62
Levenshtein distance	page 64
Tunnels of the homestead dam	page 66
Cost reduction	page 69
Icon image compression	page 71
Robot	page 73
Funtime School	page 75
Wash the uniforms	page 77
Perfect Partners	page 79
Downloads	page 81

Matrix

2018 Questions	Years	Level	Decomposition	Pattern Recognition	Abstraction	Modelling & Simulation	Algorithms	Evaluation
How many routes	3-4	A	X		X		X	
Bird house	3-4	A	X	X	X		X	
Sorting branches	3-4	A	X		X	X	X	
The way home	3-4	C	X	X	X		X	
	5-6	B						
Small program	5-6	C	X		X		X	X
	7-8	B						
Dancing man	9-10	A	X	X	X		X	
Suduko	3-4	A	X		X		X	
Worm	3-4	B	X		X	X	X	
	5-6	A						
Strawberry hunt	3-4	B	X		X	X	X	
	5-6	A						
Colour paths	3-4	A	X		X		X	
Parking lot	3-4	B	X		X		X	
	5-6	A						
Message service	3-4	C	X		X		X	
	5-6	B						
	7-8	A						
Beaver tournament	3-4	C	X		X		X	
	5-6	B						
	7-8	A						
Grandmother's jam	3-4	B	X		X		X	
	5-6	A						
Five sticks	5-6	C	X		X	X	X	
Sticks and shields	5-6	B	X		X	X	X	
	7-8	A						

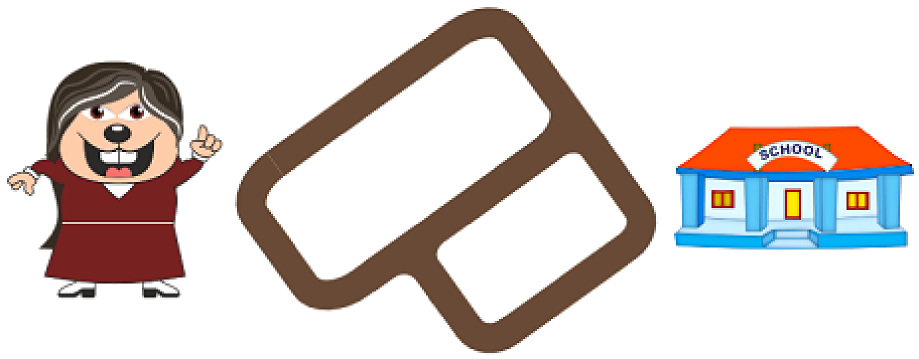
	9-10	A						
Erase walls	5-6	C	X		X		X	X
	7-8	B						
News editing	5-6	C	X		X		X	
Roundabout city	5-6	C	X		X	X	X	
Brackets	7-8	B	X	X	X		X	
	9-10	A						
Balls	7-8	B	X		X	X	X	
Cipher wheel	7-8	C	X		X		X	X
	9-10	B						
Painting wallpaper	5-6	B	X		X		X	
	7-8	A						
Railroad	9-10	A	X		X	X	X	
Commuting	11-12	A	X		X		X	
Candy maze	7-8	C	X		X		X	
Book-sharing club	9-10	A	X		X	X	X	
	11-12	A						
One too many	9-10	B	X		X	X	X	X
	11-12	A						
Sports	7-8	B	X		X		X	
	9-10	A						
Stream	5-6	A	X		X		X	
Super hero	9-10	B	X		X		X	
	11-12	B						
Digit recognition	11-12	B	X		X			
Arabot's walk	9-10	B	X		X		X	
	11-12	A						
Levenshtein distance	9-10	C	X		X		X	X
	11-12	B						

Tunnels of the homestead dam	9-10	C	X		X	X	X	
	11-12	B						
Soda shop	7-8	C	X		X		X	
Cost reduction	7-8	C	X		X	X	X	
	9-10	B						
	11-12	B						
Icon image compression	9-10	C	X	X	X		X	
	11-12	C						
Robot	9-10	C	X		X	X	X	X
	11-12	C						
Funtime school	7-8	C	X		X		X	
	11-12	A						
Wash the uniforms	11-12	C	X		X		X	X
Perfect partners	11-12	C	X		X	X	X	
Downloads	9-10	C	X		X	X		X
	11-12	C						

How many routes

Year 3+4: **A**	Year 9+10:
Year 5+6:	Year 11+12
Year 7+8:	

Beaver Jane regularly walks to school.



Jane likes to change her route each day.

She only walks on paths that take her nearer to the school.

Question:

How many different routes can Jane take to school?

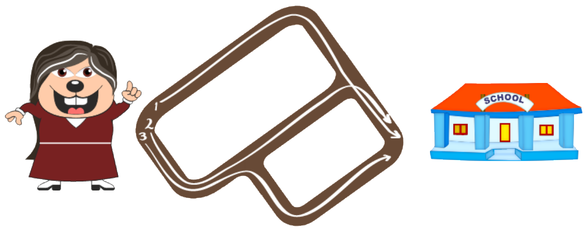
1 2 3 4 5 or 6

Answer:

The correct answer is 3.

Explanation:

This image shows all possible routes:



It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

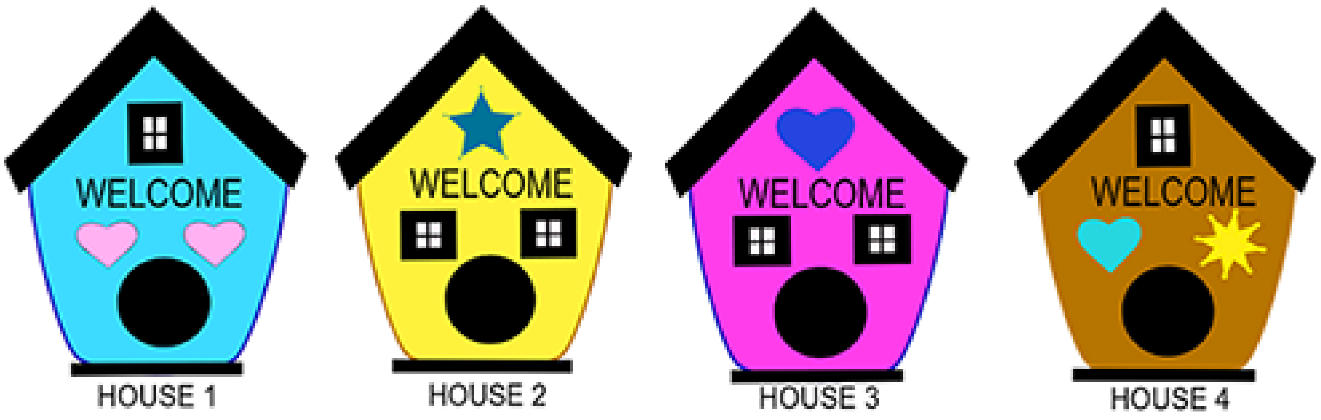
Australian Digital Technologies Curriculum Concepts: Abstraction and Algorithms

It is a common problem, in Computer Science, to have to find routes through networks in computers, and also through communication and transport networks. Although this problem is very simple, as the size of the map becomes larger, it becomes much more difficult to solve. A more common problem is to find the shortest or fastest route.

Bird house

Year 3+4: **A**
Year 5+6:
Year 7+8:
Year 9+10:
Year 11+12

A beaver wants to buy a bird house for her daughter's birthday.
Her daughter says: "I would like a bird house with 2 windows and a heart".



Question:
Which bird house should her Mum buy?

Answer:
The answer is bird house 3.

Explanation:

House 1 is not correct because it has only one window and 2 hearts. House 2 is not correct because it has 2 windows, but no heart. House 4 is not correct because it has only one window.

It's Computational Thinking:
Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

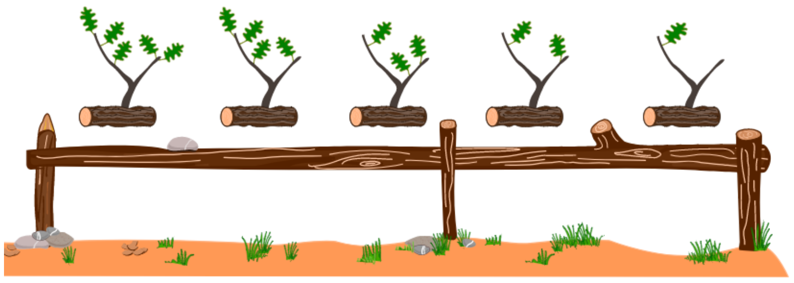
Australian Digital Technologies Curriculum Concepts: Abstraction, Data Collection, Specification and Algorithms

Abstraction is the process of extracting the most important or defining features from a problem or a challenge. The extracted features provide the information to begin examining the challenge and find a potential solution. It involves filtering out unnecessary details and only looking at information that is relevant to solving the problem. This can help by highlighting the similarities and differences in aspects of a problem. The current task focuses on identifying the objects with one heart and two windows.

Sorting branches

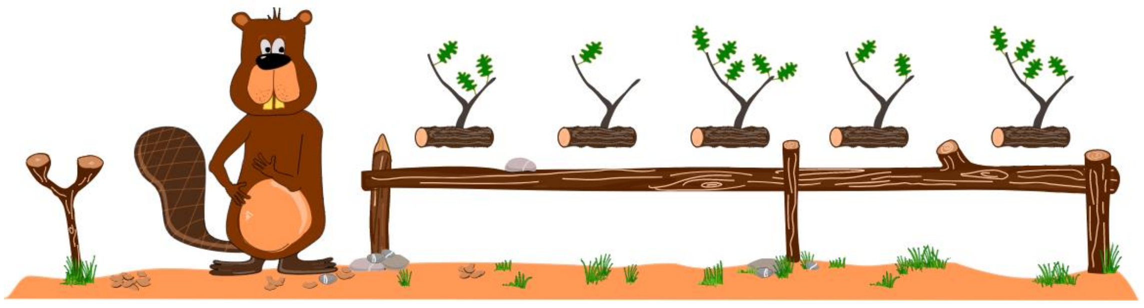
Year 3+4: **A**
Year 5+6:
Year 7+8:
Year 9+10:
Year 11+12

David wants to sort some branches into this order:



He can place one branch on the rickety stick on his right if he needs to.

Question:
Help David sort the branches below by dragging and dropping them into new positions.



Answer:

The answer is shown in the task description.

Explanation:

To solve this task it is necessary to make space on the shelves by using some temporary storage.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS) and Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Algorithms, Digital Systems and Interactions

Swapping two pieces of data by using a new temporary storage location is used widely in Computer Science. For example, we can use a sorting algorithm such as a *selection sort* when sorting a list of numbers, with a limited amount of memory. Starting with the first number in the list, swap it with the smallest number in the rest of the list, using a temporary variable. Continue across the list until the last number is reached.

The way home

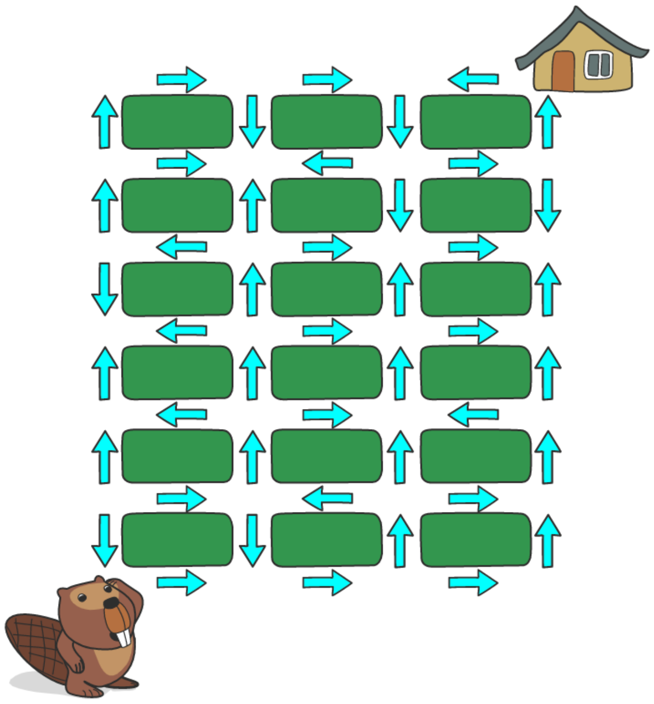
Year 3+4: **C**
Year 5+6: **B**
Year 7+8:
Year 9+10:
Year 11+12

Mark the path from the beaver to his home.

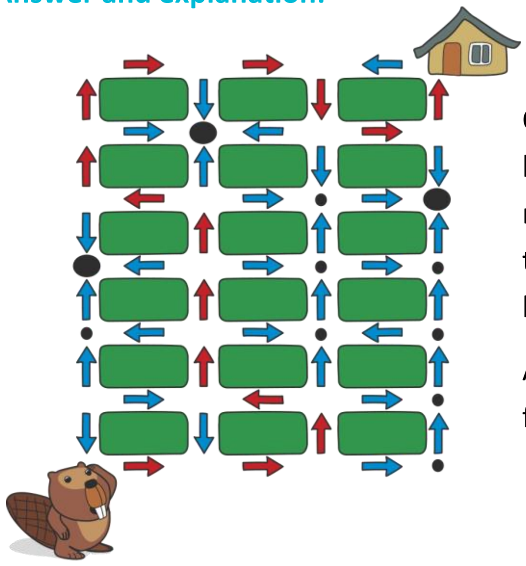
You can click on one arrow at time to make the correct sequence.

(To delete your choice, click on the arrow a second time.)

Question:
Show the path home without creating any loops.



Answer and explanation:



One way of solving this is to first identify *black holes* (see big black dots on the left) where the beaver can enter but not escape. We can also identify places that can only lead to a black hole (little black dots). The answer then becomes obvious.

An alternative strategy is to follow the arrows backwards from the house.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Pattern Recognitions (PR), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Interpretation, Specification and Algorithms

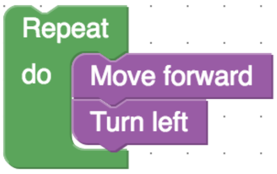
Finding a route is one of the classical problems in algorithm theory. *Backward searching* and identifying *black holes* are two algorithmic techniques used to solve such problems.

Small program

Year 3+4: **C**
Year 5+6: **B**
Year 7+8: **A**
Year 9+10:
Year 11+12

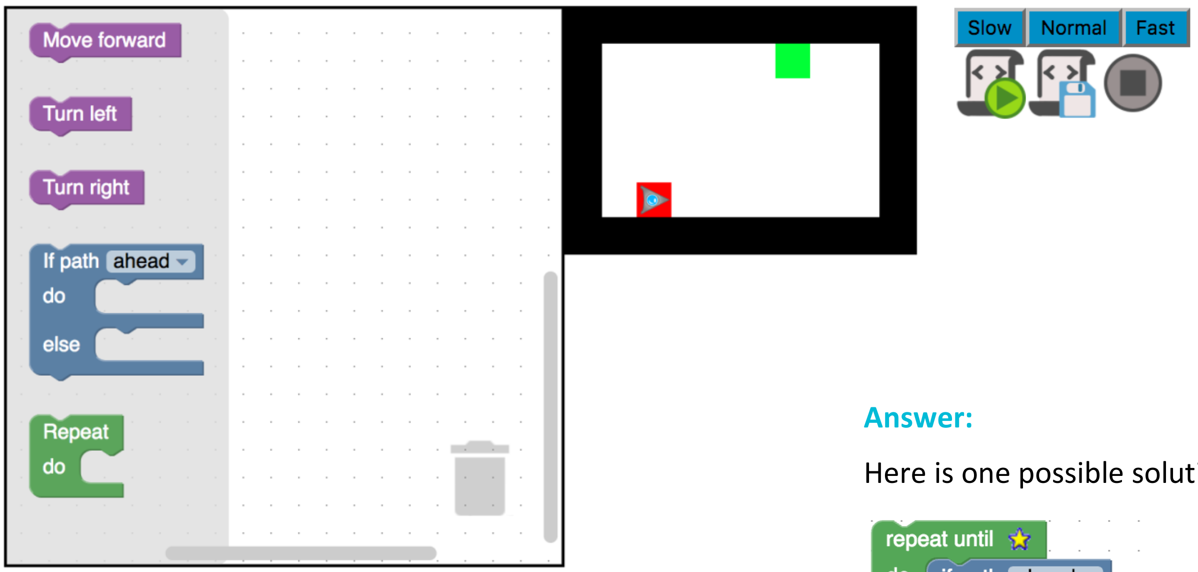
You can write a program for a robot by dragging and joining commands into the workspace below.

An example program:



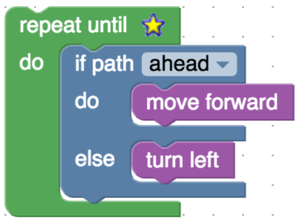
The robot has to move from the red square to the green square.

Question:
Write a program that takes the robot to the green square.
(Your program must be made from 4 blocks or less.)



Answer:

Here is one possible solution:



It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL), Evaluation (EV)

Australian Digital Technologies Curriculum Concepts: Abstraction, Specification, Algorithms, Implementation and Digital Systems

In mobile robotics, navigation is a common problem. Maze solving is not so common but requires similar Computational Thinking skills to be employed. To solve this problem, an autonomous robot is used. Mazes can be of different kinds: having loops, without any loops, grid systems or without a grid system. In this problem, the robot has very limited memory space so a loop is required to keep the program small. This does mean, however, that the robot cannot necessarily take the shortest route.

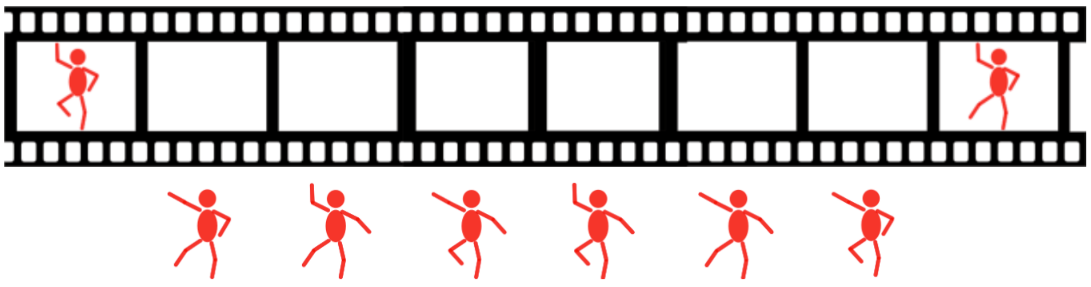
Dancing man

Year 3+4:
Year 5+6:
Year 7+8:
Year 9+10: **A**
Year 11+12

Verity makes an animation of a man dancing. So far she has only completed the first and last frame.

The man can only move one of his arms or legs at a time.
There should be only one difference between film frames that are next to each other.

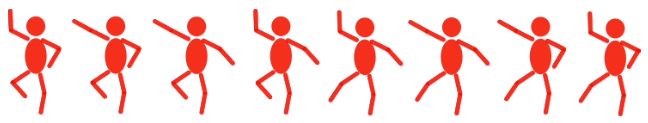
Question:
Drag and drop the images provided into the correct empty film frames below to complete the animation.



Dancing man

Answer:

One possible answer is:



Explanation:

There is only one frame sequence that satisfies the statement. To get to that sequence, start with the beginning frame and add frames that satisfy the requirements. If you are stuck and there are no frames to continue with, try to do a so-called *rollback*: remove the last added frame and add another suitable one. If this is also impossible, *rollback* again, and remove one more frame. You are unlikely to do any rollback at all; there is a high probability in this task that you will get the correct sequence the first time.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Collection, Algorithms, Implementation and Digital Systems

The *rollback* system is an example of a brute force algorithm. These are easy to write for the programmer but are inefficient in many situations.
Gray codes explanation: This is suitable for the situation when there are more than 3 possible movements.

Note that the man on all of the different frames differs only in three types of movements: with the left arm, the right arm, and with the right leg. We can encode the dance positions in the frames with three binary digits, for example, 001 will mean a dance position with the straight left arm, and bent right arm and leg.

One of the sequences may be obtained from binary representations of numbers from 0 to 7 in the following way: start the encoding with all zeros. The next encoding differs from the previous in exactly the same position where in binary representation a new "one" appears:

number	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
binary representation	000	001	010	011	100	101	110	111
dance positions	000	001	010	010	110	111	101	100

The task is based on the idea of binary Gray codes. They appeared in the design of electromechanical switches, and now they are used to correct errors in signal transmission, in encoding of numbers of hard disks paths, and in other various algorithms.

Sudoku

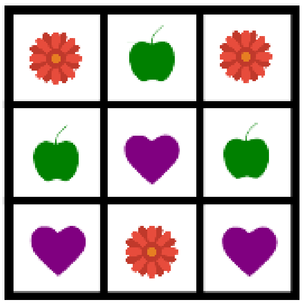
Year 3+4: **A**	Year 9+10:
Year 5+6:	Year 11+12
Year 7+8:	

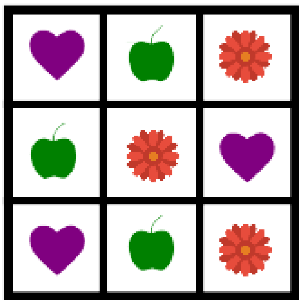
Benjamin is asked to fill a box with different shapes.
The box has 9 sections.

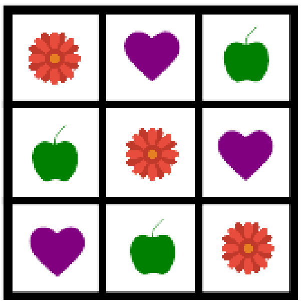
Rules:
There must be only one of the same shape in each row.
There must be only one of the same shape in each column.

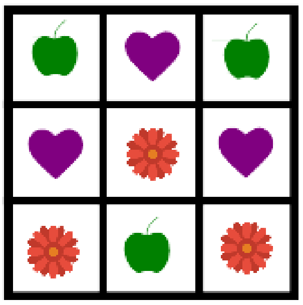
Benjamin has four goes!

Question:
Which of the following boxes is correct?


A


B


C


D

Answer:
The answer is C.

Explanation:
A is wrong because there is at least 1 column with two of the same shapes.
B is wrong because there is at least 1 row with two of the same shapes.
D is wrong because there is at least 1 row with two of the same shapes.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Collection, Specification, Algorithms and Interactions

To solve a common Sudoku exercise, we must look at all of the rows and all of the columns.

One way to do this is to look at all the rows first, and then all the columns.
Another way to do this is to look at all the columns first, and then all the rows.

If you find an incorrect row or column, it means that the box is incorrect, so you can immediately eliminate the box from the possible answers.

In life, elimination can help you to solve many problems. For example, when answering a question with multiple choice answers in a test, you can eliminate incorrect ones to minimise the possible choices.

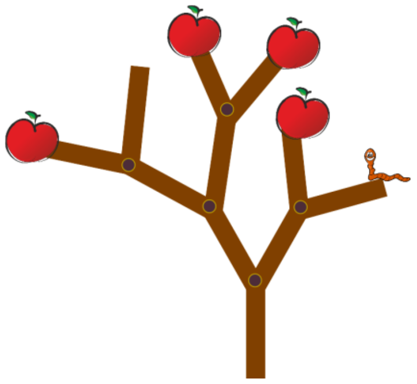
Worm

Year 3+4: **B**
Year 5+6: **A**
Year 7+8:
Year 9+10:
Year 11+12

A worm is sitting at the end of the branch on the tree shown below.

It wants to eat all of the apples by moving through the tree's branches.

(The tree is made of 1m long branch sections.)



Question:
What is the shortest distance, in metres, that the worm has to crawl to eat all of the apples?

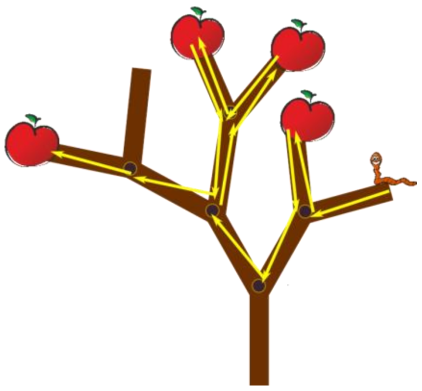
4 9 13 or 15

Answer:

The correct answer is 13.

Explanation:

The worm has to reach all apples. 4 and 9 are not the correct answers, because it is not possible to reach all apples passing 4 or 9 branch parts only. A path with length 13 allows the worm to reach all apples. First it has to reach the closest apple and then the remaining three apples. Notice, it does not matter in which order the worm will reach the remaining apples. The path with length 15 is too long.



It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Collection, Data Representation and Algorithms

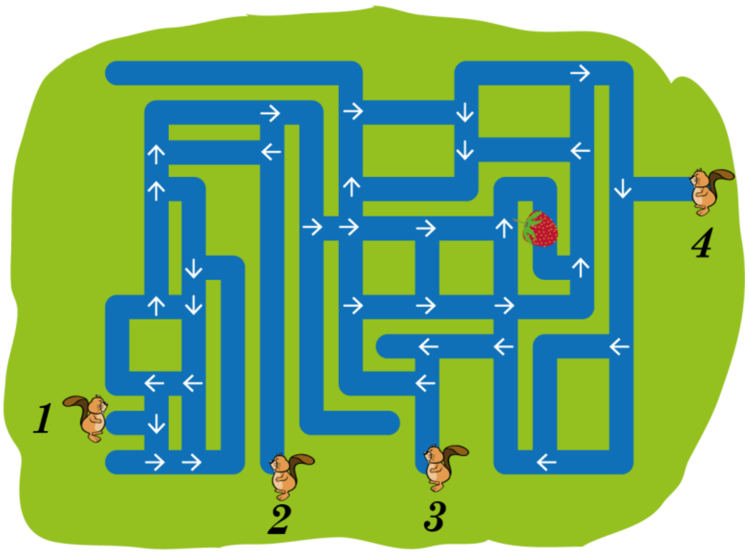
In Computer Science you not only try to plan the instructions for solving the task, but often you want to find the solution that requires the minimum amount of work. In such cases, Computer Scientists say you are optimising the solution. The tree in this task represents a special kind of diagram in which some relevant points of the tree are connected by branch sections. Computer Scientists call these diagrams *graphs*. So this task is in fact about finding a special path in a graph. The path should be the shortest one starting from the worm, and all apples have to be on the path at some point.

Strawberry hunt

Year 3+4: **B**
Year 5+6: **A**
Year 7+8:
Year 9+10:
Year 11+12

Four beavers start swimming from different places.

They can only swim forwards and always follow the arrows.



Question:
Select all the beavers who will reach the strawberry.

Beaver 1 **Beaver 2** **Beaver 3** **Beaver 4**

Strawberry hunt

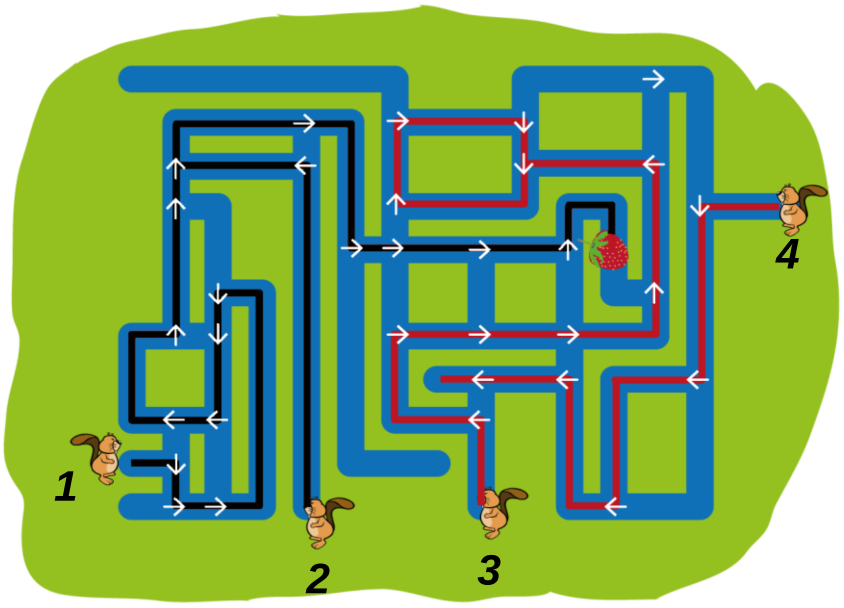
Answer:

2 beavers get to the strawberry.

Explanation:

Beavers 1 and 2 on the left can reach the strawberry. Their paths actually meet.

Beaver 3 will start swimming in a circle. Beaver 4 will swim into a dead end and cannot get out.



It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS) Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation, Specification, Algorithms and Implementation

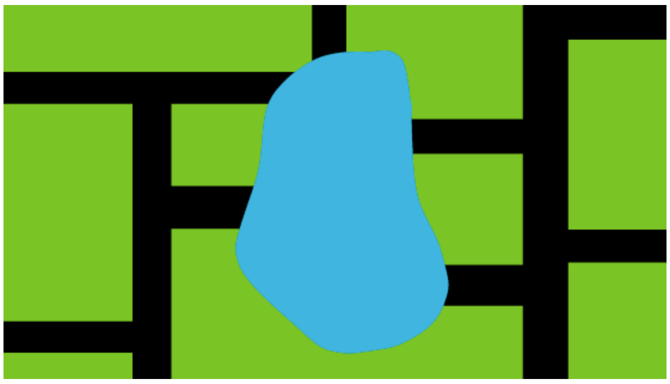
The system of canals in which the beavers are swimming has two main elements: canals (where the beavers can swim through) and crossings (where the beavers have to decide, by the arrow, into which canal to swim next). In Computer Science this system is called a *graph with edges* (the canals) and *nodes* (the crossings). In this case the nodes have extra information attached to them: which canal should the beaver swim into next.

Graphs can be used to describe situations like this task. They can also be used for programming a computer: the computer is following a path in the graph and at each crossing it receives an instruction on what to do next. In some cases it ends up solving the problem (which would be the beaver reaching the strawberry) and in some cases it ends up in a dead end or even never finishes the program (like the two other beavers).

Colour paths

Year 3+4: **A**
Year 5+6:
Year 7+8:
Year 9+10:
Year 11+12

John chooses a painting program on his computer to draw a picture of a park.
The park has a blue pond and black paths around it.



He wants to change the colour of the paths to brown.
He chooses the fill tool on his painting program and the brown colour so that he can paint the black paths brown.

Question:
What is the smallest number of clicks he needs to make to paint all the paths brown?

1 2 3 10 or 15

Colour paths

Answer:

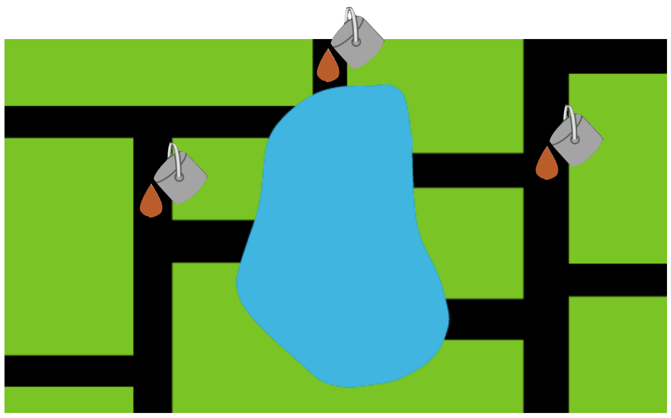
The answer is 3.

Explanation:

John had to click on the image 3 times because the fill tool only paints paths which are connected.

The image shows possible locations he clicked on when the fill tool and brown colour were selected.

There is no solution with less clicks because the pond divides the paths into three separate parts.



It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Algorithms and Digital Systems

It is often important in Computer Science to complete tasks efficiently with as few actions as possible. The same result could also be achieved by clicking on every path multiple times but this wastes effort.

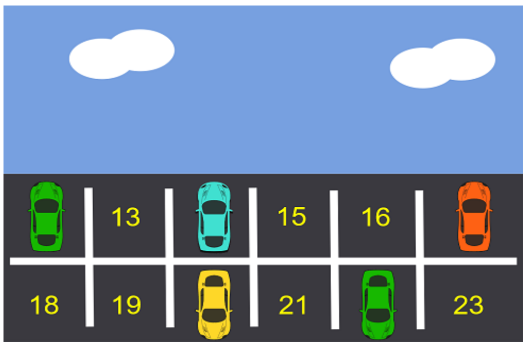
It is also important to know what the computer tools do. A computer will only do exactly what you tell it to do. You must click on the fill tool and then click on each section of the path that has adjacent pixels of the same colour. The path is separated into sections by the pond. You must repeat this for each section of the path.

Parking lot

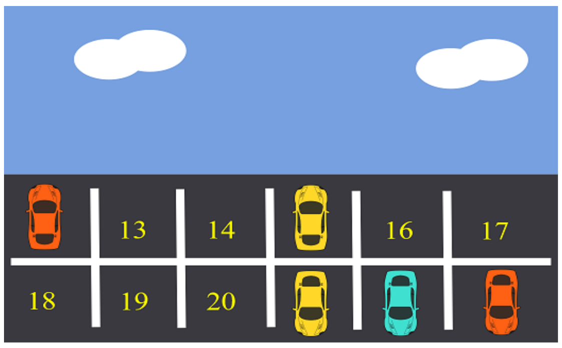
Year 3+4: **B**	Year 9+10:
Year 5+6: **A**	Year 11+12
Year 7+8:	

There are 12 spaces for cars in a parking lot. Each space is labelled with a number.
The pictures below show which spaces were used on Monday and which spaces were used on Tuesday.

Monday



Tuesday



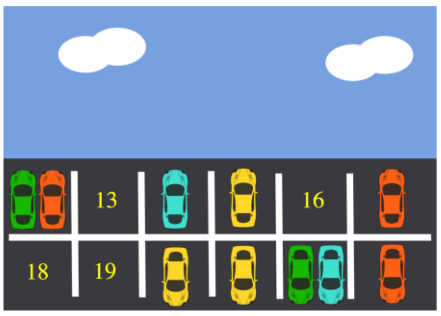
Question:

How many parking spaces were empty both on Monday and Tuesday?

3 4 5 or 6

Answer:

The answer is 4.



Explanation:

Placing the pictures of the cars from both days together in the parking spaces, gives the image on the right. Then all we have to do is count the empty spaces.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation, Specification and Algorithms

All data can be thought of as a sequence of zeros and ones. Each zero or one is called a *bit* and the sequence is called a *binary code*, *binary representation*, or *binary number*.

Here, we can model the presence of a car as a 'one' and an empty parking space as a 'zero'; so the parking space corresponds to a *bit*. We get a sequence of *bits* if we view the parking spaces in order. For example, we might move across the top row and then along the bottom row to get 101001001010 from the parking lot on Monday and 100100000111 from the parking lot on Tuesday. This task asks you to determine which of the 12 positions contain a 1 in either of these binary numbers. This is a logical operation named *OR*. Notice how we can compute the correct answer by seeing that 101001001010 *OR* 100100000111 gives 10110100111. This resulting binary number has four zeros in it.

Message service

Year 3+4: **C**
Year 5+6: **B**
Year 7+8: **A**
Year 9+10:
Year 11+12

Violet wants to send a message to Leo with the help of some beavers and some cards.

She breaks the message into groups of, at most, 3 letters on each card. She then gives one card to each beaver.

Violet knows that sometimes the beavers get distracted while transporting their cards, and they arrive at different times. Therefore, Violet also numbers the cards in the correct order before giving them to the beavers. Leo must then put then back in order to read the message.

Example:
To send the message DANCETIME, Violet creates these 3 cards:

1 DAN	2 CET	3 IME

Leo received the following sequence of cards from the beavers:

3 KEY	5 CKS	2 HOC	1 GET	4 STI

Question:
What was the original message that Violet sent to Leo?

GETSTICKYSHOCKS STICKYGETHOOKS GETHOCKEYSTICKS KEYCKSHOCGETSTI

Message service

Answer:

The original message was GETHOCKEYSTICKS

Explanation:

This problem is easily solved by putting the cards in the correct order:

1	2	3	4	5
GET	HOC	KEY	STI	CKS

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Specification, Algorithms and Digital Systems

When data, such as email messages, images or video, is sent around the internet, it is broken up into small packets, where each packet can hold the equivalent of 65536 characters: notice that 2^{16} = 65536. These packets are then sent through routers with some extra information about the ordering, the source of the packet and the destination of the packet. All of this extra information helps ensure that even if the information is received out of order or has errors in it, the original message can be reconstructed by the receiver of the packets.

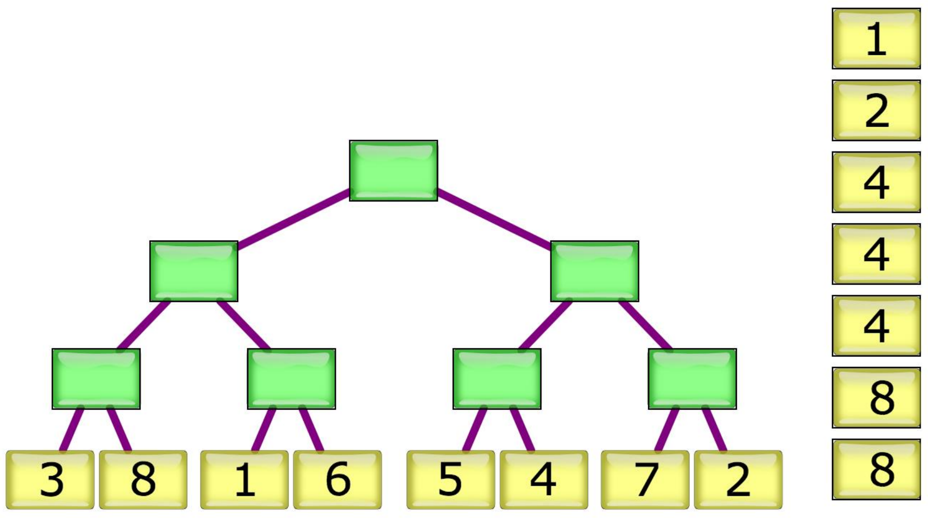
Beaver tournament

Year 3+4: **C**
Year 5+6: **B**
Year 7+8: **A**
Year 9+10:
Year 11+12

Beaver Krešo watched a tournament of races and recorded the winners of each stage on the board below.
The runners wore the same numbers, from 1 to 8, throughout the tournament.
Krešo used numbered cards to represent each runner.

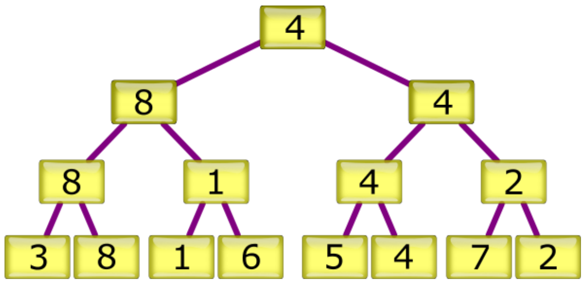
When the tournament was over his younger brother Tomo mixed up all the cards, except those from the first stage of the tournament.

Question:
Can you put the labels in the correct positions?



Beaver tournament

Answer:



Explanation:

For each race in the first round, the number of the winner should be among the mixed cards. Those that lose do not appear in the higher stages of the tournament. Hence, as we fill in the results, we simply need to look at which of the two competitors of each race exists among the remaining cards.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Interpretation and Algorithms

When solving problems and creating an algorithm, it is important to notice all the possible conditions that depend on the solution of the question. Conditions sometimes overlap, sometimes follow one another and sometimes one of them can only be executed if the previous one has or has not been executed. Careful monitoring and adherence to the necessary conditions is the basis for proper logical problem solving and the construction of an efficient algorithm solution.

In addition to checking the fulfilment of conditions (*branch structure*), *repetition* (*loops*) is commonly used when designing a computer solution. In the above tasks, the condition checking procedure must be repeated until we select one winner. Thus, the default tournament rounds also symbolise the number of repetitions required for the given action.

In this question, the condition that the student has to notice is: if the card with the number is offered, this means that the runner is a winner. The procedure is repeated until all cards are placed on the board.

Grandmother's jam

Year 3+4:
Year 5+6: **C**
Year 7+8: **B**
Year 9+10:
Year 11+12

Anna, Peter and Liza help their grandmother to make jam.

In order to make a jar of jam, they have to do three jobs:

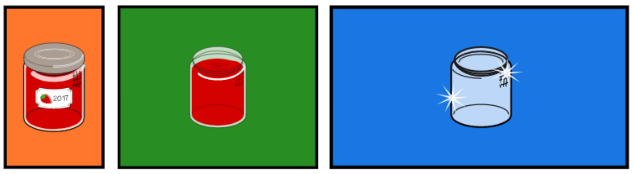
Wash a jar. This takes 3 minutes.
Put jam into a jar. This takes 2 minutes.
Close a jar. This takes 1 minute.

They have to be careful: The jar needs to be clean before they can put jam in it and they can only close the jar if it is full of jam.

Question:
Prepare a 10 minute work plan for the children so that they can prepare the maximum number of closed jars filled with jam.

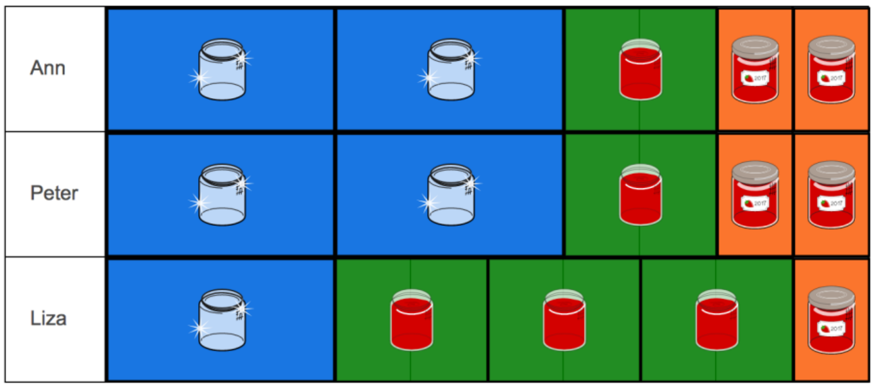
(Drag the jobs into the table.)

Ann										
Peter										
Liza										



Grandmother's jam

Answer:



Explanation:

It is important to ensure that there are always enough clean jars at each stage to fill, and enough filled jars at each age to close.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

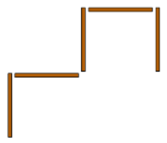
Australian Digital Technologies Curriculum Concepts: Abstraction, Specification and Algorithms

This question is equivalent to constructing a network work plan and processing several operations concurrently. The correct work arrangement is achieved with a preliminary evaluation of maximal performance limits, and an attempt to reach the optimum. A *greedy algorithm* is used for that purpose as it produces a correctly formalised task.

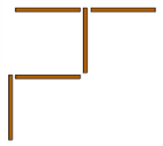
Five sticks

Year 3+4:
Year 5+6: **C**
Year 7+8:
Year 9+10:
Year 11+12

Adam has five sticks. He puts them on the table and creates this shape:



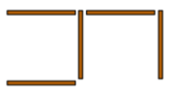
Nola comes to the table. She takes one stick and puts it in a different place:

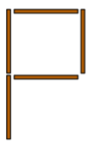


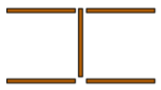
Then Bob comes to the table, he also takes one stick and puts it in a different place.

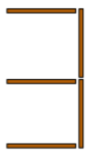
Question:

Which shape is Bob not able to make?







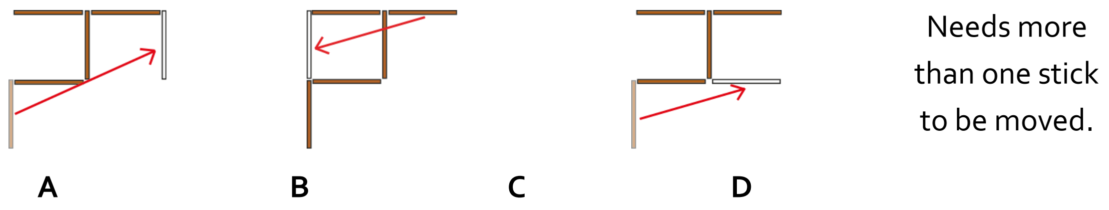


A **B** **C** **D**

Five sticks

Answer:
The answer is D.

Explanation:



It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation, Specification and Algorithms

When thinking about the shapes created from sticks we need to understand and imagine the consequence of each simple operation (move to another place). We need to be very precise to get the right answer. Just like when creating and testing computer programs.

Although the task seems to be mathematical, we must keep in mind that besides developing the spatial imagination, which is often needed to solve programming tasks, there is a need to find different possibilities for moving the stick. We need to understand how the previous steps have taken place and how the next step can be carried out.

The distance between two shapes can be defined as the number of moves or replacements you need to perform. We are looking for a shape that has a distance to the shape Nola made greater than 1.

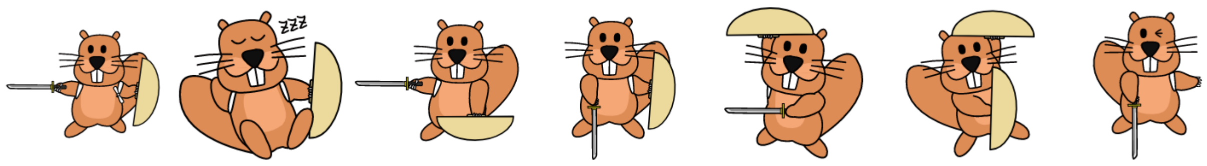
The different states you could reach could be represented as a graph. Now you are looking at points or vertices in this graph that are connected with an edge.

Sticks and shields

Year 3+4: **B**
Year 5+6: **A**
Year 7+8: **A**
Year 9+10:
Year 11+12

Lucia is playing Stick and Shields with 7 friends.

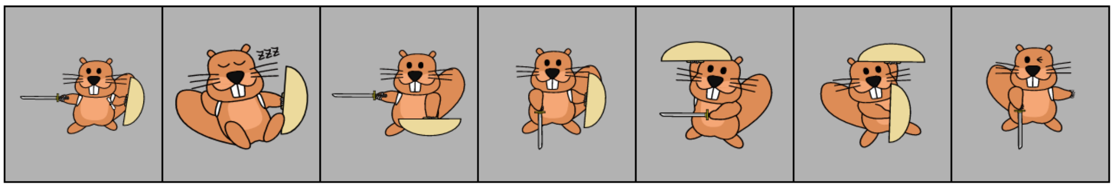
These are her friends' favourite poses:

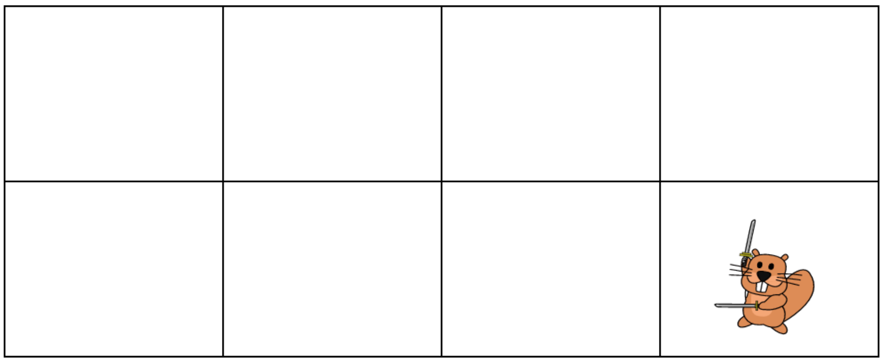


They want to have their picture taken.
In the picture, every stick should point at another beaver, and every shield should block a stick.

Lucia has already taken a spot ready for the picture.

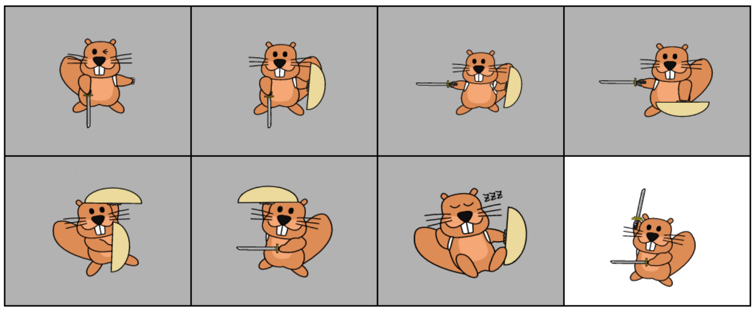
Question:
Drag the friends, shown below, into their correct positions.





Sticks and shields

Answer:



Explanation:

This is a task where the solution has to satisfy particular criteria. It is also a task where the number of possible arrangements is quite high but not many are correct. The first thing to do when solving this problem is to split the beavers into those that have to be on the top row, those that have to be on the bottom row, and those that can be anywhere. This simplifies the task somewhat, however, it is still not an easy problem.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Specification and Algorithms

This would actually be a very complicated puzzle. Just a few pictures lead to a very time-consuming search among all possible (but incorrect) solutions. If you add just one more picture to a puzzle of 6 pieces, you would have 6 times as many different possibilities of placing the 7 cards in the empty spots. For n cards, you have (n-1)! = 1 * 2 * 3 * … * (n-2) * (n-1) different possible solutions. So in this case there are 720 different possible solutions (but almost all of them are wrong).

However, using some logical thinking, the search space can be pruned by a lot. For instance, all beavers with a stick pointing down must be placed on the top row, and there is only a single beaver that can be placed right above Lucia.

A full exhaustive search can be done using an algorithm called *backtracking*. When using a backtracking algorithm the search space can get really large. This is why pruning is important.

Erase walls

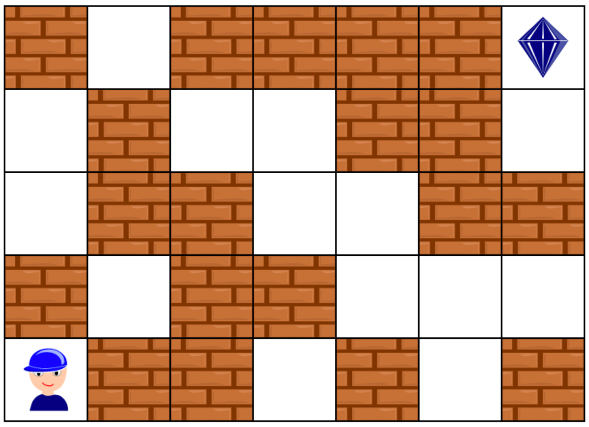
Year 3+4:
Year 5+6: **C**
Year 7+8: **B**
Year 9+10:
Year 11+12

The maze shown consists of empty squares and brick walls.

We can move from one empty square to the neighbouring empty square horizontally or vertically (not diagonally).

Walls can be removed by clicking on them. (It is also possible to rebuild them by clicking on the same square again).

Question:
Remove as few walls as possible so that it is possible to move from the bottom left corner of the maze to the top right corner.



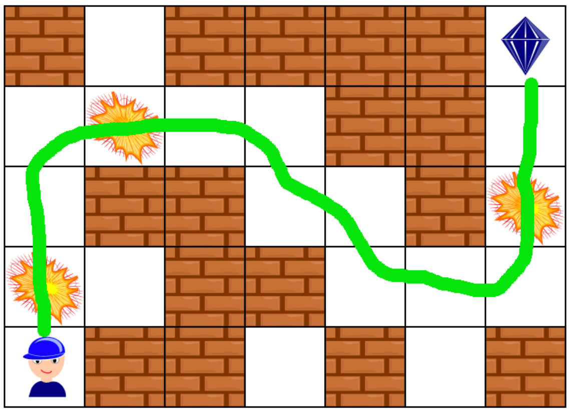
Erase walls

Answer:

The minimum number of walls that needs to be demolished is 3.

Explanation:

The image below shows the walls that need to be demolished. The green line shows the path produced.



It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL), Evaluation (EV)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation and Algorithms

Searching for a path in a maze is a known problem in Computer Science. This question uses ideas from other similar tasks, but adds an additional requirement which is to demolish as few walls as possible.

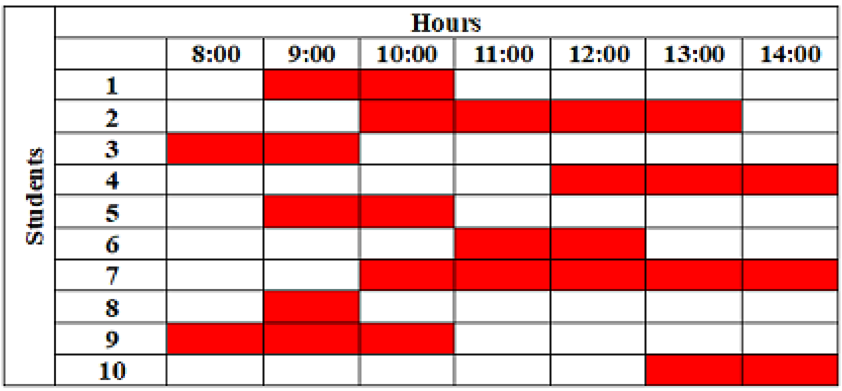
Answering this question systematically requires algorithmic thinking by scanning the cells one by one from the starting position. Marking all the cells produces an *array* that contains valuable information: the number of walls needed to be demolished to reach each cell. In Computer Science, an *array* is a data structure consisting of a collection of elements, such as values or variables.

News editing

Year 3+4:
Year 5+6: **C**
Year 7+8:
Year 9+10:
Year 11+12

There are 10 students working on the school's newspaper. Every Friday they write or edit their own articles.

The red cells, on the plan below, show when the students need a computer.
The computers are all the same.
During any one hour, only one student at a time can work on a computer.



Students	Hours						
		8:00	9:00	10:00	11:00	12:00	13:00	14:00
	1		🟥	🟥				
	2			🟥	🟥	🟥	🟥	
	3	🟥	🟥					
	4					🟥	🟥	🟥
	5		🟥	🟥				
	6				🟥	🟥		
	7			🟥	🟥	🟥	🟥	🟥
	8		🟥					
	9	🟥	🟥	🟥				
	10						🟥	🟥

Question:
What is the minimum number of computers needed for all of the students to work according to the plan shown above?

4 5 6 or 10

News editing

Answer:

The correct answer is 5.

Explanation:

At 09:00 and 10:00, 5 students need a computer – we can't solve the problem with fewer than 5 computers.

Student		8:00	9:00	10:00	11:00	12:00	13:00	14:00
	1		PC 3	PC 3				
	2			PC 1	PC 1	PC 1	PC 1	
	3	PC 1	PC 1					
	4					PC 3	PC 3	PC 3
	5		PC 4	PC 4				
	6				PC 2	PC 2		
	7			PC 5	PC 5	PC 5	PC 5	PC 5
	8		PC 5					
	9	PC 2	PC 2	PC 2				
	10						PC 2	PC 2

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation, Algorithms and Interactions

In order to understand large quantities of data and the relationship between different types of data, it is best to create a method of data representation, for example, a table, chart or diagram. Depending on the information of interest, we can represent the same data differently.

Scheduling and optimal use of resources is often a problem in real life, too. For example, in a hotel, in the room reservation system, it is very important for two reservations not to overlap and the allocation of the rooms to be optimal. Also, when arranging a school timetable and ensuring that all the teachers and students have classrooms to have lessons in, an optimal schedule can be helpful.

Another way to represent this data is to create a graph, where the nodes are intervals and there is an edge between two nodes, corresponding to intervals that intersect. This graph is called an *interval graph*.

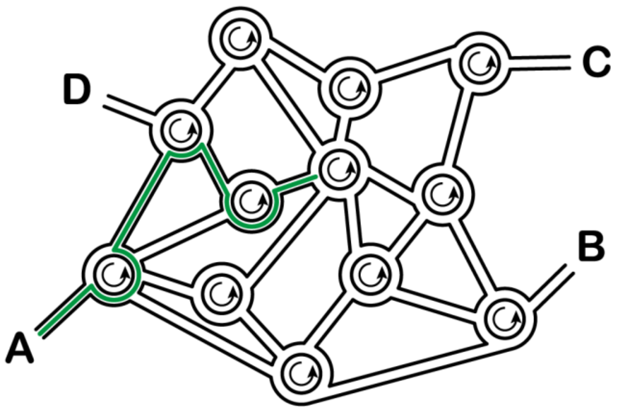
Roundabout city

Year 3+4:
Year 5+6: **C**
Year 7+8:
Year 9+10:
Year 11+12

In Roundabout City, the navigation software does not give instructions like:

- At the next roundabout, take the 4th exit.
- At the next roundabout, take the 1st exit.
- At the next roundabout, take the 2nd exit.

Instead, it gives you a sequence of numbers, like "4 1 2" which would make you go this way:



Question:
If we start from A and follow the sequence 3 1 3 2 3, where will we end up?

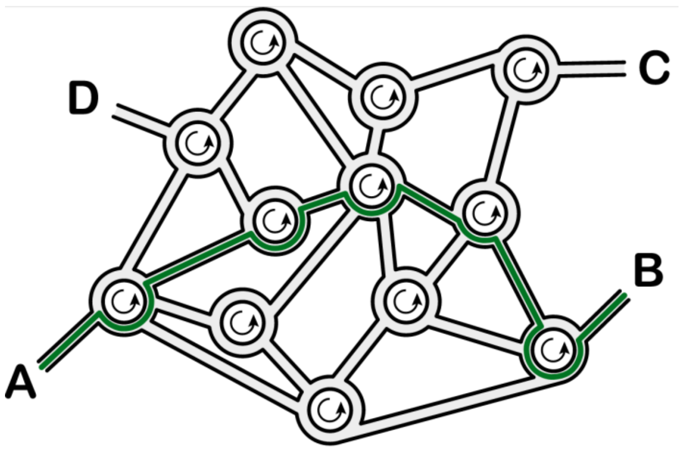
A **B** **C** **or** **D**

Roundabout city

Answer:

The exit point will be B.

Explanation:



Note that the arrows on the roundabout show a clockwise navigation. Not all countries navigate on the same side of the road to Australia. Computer programmers need to pay special attention to details. Can you imagine what would happen if our traffic signals were programmed incorrectly?

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS) Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Algorithms, Digital Systems and Interactions

This task introduces one important element of many computer programming languages: *sequential composition*, which means following instructions one after another in order.

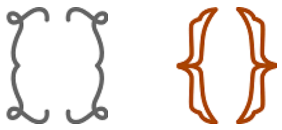
A sequence of instructions - what we gave to the car - is the basis for a computer program. In this question we have a very simple set of instructions - it means that we have a very simple programming language.

Brackets

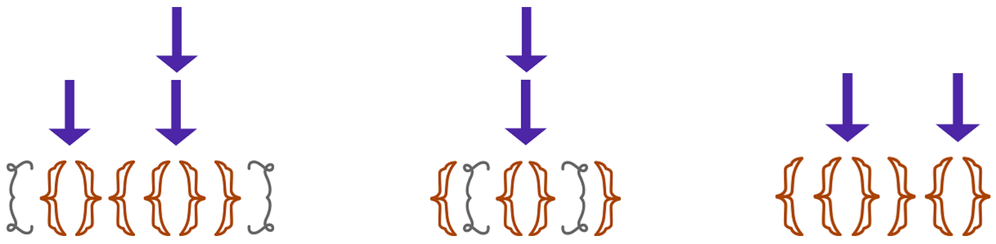
Year 3+4:
Year 5+6:
Year 7+8: **B**
Year 9+10: **A**
Year 11+12

A jewellery shop produces bracelets.

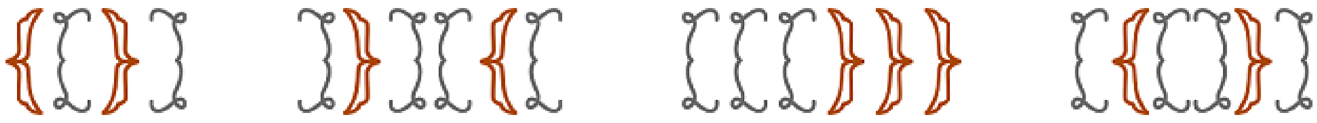
They use bracket-shaped ornaments that come in pairs. To make a bracelet you start with one of these pairs:



Additional bracket pairs are inserted repeatedly at any place in the bracelet, as you can see in the three examples below:



Question:
Which of the following bracelets is made with the method described?



A B C D

Brackets

Answer:

The correct bracelet is D.

Explanation:

D is correct because it started with two paired brackets. Between these were placed a pair of other brackets and another pair was placed between the second pair.

All other bracelets were not made according to the method described:

A. Position 3 is wrong: they placed the right side of ornament 2 before the right side of ornament 1;

B. Position 1 is wrong: you start off with a right side of ornament instead of a left side, which is not correct;

C. Position 2 is wrong: you see 3 left sides of one ornament and then 3 right sides of another ornament, so there aren't any pairs used.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation and Algorithms

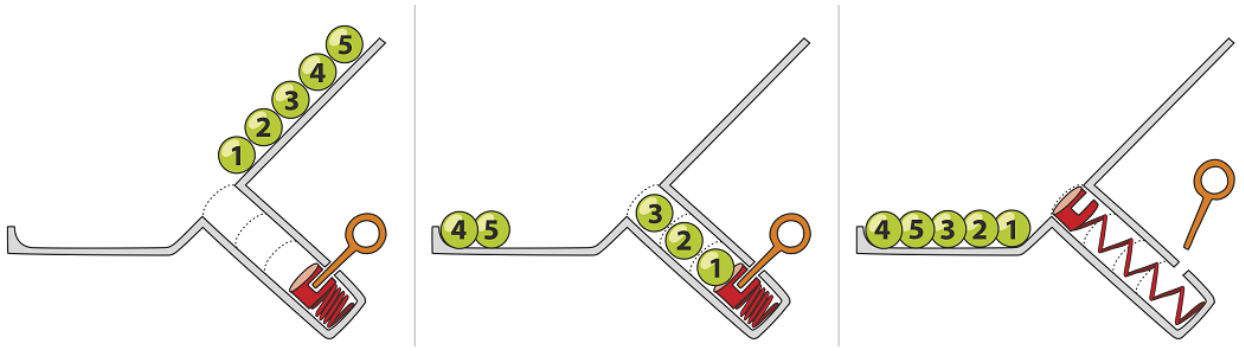
The rules of the bracelets are exactly like the rules for bracket expressions. Computer Scientists call correct expressions *well-formed*. Expressions containing errors are called *malformed*. An expression that is well-formed can also be called *syntactically correct*, meaning that it obeys the required syntax. Syntax errors can be hard to find in complex expressions but syntax errors are generally much easier to find than *semantic errors*, which are logical mistakes made by the programmer.

Balls

Year 3+4:	Year 9+10:
Year 5+6:	Year 11+12
Year 7+8: **B**	

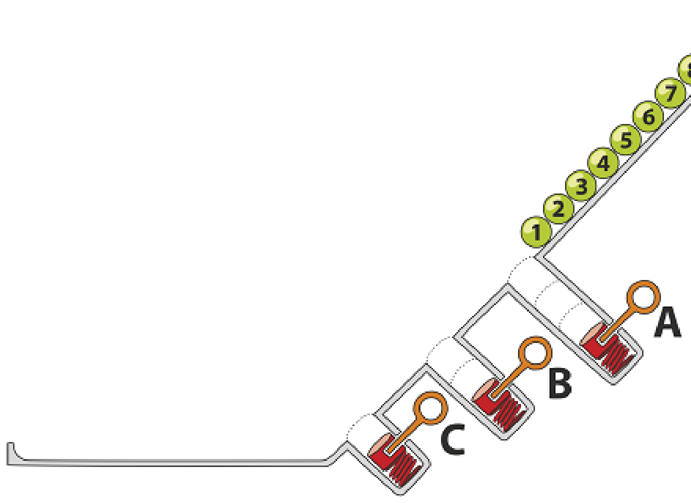
Numbered balls roll down ramps. The order of the balls changes as they fall into holes. When a ball comes to a hole, if there is enough space, the ball falls in, otherwise, the ball rolls past the hole. A pin at the bottom of each hole can be pulled which ejects the balls.

Here is an example:

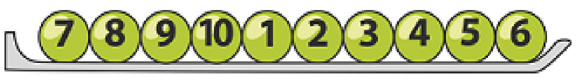


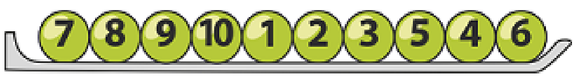
before balls start rolling   after balls have stopped rolling   final result after pulling the pin

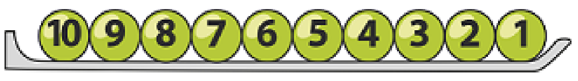
Ten balls roll down the ramp shown below.
Three holes A, B and C have space for 3, 2 and 1 balls as shown.
The pins are pulled in the order A, B, C but, each time, only after all the balls have stopped rolling.

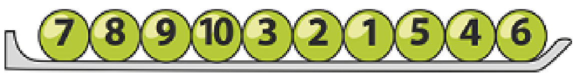


Question:
Which of the following is the final result?



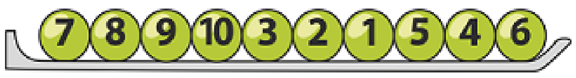






Balls

Answer:



Explanation:

Hole A has space for three balls, so balls 4 to 10 roll past it in order. Hole B has space for two balls, so balls 6 to 10 roll past it in order. Hole C has space for one ball, so balls 7 to 10 roll past it in order. Then the pin in Hole A is pulled and balls are ejected in the order 3,2,1 and roll to the bottom. At this point, the balls are at the bottom in the order 7,8,9,10,3,2,1.

Then the pin in Hole B is pulled and balls are ejected in the order 5,4. At this point, the balls are at the bottom in the order 7,8,9,10,3,2,1,5,4.

Finally, the pin is pulled in Hole C and ball 6 rolls to the bottom giving the correct answer.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation, Specification and Algorithms

The holes act like a stack which is a *data structure*. That is, it is a way of organising data. It is based on the *last-in first-out principle* (LIFO). For example, the first ball to fall into a hole, is the last ball ejected from the hole. Despite being a very simple idea, it is useful in many different situations.

For example, you might want to research how a stack can be used to check that the brackets in ((1+2)*3) are balanced but the brackets in ((4+5)*(6-7) are not. The idea is that all opening brackets are put on a stack (this operation is called a *push*) and when its matching closing bracket is found, the opening bracket is removed from the top of the stack (this operation is called a *pop*). Instead of inventing a complicated algorithm or using a more sophisticated way to organise the data, it turns out that we only need this *last-in first-out* principle.

Cipher wheel

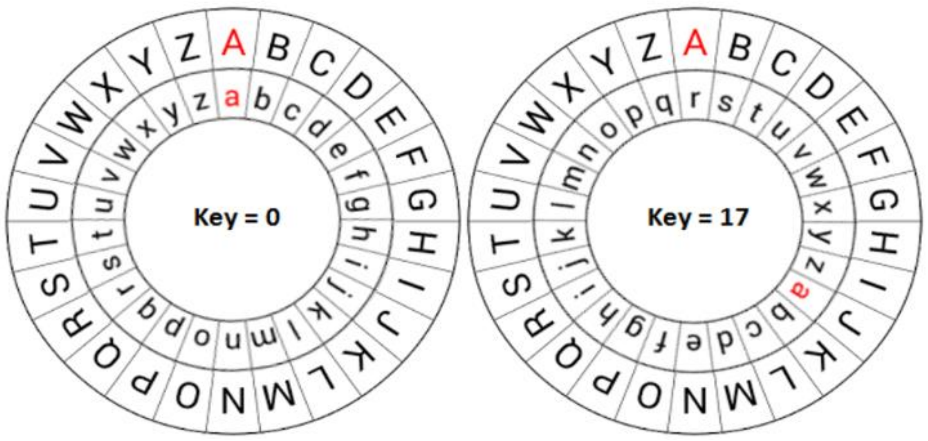
Year 3+4:
Year 5+6:
Year 7+8: **C**
Year 9+10: **B**
Year 11+12

A beaver left a secret message on his tombstone by using a Cipher Wheel and we want to figure out what it means.

The wheel works such that only the inner wheel (with small letters) can be rotated. The outer wheel is for the actual message.

As you can see in the first image, when the key is 0 ‘A’ is encoded as ‘a'.

The second image shows that when the key is 17 (because the inner wheel has been rotated by 17 positions counter-clockwise) 'A' is encoded as 'r'.



With the key equal to 17, we can encode the message WHO ARE YOU as nyf riv pfl

The message j cp fjgcma is received. We know that this was encrypted in a clever way:
For the first letter the key was 1, for the second letter the key was 2, the key for the third letter was 3, etc.

Question:
Decipher the encrypted message and enter the original message as your answer.

Cipher wheel

Answer:

The answer is I AM BEAVER.

Explanation:

You need to pay attention to the change of the key for every letter. The key actually means how many steps you need to go through the alphabet for encoding. Also note that by deciphering you need to go the other way so for *j* you get *I* because you take 1 step backwards.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL), Evaluation (EV)

Australian Digital Technologies Curriculum Concepts: Abstraction, Specification, Algorithms and Digital Systems

The Cipher Wheel is similar to the cipher disk which is an encrypting and decrypting tool developed in 1470 by the Italian architect and author Leon Battista Alberti. He constructed a device, called the Alberti cipher disk, consisting of two concentric circular plates mounted one on top of the other. The larger plate is called the *stationary* and the smaller one the *moveable* since the smaller one could move on top of the *stationary*.

Nowadays, cryptography (secure communication processes) is usually done digitally using computers, and can become very complex. An encrypting and decrypting tool, such as the cipher wheel, is an example how effective cryptography can also be done mechanically.

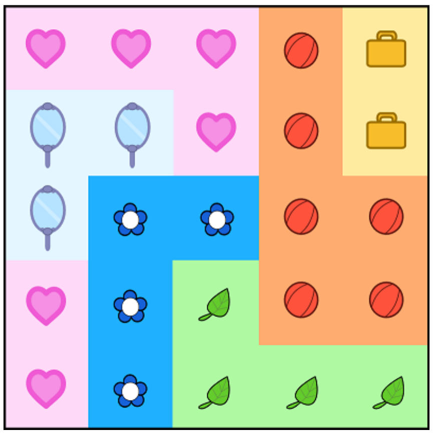
Painting wallpaper

Year 3+4: **C**
Year 5+6: **B**
Year 7+8: **A**
Year 9+10:
Year 11+12

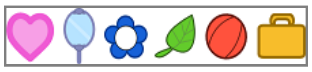
Robyn is wallpapering.

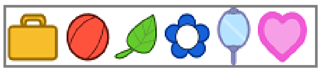
She uses rectangular wallpaper pieces of different sizes.
Each wallpaper piece has only one colour with one pattern on it.
She never puts wallpaper beyond the edge of the wall.

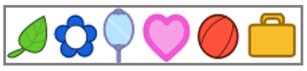
Sometimes, Robyn covers part of one piece of wallpaper with a new rectangular piece of a different colour.

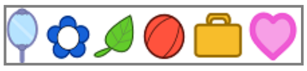


Question:
In which order has Robyn placed the wallpaper?

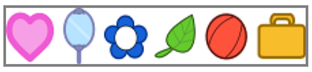








Answer:



Explanation:

The yellow wallpaper with the briefcases is the only wallpaper that isn’t cut off by another one, so that should be the last one. You can see the suitcase cuts off the basketball wallpaper, the basketball wallpaper cuts off the leaf wallpaper, the leaf wallpaper cuts off the flower wallpaper, the flower wallpaper cuts off the mirrors and the mirrors cut off the hearts.

It’s Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Collection and Algorithms

Using the same wallpaper rectangles in a different order would have given a totally different image: sequences are very important in informatics. The order in which you do things, and thus the order in which you command your computer to do things, give very specific outcomes.

This is also an example of layers: digital image editing uses layering a lot to alter parts of an image.

Railroad

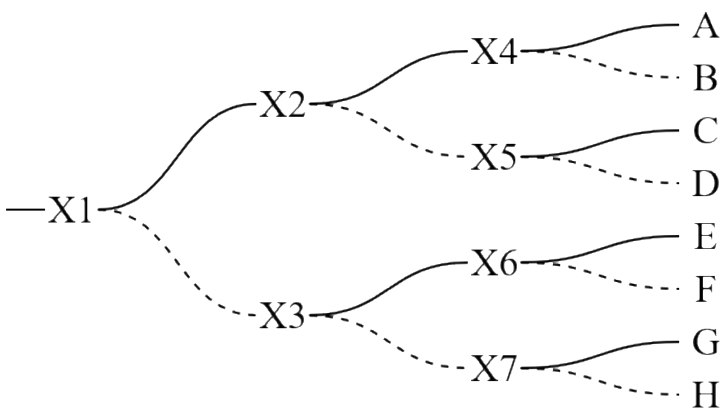
Year 3+4:	Year 9+10: **A**
Year 5+6:	Year 11+12
Year 7+8:	

8 trains (named a to h) enter the switch X1 from the left on the figure below.

Train a needs to go to station A, train b to station B, train c to station C, etc.

Each of the switches X1 to X7 are initially set to direct trains to the left.
After a train has passed a switch, the switch reverts to the other direction.

The Railroad Director needs to ensure that all the trains go to their correct stations.



Question:
What is the correct order for the trains to pass through switch X1?

aecgbfdh **adcgbfeh** **agcdbfeh** **or** **acedfghb**

Railroad

Answer:

The correct order for the trains to pass switch X1 is aecgbfdh.

Explanation:

For each switch, let the status 0 mean a turn to the left and status 1 mean turn to the right.

The switch X1 changes its status after each train passing by – as is shown in the second column in the table below. Based on the status of X1, exactly one of the switches X2 and X3 changes its status, as shown in the third column (X2 for odd-numbered and X3 for even-numbered trains). And finally exactly one of the switches X4, X5, X6, and X7 changes.

In each row of the table we get a three-digit binary number. The numbers correspond to the destinations of the trains (last column of the table).

Train order	Passing X1	Passing X2 or X3	Passing X4 or X5 or X6 or X7	Decimal number	Station letter
First	0	0	0	**0**	**A**
Second	1	0	0	**4**	**E**
Third	0	1	0	**2**	**C**
Fourth	1	1	0	**6**	**G**
Fifth	0	0	1	**1**	**B**
Sixth	1	0	1	**5**	**F**
Seventh	0	1	1	**3**	**D**
Eighth	1	1	1	**7**	**H**

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS), Algorithms (AL)

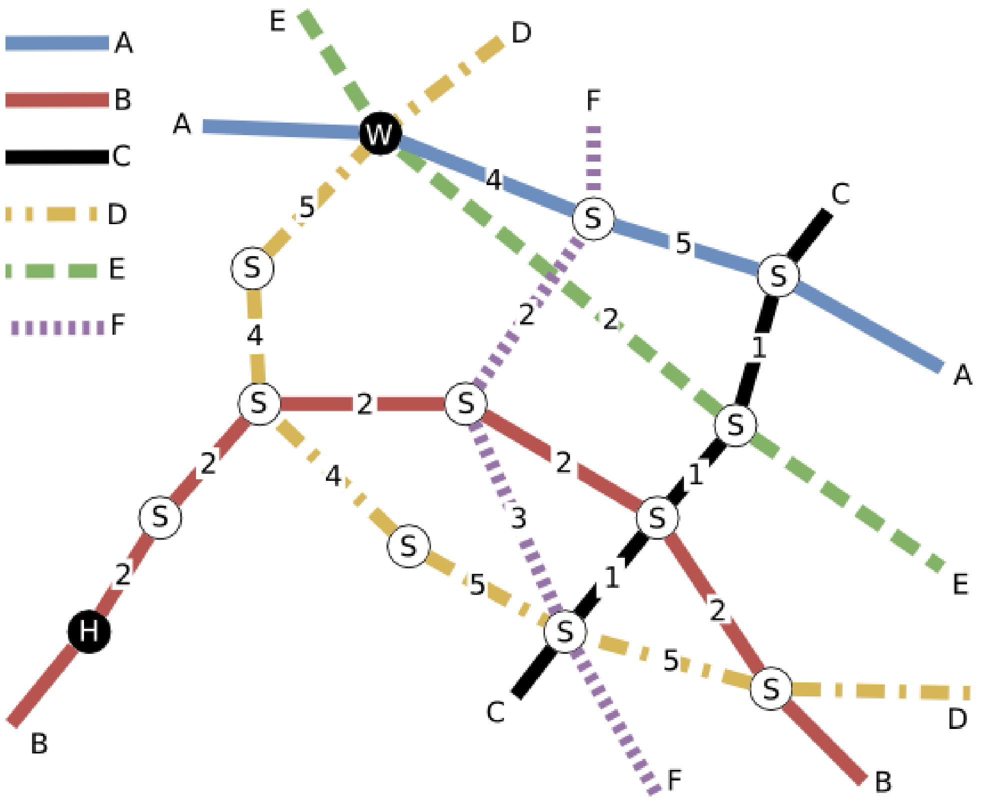
Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation, Algorithms, Digital Systems and Impacts

Using the binary system makes it possible to easily solve a rather complicated problem.

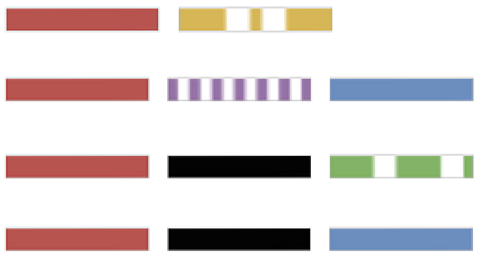
Commuting

Year 3+4: Year 9+10:
Year 5+6: Year 11+12: **A**
Year 7+8:

Beaver Martina goes to work by train every day. There is no direct line, so Martina has to switch between several lines. The map below shows the available lines, with the travel time between any two stations (Martina's home is marked with "H", her workplace is marked with "W", and the stations, where it is possible to change line, are marked with "S").

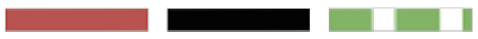


Question:
Assuming that changing line takes no time, which lines should Martina take in order to arrive at work as fast as possible?



Commuting

Answer:



Explanation:

Distances for each route are, respectively: 13, 12, 11 and 19.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation and Algorithms

Graphs are a very important data structure in Computer Science as they can be used to represent networks of several different types (social, preferences, transportation...).

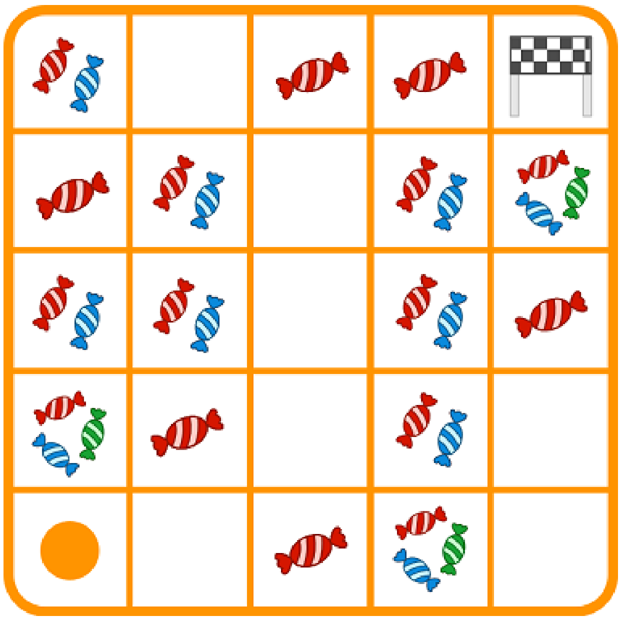
In this question, the goal is to find the shortest path between two points in a graph, given weights on the *edges*. This problem is known as the shortest path problem and can be solved using the *Dijkstra algorithm*.

Candy maze

Year 3+4:
Year 5+6:
Year 7+8: **C**
Year 9+10:
Year 11+12: **A**

A robot is programmed to collect as many sweets as possible. It does this while walking through cells. Each cell in the grid on the right has either 0, 1, 2 or 3 sweets.

The robot begins in the bottom-left and ends in the top-right. The robot can only move to the right or upwards.



Question:

How many sweets will the robot collect in this grid?

10 **12** **14** **or** **16**

Candy maze

Answer:

The answer is 14.

Explanation:

One approach is to fill in a table of the "best" possible amounts of sweets that can be gathered by filling in a "diagonal sweep" of the table. Initially, we have 0 sweets, so we can think of the table as shown on the right, where the bold element is the maximum number of sweets we can possibly attain in each cell.

2	0	1	1	F
1	2	0	2	3
2	2	0	2	1
3	1	0	2	0
0	0	1	3	0

Going up will gain 3 sweets, and going right will gain 0 sweets, so we can update our totals

2	0	1	1	F
1	2	0	2	3
2	2	0	2	1
3	1	0	2	0
0	**0**	1	3	0

Notice the cell which is to the right of the bold 3 and above the bold 0. What is the maximum amount of sweets we could have gathered to get to this cell? We should get to this cell after having gathered 3, rather than 0, sweets. Thus, we could be in this cell after gathering a total of 4 sweets.

2	0	1	1	F
1	2	0	2	3
2	2	0	2	1
3	**4**	0	2	0
0	**0**	1	3	0

Continuing in this way, we can see that the maximum number of sweets we can gather in a cell is the number of sweets in this cell plus the larger of the "left" maximum and "below" maximum.

Because you always want to look to the left and to the bottom, you need to add a column of zeros to the left, and a row of zeros to the bottom of the table. Applying this relationship, we can fill in the rest of the table as shown.

0	**8**	**9**	**10**	**12**	**14**
0	**6**	**9**	**9**	**11**	**14**
0	**5**	**7**	**7**	**9**	**10**
0	**3**	**4**	**4**	**6**	**6**
0	**0**	**0**	**1**	**4**	**4**
0	*0*	*0*	*0*	*0*	*0*

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Collection, Specification and Algorithms

Determining the "best" solution among a set of possible solutions is a difficult problem. For this particular sweet gathering problem, we could try all possible paths, which is the brute force solution. Unfortunately, there are 70 possible different paths.

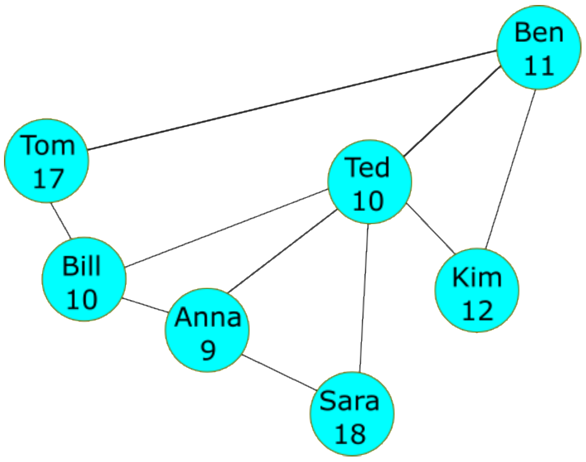
In this special case however, you can try to find some worthwhile parts of the task and find the best solution from there. Since the grid is relatively small, you can deduce that all other possibilities must be worse.

A more efficient solution involves filling in the table, as we did, using a technique called memorisation of the dynamic programming recurrence. That is, once we have derived a formula/relationship for the "best" solution of a current cell, based on the cell to the left or the cell below, we can perform 25 calculations, in this case, to compute the maximum number of sweets available, building up the larger solution from the initial conditions.

Book-sharing club

Year 3+4:
Year 5+6:
Year 7+8:
Year 9+10: **A**
Year 11+12:

This diagram shows the relationship between 7 students in a book-sharing club. Their names and ages are also shown.



The club has some regulations for members:

When you receive a book you read it (if you haven't already done so) and then pass it to the youngest friend who has not read it yet. If, however, all your friends have read the book then you should pass it to the friend who first gave it to you.

Now Ben has read a new book and wants to share it with his friends.

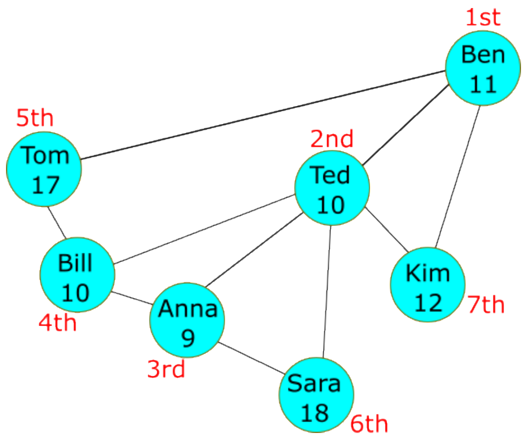
Question:
Who will read the book last?

Tom **Sara** **Bill** **or** **Kim**

Book-sharing club

Answer:
The correct answer is Kim.

Explanation:
Ben gives to Ted. Ted gives to Anna. Anna gives to Bill. Bill gives to Tom. Tom gives back to Bill. Bill gives back to Anna. Anna gives to Sara. Sara gives back to Anna. Anna gives back to Ted. Ted gives the book to Kim who is the last person to read the book.



It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS) Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation, Algorithms, Digital Systems and Interactions

A lot of the data Computer Scientists work with is interconnected. We can think about our favourite social network as a set of elements (people) and connections (friendship relationships). We can think about a road grid plan as a set of elements (cities) and connections (roads). We can think about the food chain as a set of elements (living beings) and connections (predator-prey relationships). So, a lot of systems from the real world can be thought of as a network.

To represent networks, Computer Scientists use graphs as data structures, which are simply sets of connections. To analyse these graphs, there are multiple algorithms that evaluate different properties. One well-known algorithm is called the *Depth-First Search*, which is an algorithm for traversing or searching a graph data structure in a specific order. This is very useful to compute several properties of the graph. For example, how many connected components does the graph have (a connected component is a set of elements of the graph that can all be reached from one another by following the graphs' connections)? If we perform a *depth-first search* from one element we can find out which other elements are reachable from this node, so we can do the same with all other elements.

In this problem we need to track a book through a network of friends. In fact, the path the book takes through the network of friends is very close to how the order of a depth-first search would look in the graph of friends.

One too many

Year 3+4:
Year 5+6:
Year 7+8:
Year 9+10: **B**
Year 11+12: **A**

I recently discovered a silly mistake I made in a long piece of alphanumeric text.

All 1's should be 11's and all 11's should be 1's. Luckily, I can be rather smart and I have an editor which enables me to replace a sequence of characters with another sequence.

See what happops in a soptopce in which I replace all occuropces (except the last one) of en with op! Of, wofse, feplacing all occuffences of r (except the last one) with f.

Question:
What should I do to fix my text?

Replace all 11's with 1's and then all 1's with 11's

Replace all 1's with 11's and then all 11's with 1's

Replace all 1's with $s and then all $s with 11's and then all 11's with 1's

Replace all 11's with $s and then all 1's with 11's and then all $s with 1's

Answer:
Replace all 11's with $s and then all 1's with 11's and then all $s with 1's

Explanation:

In A, after we replace 11's with 1's, we can no longer distinguish between the original 1's which now have to be replaced with 11's, and those 1's that we just made from 11's. In the end, we would have just 11's.
In B, we don't only change 1's into 11's, but also 11's into 1111's and then back, so we end up where we started.
C will let us end up with only 1s.
D works.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS), Algorithms (AL), Evaluation (EV)

Australian Digital Technologies Curriculum Concepts: Abstraction, Algorithms and Implementation

Computational thinkers will always look for ways of automating things even in word processors.

It is not uncommon for developers to have, as input, a set of data that is in the wrong format for their needs. Before being able to use the data it is necessary to perform a process like this one.

Sports

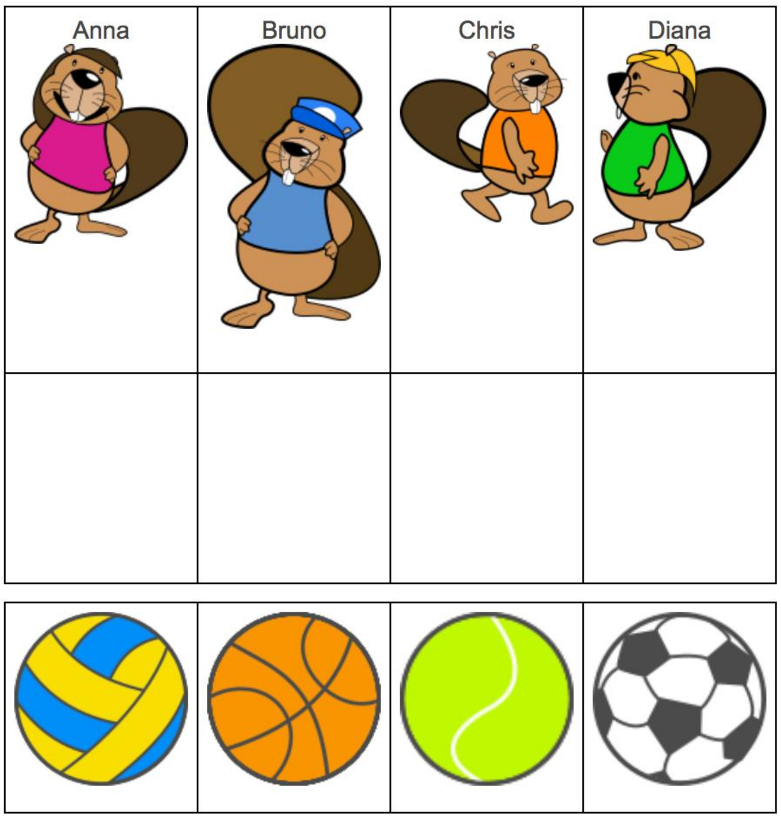
Year 3+4:
Year 5+6:
Year 7+8: **B**
Year 9+10: **A**
Year 11+12:

The *BebraFitness* sports centre has a volleyball court, a basketball court, a tennis court and a football field.

Four beaver friends, Anna, Bruno, Chris and Diana, occasionally come to *BebraFitness* to play their favourite games.

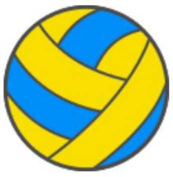
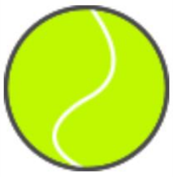
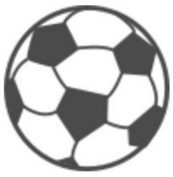
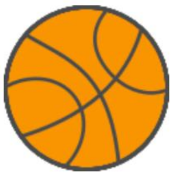
It is known that Anna and Chris do not use rackets. The volleyballer, the footballer and Diana have their trainings on the same day. The footballer plans to watch Chris's match. In the mornings, Bruno and the footballer go for a run. Diana lives in the same den with the tennis player.

Question:
Match the players with their favourite sports by dragging the correct balls into the spaces below them.



Sports

Answer:

Anna	Bruno	Chris	Diana
			

Explanation:

The question can be solved by marking exclusion in the matrix below while reading each sentence.

	Volleyball	Basketball	Tennis	Football
Anna			no	
Bruno				no
Chris			no	no
Diana	no		no	no

Now it's clear that Diana can play only basketball (and no other sport). Football must be played by Anna and tennis by Bruno (no other player). The leftovers are volleyball and Chris.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

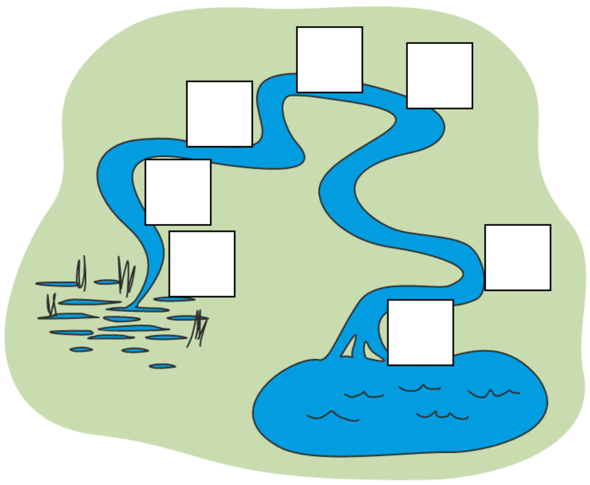
Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation, Algorithms and Digital Systems

The question is an example of finding a solution using *Boolean expressions* ('true' and 'false' values). Using binary encoding of information and processing it with Boolean operations are the fundamental principles of Computer Science.

Stream

Year 3+4: **B**
Year 5+6: **A**
Year 7+8:
Year 9+10:
Year 11+12:

A river flows from a swamp to a lake. Half way down the river is a pier with a cafe. Visitors at the cafe enjoy a view of branches on the river next to the pier. At the café, visitors can also book a steamboat trip to the carousel which is located by the mouth of the river. On the return trip from the lake to the pier the steamboat will stop at a barbecue near the carousel for a lunch of smoked fish. As the steamboat returns to the pier, to the left, visitors will enjoy a view of the hundred year old oak tree which is in the swamp and a hill which is located between the pier and a windmill.



Question:

Drag and drop the objects below to their correct places along the river bank above.

oak tree	barbecue	windmill	carousel	branches	pier	hill

Stream

Answer:

The correct order by following the stream from left to right is:
oak tree, windmill, hill, pier, branches, barbecue, carousel.

Explanation:

1. We know that in the middle of the tour (4th position by the stream) is the pier.
2. We read in the text that the oak tree is in the swamp – 1st position looking by the stream.
3. In the text it is said that the carousel is near the mouth of the river – in the 7th position.
4. The barbecue is near the carousel. It is in the 6th position.
5. On the trip back to the pier is seen the oak tree, the hill which is between the windmill and the pier. Therefore, the windmill is in the 2nd position and the 3rd position is left for the hill.
6. The only position is left is 5th for branches which is next to pier.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Collection, Specification and Algorithms

Often a team of problem solvers need to get information from a user's chaotic description. The programmers need to change a vaguely described sequence of actions into an exact and logically ordered sequence in order to develop a computer program. The users often describe their ideas of what the program should do by lengthy sentences and descriptions and often do not use a clear order in which things must happen. Therefore, the programmers must be ready to transform such descriptions into a more exact form.

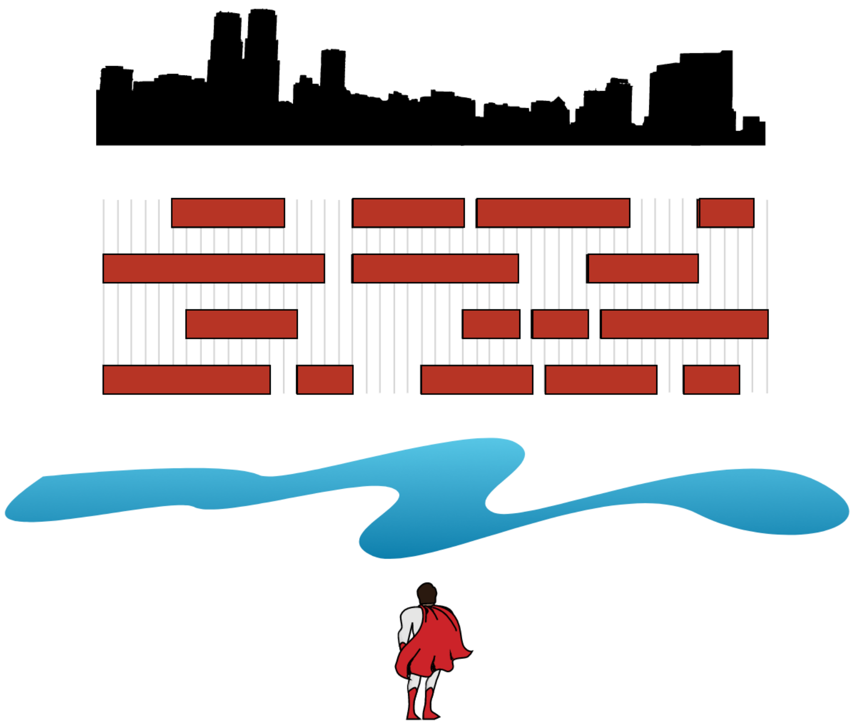
Super hero

Year 3+4:
Year 5+6:
Year 7+8:
Year 9+10: **B**
Year 11+12: **B**

A super hero watches over Beaver City from a straight path across the river.
From every point along the path, the super hero needs to be able to see the point in the city directly across the river.

Unfortunately, 16 walls of varying lengths stand between the river and the city.
Fortunately, the super hero has X-ray vision and can see through a wall.
Unfortunately, the super hero can only see through one wall at a time.
Fortunately, the super hero is strong enough to destroy walls.
Unfortunately, destroying a wall makes the super hero very tired.

Question:
Click on the walls below to remove the least number of walls that the super hero needs to destroy.



Super hero

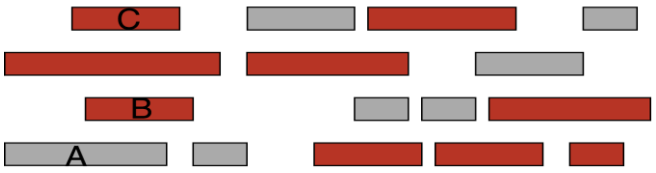
Answer:

The answer is 9.

Explanation:

Instead of determining the fewest number of walls that the super hero needs to destroy, we can consider the largest number of walls that the super hero can keep (not destroy). This is equivalent. We will show that the largest number of walls that the super hero can keep is seven. Since there are a total of 16 walls, this means the super hero needs to destroy 16−7=9 walls.

Consider the seven grey walls in the picture below.



No two of these seven walls "overlap". That is, from every point along the path, at most one of these walls will be between the super hero and the point in the city directly across the river. Thus we know that the super hero can keep at least seven walls. We must show that the super hero cannot keep more than seven walls. There are walls of four lengths. Let's name these lengths 1, 2, 3 and 4 (which happen to be proportional to the actual lengths in the picture).

Interestingly, instead of the leftmost wall "A" of length 3 in our solution, the super hero could keep either of the two leftmost walls of length 2, "B" or "C". So there is an alternative set of seven walls of lengths 2, 1, 2, 1, 1, 2 and 1 that the super hero could keep. The super hero cannot do better than this because there are only five walls of length 1 and two of them "overlap".

Why did we choose a wall of length 3 in our solution? We did this because it comes from a way to solve this problem for any set and placement of walls, not just those in this particular task. More generally, we can find the walls to keep by starting from the left and always selecting the next (non-overlapping) wall that ends closest to the left. (Can you see why this always works?)

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

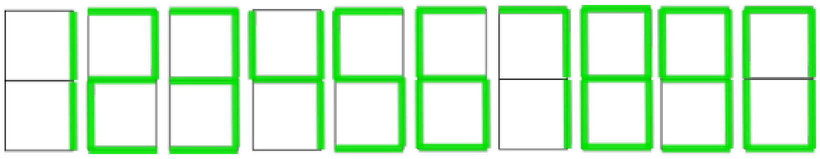
Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation, Algorithms and Interactions

The general solution to this problem is a *greedy algorithm*. That is, it is a sequence that solves a problem by always making a choice at each step that seems best "locally" or "at the moment". It does not take into account the "big picture" or full set of information. This particular problem is a model of a scheduling problem which computer scientists are interested in solving with efficient algorithms. To see this, think of time moving forward as you move towards the right and think of the walls as tasks that need to be completed with a fixed start and end time. The restriction that the super hero can only see through one wall corresponds to the fact that only one task can be completed at any one time. In our problem, the super hero is trying to solve as many tasks as possible.

Digit recognition

Year 3+4:
Year 5+6:
Year 7+8: **C**
Year 9+10:
Year 11+12: **B**

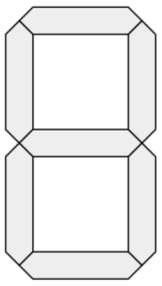
A digit recognition system understands digits that look like these:



Each digit is made up of up to 7 segments.

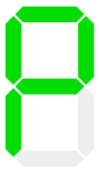
Not all segments are necessary to recognise a digit. It is possible to understand a digit if only some of the segments are visible.

Question:
Select those segments that are absolutely necessary to identify all of the ten digits (0...9) unambiguously.

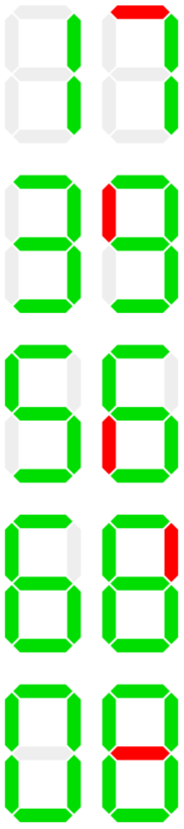


Digit recognition

Answer:



Explanation:



To understand whether 5 segments are enough, let us carry out the following reasoning. Since 0 and 8 differ only in one segment, this segment must be used in recognition, and we must select it.

All the digits may be split into two sets depending on whether this segment is present in a digit: for {0;1;7} it is not present, and for {2;3;4;5;6;8;9} it is present. Next note than 1 and 7 also differ in only one segment, thus we also must select it.

Now, all the digits are split into 4 sets depending on whether they contain or not contain the two selected segments: {1}, {0;7}, {4}, {2;3;5;6;8;9}. So, the two selected segments already allow us to recognise digits 1 and 4. Now, let us find more digits that differ in exactly one element. These are 8 and 9. Let us select this segment also.

Digits are split in the following sets with identical elements: {1}, {0}, {7}, {4}, {2;6;8}, {3;5;9}. Similarly, 6 and 8 also differ in only one element, let us select it.

After that, the four segments divide digits into the following subsets: {1}, {0}, {7}, {4}, {2;8}, {6}, {3;9}, {5}. Thus, four segments are not enough. 3 and 9 differ only in one segment. This segment also allows us to distinguish between 2 and 8, so we may select this element, and all the digits will be separated.

If we now overlay the segments that we have identified as being unique, we get our solution.

It's Computational Thinking:

<u>Computational Thinking Skills:</u> Decomposition (DE), Abstraction (AB)

<u>Australian Digital Technologies Curriculum Concepts:</u> Abstraction, Data Representation and Digital Systems

Pattern recognition algorithms generally aim to provide a reasonable answer for all possible inputs and to perform "most likely" matching of the inputs while taking into account their statistical variation. Pattern recognition is a branch of machine learning that focuses on the recognition of patterns and regularities in data. One approach to recognition is to extract specific features that allow uniquely identifiable objects.

Computer vision is an actively developing field of information technologies.

Arabot's walk

Year 3+4:
Year 5+6:
Year 7+8:
Year 9+10: **B**
Year 11+12: **A**

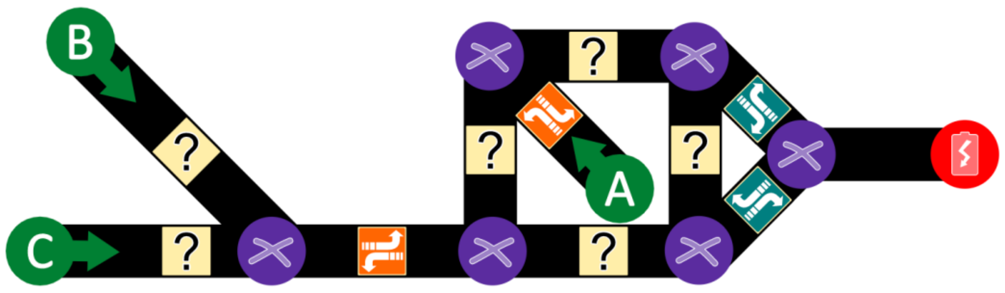
An Arabot is a robot that can follow black lines drawn on a piece of paper.

On every line there is a label, which tells it to turn left ( ) or right ( ) at the next junction ( ).
Some labels are already chosen, but there are still some labels left to choose.

Jane wants to be able to start her Arabot at A, B or C.
She also wants her Arabot to always end up at the charging station ( ).

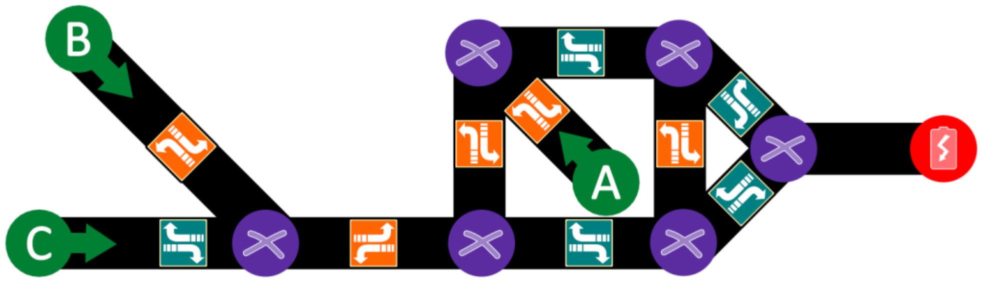
If the Arabot ends up at A, B or C, it doesn't know how to continue and turns itself off.

Question:
Help Jane finish labelling the lines so that the Arabot always ends up at the charging station ( ).
(Labels can be placed or removed by repeatedly clicking on the question marks.)



Arabot's walk

Answer:



Explanation:

For the two leftmost lines you have to make sure that the Arabot continues to the right, so for the top one you have to choose and for the bottom one you have to choose .

In the right triangle the only way to get close to is by taking the bottom line to the right, because the top line will take you back around and in the end to place B. So in the middle rectangle you want to choose for the bottom line and also for the top line. For the left line of that rectangle you need to make sure that you don't erroneously go to C, so you have to choose .

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Algorithms, Digital Systems and Implementation

The activity is about encoding several different path-tasks ("A" to , "B" to and "C" to) into a single labeled structure (a graph in this case). In Computer Science this would be called a *data structure.* In traversing a path (for example, from "A" to) each step Arabot takes along the way involves reading and interpreting an instruction: which way to turn at the next intersection, with each turn leading to a new instruction. This is similar to the way computer hardware functions at a basic level in its electronics.

This puzzle raises a number of interesting mathematical questions about how difficult it is to work out such requested labelings. Some of these questions are unresolved and are of current interest to computer science researchers in the area of algorithms and computational complexity. Related puzzles have applications in computational biology and computational medicine.

Levenshtein distance

Year 3+4:	Year 9+10: **C**
Year 5+6:	Year 11+12: **B**
Year 7+8:	

We define a basic operation as one of the following:

- insert one character into a word,
- remove one character from a word,
- replace one character with another.

We define the distance between two words as the minimum number of basic operations which allows us to change the first word into the second.

For example, the distance between kitten and sitting is equal to 3. The basic operations necessary are:

1. kitten → sitten (change k to s),
2. sitten → sittin (change i to e),
3. sittin → sitting (insert g at the end).

Question:
What is the minimum distance between length and french?

Levenshtein distance

Answer:

The correct answer is 4.

Explanation:

There is one possible solution:

1. length → fength (change l to f)
2. fength → frength (insert r)
3. frength → frencth (change g to c)
4. frencth → french (remove t)

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL), Evaluation (EV)

Australian Digital Technologies Curriculum Concepts: Abstraction, Algorithms and Digital Systems

The *Levenshtein distance* is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other. It is named after Vladimir Levenshtein, who studied this problem in 1965.

Levenshtein distance may also be referred to as *edit distance*.

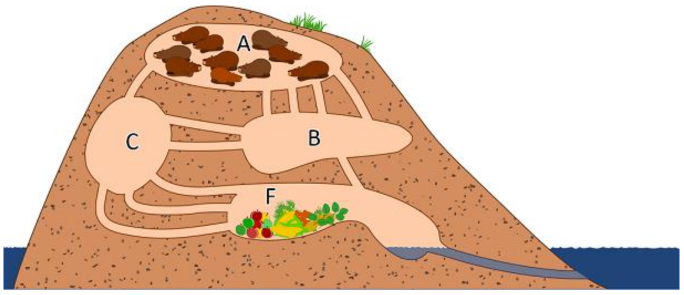
The Levenshtein distance can also be computed between two longer strings, but the cost to compute it, which is roughly proportional to the product of the two string lengths, makes this impractical.

Tunnels of the homestead dam

Year 3+4:	Year 9+10: **C**
Year 5+6:	Year 11+12: **B**
Year 7+8:	

Homestead Dam has tunnels that connect four rooms: A, B, C, and F.

A, B, and C are living rooms.
F is where food is stored. (see image)



10 beavers are staying in room A, they are becoming hungry and they want to go to room F to eat. Since all the beavers are very hungry, they all want to arrive in the food storage as soon as possible.

It takes 1 minute to get through a tunnel and only one beaver may do this at the same time (not several beavers following each other).

The connections between the rooms consist of a certain number of tunnels:

- Between A and B: 4 tunnels.
- Between A and C: 1 tunnel.
- Between B and C: 2 tunnels.
- Between B and F: 1 tunnel.
- Between C and F: 3 tunnels.

All the rooms can fit as many of the beavers as want to be there.

Question:
In the best case, after how many minutes can all the beavers be in the food storage?

Tunnels of the homestead dam

Answer:
In the best case, all beavers are in the food storage after *4 minutes*.

Explanation:
The graph has two shortest routes. Both have the capacity of 1 beaver and both require 2 minutes total travel time:

- A to B to F
- A to C to F

There is a route that has higher capacity (2 beavers) but requires 3 minutes total travel time:

- A to B to C to F

To optimise overall time, the connection A to B must be used at reduced capacity. The optimal solution sends 3 beavers at the first minute (which is very counter intuitive) and 3 at the second minute.

Given that only 3 tunnels exit room B, if the connection A to B is used at full capacity (4 beavers), connections B to C and B to F become a bottleneck and one beaver stays stuck in room B.

The following table explains the actions taken minute after minute, until all beavers are in the food storage. Although only one optimal solution for this task exists (4 minutes), there are several ways to achieve it. Here we consider a way in which the beavers don't have to wait in room B:

Action / Situation	Number of Beavers in Rooms (after the action)			
	A	B	C	F
Situation at Start	**10**	**0**	**0**	**0**
3 Beavers go from A to B (reduced capacity)				
1 Beaver goes from A to C				
Situaton after 1 Minute	**6**	**3**	**1**	**0**
3 Beavers go from A to B (reduced capacity)				
1 Beaver goes from B to F				
2 Beavers go from B to C				
1 Beaver goes from C to F				
1 Beaver goes from A to C				
Situation after 2 Minutes	**2**	**3**	**3**	**2**
1 Beaver goes from A to B (shortest route selected)				
1 Beaver goes from B to F				
2 Beavers go from B to C				
1 Beaver goes from A to C (shortest route selected)				
3 Beavers go from C to F				
Situation after 3 Minutes	**0**	**1**	**3**	**6**
1 Beaver goes from B to F				
3 Beavers go from C to F				
Situation after 4 Minutes	**0**	**0**	**0**	**10**

Tunnels of the homestead dam

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation, Algorithms and Digital Systems

We can imagine the network of tunnels in graph theory as a flow network. That is a so called directed graph where each edge has a capacity (the maximal number of tunnels or beavers between the rooms) and each edge receives a flow. The amount of flow on an edge cannot exceed the capacity of the edge.

The goal here is to optimise the flow of the beavers through the network so that as many beavers as possible arrive at the food in the shortest time. For example, such a network can be used to model traffic in a road system. There are several algorithms for these problems, one of them is the *Ford-Fulkerson algorithm.*

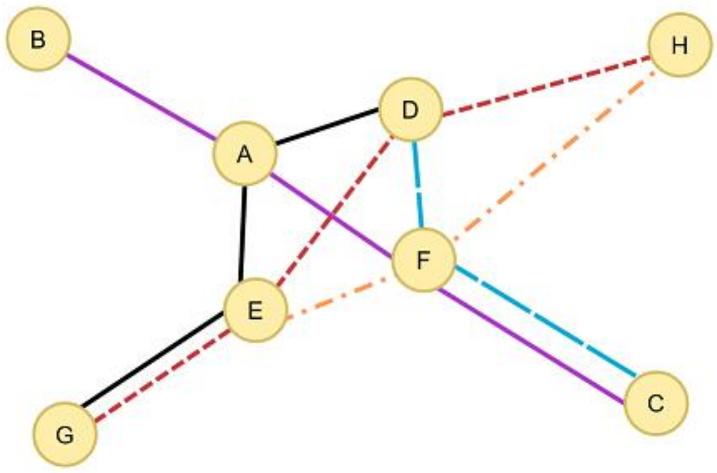
Our problem is a special version of this problem, because we allow beavers to wait in B and C if there is no possibility to continue. In the classical flow network problem nothing can stay in one place. Everything coming in must immediately go out through other channels. Additionally, in the classical version of the problem, prescribed directions to each tunnel exist. Here we are free in which direction we run through the tunnels.

Cost reduction

Year 3+4:
Year 5+6:
Year 7+8: **C**
Year 9+10: **B**
Year 11+12: **B**

Beaverland has eight railway stations and five railway lines. The lines are shown in the diagram below, each with a different colour. Note that it is possible to travel from any station to any other station using at most one train transfer. For example, to get from B to H one could follow the purple line from B to F, transfer to the orange line and go to H.

Because the railway company wants to reduce costs, they plan to shut down one or more rail lines. They must do this in such a way that all stations stay connected to the railway network and that travelling from any station to any other station can still be done using, at most, one train transfer.



Question:
How many railway lines can, at most, be shut down by the railway company?

Cost reduction

Answer:

The correct answer is, at most, 2 lines can be removed (3 lines remain).

Explanation:

If you remove the red line (GEDH) and the light blue line (DFC) all conditions are still satisfied. Indeed the lines that remain (BAFC, GEAD and EFH) clearly serve all eight stations and have the additional property that each pair of them has (at least) one station in common. Therefore, you can go from any station to any other with at most one transfer.

Would it be possible to remove 3 lines instead of 2? In other words, could all the conditions still be satisfied with using only two lines?

The answer is no. Note that the purple line (BAFC) is the only line that serves station B, so it must remain. The other line then must serve the four remaining stations and that leaves red (GEDH) as the only possibility. But these two lines do not have a station in common, and therefore it is, for instance, not possible to travel by train from B to H let alone with, at most, a single transfer. So two lines are not sufficient.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS), Algorithms (AL)

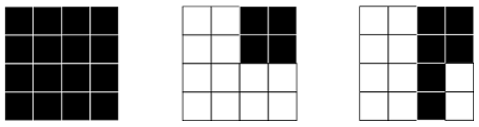
Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation and Algorithms

This task is about a graph that represents a railway network. In Computer Science, graphs are often used to represent complex problems.

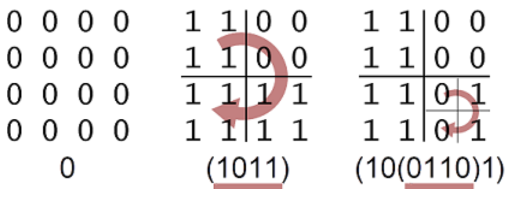
Icon image compression

Year 3+4:
Year 5+6:
Year 7+8:
Year 9+10: **C**
Year 11+12: **C**

Look at the following 4x4 black and white pixel images:



This could be stored using binary digits: “1” for white pixels and “0” for black pixels.
For a 4x4 image we would have to store 16 digits. The following image compression method allows us to store images using less space, especially for simple patterns:



The binary digits are arranged in a grid like the pixels in the images.

The compression method is applied to this grid as follows, producing a string of digits:

1. If all the digits in the grid are 0, the result is ”0” (see left image).
   If all the digits in the grid are 1, the result is ”1”.
2. Otherwise, the grid is divided into quarters. The compression method is applied to each quarter sub-grid from the top left in clockwise order. The results are combined and surrounded by round brackets. Two different examples can be seen in the centre and on the right above.

Note that a sub-grid may consist of one digit only; see the right image, bottom right corner.
In this case, the method will use step 1 only.

Question:

On the right is the binary digit grid for an 8×8 image.
The above compression method is applied to this grid.
Which string of digits can represent this image?

```
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 0 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
```

(1110)

(11(1011)1)

(111(1(1101)11))

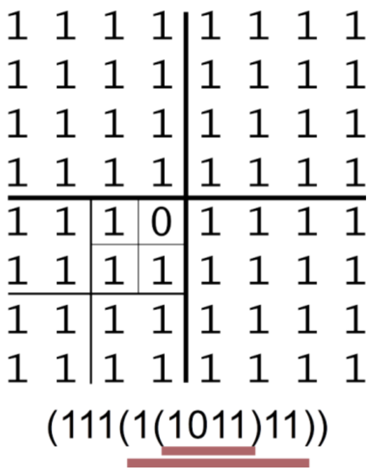
(111(1(1011)11))

Icon image compression

Answer:

The correct answer is D.

Explanation:



It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Evaluation (EV)

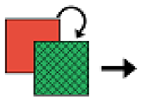
Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation, Implementation and Digital Systems

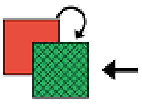
Quad tree compression saves a lot of data storage space but is only suited for certain classes of images.

Robot

Year 3+4:
Year 5+6:
Year 7+8:
Year 9+10: **C**
Year 11+12: **C**

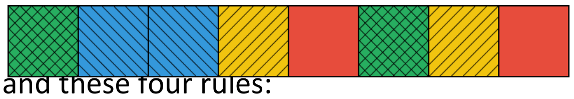
Milan has built a robot that reads coloured squares, changes their colours and moves one square to the left or one square to the right.
The robot acts according to rules like these:


If you see a red square, change its colour to green and move one square to the right.

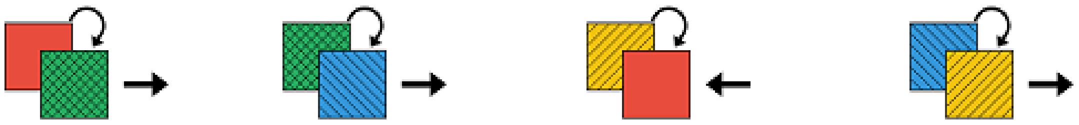

If you see a red square, change its colour to green and move one square to the left.

At the beginning, the robot is standing on the leftmost square. It detects the colour of the square, finds the rule that starts with this colour, changes the colour of the square according to the rule and moves according to the rule. Then, the robot repeats the same procedure for the square that it is standing on, and so on. If it doesn't find an appropriate rule or it goes outside of the squares, it stops.

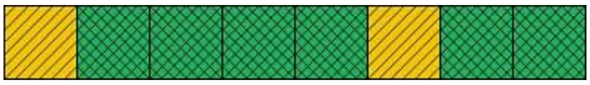
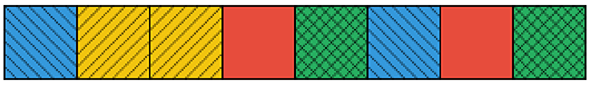
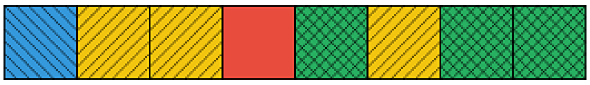
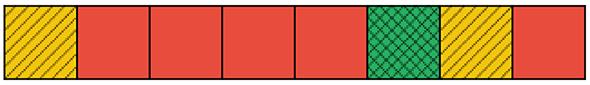
The robot was given this sequence of squares:



and these four rules:

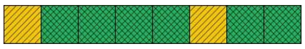


Question:
What will the squares look like when the robot stops?

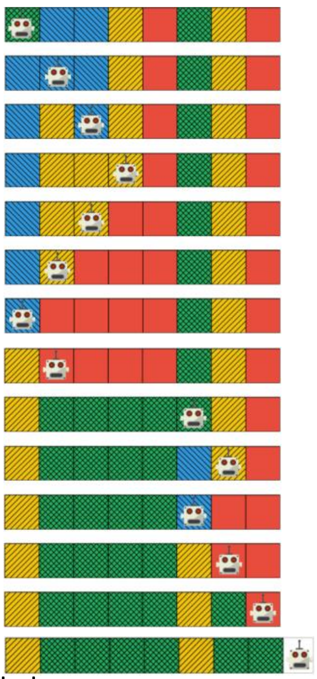





Robot

Answer:



Explanation:



Start:

Following the rule 🟩🟦 → we get

Following the rule 🟦🟨 → we get

Following the rule 🟦🟨 → we get

Following the rule 🟨🟥 ← we get

Following the rule 🟨🟥 ← we get

Following the rule 🟨🟥 ← we get

Following the rule 🟦🟨 → we get

Following the rule 🟥🟩 → 4 times we get

Following the rule 🟩🟦 → we get

Following the rule 🟨🟥 ← we get

Following the rule 🟦🟨 → we get

Following the rule 🟥🟩 → we get

Following the rule 🟥🟩 → we get

Now the robot is outside of the squares, it stops.

It's Computational Thinking:

<u>Computational Thinking Skills:</u> Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS), Algorithms (AL), Evaluation (EV)

<u>Australian Digital Technologies Curriculum Concepts:</u> Abstraction, Data Representation, Algorithms and Digital Systems

In Computer Science it is important to define a model of computation to work with. A *model of computation* is simply a set of rules and structures to comply with. So, for example, when we are programming software our model of computation is the programming language we are using.

The problem in this task defines a model very similar to a well-known model called a *Turing machine*. A Turing machine is a useful model of computation for Computer Scientists because even though it is very simple, it is equivalent to most programming languages, meaning we can turn any software program into a Turing machine and conversely turn any Turing machine into a program.

Funtime school

Year 3+4:
Year 5+6:
Year 7+8: **C**
Year 9+10:
Year 11+12: **A**

The teachers at Funtime School like to include games in their lessons.

At the end of one day, one teacher invites his students to play a game. The winner gets to leave school before the others are dismissed.

Rules of the game:
The school has one corridor with five doors in a row. The students form a queue and take turns to walk down the corridor. When they get to an open door, they must close it and move to the next door, When they get to a closed door, they must open it, go into the classroom, leave the door open and wait there until the teacher dismisses them.

At the start of the game all the doors are closed.

If a student finds all the doors are open, after shutting each of them, they can head home for their dinner.

Question:
If the students are numbered 1 to 35, which student gets to leave school first?

Funtime school

Answer:

The 32nd student leaves school first.

Explanation:

Let “0” represent closed doors and let “1” represent open doors. The following table represents the state of the corridor at the beginning and for the first eight students, assuming that the entrance into the corridor is on the right.

Start: 00000 All doors closed

1st student:	00001 The first is closed; open and enter
2nd student:	00010 Shut the first door, the second is closed, open and enter
3rd student:	00011 The first is closed, open and enter
4th student:	00100 Slams the first two, the third is closed, open and enter
5th student:	00101 The first is closed, open and enter
6th student:	00110 Slams the first, the second is closed, open and enter
7th student:	00111 The first is closed, open and enter
8th student:	01000 Slams the first three, the fourth is closed, open and enter

If this reminds you of binary counting, you're right. This will lead to:

31st student: 11111

So the 32nd student will shut all the doors and leave.

It’s Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Data Representation, Algorithms and Digital Systems

Computers can, at their most basic level, only store representations of binary numbers. The amazing thing is that with groups of binary numbers they can represent numbers, letters, images, sounds, etc. and manipulate them and do calculations.

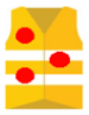
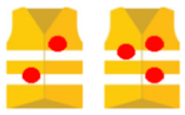
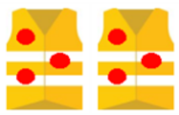
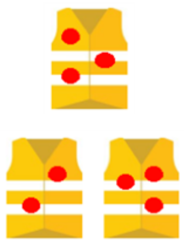
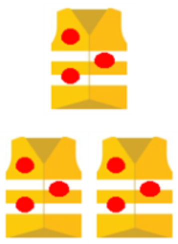
Wash the uniforms

Year 3+4:
Year 5+6:
Year 7+8:
Year 9+10:
Year 11+12: **C**

All uniforms used at a tomato festival need to be washed using a single washing machine. The washing machine can wash up to three uniforms at the same time during each wash cycle.

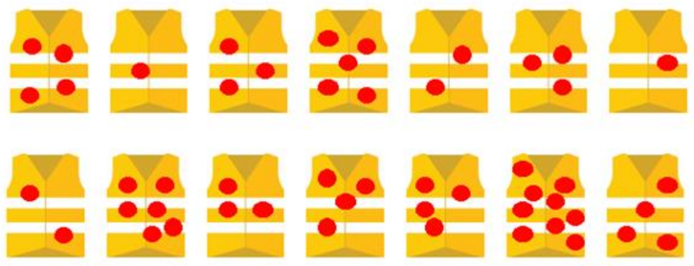
- The number of hours it takes to wash a single uniform is exactly the same as the number of tomato stains on that uniform.
- The number of hours it takes to wash two uniforms at the same time is the same as the number of tomato stains on the dirtier uniform of the two.
- The number of hours it takes to wash three uniforms at the same time is the same as the number of tomato stains on the second dirtiest of the three.

The following table shows possible examples of a laundry process that takes three hours.

1 uniform	2 uniforms	2 uniforms	3 uniforms	3 uniforms
				

Question:

How long does it take to wash the 14 uniforms (shown below) in the quickest way?



Wash the uniforms

Answer:
The correct answer is 14.

Explanation:
The question focuses on the optimisation process of washing all 14 uniforms in one washing machine by grouping the uniforms in groups of one, two or three for each cycle.

In this problem, it is not possible to do less than 5 cycles, thus our optimal solution will have exactly 5 cycles. Since 14 is not divisible by 3, we need four cycles with 3 uniforms and one cycle with 2 uniforms.

To minimise the total wash time, we want the uniforms with the fewest stains to be the dirtiest uniforms in the cycles that use two uniforms and the second dirtiest uniforms in the cycles that use three uniforms (since these are the ones that determine the wash time of a cycle). Thus, the best way to distribute uniforms is to group the uniforms with the fewest stains in pairs until we have five pairs. The uniform with the most stains in each pair will determine the wash time of a cycle, so we can simply distribute the remaining uniforms by the existing pairs in order to get the cycles of our optimal solution.

To do that, it is useful to first sort the uniforms in ascending order:

1 1 2 2 3 3 3 4 4 4 4 5 6 9

Now, we group the first two uniforms of the sorted sequence, then the second two and so on, getting the following five pairs:

(1,1) (2, 2) (3, 3) (3, 4) (4, 4)

If we distribute the rest of the uniforms into the created pairs, we get the following solution (there are other possible solutions with the same total wash time, obtainable by distributing the remaining uniforms differently):

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Algorithms (AL), Evaluation (EV)

Australian Digital Technologies Curriculum Concepts: Abstraction, Specification, Algorithms and Digital Systems

Optimisation problems focus on figuring out the best solution amongst all available solutions.

One of the best known algorithms for obtaining an optimal solution is the *Greedy algorithm*. This algorithm solves the problems of optimal solutions by building the final solution step by step. Usually, the Greedy algorithm performs a data sort to determine the selection order of items that will be part of the optimal solution. The globally optimised solution is built through successive estimations of the local optimal solutions.

Greedy algorithms are simple, in general, and used in optimisation problems, like, finding the optimal sequence of tasks performed on a computer or finding the shortest path in a graph.

Perfect partners

Year 3+4:
Year 5+6:
Year 7+8:
Year 9+10:
Year 11+12: **C**

Andy, Bert, Chris, David and Eric are professional male ballroom dancers that take part in a TV show.

Amy, Brenda, Carol, Dianna and Emma are female contestants that will learn to dance during this show.

Each professional will be assigned a single contestant to teach.

Before the show, the producer organises a party where everybody meets.

After the party the professionals and contestants fill out a questionnaire:

- each professional ranks the contestants in the order that he thinks they can be successful
- each contestant ranks the professionals in the order of how fast she can learn from him

(1 = 1st choice, 2 = 2nd choice, etc.)

Here are the results of these choices:

Professional dancers' preferences

	Amy	Brenda	Carol	Dianna	Emma
Andy	1	3	2	5	4
Bert	1	2	3	4	5
Chris	2	1	4	5	3
David	5	4	3	2	1
Eric	4	5	2	3	1

Contestants' preferences

	Andy	Bert	Chris	David	Eric
Amy	4	3	5	2	1
Brenda	3	4	1	2	5
Carol	2	4	1	5	3
Dianna	5	2	3	4	1
Emma	5	2	3	1	4

The producers want to match the professionals with the contestants in such a way that there will not be any envy.

You are asked to match the professionals with the contestants so that every beaver has a partner. You must also make sure that all unmatched beaver pairs would not be happier if they were actually matched together.

Question:
When you have finished assigning partners, who is Eric's partner?

Amy **Brenda** **Carol** **or** **Dianna**

Perfect partners

Answer:
Eric's partner is Dianna.

Explanation:
It turns out that given this set of preferences there is only one set of partners:

Andy & Carol

Bert & Amy

Chris & Brenda

Dave & Emma

Eric & Dianna

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS), Algorithms (AL)

Australian Digital Technologies Curriculum Concepts: Abstraction, Algorithms, Interactions and Impacts

This is an important problem in Computer Science. Although there are not many problems that require this solution the few that do are important. One example where this solution is applicable is that of assigning trainee doctors to their preferred hospital practice. A very important example in Computer Science is assigning users to servers in a large distributed internet service.

Using this algorithm, and Am to represent a male dancer starting with A (ie Andy) and Af to represent a female dancer with initial A (ie Amy), etc., the solution above is produced like this:

Round 1:
Am asks Af
Bm asks Af
Cm asks Bf --> provisional Y
Dm asks Ef --> provisional Y
Em asks Ef

Round 2:
Am asks Cf --> provisional Y
Bm asks Bf
Em asks Cf

Round 3:
Bm asks Cf
Em asks Df --> provisional Y

Round 4:
Bm asks Df

Round 5:
Bm asks Ef

Round 6:
Bm asks Af --> provisional Y

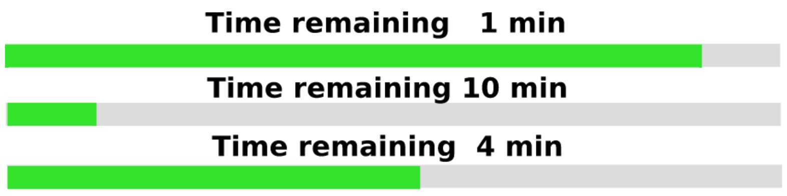
As this particular task only needs solving for five pairings, there are other simpler methods of coming up with the same answer.

Downloads

Year 3+4:
Year 5+6:
Year 7+8:
Year 9+10: **C**
Year 11+12: **C**

When downloading files from a server, the download speed is limited.
For example, when 10 files are downloaded simultaneously, the download speed for each file is 10 times slower than it would be for only one file.

A user simultaneously downloads three files from a server.
The picture below shows the current download state.



Note that the time remaining for each file is computed based only on the current speed and does not depend on any history.

Question:
How many minutes will it take to download all the files?

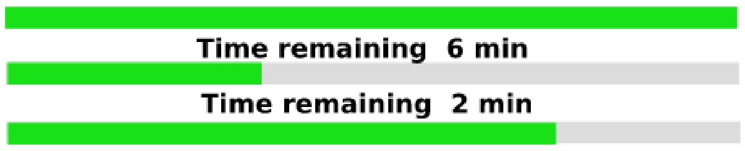
Downloads

Answer:

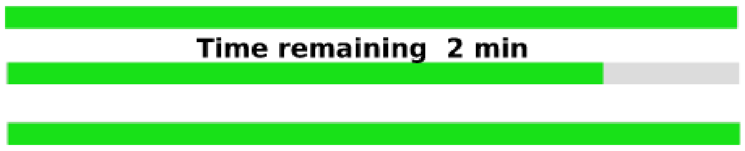
The correct answer is 5.

Explanation:

After one minute the first file will be downloaded, so the speed will increase by a factor of 3/2 (that is, the 3 downloading files became 2 downloading files). The progress will look like:



After a further two minutes the third file will be downloaded, and the progress will look like:



There are two more minutes needed to download the last file. So, after 1 + 2 + 2 = 5 minutes all the files will be downloaded.

It's Computational Thinking:

Computational Thinking Skills: Decomposition (DE), Abstraction (AB), Modelling and Simulation (MS), Evaluation (EV)

Australian Digital Technologies Curriculum Concepts: Abstraction, Digital Systems, Interaction and Impacts

A progress bar is an element of a visual user interface (sometimes called a *widget*), that is usually represented as a rectangle or any other area partly filled by a colour or a pattern. The size of a filled part represents a fraction of time passed from the start of some operation to the total time it would take. Progress bars are used when remaining time to finish an operation is known (at least approximately).

Many aspects of user interfaces require calculations. Another example is when you resize a window, the contents contained in the window as well as other elements on the screen may need to be adjusted, and these adjustments can involve detailed computation. These issues are part of the fields of human computer interaction or user interfaces.