

Bebras Australia Computational Thinking Challenge

2020 Solutions Guide Round 1



Bebras Australia Computational Thinking Challenge

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

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

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

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

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

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

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

Engaging young minds for Australia's digital future

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

digitalcareers.csiro.au

622

Australian schools participated in Bebras in 2019



43 163

Australian students participated in Bebras in 2019



2.9million

Students participate worldwide



digital
careers

Contents

What is a Solutions Guide?	4	Bridges and Islands	51
What is Computational Thinking?	5	Ancient Message	52
Computational Thinking skills alignment	6	Seating Arrangement	53
Digital Technologies key concepts	8	Wood Allergies	54
Digital Technologies alignment	9	Bottle Recycling	56
		Ants in a Pond	57
		Tunnels	59
		Aircraft Scheduling	60
		Beaver Ball Game	62
		Many Beavers	63
		Bee Hive	64
Years 3+4		Years 9+10	
Art Scratch Paper	12	Ancient Message	66
Crossroads	13	Greener Flight Routes	67
Rubbish Robot	14	Wood Allergies	69
Hats for Snowmen	15	Ants in a Pond	71
Smoke Signals	16	Shapes	73
Eddie's Lunch Maze	17	Seating Arrangement	74
Plates	18	Embroidery Machine	75
Double Parked	19	Scientists in the Lab	77
Space Travel	20	Many Beavers	79
Box of Marbles	21	Bee Hive	80
Binary Bulbs	22	Knight's Moves	81
Gears in Motion	23	Encoded Treasure Map	82
Meal Planning	24	Willows and Poplars	83
Animals	25	Sawmill	85
Hamburgers	26	Rescue Mission	86
		Years 11+12	
		Embroidery Machine	88
		Bee Hive	90
		Rescue Mission	91
		Drawing Triplets	92
		Scientists in the Lab	94
		Railway Electrification	96
		Willows and Poplars	97
		Encoded Treasure Map	99
		Way To Go Home	100
		Warehouses	101
		Video Compression	102
		Friendship Bracelets	103
		Shoe Shopping	105
		Rotate Rotate Rotate	107
		Water Pouring	108
Years 5+6			
Art Scratch Paper	28		
Hats for Snowmen	29		
Box of Marbles	30		
Eddie's Lunch Maze	31		
Well Placed Towers	32		
Binary Bulbs	33		
Animals	34		
Gears in Motion	35		
Ancient Message	36		
Meal Planning	37		
Trap Island	38		
Many Beavers	39		
Wood Allergies	40		
Ants in a Pond	42		
Safe	44		
Years 7+8			
Library Break-in	47		
Meal Planning	48		
Wizard Bibraxus	49		
Eddie's Lunch Maze	50		

What is a Solutions Guide?

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

Within this Solutions Guide you will find all of the questions and tasks from Round 1 of the Bebras Australia Computational Thinking Challenge 2020. On each page above the question you will find the age group, level of difficulty, country of origin and key Computational Thinking skills.

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

What is Computational Thinking?

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



DECOMPOSITION

Breaking down problems into smaller, easier parts.



PATTERN RECOGNITION

Using patterns in information to solve problems.



ABSTRACTION

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



MODELLING AND SIMULATION

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



ALGORITHMS

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



EVALUATION

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

More Computational Thinking resources

Visit digitalcareers.csiro.au/CTIA to download the Computational Thinking in Action worksheets.

These can be used as discussion prompts, extension activities or a framework to build a class project.

Each resource was designed to develop team work; critical and creative thinking; problem solving; and computational thinking skills.



Computational Thinking skills alignment

2020 Round 1 Questions	Grade level	Decomposition	Pattern Recognition	Abstraction	Modelling & Simulation	Algorithms	Evaluation
Art Scratch Paper	3+4 Easy 5+6 Easy						
Crossroads	3+4 Easy						
Rubbish Robot	3+4 Easy						
Hats for Snowmen	3+4 Easy 5+6 Easy						
Smoke Signals	3+4 Easy						
Eddie's Lunch Maze	3+4 Medium 5+6 Easy 7+8 Easy						
Plates	3+4 Medium						
Double Parked	3+4 Medium						
Space Travel	3+4 Medium						
Box of Marbles	3+4 Medium 5+6 Easy						
Binary Bulbs	3+4 Hard 5+6 Medium						
Gears In Motion	3+4 Hard 5+6 Medium						
Meal Planning	3+4 Hard 5+6 Medium 7+8 Easy						
Animals	3+4 Hard 5+6 Medium						
Hamburgers	3+4 Hard						
Well Placed Towers	5+6 Easy						
Ancient Message	5+6 Medium 7+8 Medium 9+10 Easy						
Trap Island	5+6 Hard						
Many Beavers	5+6 Hard 7+8 Hard 9+10 Medium						
Wood Allergies	5+6 Hard 7+8 Medium 9+10 Easy						
Ants in a Pond	5+6 Hard 7+8 Medium 9+10 Easy						

Safe	5+6 Hard						
Library Break-in	7+8 Easy						
Wizard Bibraxus	7+8 Easy						
Bridges and Islands	7+8 Easy						
Seating Arrangement	7+8 Medium 9+10 Medium						
Bottle Recycling	7+8 Medium						
Tunnels	7+8 Hard						
Aircraft Scheduling	7+8 Hard						
Beaver Ball Game	7+8 Hard						
Bee Hive <i>multiple choice</i>	7+8 Hard						
Greener Flight Routes	9+10 Easy						
Shapes	9+10 Easy						
Embroidery Machine	9+10 Medium 11+12 Easy						
Scientists in the Lab	9+10 Medium 11+12 Easy						
Bee Hive <i>open answer</i>	9+10 Medium 11+12 Easy						
Knight's Moves	9+10 Hard						
Encoded Treasure Map	9+10 Hard 11+12 Medium						
Willows and Poplars	9+10 Hard 11+12 Medium						
Sawmill	9+10 Hard						
Rescue Mission	9+10 Hard 11+12 Easy						
Drawing Triplets	11+12 Easy						
Railway Electrification	11+12 Medium						
Way To Go Home	11+12 Medium						
Warehouses	11+12 Medium						
Video Compression	11+12 Hard						
Friendship Bracelets	11+12 Hard						
Shoe Shopping	11+12 Hard						
Rotate Rotate Rotate	11+12 Hard						
Water Pouring	11+12 Hard						

Australian Digital Technologies curriculum key concepts

Abstraction

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

Data Collection

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

Data Representation

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

Data Interpretation

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

Specification

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

Algorithms

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

Implementation

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

Digital Systems

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

Interactions

Human–Human Interactions: How users use digital systems to communicate and collaborate.

Human–Computer Interactions: How users experience and interface with digital systems.

Impact

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

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

Digital Technologies

key concepts alignment

2020 Round 1 Questions	Abstraction	Data Collection	Data Representation	Data Interpretation	Specification	Algorithms	Implementation	Digital Systems	Interactions	Impacts
Art Scratch Paper										
Crossroads										
Rubbish Robot										
Hats for Snowmen										
Smoke Signals										
Eddie's Lunch Maze										
Plates										
Double Parked										
Space Travel										
Box of Marbles										
Binary Bulbs										
Gears In Motion										
Meal Planning										
Animals										
Hamburgers										
Well Placed Towers										
Ancient Message										
Trap Island										
Many Beavers										
Wood Allergies										
Ants in a Pond										
Safe										
Library Break-in										

2020 Round 1 Questions	Abstraction	Data Collection	Data Representation	Data Interpretation	Specification	Algorithms	Implementation	Digital Systems	Interactions	Impacts
Wizard Bibraxus										
Bridges and Islands										
Seating Arrangement										
Bottle Recycling										
Tunnels										
Aircraft Scheduling										
Beaver Ball Game										
Bee Hive										
Greener Flight Routes										
Shapes										
Embroidery Machine										
Scientists in the Lab										
Bee Hive										
Knight's Moves										
Encoded Treasure Map										
Willows and Poplars										
Sawmill										
Rescue Mission										
Drawing Triplets										
Railway Electrification										
Way To Go Home										
Warehouses										
Video Compression										
Friendship Bracelets										
Shoe Shopping										
Rotate Rotate Rotate										
Water Pouring										

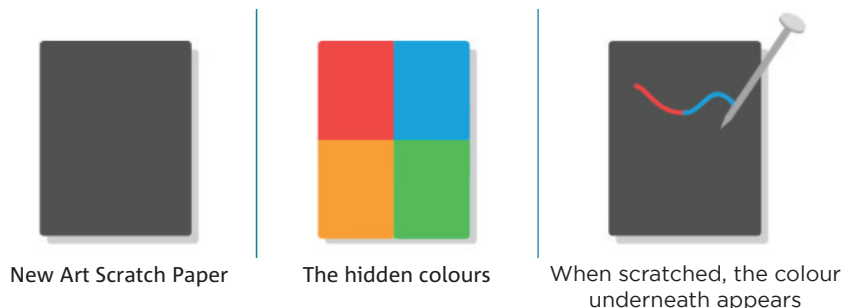
Bebras Challenge 2020 Round 1

Years 3+4



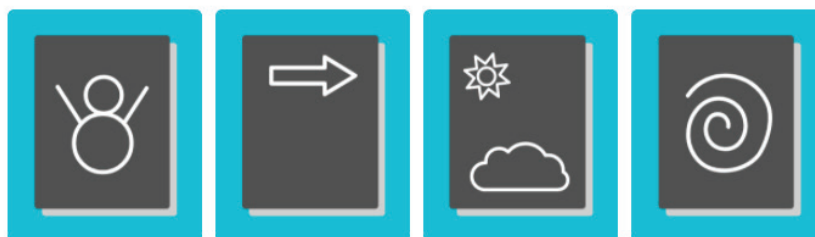
Art Scratch Paper

Art Scratch Paper is special. It is dark grey but has four colours hidden below. You can draw a pretty picture by scratching it with a sharp object:



Question

If using Art Scratch Paper, which of these drawings will show exactly three colours?



Answer

Only the drawing of the sun and cloud touches exactly three colours.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Pattern Recognition, Abstraction

Concepts: Abstraction, Data Representation, Data Interpretation

When you work with computer software to create images, you often have to deal with layers that look like sheets of paper in a stack. The images in each layer can mix with one another. Paint and design programs have even more possibilities: they allow you to change the order of the layers, and they also allow you to take parts of the layer away. This is called transparency.

If you look at any website you will see many things on top of each other: the background image, colours, banners, images, text, etc. In order to display this correctly the computer has to know in which order things are put on top of each other. Computers call this the z-index; a number that says where a layer is located in the stack.



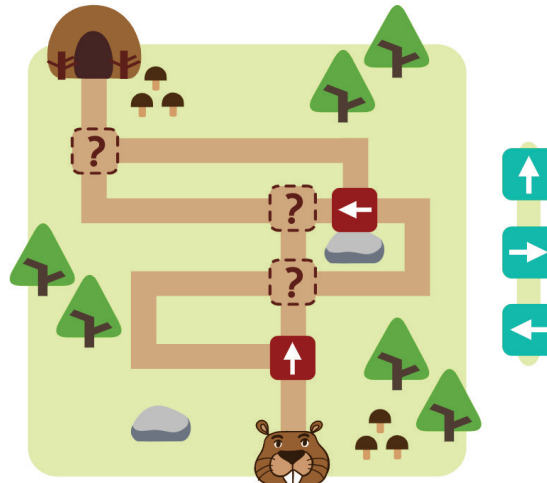
Crossroads

Jordan wants to go home.

Normally he follows the directions on the signs, but a strong wind has blown 3 of them away.

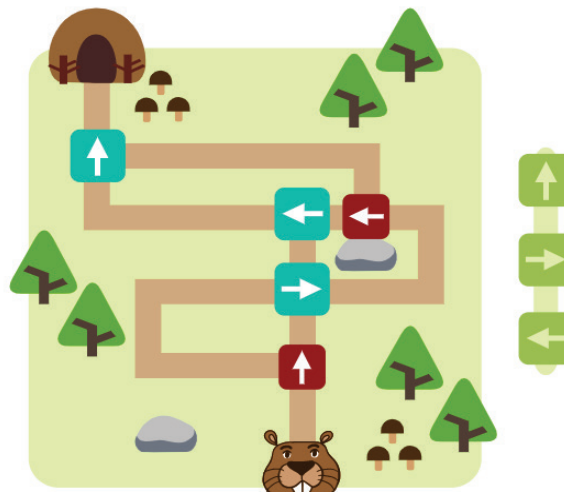
Question

Help Jordan find his way home by dragging the new blue signs into their correct places.



Answer

This answer can be found by working backwards. At the last intersection you have to select the up arrow, otherwise Jordan will go in a loop. At the first intersection you can only choose the right arrow because the left arrow will lead Jordan in another loop. As a result, the left arrow is all that remains for the second intersection.



It's Computational Thinking

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

Concepts: Abstraction, Specification, Algorithms

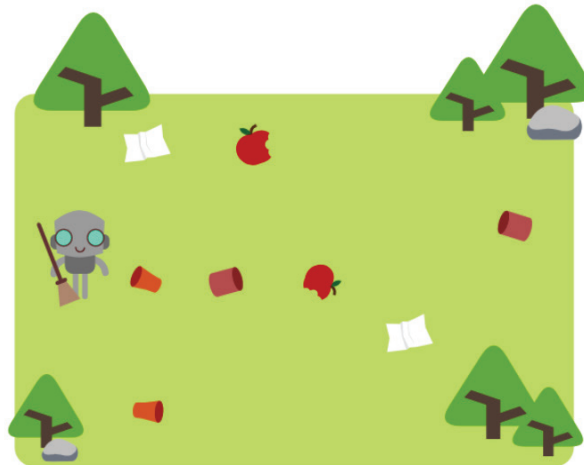
An algorithm is a method used to solve problems and represent a succession of operations and decisions that must be made. Starting from one point, reaching another point, corresponding to a correct solution to the problem.

In this task, decisions must be made at each intersection by selecting the correct arrow sign. Depending on their sequence, other ways to go from the start point to the end point might appear. There is also the risk of entering a loop, which determines an algorithm will never end and never reach the end point.



Rubbish Robot

A robot picks up rubbish. The robot moves to the closest piece of rubbish and picks it up. It then moves to the next closest piece of rubbish and picks that up. It keeps going in this way until all of the rubbish has been picked up.



Question

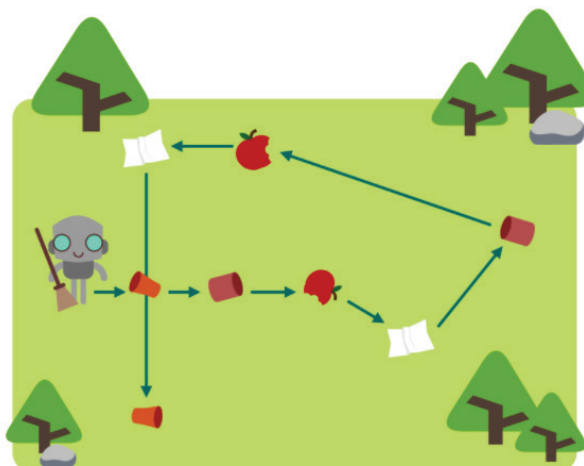
Which type of rubbish will the robot pick up last?



Answer

The type of rubbish picked up last is a paper cup.

The image shows the route the robot takes according to the instructions. It starts by picking up the paper cup as it is the closest piece of rubbish. It then moves on to the tin as it is closest and picks it up. It then follows the route shown in the image below.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

The behaviour of a robot is controlled by a program, running on the computer that is built into the robot. A computer program consists of instructions written in a language that a computer can understand.

Robots have cameras and sensors that enable them to identify objects in their environment and calculate distances to these objects. Robot control programs process the inputs of these sensors and enable robots to behave autonomously, i.e. without human interference.



Hats for Snowmen

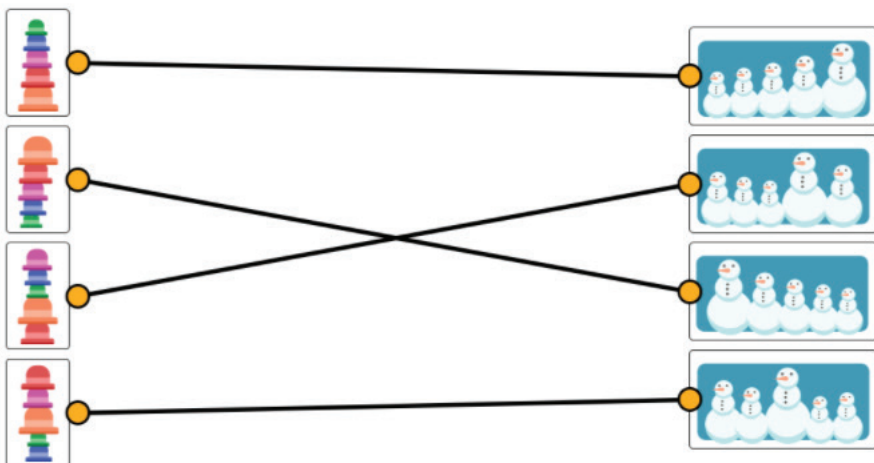
Five snowmen are standing in line. They all want a hat that fits them. The snowman on the left gets the hat on top of the pile. The next snowman gets the next hat in the pile. This continues until the snowman on the right gets the bottom hat.

Question

Connect the piles of hats with the correct line of snowmen so that every snowman gets a hat that fits him correctly.



Answer



The first pile of hats belongs to the first line of snowmen. The first line of snowmen is ordered from smallest to largest and so is the first pile of hats. The second pile of hats belongs to the third line of snowmen. The snowmen are in line from largest to smallest and so are the hats. The third pile of hats belongs to the second line of snowmen. Here, the first snowman is the third largest and the first hat is the third largest hat. The fourth pile of hats belongs to the fourth line of snowmen. The first snowman is the second largest and so is the top hat.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms, Evaluation

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

The hats are presented as a stack. This models a way that some computer data is stored - the abstract data structure (ADT) stack. With stacks you can only do two things or operations: take away the topmost element or put a new element on top of the stack.

Since we don't know how the line of snowmen was made, it could be that the line was created from the left or right. That is called a queue, which has the following operations: take away the front-most element or add a new element at the end of the queue.



Smoke Signals

The Weather Beaver uses smoke signals to send messages from the top of a mountain to the beavers in the valley below.

She uses small and large smoke clouds and the following code to send her messages:

Thunderstorm	Light Rain	Cloudy	Sunny
			

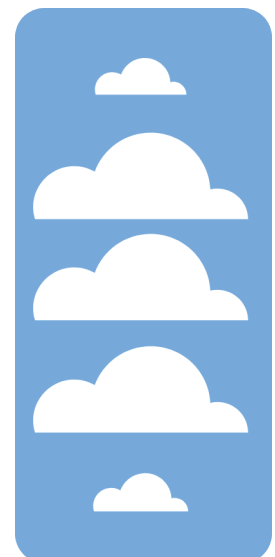
One day, the beavers in the valley see the following message:

But something has gone wrong! She mistakenly sent either a small cloud instead of a large cloud, or a large cloud instead of a small one.

Question

Which message did the Weather Beaver mean to send?

Thunderstorm	Light rain	Cloudy	Sunny
--------------	------------	--------	-------



Answer

Cloudy.

If you change the third cloud from large to small you will get the code for Cloudy. Thunderstorm is not the correct answer because the first and the last cloud would need to be changed. Light Rain is also not correct, because the first, second and fourth cloud would need to change and Sunny is incorrect, because all except the first cloud would need to change.

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

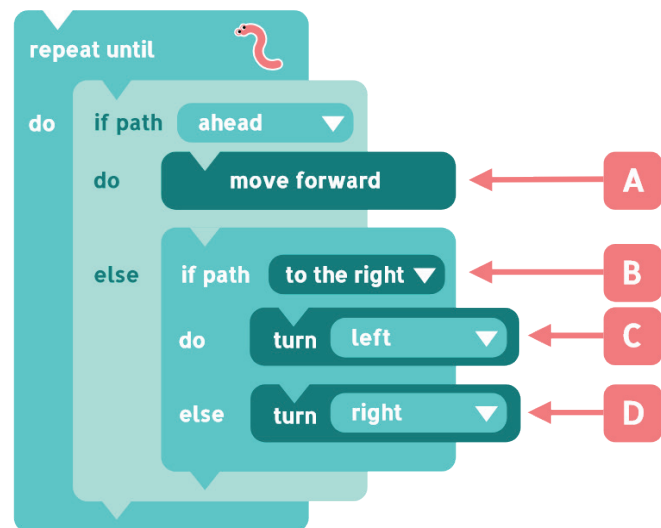
If you design a sequence of symbols to be used for communication (by humans or computers) it is better to design the sequence in a way that the information can be reconstructed even if some parts of the message are lost or damaged. The basic principle behind this code is that these four cloud sequences differ from one another, in at least three positions.

This is done by sending more information than necessary so that the essential information is redundant. If the original meaning of a sequence of symbols can be reconstructed even if n errors occurred, the design of the sequences of symbols is called an n error correcting code.



Eddie's Lunch Maze

It's lunch time for Eddie the bird! The block code should take Eddie to the worm by the path that you see in the image below. But there is a mistake in the code!



Question

Which part of the block code contains a mistake?



Answer

In answer B, change **if path right** to **if path left**

The program needs to be run through to see where it is getting stuck. If you slow the program down and follow the movement along with the code you can identify when things go wrong.

It's Computational Thinking

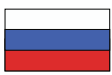
Computational Thinking Skills: Decomposition, Pattern Recognition, Abstraction, Algorithms, Evaluation

Concepts: Abstraction, Specification, Algorithms

An algorithm is a sequence of instructions or set of rules used to achieve a task. Sometimes, when we need to make decisions we use IF THEN ELSE statements. These are known as branching statements.

We also use repeat loops to make our program more efficient.

In this question, we were looking for a mistake. This is called debugging, which is an important skill for computer programmers to develop.



This question comes from
Russia

Years 3+4 Medium

Years 5+6

Years 7+8

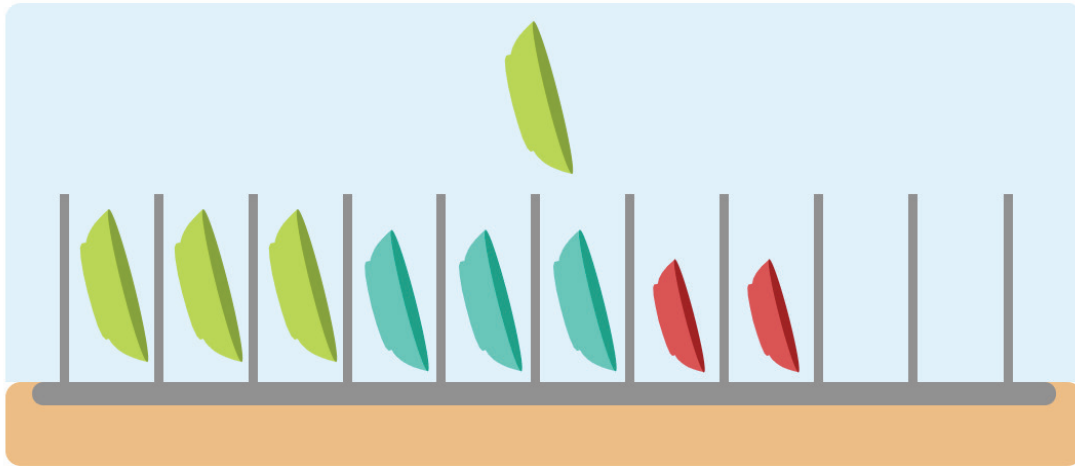
Years 9+10

Years 11+12



Plates

It looks nice when plates are placed in the rack in order of size: the largest ones first, then the medium ones, and finally the small ones. You need to put another large plate into the rack.



Question

Drag the plates to arrange them in the correct order using the fewest number of moves possible.

Answer

The plates can be arranged in correct order with only three moves.

It's Computational Thinking

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

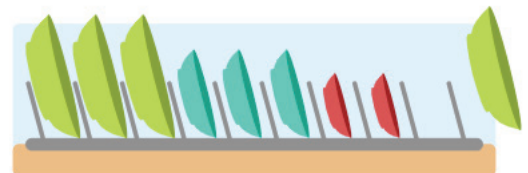
Concepts: Abstraction, Specification, Algorithms

A very straightforward solution to this problem is to move all the medium and the small plates 1 slot to the right. This results in an empty space just where we need it. In our task, this means that we would have to move 6 plates.

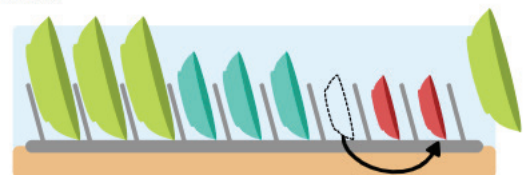
This straightforward method (algorithm) is used in computers as well. The main difference is that computers may work with far more elements - up to billions of them. This means that if we want to insert one element into a list of sorted values, it is possible that we have to move billions of other values. That is slow even for a computer. To solve this, computer scientists had to come up with better algorithms just like the one used to solve this task.

If there are identical items in the list, and we know the number of items for each different value, then in order to insert a new item, it is enough to move as many elements as the number of unique values. In our case this means 3 plates, since there are three different sizes of plates.

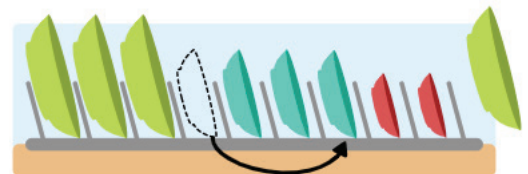
Start:



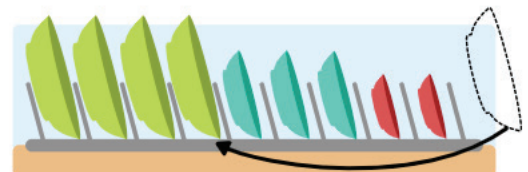
Move 1:



Move 2:



Move 3:





Double Parked

In the parking lot below, cars can be parked either in the parking spaces or in front of these parking spaces.

When a car wants to leave its parking space, any cars that are parked in front of it must move forward or backwards to let the car out.

For example, in the image below:

Car A is not blocked and can get out of its parking space.

Car L is blocked by car M.

Car M must move backwards before car L can get out of its parking space.

Question

Find the one car that can only get out if two cars move (either forward or backwards).



Answer

Car I

Car I is blocked by car N. There is not enough space to push car N away so that car I can leave its parking space. Therefore, car O must be pushed to the left or car M must be pushed to the right. Then, there is space to push car N away for car I to leave its parking space.

There are no other cars for which two cars must be moved. Cars A, D, E, J and Q can leave their parking spaces immediately. Cars B, C, F, G, H, K and L can leave their parking spaces when only one car is moved.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms, Evaluation

Concepts: Abstraction, Specification, Algorithms, Interactions, Data Collection, Data Interpretation

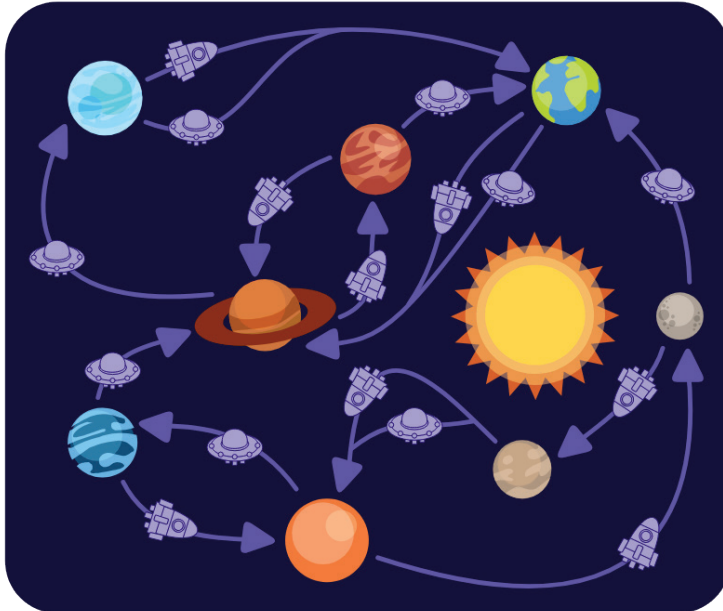
This task involves two important aspects related to Computational Thinking and computer science:

1. An exhaustive search (brute-force) algorithm to search through all possible candidate cars and checking which is the one that has the required property, i.e. 'which can only leave its space after two other cars are moved'.
2. The autonomous (automatic) parking algorithms, which are becoming readily available in cars nowadays. A lot of research is being done in the field of public transportation/parking systems based on autonomous vehicles. One of the advantages is that autonomous vehicle parking can be done very efficiently.



Space Travel

Astronauts can travel between planets using a rocket  or a spaceship  .
The map below shows possible journeys.



Example

An astronaut wants to travel from Venus  to Saturn .

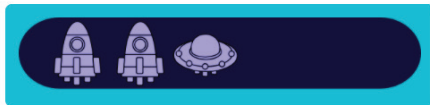
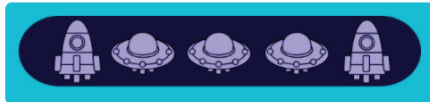
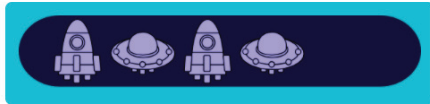
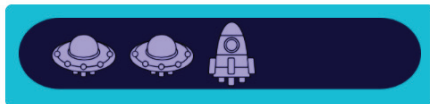
She could take a rocket to Jupiter ,
then a spaceship to Neptune ,
and finally a spaceship to Saturn .

The astronaut shortens this description by
writing the route as:   

Joanna, an astronaut, is stuck on Neptune
and wants to come back home to  Earth.

Question

Which of these routes will **NOT** bring Joanna back to Earth?



Answer

The correct answer is B.

If Joanna used this travel scenario, she would end up back on Neptune. First she would fly to Jupiter in a rocket, then back to Neptune in a spaceship, then again to Jupiter in a rocket, and finally back to Neptune in a spaceship.

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

This task is based on how a computer executes a sequence of commands. In computer science, we call it Deterministic Finite Automaton (DFA). The map of space in our task is an example of a DFA.

DFA have:

- An input alphabet (in our case the rocket and the spaceship)
- A finite set of states (planets)
- An initial state (where our astronaut starts her journey)
- A set of final states (where the astronaut should end her journey)
- Transitions from one state to another (in our case the possible flights between planets)

A DFA accepts a sequence of symbols only if it leads from the initial state to a final state. For example, a DFA can model software that decides whether or not user input (such as email addresses) are valid.



Box of Marbles

A toy to teach simple programs consists of:

- A see-through tube which is closed on the left and open on the right.
- Seven different marbles.
- A simple programming system.



The program can only move one marble at a time.

Example

To put the  marble in between  and  above, this program could be written:



Question

If the toy looks like this:



Write a program to move the marbles into this order:



Answer

The correct answer is:



It's Computational Thinking

Computational Thinking Skills: Algorithms

Concepts: Specification, Algorithms, Implementation, Digital Systems, Impacts

This question is related to a basic Data Structure in computer science called a stack. A stack is a linear structure represented by a pile.

The box in the question only has one opening, so we need to follow the rule 'first in, last out' (FILO) to put in/take out marbles. In our daily life, the stack is like a pile of books. When you remove a book from a pile, you need to remove all books above it so it doesn't come crashing down. In the same way, to remove marbles from the box, you must remove every marble that comes before the marble you want.

In Computer Science, a stack can also be used to convert infix expression to postfix expression. So it will be easier for a program to compute the answer.



Binary Bulbs

Beavers use binary bulbs to send messages. To send a message they use the following table and rules. The beavers sent the following message:

A	01000001	J	01001010	S	01010011
B	01000010	K	01001011	T	01010100
C	01000011	L	01001100	U	01010101
D	01000100	M	01001101	V	01010110
E	01000101	N	01001110	W	01010111
F	01000110	O	01001111	X	01011000
G	01000111	P	01010000	Y	01011001
H	01001000	Q	01010001	Z	01011010
I	01001001	R	01010010		



Rules:



Question

Which message did they send?

Happy

Honey

Horse

House

Answer

The correct answer is HOUSE.

Following the table and the above rules, we can conclude the following:



It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

Concepts: Abstraction, Data Interpretation, Specification, Algorithms, Digital Systems

Binary is the base two number system invented by Gottfried Leibniz that is made up of only two numbers: 0 and 1. This is the basis for all binary code, which is used to write data such as the instructions that computer processors use, or the digital text you read every day.

It's not difficult to find a binary system in our life. Bulbs and switches are good examples as you can see in this task. We can make codes and messages with this system, and can send messages and communicate with our friends. Morse code is another example.



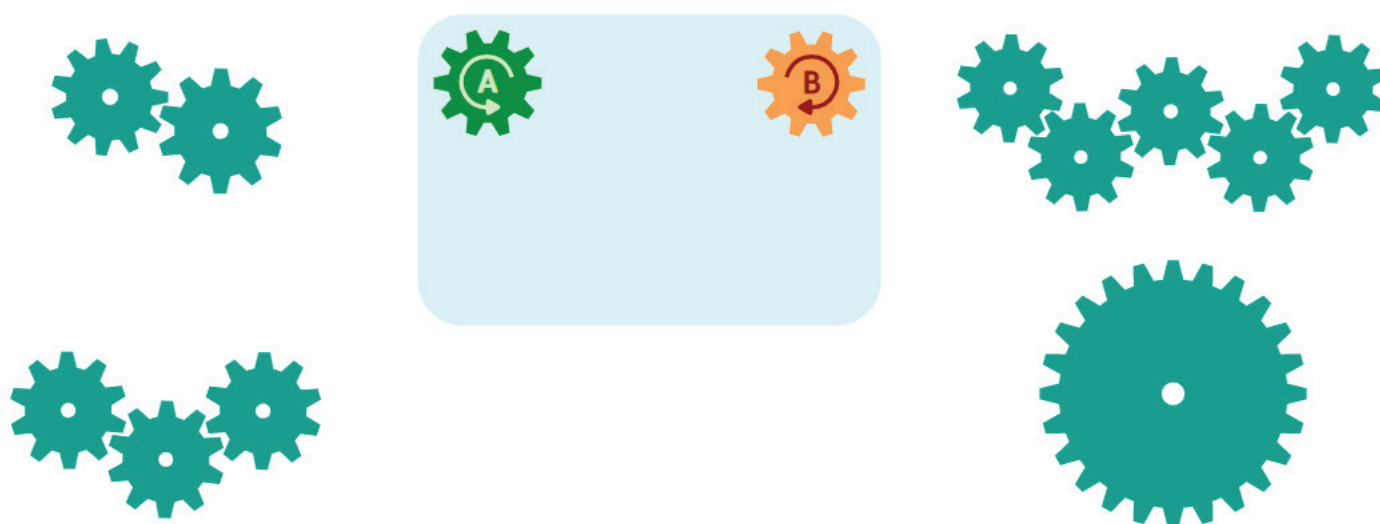
Gears in Motion

Look at the gears below.

When the green gear A turns clockwise as shown by the arrow, the orange gear B must turn in the opposite direction as shown by its arrow.

Question

Which set of gears should be inserted below to allow B to turn in the direction shown?



Answer

The block of two gears is required.

The way to solve this problem is to think about which way gear A turns the next one and so on. If you place the single big gear in between A and B, it will make B turn in the wrong direction. This will happen with all odd numbers of gears. So the only correct answer is placing two gears in between A and B.

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

It is common for computer programmers to make simulations of real-world machines. There are two ways we can use abstraction to make this problem easier to solve:

1. Re-draw the gears in a single line.
2. Realise that alternating gears turn in opposite directions.



Meal Planning

Raju lives in India. He takes 2 lunchboxes to school each day: one for the snack break, and the other for the lunch break.

He only has 30 minutes to prepare both boxes. Below you will find all the available items for the boxes and the preparation time each food needs:

		
Rice Preparation time: 15 min	Poha Preparation time: 20 min	Sandwich Preparation time: 15 min
		
Roti Preparation time: 20 min	Paratha Preparation time: 20 min	Bhel Preparation time: 10 min

Question

Which one of the following statements is **false**?

If Raju packs Roti for lunch, he cannot also pack a Sandwich for his snack.

If Raju packs Paratha for lunch, he can only pack Bhel for his snack.

If Raju packs Rice for lunch, he can pack Poha for his snack.

If Raju packs Rice for lunch, he can pack either a Sandwich or Bhel for his snack.

Answer

If Raju packs Rice for lunch, he can pack Poha for his snack.

Rice takes 15 minutes, and Poha takes 20 minutes, which makes it 35 minutes. So this statement is False. All the other statements have combinations which take 30 minutes or less, so they are true.

Raju has 30 minutes to pack both boxes. The correct combinations of dishes are those which take less than or at most 30 minutes to pack. As this is a multi-choice question there are only a few combinations to consider, so for each combination we have to add up the time required and compare it with the set 30 minutes.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

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

Constraints are the conditions which reduce the choices available in a situation. Solutions need to satisfy all the constraints.

In complex tasks, constraints may be combined by logical operators. In this problem, the constraint was that the total time for preparation is limited to 30 minutes.



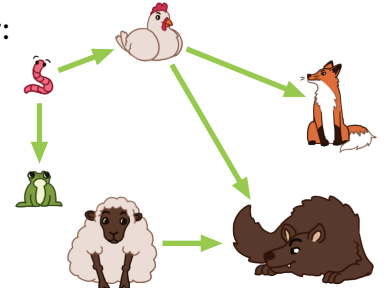
Animals

Six animals need to be put in six pens.

An animal cannot be in a pen that shares a fence with an animal that will eat it.

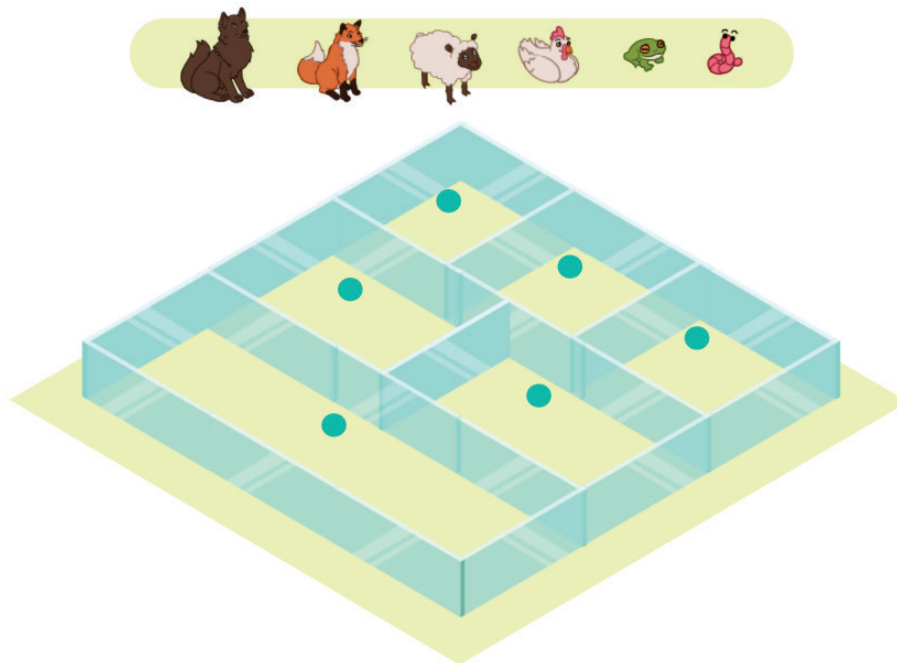
The arrows in the diagram on the right show which animals eat each other:

- The worm gets eaten by the frog or the chicken
- The sheep gets eaten by the wolf
- The chicken gets eaten by the fox or the wolf



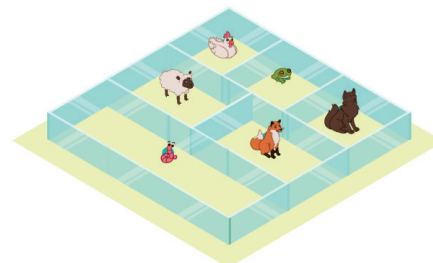
Question

Drag the animals below to the correct pen, so that none of the animals get eaten.



Answer

There are multiple correct answers. This is one:



It's Computational Thinking

Computational Thinking Skills: Algorithms, Evaluation

Concepts: Data Interpretation, Specification, Algorithms, Digital Systems, Interactions, Impacts

To solve this problem, we choose an animal who has several predators, for example the chicken who is hunted by both the wolf and fox. We check if the chicken is safe. We continue to test the animals with the most to least predators until we are sure each is safe.

In Informatics, this is called a 'constraint satisfaction problem'. The solution is one where a set of variables (all the animals) must follow a set of rules or 'constraints' (don't live beside a predator).



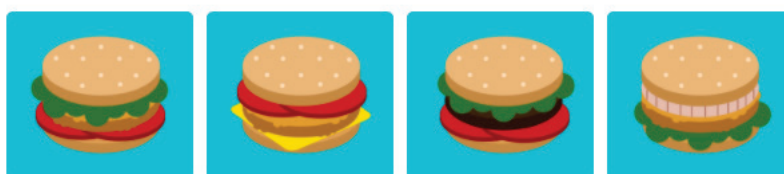
Hamburgers

BeaverKing uses six types of fillings (A, B, C, D, E, and F) in order to make a burger. The following table shows the different types of burgers and their fillings. The fillings are not listed in any particular order.

Burger				
Fillings	C, F	A, B, E	B, E, F	B, C, D

Question

Which of the burgers below has fillings A, E, and F?



Answer

The correct answer is

We can match the letters and fillings by comparing the similarities in the burgers:

Compared Burgers	Common Letter	Common Filling
C,F B,E,F	F	green "lettuce"
C,F B,C,D	C	brown "meat"
A,B,E B,C,D	B	yellow "cheese"
B,E,F A,B,E	E	orange "chicken"

For each burger below, since we already know all but one filling, the remaining one must be the unknown filling.

Burger	Unique Letter	Unique Filling
A,B,E	A	red "tomato"
B,C,D	D	white "onion"

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

Logic is the base of many aspects of Computer Science. This problem can be solved with logical reasoning by identifying known information such as common fillings in burgers, the unknown information can be derived.

In set theory, a set is a group of members (such as different types of fillings in this case). Sets can undergo operations such as intersection and difference:

- The **intersection** of the sets of fillings produces the common fillings.
- The **difference** between all fillings and the known fillings in a burger produces the unknown fillings (removing yellow 'cheese' and orange 'chicken' fillings results in the remaining red 'tomato' filling).

If the intersection or difference only contains one member we can directly determine the unknown filling.



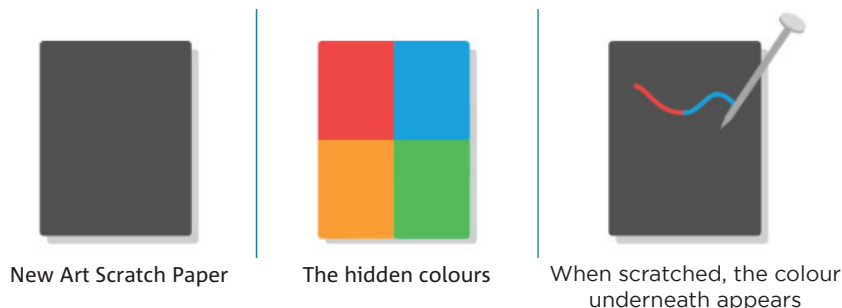
Bebras Challenge 2020 Round 1

Years 5+6



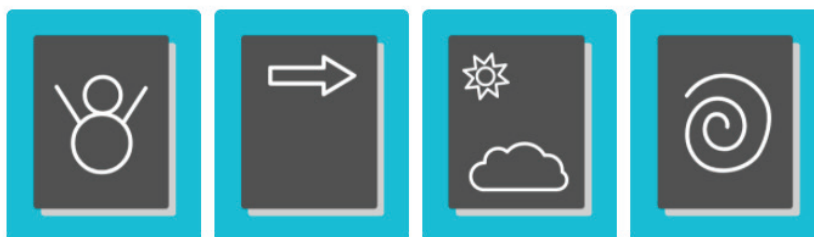
Art Scratch Paper

Art Scratch Paper is special. It is dark grey but has four colours hidden below. You can draw a pretty picture by scratching it with a sharp object:



Question

If using Art Scratch Paper, which of these drawings will show exactly three colours?



Answer

Only the drawing of the sun and cloud touches exactly three colours.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Pattern Recognition

Concepts: Abstraction, Data Representation, Data Interpretation

When you work with computer software to create images, you often have to deal with layers that look like sheets of paper in a stack. The images in each layer can mix with one another. Paint and design programs have even more possibilities: they allow you to change the order of the layers, and they also allow you to take parts of the layer away. This is called transparency.

If you look at any website you will see many things on top of each other: the background image, colours, banners, images, text, etc. In order to display this correctly the computer has to know in which order things are put on top of each other. Computers call this the z-index; a number that says where a layer is located in the stack.



Hats for Snowmen

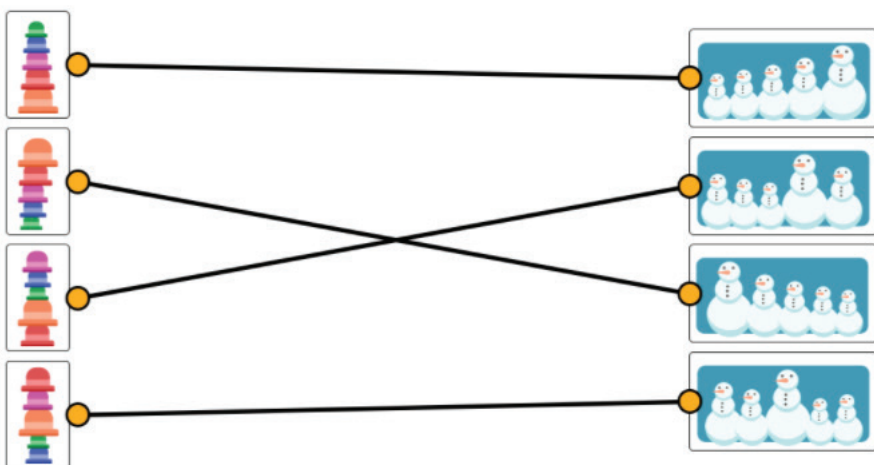
Five snowmen are standing in line. They all want a hat that fits them. The snowman on the left gets the hat on top of the pile. The next snowman gets the next hat in the pile. This continues until the snowman on the right gets the bottom hat.

Question

Connect the piles of hats with the correct line of snowmen so that every snowman gets a hat that fits him correctly.



Answer



The first pile of hats belongs to the first line of snowmen. The first line of snowmen is ordered from smallest to largest and so is the first pile of hats. The second pile of hats belongs to the third line of snowmen. The snowmen are in line from largest to smallest and so are the hats. The third pile of hats belongs to the second line of snowmen. Here, the first snowman is the third largest and the first hat is the third largest hat. The fourth pile of hats belongs to the fourth line of snowmen. The first snowman is the second largest and so is the top hat.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms, Evaluation

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

The hats are presented as a stack. This models a way that some computer data is stored - the abstract data structure (ADT) stack. With stacks you can only do two things or operations: take away the topmost element or put a new element on top of the stack.

Since we don't know how the line of snowmen was made, it could be that the line was created from the left or right. That is called a queue, which has the following operations: take away the front-most element or add a new element at the end of the queue.



Box of Marbles

A toy to teach simple programs consists of:

- A see-through tube which is closed on the left and open on the right.
- Seven different marbles.
- A simple programming system.



The program can only move one marble at a time.

Example

To put the  marble in between  and  above, this program could be written:



Question

If the toy looks like this:



Write a program to move the marbles into this order:



Answer

The correct answer is:



It's Computational Thinking

Computational Thinking Skills: Algorithms

Concepts: Specification, Algorithms, Implementation, Digital Systems, Impacts

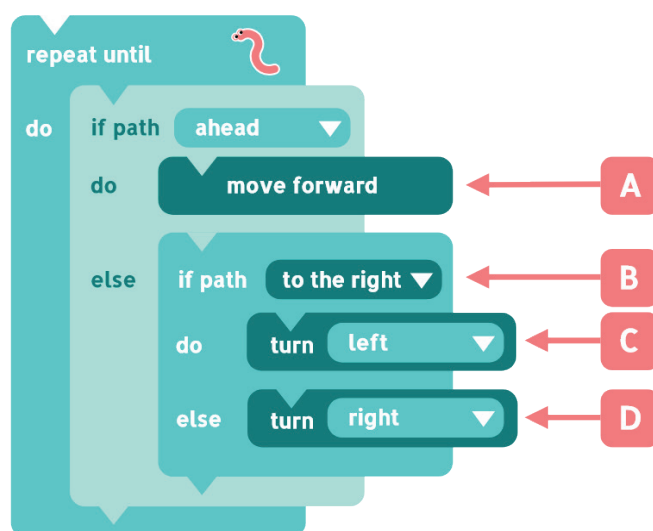
This question is related to a basic Data Structure in computer science called a stack. A stack is a linear structure represented by a pile.

The box in the question only has one opening, so we need to follow the rule 'first in, last out' (FILO) to put in/take out marbles. In our daily life, the stack is like a pile of books. When you remove a book from a pile, you need to remove all books above it so it doesn't come crashing down. In the same way, to remove marbles from the box, you must remove every marble that comes before the marble you want.

In Computer Science, a stack can also be used to convert infix expression to postfix expression. So it will be easier for a program to compute the answer.

Eddie's Lunch Maze

It's lunch time for Eddie the bird! The block code should take Eddie to the worm by the path that you see in the image below. But there is a mistake in the code!



Question

Which part of the block code contains a mistake?

A diagram showing four boxes labeled A, B, C, and D, arranged horizontally. Each box is a light blue rectangle with its label in the center.

Answer

In answer B, change **if path right** to **if path left**

The program needs to be run through to see where it is getting stuck. If you slow the program down and follow the movement along with the code you can identify when things go wrong.

It's Computational Thinking

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

Concepts: Abstraction, Specification, Algorithms

An algorithm is a sequence of instructions or set of rules used to achieve a task. Sometimes, when we need to make decisions we use IF THEN ELSE statements. These are known as branching statements.

We also use repeat loops to make our program more efficient.

In this question, we were looking for a mistake. This is called debugging, which is an important skill for computer programmers to develop.

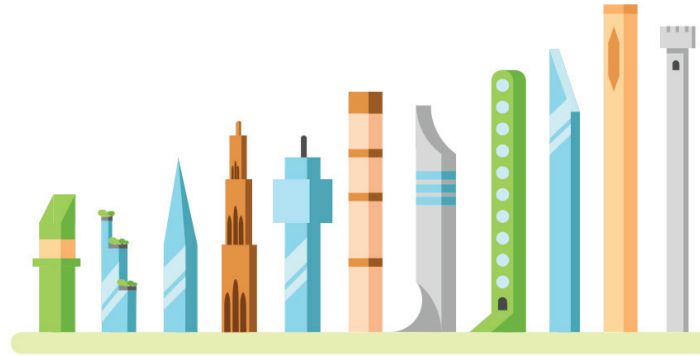


Well Placed Towers

Look at the towers shown below. A tower is 'well placed' if all the towers to the left of it are shorter, and all the towers to its right are taller.

Question

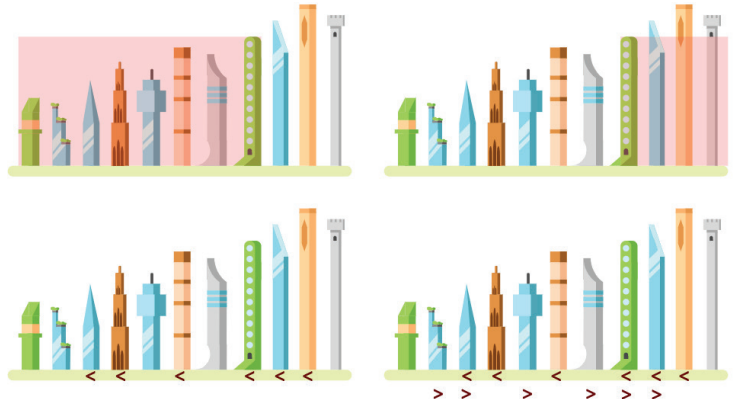
Select ALL of the 'well placed' towers below by clicking on them.



Answer

There are 3 towers that are 'well placed' towers. A tower is well placed if:

- All the tips of the towers to the left are within the coloured rectangle (see top left image).
- If all the tips of the towers to the right are outside the coloured rectangle (see top right image).
- This is shown for the eighth tower in the top images of the example below.



One possible way of finding the solution is then to move through all the towers, one by one, and mark a tower if all to the left of it are shorter (see bottom left image). Then make a similar second pass, marking a tower if all to the right are taller (see bottom right image). All towers marked twice are 'well placed' towers.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Algorithms

Concepts: Data Collection, Data Interpretation, Specification, Algorithms

This task focuses on the order of towers based on their height. In general, placing objects in an order is called sorting which is among the most well-known and studied problems in Computer Science. There are many different sorting algorithms. The quicksort algorithm is one of the most frequently used sorting methods because it tends to be very fast. The quicksort algorithm works as follows:

- A random element from the list is chosen. This element is called the 'pivot'.
- All elements smaller than the pivot are moved to the left side of the pivot and all elements greater than the pivot are moved to the right side of the pivot.
- Now there are two resulting lists, one to the left and one to the right and the same process is repeated for both of them recursively.
- After every step of the process, the resulting lists become smaller and smaller.
- This process ends if the resulting lists only contain 1 element, meaning that the whole list has then been sorted.

In our task, the special towers represent the pivots after moving the smaller elements to the left and the larger elements to the right.



Binary Bulbs

Beavers use binary bulbs to send messages. To send a message they use the following table and rules. The beavers sent the following message:

A	01000001	J	01001010	S	01010011
B	01000010	K	01001011	T	01010100
C	01000011	L	01001100	U	01010101
D	01000100	M	01001101	V	01010110
E	01000101	N	01001110	W	01010111
F	01000110	O	01001111	X	01011000
G	01000111	P	01010000	Y	01011001
H	01001000	Q	01010001	Z	01011010
I	01001001	R	01010010		



Rules:



Question

Which message did they send?

Happy

Honey

Horse

House

Answer

The correct answer is HOUSE.

Following the table and the above rules, we can conclude the following:



It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

Concepts: Abstraction, Data Interpretation, Specification, Algorithms, Digital Systems

Binary is the base two number system invented by Gottfried Leibniz that is made up of only two numbers: 0 and 1. This is the basis for all binary code, which is used to write data such as the instructions that computer processors use, or the digital text you read every day.

It's not difficult to find a binary system in our life. Bulbs and switches are good examples as you can see in this task. We can make codes and messages with this system, and can send messages and communicate with our friends. Morse code is another example.



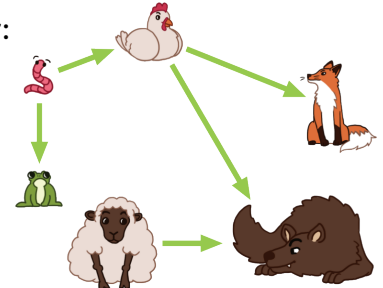
Animals

Six animals need to be put in six pens.

An animal cannot be in a pen that shares a fence with an animal that will eat it.

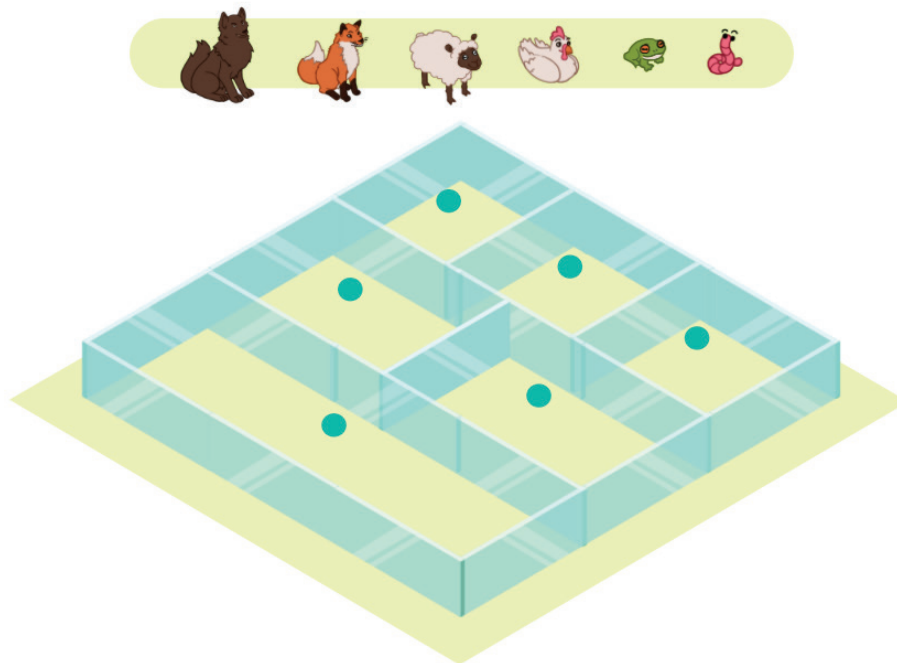
The arrows in the diagram on the right show which animals eat each other:

- The worm gets eaten by the frog or the chicken
- The sheep gets eaten by the wolf
- The chicken gets eaten by the fox or the wolf



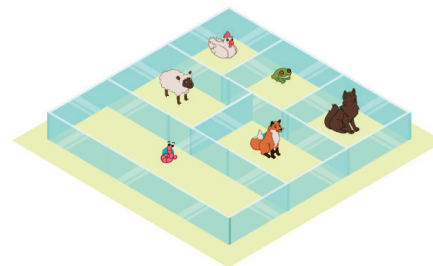
Question

Drag the animals below to the correct pen, so that none of the animals get eaten.



Answer

There are multiple correct answers. This is one:



It's Computational Thinking

Computational Thinking Skills: Algorithms, Evaluation

Concepts: Data Interpretation, Specification, Algorithms, Digital Systems, Interactions, Impacts

To solve this problem, we choose an animal who has several predators, for example the chicken who is hunted by both the wolf and fox. We check if the chicken is safe. We continue to test the animals with the most to least predators until we are sure each is safe.

In Informatics, this is called a 'constraint satisfaction problem'. The solution is one where a set of variables (all the animals) must follow a set of rules or 'constraints' (don't live beside a predator).



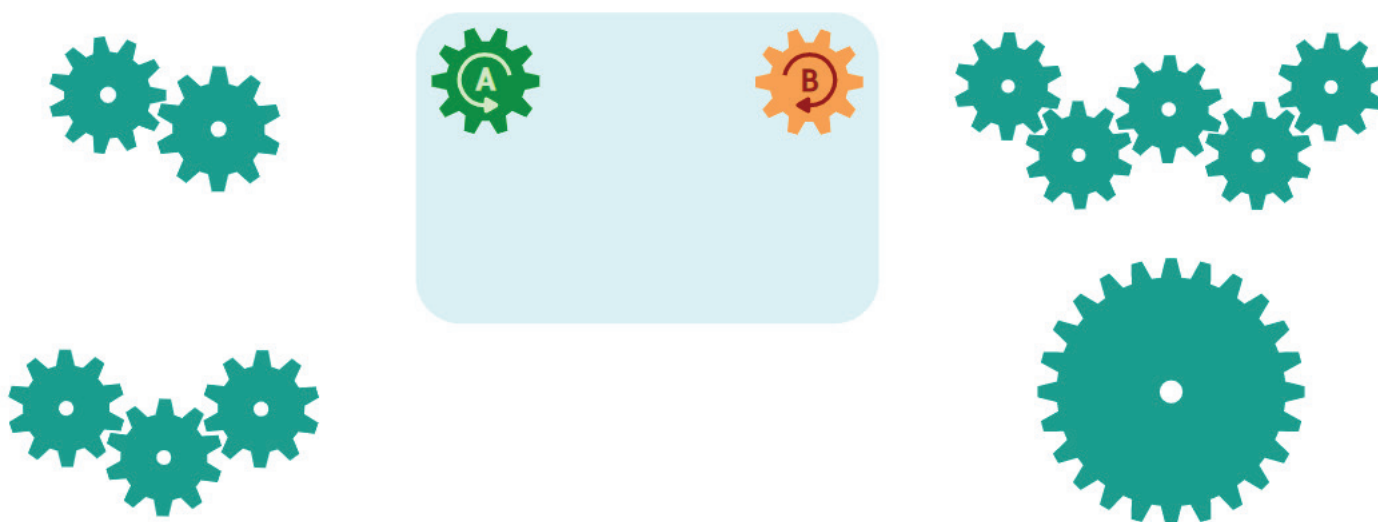
Gears in Motion

Look at the gears below.

When the green gear A turns clockwise as shown by the arrow, the orange gear B must turn in the opposite direction as shown by its arrow.

Question

Which set of gears should be inserted below to allow B to turn in the direction shown?



Answer

The block of two gears is required.

The way to solve this problem is to think about which way gear A turns the next one and so on. If you place the single big gear in between A and B, it will make B turn in the wrong direction. This will happen with all odd numbers of gears. So the only correct answer is placing two gears in between A and B.

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

It is common for computer programmers to make simulations of real-world machines. There are two ways we can use abstraction to make this problem easier to solve:

1. Re-draw the gears in a single line.
2. Realise that alternating gears turn in opposite directions.



Ancient Message

Lena discovered an ancient table of letters and symbols. This will help her decode the ancient message she found last year.

Lena now knows that letters have symbols! Each letter symbol is a combination of its column and row symbols.

Example

The letter H is made like this:

	I	II	III	IIII	○	⦿	⦿	⦿	⦿
⦿	A	B	C	D	E	F	G	H	I
⦿	J	K	L	M	N	O	P	Q	R
⦿	S	T	U	V	W	X	Y	Z	

	I	II	III	IIII	○	⦿	⦿	⦿	⦿
⦿	A	B	C	D	E	F	G	H	I
⦿	J	K	L	M	N	O	P	Q	R
⦿	S	T	U	V	W	X	Y	Z	

$$\text{⦿} + \text{⦿} = \text{⦿}$$

Lena can now decode this message:



Question

What is the message?

LOVEWATER

SLEEPDAYS

LOVEMYSUN

CAREFORME

Answer

LOVEWATER

At first, we check whether the length of the message gives us a hint. But all solutions are of the same length. Next we decode the first letter and find out that it is an "L". This tells us that it cannot be SLEEPDAYS nor CAREFORME. However answers LOVEWATER and LOVEMYSUN are still possible.

Then we look at the differences between LOVEWATER and LOVEMYSUN. On the fifth letter they start to differ. So we decode the fifth sign, and find that it is a "W". Therefore we know that solution LOVEWATER is correct.

To be certain that LOVEWATER is the correct solution, we decode another sign, for example the last sign. Here we find that it is a "R". This fits the expected solution.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

Data security is a big issue in society today. One of the methods to protect data from unauthorized people is to use secret ciphering. Cryptology started some 3500 years ago and one of the first methods was to replace each letter by another letter. In this example, new signs are created for the letters of the known alphabet, in such a way that one can easily remember the cipher system.

In this task, if there was no table to describe the encoding, and we were just given the symbols, the decoding of pictures into words would be much more difficult. Cryptanalysts, who try to break codes, use techniques like frequency analysis and pattern recognition in order to determine what encoded symbols should be.



Meal Planning

Raju lives in India. He takes 2 lunchboxes to school each day: one for the snack break, and the other for the lunch break.

He only has 30 minutes to prepare both boxes. Below you will find all the available items for the boxes and the preparation time each food needs:

		
Rice Preparation time: 15 min	Poha Preparation time: 20 min	Sandwich Preparation time: 15 min
		
Roti Preparation time: 20 min	Paratha Preparation time: 20 min	Bhel Preparation time: 10 min

Question

Which one of the following statements is **false**?

If Raju packs Roti for lunch, he cannot also pack a Sandwich for his snack.

If Raju packs Paratha for lunch, he can only pack Bhel for his snack.

If Raju packs Rice for lunch, he can pack Poha for his snack.

If Raju packs Rice for lunch, he can pack either a Sandwich or Bhel for his snack.

Answer

If Raju packs Rice for lunch, he can pack Poha for his snack.

Rice takes 15 minutes, and Poha takes 20 minutes, which makes it 35 minutes. So this statement is False. All the other statements have combinations which take 30 minutes or less, so they are true.

Raju has 30 minutes to pack both boxes. The correct combinations of dishes are those which take less than or at most 30 minutes to pack. As this is a multi-choice question there are only a few combinations to consider, so for each combination we have to add up the time required and compare it with the set 30 minutes.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

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

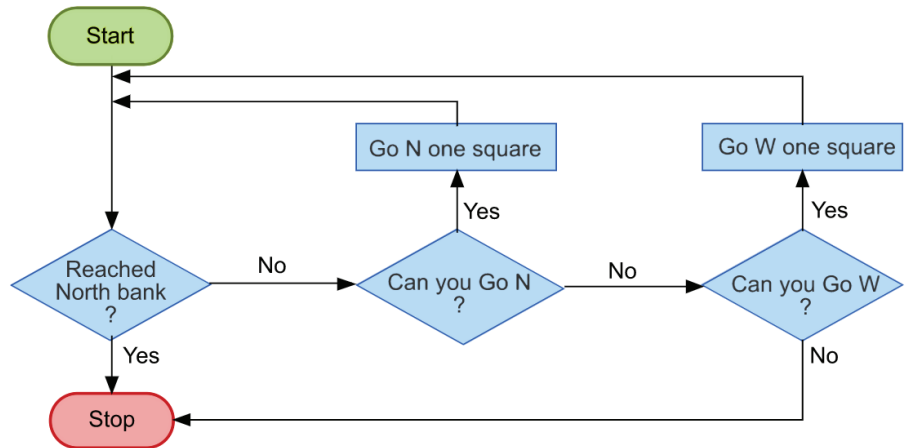
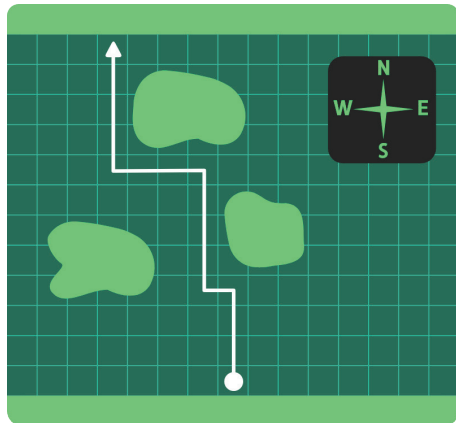
Constraints are the conditions which reduce the choices available in a situation. Solutions need to satisfy all the constraints.

In complex tasks, constraints may be combined by logical operators. In this problem, the constraint was that the total time for preparation is limited to 30 minutes.



Trap Island

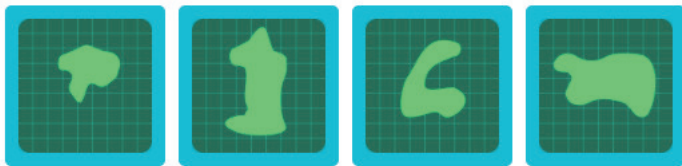
A robot boat goes from the south bank of the river to the north bank. It has to go around the islands in the river. The boat uses a digital map to find its way from one square to the next. The boat can only go to a square that is not blocked by any part of an island. The boat uses these instructions:



Pirates install an artificial island to trap the boat. Coming close to the island may stop the boat while it follows its instructions.

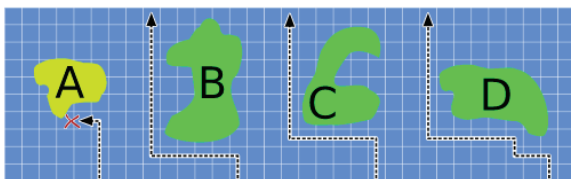
Question

Which of one of these islands could be a trap for the robot boat?



Answer

The correct answer is A below:



If the robot boat approaches the island on the left as shown in the image, it halts without reaching the northern bank. The reason is that it cannot go north or west. While the island can be approached from a different starting point, where the boat will not be trapped, it is impossible to be trapped in any of the other islands. At the southern coasts of these islands there is no square, with the northern and the western square not free. So, a robot boat approaching one of these islands from the south can always dodge to the west.

It's Computational Thinking

Computational Thinking Skills: Modelling and Simulation, Algorithms, Evaluation

Concepts: Data Interpretation, Specification, Algorithms Implementation, Digital Systems

The flowchart controlling the robot boat works fine with the existing round shaped islands. However, the flow chart did not take into account certain conditions that would lead to a halt at an island instead of the Northern river bank. Thus, there is a weakness in the flow chart, a security breach, which can be exploited by malicious people.

Computer programmers try to develop software that works fine under any condition. For example, an interactive website with input fields checks the input data carefully and refuses any input that might be dangerous for the software or the computer.

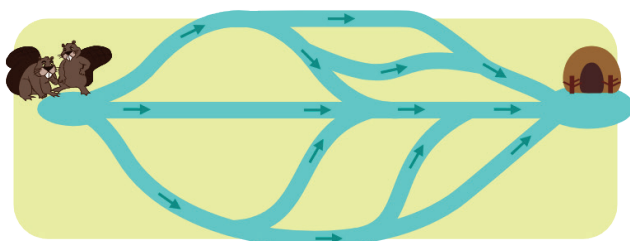
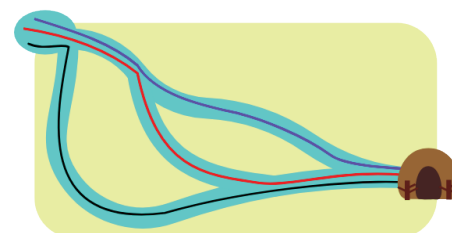


Many Beavers

Beavers are very ambitious: each beaver must have its own path to their home but are too lazy to swim upstream.

For example, in the river system pictured in the image below, we have at most 3 different paths from start to finish.

So, we can have at most 3 beavers living on this river.



There is a beaver couple living on a different river system shown on the picture below.

The beaver couple wants to have children and expand their family.

Question

What is the maximum number of kids they can have if each family member must have its own path to their home?

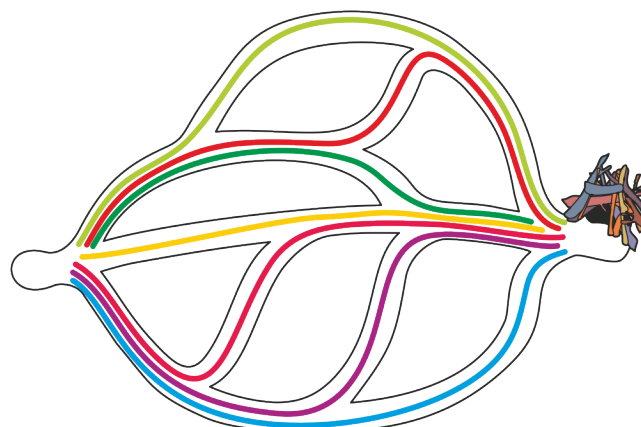
Answer

5

There are 7 different paths that lead to their home, 2 of which are reserved for the parents. We can count these paths in two different ways. One way is to look for all the paths in the picture and draw them in different colours.

A second way is to use a dynamic programming method. We look for all the junctions in the picture. We count how many paths lead to each junction and write the appropriate number on it.

To do it, we go from left to right and sum up the numbers on the junctions that are directly connected to the current junction by incoming arrows.



It's Computational Thinking

Computational Thinking Skills: Decomposition

Concepts: Data Representation, Specification, Algorithms, Implementation

Different tasks require different data representation. One of the ways is to use graphs. The water system in the picture can be represented as a directed graph. The edges represent the river sections and the nodes represent the junctions.

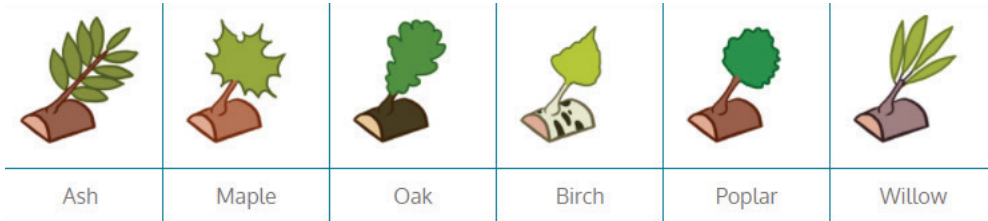
A directed graph is a convenient data structure that can help us find whether we can move from one node to another. We can write the length of each river section to the corresponding edge and get a weighted graph. We use weighted graphs to find the optimal path between starting and ending position.

The systematic approach in which we build our solution step by step using the previous steps is called dynamic programming. This approach is widely used to solve a lot of modern problems. You can apply dynamic programming only in the case where each next step depends only on the current situation and not on the way we reached it.



Wood Allergies

Some beavers are allergic to some types of wood and get really sick if they eat them. George is making wood snacks for a party and wants to make sure that everyone will have something to eat without getting sick. Each snack is made from one type of wood:



Name	Wood
Ann	Willow, Oak, Ash, Maple
Benjamin	Willow, Oak, Poplar
Cecil	Oak
Danny	Ash, Birch
Emma	Willow, Maple, Birch
Fred	Oak, Ash
George	Poplar, Maple

The beavers are happy to share snacks.

George has a list of the beavers attending the party and the types of wood that they can eat without getting sick.

To save time, George does not want to have to make snacks out of all six different types of wood, if possible.

Question

What are the fewest types of wood George can use to make snacks so that everyone can eat without getting sick?

1

2

3

4

5

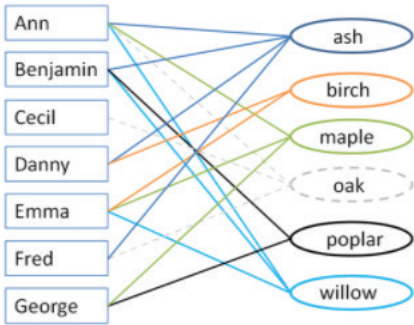
6

Answer

3

George will obviously have to make a dish from Oak for Cecil. This will also make Ann, Benjamin and Fred happy. There is no common dish for the remaining three beavers, so George will have to prepare at least two more. This will fortunately be enough. For example, George could make Oak, Ash and Maple.

This task is a classic example of a bipartite graph. Here, the first partition is the set of beavers, and the second partition is types of wood.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms, Evaluation
Concepts: Abstraction, Data Interpretation

This task is known in algorithm theory as the railway optimisation problem, in which the input is a schedule of trains and their stops and the goal is to find a set of train stations as small as possible so that every train visits at least one of the chosen stations.

Continued on next page



This question comes from
Slovenia

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



Wood Allergies - continued

It's Computational Thinking - continued

The problem is also known as a Dominating Set cover problem. Imagine six sets, each representing a different dish. Every set contains the beavers who like that dish. The task is to select the smallest number of sets (dishes) such that the union of these sets (all beavers who like any of the dishes represented by the chosen sets) contains all elements (beavers). In other words, the task is to cover all elements with the smallest number of sets.

The set cover problem is known as one of the hardest problems in computer science, a so-called NP-complete task. For these kinds of tasks, we do not know of any efficient algorithms that can be used to solve them. The only general solution is to try all different possibilities and even with a rather small number of sets the number of all possible combinations can be extremely large. With 240 dishes the number of different combinations are already about the same as the number of atoms in the entire universe!

Why have we been able to solve this task then? First, it was rather small. Second, it was made easier because Cecil had limited options. Finally, we used some simple logic that worked for this particular data, but wouldn't necessarily help us in general.



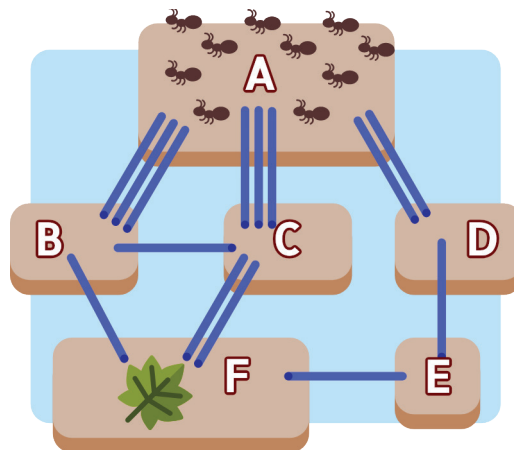
Ants in a Pond

Ten ants are on Stone A.

The ants want to get to the food on Stone F.

Only one ant can walk on a straw at a time.

It takes 1 minute for an ant to walk from one stone to another.



Question

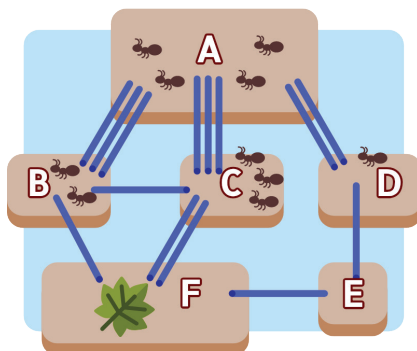
What is the maximum number of ants that can reach Stone F in 3 minutes?

Answer

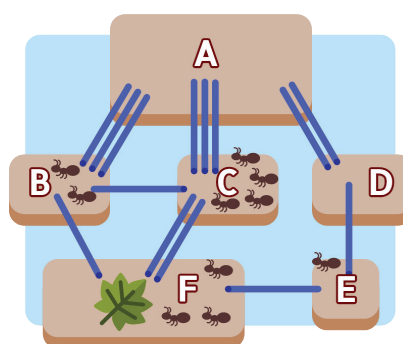
Correct answer is 7.

A possible situation after each minute is presented in the images below.

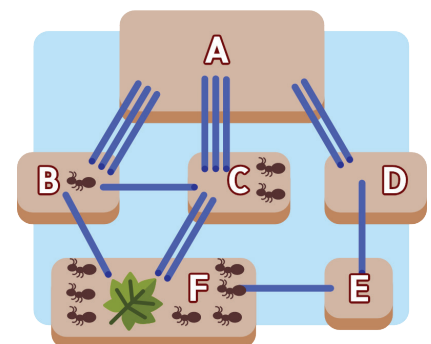
After one minute:



After two minutes:



After three minutes:



It's Computational Thinking

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

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

The goal here is to optimise the flow of the ants through the network so that as many ants as possible arrive at the food in 3 minutes time. This is called an optimization problem.

Ants that are not aware of the structure of the network they travel will not be able to find the best solution. But an observer, that can see the networks structure, can find an optimal strategy.

In this task, we assume that the ants are aware of the structure of the network, resulting in them choosing to move along a specific pathway.

Continued on next page



Ants in a Pond - continued

It's Computational Thinking - continued

Graphs are abstract data structures that are used to model networks and there are many existing algorithms to optimize the flow under specific circumstances. Alternatively, without making use of a pre-existing algorithm we can assert that:

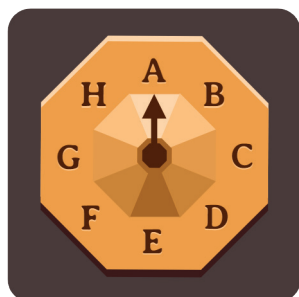
- It makes no sense to send more than one ant via the stones D-E
- It makes no sense to send more than two ants via the stones A-B
- The straw between B-C does not contribute to the flow and can be ignored
- The limiting factor of the flow are the straws B-F and C-F

In our case, these considerations enabled us to find the optimal solution.



Safe

A safe is unlocked using an octagonal (8-sided) knob. The knob has a pointer which can point to one of eight letters.



To unlock the safe, the pointer is first turned to point at A and then the password must be spelled out using the pointer.

The knob must be turned clockwise to the first letter, then counterclockwise to the next letter, clockwise to the third, and so on.

A password can be written down using a set of instructions.

Example

1↻ 2↻ means turn clockwise one letter and then counterclockwise two letters. These instructions spell BH.

Question

If the password is CHEFDG, which of the following instructions will unlock the safe?

2↻ 5↻ 5↻ 1↻ 6↻ 3↻

6↻ 3↻ 3↻ 7↻ 2↻ 5↻

2↻ 3↻ 5↻ 7↻ 6↻ 5↻

2↻ 1↻ 4↻ 3↻ 3↻ 2↻

Answer

The correct answer is:

and do not follow the instructions of alternating directions.

only works if the knob goes back to the starting position (letter A) after every letter, but it does not.

The safe lock only starts recording the letters spelled after the first turn of the knob.

The starting position is not included as part of the password.

It's Computational Thinking

Computational Thinking Skills: Pattern Recognition, Evaluation

Concepts: Data Representation, Data Interpretation, Specification, Algorithms

When working with an object, it is often needed to keep track of its position or state. The position of the arrow is part of the state of the safe lock in this task.



Safe - continued

It's Computational Thinking - continued

The state of an object can also include the history of the actions done on it. As the knob is turned, the lock must remember which letters of the password have already been spelled. The letters that have already been spelled form part of the state of the lock.

Doing the same action on an object may not always have the same effect. Actions done on an object can have different effects based on its state. For example, turning the knob to the letter G after spelling "CHEFD" will open the lock above. But turning it to the same letter after spelling only "CHE" will not.

This also applies to computers. For example, when you're drawing a picture, the parts that you have already drawn are the state of your picture. Adding new lines or deleting some lines change the state of the picture and the drawing program needs to keep track of that. Using the fill tool can change the colour of bigger or smaller parts of the picture depending on the lines already drawn.

Another example is when you're walking with your phone giving directions. Your location is the state that the navigation app on your phone has to keep track of to give you correct directions and to give a warning when you go in the wrong direction. The navigation app might also keep a history of places you frequently visit and suggest new places to go based on your history.

When writing a computer program, the programmer has to decide what the state of the system they are working with is and write the program to correctly keep and update that state. When the programmer gets this wrong, the program will not work correctly.

Notice that in this task, the starting position is always at letter A. Before we unlock the safe, the knob must always first be turned to the letter A. If it's not allowed to turn the knob eight or more times, then it is impossible to have a password starting with the letter A. This is because to spell the letter A, we must first turn the knob to a different letter, which means that the password spelled will start with that letter.

When designing systems, computer programmers must always be careful to avoid giving users not-so-nice restrictions

Bebras Challenge 2020 Round 1

Years 7+8



Library Break-in

A thief (or thieves) stole a large number of books from the school library and drove away with them.

The police identified three well-known suspects.

They were asked to come to the police station for questioning.

The following information was revealed during the questioning:

1. No one apart from Adam, Bob or Claire could have taken part in the theft.
2. Claire never participates in any activity unless Adam also does.
3. Bob cannot drive.

Question

Is Adam definitely guilty?

Yes, Adam is guilty.

No, Adam is not guilty.

It is impossible to tell.

Answer

Yes, Adam must be guilty.

First at least one out of Adam and Claire has to be proven guilty.

- If Bob is innocent, then either Adam or Claire must be the perpetrator.
- If Bob was involved, then he should have at least one accomplice since he can't drive. In this way either Adam or Claire must be guilty.

So, we can conclude that at least one of Adam and Claire is guilty.

- If Claire is innocent, Adam must be guilty.
- If Claire is guilty, then Adam must be guilty according to no.2 of the conditions above.

So Adam must be guilty.

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

A table like this is called a truth table. Drawing truth tables can help us figure out the validity of statements for complicated situations.

A	B	C	$A \vee B \vee C$	$C \Rightarrow A$	$B \Rightarrow (A \vee C)$
False	False	False	False	True	True
False	False	True	True	False	True
False	True	False	True	True	False
False	True	True	True	False	True
True	False	False	True	True	True
True	False	True	True	True	True
True	True	False	True	True	True
True	True	True	True	True	True



Meal Planning

Raju lives in India. He takes 2 lunchboxes to school each day: one for the snack break, and the other for the lunch break.

He only has 30 minutes to prepare both boxes. Below you will find all the available items for the boxes and the preparation time each food needs:

		
Rice Preparation time: 15 min	Poha Preparation time: 20 min	Sandwich Preparation time: 15 min
		
Roti Preparation time: 20 min	Paratha Preparation time: 20 min	Bhel Preparation time: 10 min

Question

Which one of the following statements is **false**?

If Raju packs Roti for lunch, he cannot also pack a Sandwich for his snack.

If Raju packs Paratha for lunch, he can only pack Bhel for his snack.

If Raju packs Rice for lunch, he can pack Poha for his snack.

If Raju packs Rice for lunch, he can pack either a Sandwich or Bhel for his snack.

Answer

If Raju packs Rice for lunch, he can pack Poha for his snack.

Rice takes 15 minutes, and Poha takes 20 minutes, which makes it 35 minutes. So this statement is False. All the other statements have combinations which take 30 minutes or less, so they are true.

Raju has 30 minutes to pack both boxes. The correct combinations of dishes are those which take less than or at most 30 minutes to pack. As this is a multi-choice question there are only a few combinations to consider, so for each combination we have to add up the time required and compare it with the set 30 minutes.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

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

Constraints are the conditions which reduce the choices available in a situation. Solutions need to satisfy all the constraints.

In complex tasks, constraints may be combined by logical operators. In this problem, the constraint was that the total time for preparation is limited to 30 minutes.



Wizard Bibraxus

In front of Wizard Bibraxus is a table with 49 coins. While his eyes are covered, you are allowed to flip a coin over.

Once his eyes are uncovered, Bibraxus immediately knows which coin has been flipped over! He tells you his secret:

"In every row and in every column there is an even number of coins with a star on top.

After flipping over a coin, there is exactly one row and one column with an odd number of coins showing the star on top.

Where this row and this column meet is the changed coin."

He would like for you to assist him on his next show.

Question

Which coin was flipped over?

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							

Answer

The correct answer is D6.

Column D contains five coins with a star symbol. Row 6 also contains five coins with a star symbol.

D6 is the coin, where row 6 and column D meet. All other columns and rows have an even number of coins where the star is visible.

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

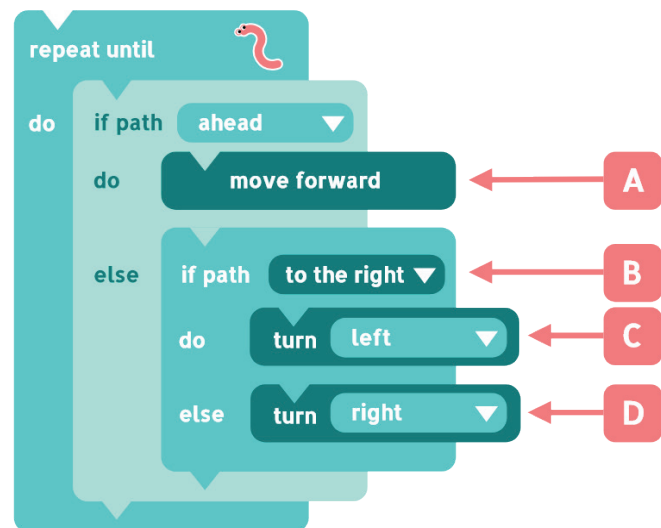
This magic trick has a special start position where changes can be easily detected. When images, text files, videos, etc. are sent over the internet or even from one part of a computer to another the information is actually sent as 0s and 1s. This is similar to this task where the pattern is represented by heads and tails. A similar system is used so that computers can check that the data has been transferred accurately.

Older students may be interested in how this works: The information in images is transferred as bytes where the first seven bits contain the information about the file and the eighth bit is chosen such that all the 1s in every byte is even. This is called even parity. Sometimes computers use odd-parity by using the eighth bit to ensure all bytes are odd (odd parity). When the file is received the computer runs a parity check on the data and if there are any bytes that are odd when they should be even, the data package is requested to be resent.



Eddie's Lunch Maze

It's lunch time for Eddie the bird! The block code should take Eddie to the worm by the path that you see in the image below. But there is a mistake in the code!



Question

Which part of the block code contains a mistake?

A

B

C

D

Answer

In answer B, change **if path right** to **if path left**.

The program needs to be run through to see where it is getting stuck. If you slow the program down and follow the movement along with the code you can identify when things go wrong.

It's Computational Thinking

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

Concepts: Abstraction, Specification, Algorithms

An algorithm is a sequence of instructions or set of rules used to achieve a task. Sometimes, when we need to make decisions we use IF THEN ELSE statements. These are known as branching statements.

We also use repeat loops to make our program more efficient.

In this question, we were looking for a mistake. This is called debugging, which is an important skill for computer programmers to develop.

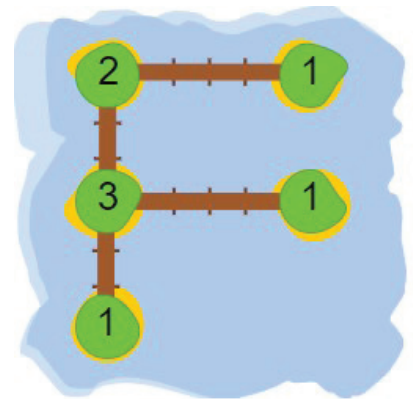


Bridges and Islands

To travel between islands a beaver needs to build bridges. The bridges must make it possible to travel from any island to any other island. The numbers in each circle tell the beavers how many bridges must connect to that island. Bridges can only be drawn horizontally or vertically.

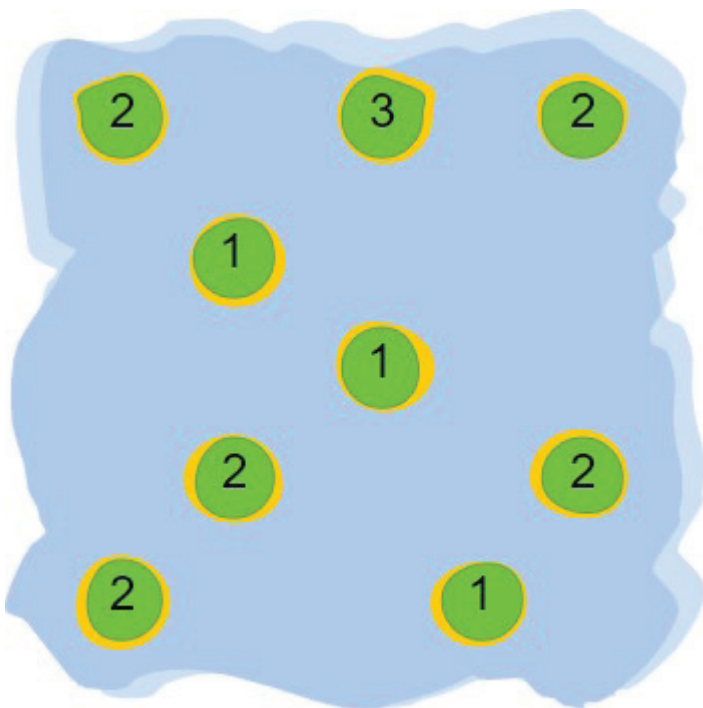
Example

Here is a bridge system that follows these rules:



Question

How many bridges need to be built, given the map below?



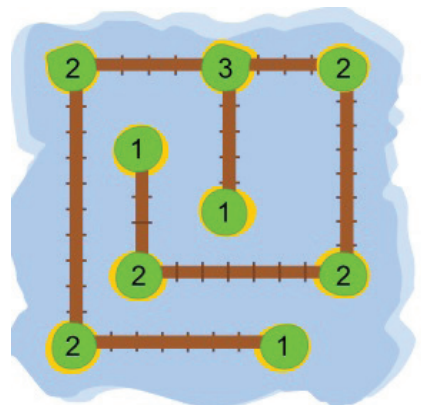
Answer

In any completed bridges system, each bridge would be counted on the two islands it connects. So to complete the given map, the number of bridges should be half of the total of all the numbers in the map, which is $(2+3+2+1+1+2+2+2+1)/2=8$. To complete the answer, it must be shown that a complete bridges system is possible for the given map.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

Concepts: Abstraction, Data Representation, Specification, Algorithms



In graph theory, a branch of mathematics, objects represented by points are connected by lines to signify pairwise relation. The degree of a vertex is the number of edges incident to the vertex, similar to the number of bridges connected to an island, and the sum of all the degrees in a graph is equal to twice the number of edges, since each edge has two endpoints.

This can be extended to the handshaking lemma which states that every finite undirected graph has an even number of vertices with odd degree (the number of edges touching the vertex). In more colloquial terms, in a party of people some of whom shake hands, an even number of people must have shaken an odd number of other people's hands.



Ancient Message

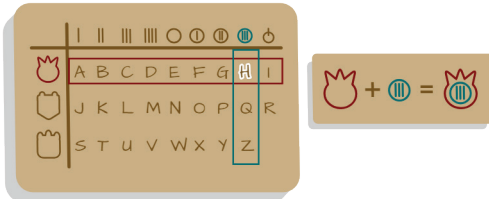
Lena discovered an ancient table of letters and symbols. This will help her decode the ancient message she found last year.

Lena now knows that letters have symbols! Each letter symbol is a combination of its column and row symbols.

Example

The letter H is made like this:

	I	II	III	IIII	O	⓪	Ⓛ	Ⓜ	Ⓝ
Ⓜ	A	B	C	D	E	F	G	H	I
Ⓛ	J	K	L	M	N	O	P	Q	R
⓪	S	T	U	V	W	X	Y	Z	



Lena can now decode this message:



Question

What is the message?

LOVEWATER

SLEEPDAYS

LOVEMYSUN

CAREFORME

Answer

LOVEWATER

At first, we check whether the length of the message gives us a hint. But all solutions are of the same length. Next we decode the first letter and find out that it is an "L". This tells us that it cannot be SLEEPDAYS nor CAREFORME. However answers LOVEWATER and LOVEMYSUN are still possible.

Then we look at the differences between LOVEWATER and LOVEMYSUN. On the fifth letter they start to differ. So we decode the fifth sign and find that it is a "W". Therefore we know that solution LOVEWATER is correct.

To be certain that LOVEWATER is the correct solution, we decode another sign, for example the last sign. Here we find that it is a "R". This fits the expected solution.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

Data security is a big issue in society today. One of the methods to protect data from unauthorized people is to use secret ciphering. Cryptology started some 3500 years ago and one of the first methods was to replace each letter by another letter. In this example, new signs are created for the letters of the known alphabet, in such a way that one can easily remember the cipher system.

In this task, if there was no table to describe the encoding, and we were just given the symbols, the decoding of pictures into words would be much more difficult. Cryptanalysts, who try to break codes, use techniques like frequency analysis and pattern recognition in order to determine what encoded symbols should be.



Seating Arrangement

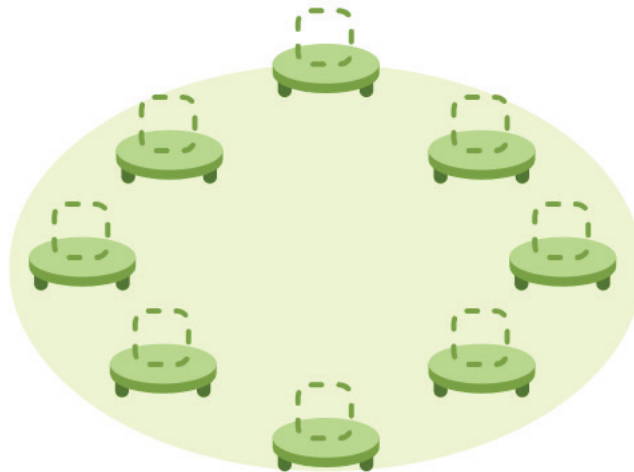
Eight friends are sitting in a circle. They are all facing inwards. We know the following facts about where they are sitting:

1. Alice is sitting directly opposite of David.
2. Henry is sitting between Greta and Eugene.
3. Franny is not next to Alice or David.
4. There is one person between Greta and Claire.
5. Eugene is sitting immediately to David's left.

Question

Place the friends in their correct places in the circle by dragging the letter next to their name to their chair.

There may be multiple correct solutions; you only need to find one.

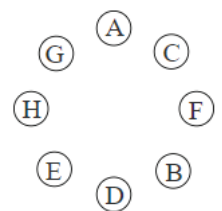


A	Alice
B	Bruce
C	Claire
D	David
E	Eugene
F	Franny
G	Greta
H	Henry

Answer

The following answer assumes that Alice is placed in the top seat. There are in fact 8 different answers, as an answer can be found corresponding to Alice being in any of the seats.

- Fact 1, Alice sits directly opposite to David, enables us to seat David.
- Now, Fact 5, Eugene is beside David, on David's left, enables us to seat Eugene.
- At this stage, Fact 2, Henry sits between Greta and Eugene, tells us where Henry sits.
- Knowing where Henry sits we now can place Greta.
- With five friends placed, Fact 3, Franny is not beside Alice or David, leaves only one space for Franny.
- Fact 4, there is one person between Greta and Claire, now tells us where Claire sits.
- Finally there is only one seat and one friend to place so we can place Bruce.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Algorithms, Evaluation

Concepts: Data Interpretation, Specification, Algorithms, Implementation

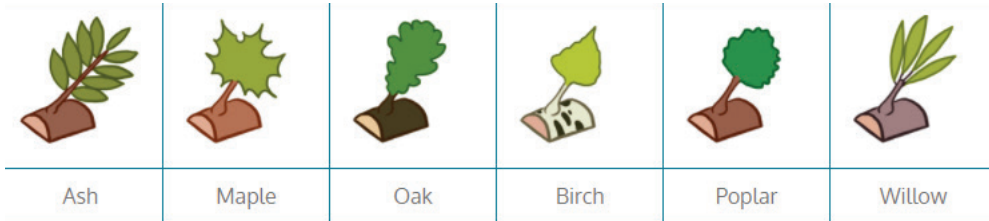
Logic is about following rules, understanding ordering and negation. For example, the key logic operator used in this question is negation, which is used whenever we say NOT. So, the fact that states that 'Franny is not beside Alice or David' can be stated differently as 'Franny not beside (Alice or David)' which is the same as '(Franny not beside Alice) and (Franny not beside David)'.

The negation operator, not, changes the logical expression of not (A or B) to an equivalent (not A) and (not B) which is an example of De Morgan's law. Knowing how logical expressions can be rewritten, combined, or simplified is a very useful skill in computer science.



Wood Allergies

Some beavers are allergic to some types of wood and get really sick if they eat them. George is making wood snacks for a party and wants to make sure that everyone will have something to eat without getting sick. Each snack is made from one type of wood:



Name	Wood
Ann	Willow, Oak, Ash, Maple
Benjamin	Willow, Oak, Poplar
Cecil	Oak
Danny	Ash, Birch
Emma	Willow, Maple, Birch
Fred	Oak, Ash
George	Poplar, Maple

The beavers are happy to share snacks.

George has a list of the beavers attending the party and the types of wood that they can eat without getting sick.

To save time, George does not want to have to make snacks out of all six different types of wood, if possible.

Question

What are the fewest types of wood George can use to make snacks so that everyone can eat without getting sick?

1

2

3

4

5

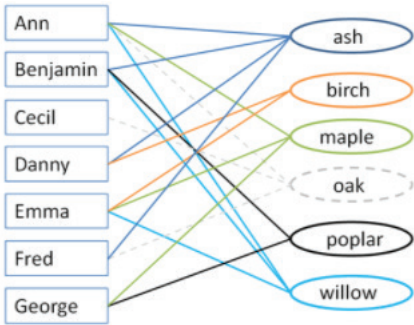
6

Answer

3

George will obviously have to make a dish from Oak for Cecil. This will also make Ann, Benjamin and Fred happy. There is no common dish for the remaining three beavers, so George will have to prepare at least two more. This will fortunately be enough. For example, George could make Oak, Ash and Maple.

This task is a classic example of a bipartite graph. Here, the first partition is the set of beavers, and the second partition is types of wood.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms, Evaluation
Concepts: Abstraction, Data Interpretation

This task is known in algorithm theory as the railway optimisation problem, in which the input is a schedule of trains and their stops and the goal is to find a set of train stations as small as possible so that every train visits at least one of the chosen stations.

Continued on next page



This question comes from
Slovenia

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



Wood Allergies - continued

It's Computational Thinking - continued

The problem is also known as a Dominating Set cover problem. Imagine six sets, each representing a different dish. Every set contains the beavers who like that dish. The task is to select the smallest number of sets (dishes) such that the union of these sets (all beavers who like any of the dishes represented by the chosen sets) contains all elements (beavers). In other words, the task is to cover all elements with the smallest number of sets.

The set cover problem is known as one of the hardest problems in computer science, a so-called NP-complete task. For these kinds of tasks, we do not know of any efficient algorithms that can be used to solve them. The only general solution is to try all different possibilities and even with a rather small number of sets the number of all possible combinations can be extremely large. With 240 dishes the number of different combinations are already about the same as the number of atoms in the entire universe!

Why have we been able to solve this task then? First, it was rather small. Second, it was made easier because Cecil had limited options. Finally, we used some simple logic that worked for this particular data, but wouldn't necessarily help us in general.



Bottle Recycling

Three machines can make new bottles from old plastic bottles as shown below:



Makes a large white bottle if two white bottles are inserted. Any other combination will make a green bottle.



Makes a large green bottle if two green bottles are inserted. Any other combination will produce a white bottle.

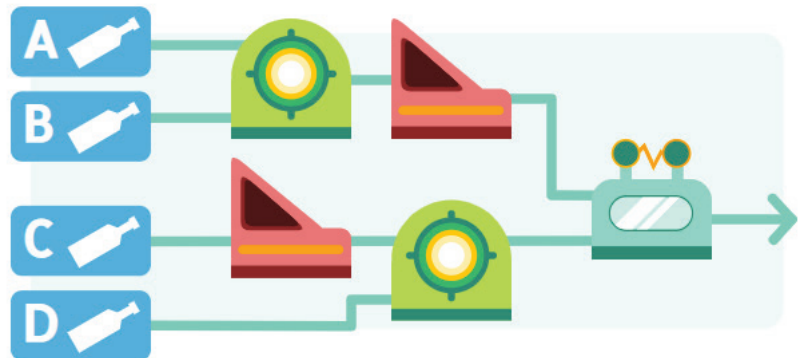


Turns a green bottle into a white bottle or a white bottle into a green one.

Question

What kind of plastic bottles should A, B, C and D be to get one new white bottle out of the system?

There may be multiple correct solutions; you only need to find one.



Answer

There are three possible correct answers that produce the desired outcome here:

A = green, B = green, C = green, D = green

A = green, B = green, C = green, D = white

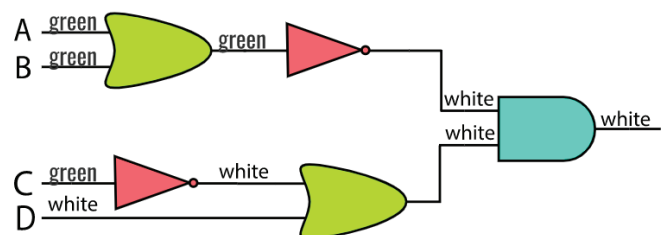
A = green, B = green, C = white, D = white

In the following diagram the machines have been replaced with the logic gates they represent. The colours of the machines have remained the same. The easiest way of checking your answer is to write the colour of bottles above the lines coming out of each machine starting from the left. This is also shown on the diagram below for the correct answer A = green, B = green, C = green, D = white.

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms



All computers contain circuits, that consist of different types of small elements called gates. Some of the most popular gates are NOT, OR, AND, XOR.

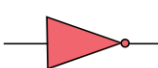
We have used AND, OR and NOT gates here. Their graphical representation in the answer is the same as used in engineering. In engineering the elements work using electrical signals. We note them as 1 if there is a signal, and 0 if there is no signal. In logic we refer to signal 1 as TRUE and to signal 0 as FALSE. In this task we consider white plastic as TRUE (or 1) and dark green plastic as FALSE (or 0).



AND gate requires both inputs to be TRUE (white bottle) to get TRUE as the output.



OR gate requires at least one of the inputs to be TRUE to get TRUE as the output.



NOT gate reverses values, so if the input is TRUE, the output is FALSE, and if the input is FALSE, the output is TRUE.

Logic gates are used, among other things, in microprocessor architecture in order to perform arithmetic and logic operations.



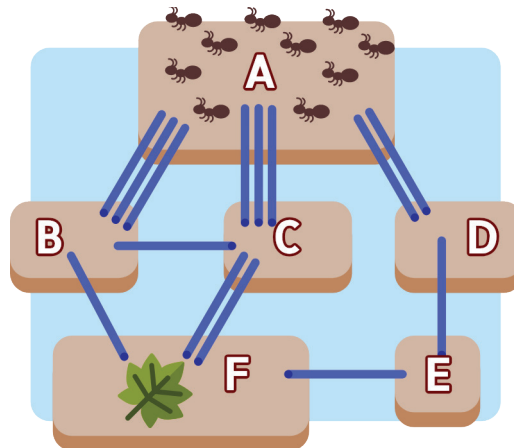
Ants in a Pond

Ten ants are on Stone A.

The ants want to get to the food on Stone F.

Only one ant can walk on a straw at a time.

It takes 1 minute for an ant to walk from one stone to another.



Question

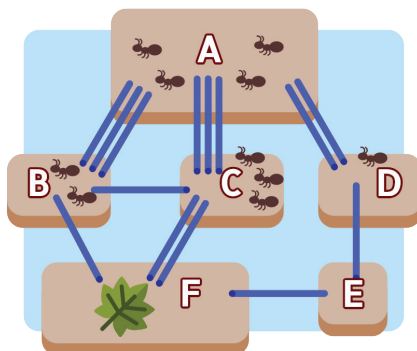
What is the maximum number of ants that can reach Stone F in 3 minutes?

Answer

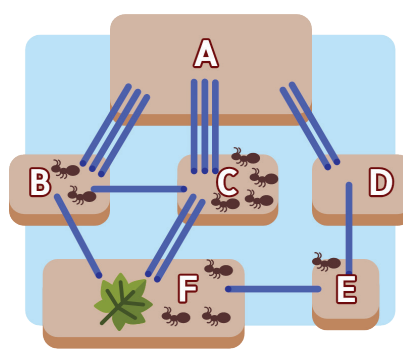
Correct answer is 7.

A possible situation after each minute is presented in the images below.

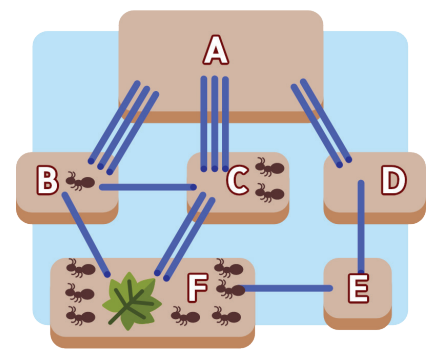
After one minute:



After two minutes:



After three minutes:



It's Computational Thinking

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

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

The goal here is to optimise the flow of the ants through the network so that as many ants as possible arrive at the food in 3 minutes time. This is called an optimization problem.

Ants that are not aware of the structure of the network they travel will not be able to find the best solution. But an observer, that can see the networks structure, can find an optimal strategy.

In this task, we assume that the ants are aware of the structure of the network, resulting in them choosing to move along a specific pathway.



Ants in a Pond - continued

It's Computational Thinking - continued

Graphs are abstract data structures that are used to model networks and there are many existing algorithms to optimize the flow under specific circumstances. Alternatively, without making use of a pre-existing algorithm we can assert that:

- It makes no sense to send more than one ant via the stones D-E
- It makes no sense to send more than two ants via the stones A-B
- The straw between B-C does not contribute to the flow and can be ignored
- The limiting factor of the flow are the straws B-F and C-F

In our case, these considerations enabled us to find the optimal solution.

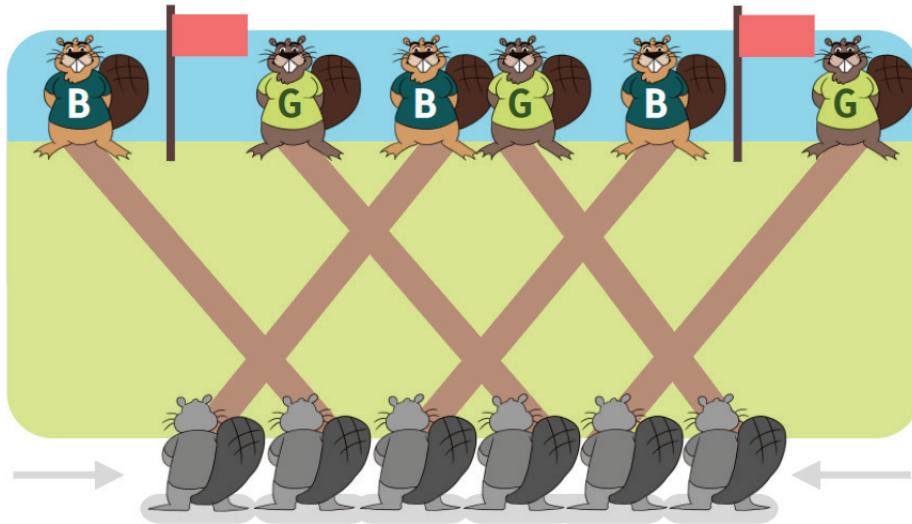


Tunnels

The beavers have built a network of underground tunnels with six entrances and six exits. Exactly one beaver may enter at each entrance.

There are two teams of beavers: green and blue. If two beavers meet at an intersection and they are of two different colours, the blue beaver will go to the right while the green beaver will go to the left.

If they are from the same coloured team, they continue on their path. Six beavers enter the network of passages at the same time and travel at the same speed.



Question

The following order of beavers exited the network: BGBGBG.

Click on the beavers entering the network to provide the correct blue 'B' and green 'G' order in which they entered. There may be multiple correct solutions; you only need to find one.

Answer

There are two possible correct answers for this question: BBGBGG or BBBGGG.

Any other answer would not provide the exit order needed.

In order to have a B at the left-most exit, it is necessary for the two left entries to be BB.
Any other combination would produce G at the most left exit.

In order to have a G at the right-most exit, it is necessary for the two right entries to be GG.
Any other combination would produce B at the most right exit.

For the two middle entrances we are left with only two combinations: BG or GB.
It's easy to check that both are correct.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

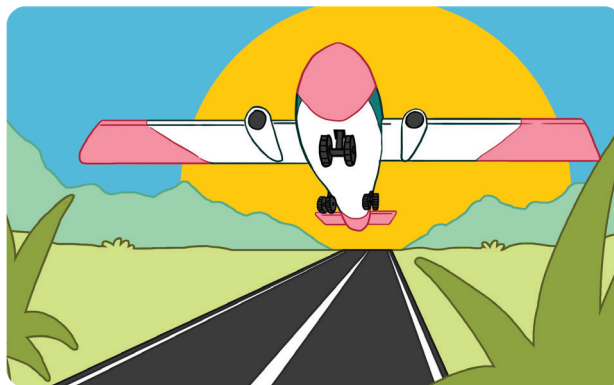
Concepts: Abstraction, Specification, Algorithms

This task is about beavers moving through a network according to some rules. This corresponds with the flow of data through computer networks. This is called IP routing. IP routing is the process of moving data packets between different networks. By default two different IP networks cannot communicate with each other. They need a mediator device that can switch packet between them. A router takes care of this. Router interfaces are associated with different networks. This association is kept in a routing table. Routers use these tables to make switching decisions.

Aircraft Scheduling

When aircraft land or take off from an airport they are assigned corridors to avoid accidents. These are designated airspaces which separate each aircraft.

At the Bebrasland airport, aircraft cannot have the same corridor if they are landing or taking off within 15 minutes.



Example

Flight #1 at 6:10 AM and Flight #2 at 6:25 AM cannot be assigned the same corridor whereas Flight #3 at 6:26 am can be assigned the same corridor as Flight #1.

You are the Air Traffic Controller at the airport today and it is your job to ensure that you schedule flights which are within 15 minutes of each other to different corridors.

A timetable is provided with arrival and departure times for flights.

Flight	Arrivals	Flight	Departures
9W2400	7:00	EK427	7:03
SG147	7:12	AI561	7:20
AI620	7:18	9W1321	7:21

Question

What is the minimum number of corridors needed to ensure that all the above flights land and take off safely?

Answer

The correct answer is 4.

Let us list flights and flight times in ascending order of landing time:

1. 9W2400 7:00
2. EK427 7:03
3. SG147 7:12
4. AI620 7:18
5. AI561 7:20
6. 9W1321 7:21

The first flight 9W2400 is at 7:00 so we assign it to corridor 1. The next two flights EK427 and SG147 are within 15 minutes of the first flight and the 3rd flight is within 15 minutes of 2nd flight as well, so clearly, all three need to be assigned separate corridors. So we need 3 corridors so far.

For each of the following flights, we try to assign it to already used corridors if possible. The 4th flight AI620 at 7:18 is more than 15 minutes away from the first flight 9W2400 and can be assigned the same corridor as flight 9W2400. Similarly, the 5th flight AI561 cannot be assigned corridor 1 as it is within 15 minutes of AI620 but can be assigned corridor 2 as it is more than 15 minutes away from EK427.

Continued on next page



Aircraft Scheduling - continued

Answer - continued

The 6th flight 9W1321 at 7:21 is less than 15 minutes from flights SG147 (7:12), AI620 (7:18), and AI561 (7:20) that are assigned to the 3 corridors so far. So we must assign another corridor for this flight. Thus giving us the following flight assignments to corridors:

- Corridor 1: 9W2400 (7:00), AI620 (7:18)
- Corridor 2: EK427 (7:03), AI561 (7:20)
- Corridor 3: SG147 (7:12)
- Corridor 4: 9W1321 (7:21)

While we can assign more than 4 corridors, it is clear that less than 4 may result in accidents.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms, Evaluation

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

In the above problem, you have conflicts (two aircraft cannot use the same corridor if landing within 15 minutes of each other) and resources (corridors) to manage. There are many problems of this type and in informatics special approaches have been developed to solve such problems.

One of them is to use a graph to represent it. A graph has nodes and edges that connect nodes. In this case, the nodes represent the objects (flights) and the edges the conflicts: if two objects are in conflict with each other, their nodes are connected by an edge, otherwise, they aren't. The problem of the task can be considered as a graph colouring problem: what is the minimum number of colours we need to colour each node in the graph in such a way that there are no same coloured nodes connected by an edge.

When applied to the aircraft scheduling problem, each colour will represent a corridor. Graph colouring has applications like scheduling aircraft to flight routes, bandwidth allocation to radio stations, pattern matching, sports scheduling, designing seating plans, exam timetabling etc.

There are many possible ways to solve this problem, the difference is in the data structures (how data is represented) used and the time needed to arrive at the solution. One of such methods uses a greedy algorithm. The word 'greedy' means that we try to sort all objects in descending order of the 'Price' we assign to it using some common sense. In this case, we use time of landing as the Price (the earlier the flight the higher the Price). Greedy algorithms don't always give us the right answers, but most of them provide us with suitable solutions.

Here is the solution using a greedy graph colouring algorithm. Flights have a relationship if they are within 15 minutes of each other. We start with flight 9W2400. We connect it to all the flights that are related i.e. less than 15 minutes away – EK427 and SG147. We repeat this for every flight making a new connection with another flight within 15 minutes if a connection does not already exist.

Let us assign the flight corridors, say colour red, to unconnected flights. 9W2400 gets assigned flight corridor 1 as it is about to land. All flights not connected to this get same flight corridor – colour red as long as they are not connected to a flight in corridor 1 marked as red. While flights AI561 and 9W1321 are not connected to 9W2400, they cannot be coloured red as they are connected to flight AI620 which is red.

The next flight corridor is 2, let's say green. Assign it to EK427 and to any flight(s) not connected to it which is AI561. 9W1321 cannot get coloured green as it is connected to AI561. Assign the third flight corridor, yellow, to SG147 and the fourth flight corridor, blue, to 9W1321. It is clear from the above that we need no more than 4 corridors for the above problem to ensure that all flights can land safely.



Beaver Ball Game

The beavers are playing a ball game where they throw the ball around in a circle, skipping the beaver next to them. The beaver between the thrower and the catcher is out of the game. The text below describes how the game works for 5 beavers. You can press on the play button in the image above to see it visually.

Beaver 1 always starts. He throws the ball to Beaver 3. Beaver 2 is now out of the game.
Beaver 3 throws the ball to Beaver 5. Beaver 4 is now out of the game.
Beaver 5 throws the ball to Beaver 3. Beaver 1 is now out of the game.
Beaver 3 throws the ball around the whole circle back to herself.
Beaver 5 is now out of the game. Since Beaver 3 is the last beaver standing, she wins the game!

Question

If 34 beavers play this game, and they start again with beaver 1, which beaver will win the game?



Answer

Beaver 5 will win the game.

It's Computational Thinking

Computational Thinking Skills: Modelling and Simulation, Algorithms

Concepts: Data Interpretation, Specification, Algorithms

This is a classical theoretical problem about counting out participants in a game. This game is known as the Josephus problem.

If you manually work out this game starting with a low number of beavers, you will quickly identify the property that for an even number beavers the same beaver will hold the boomerang. This way of halving the problem is a nice way of getting to a solution.

Even though solving the Josephus problem for all cases is more of a mathematical problem than a computational thinking problem, the hint of using a number of participants that is slightly above a power of two, and the fact that we are only asking for a single value, makes this more of a thinking problem about halving numbers.

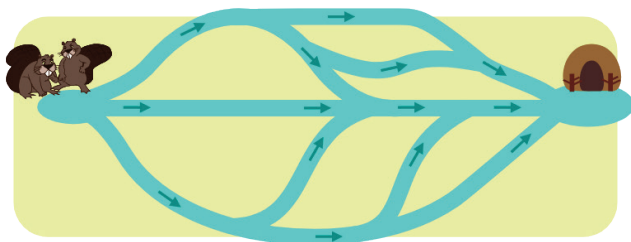
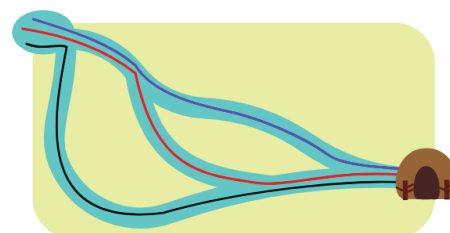


Many Beavers

Beavers are very ambitious: each beaver must have its own path to their home but are too lazy to swim upstream.

For example, in the river system pictured in the image below, we have at most 3 different paths from start to finish.

So, we can have at most 3 beavers living on this river.



There is a beaver couple living on a different river system shown on the picture below.

The beaver couple wants to have children and expand their family.

Question

What is the maximum number of kids they can have if each family member must have its own path to their home?

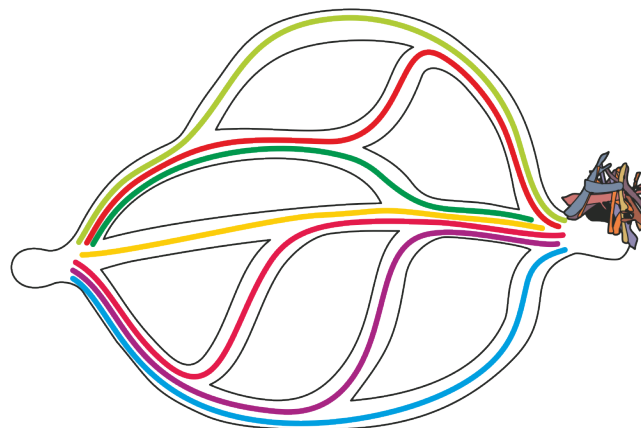
Answer

5

There are 7 different paths that lead to their home, 2 of which are reserved for the parents. We can count these paths in two different ways. One way is to look for all the paths in the picture and draw them in different colours.

A second way is to use a dynamic programming method. We look for all the junctions in the picture. We count how many paths lead to each junction and write the appropriate number on it.

To do it, we go from left to right and sum up the numbers on the junctions that are directly connected to the current junction by incoming arrows.



It's Computational Thinking

Computational Thinking Skills: Decomposition

Concepts: Data Representation, Specification, Algorithms, Implementation

Different tasks require different data representation. One of the ways is to use graphs. The water system in the picture can be represented as a directed graph. The edges represent the river sections and the nodes represent the junctions.

A directed graph is a convenient data structure that can help us find whether we can move from one node to another. We can write the length of each river section to the corresponding edge and get a weighted graph. We use weighted graphs to find the optimal path between starting and ending position.

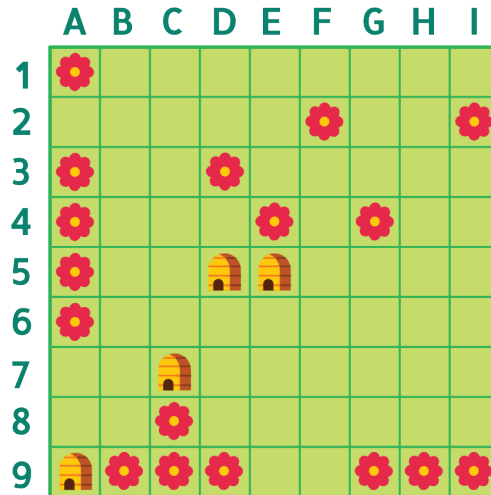
The systematic approach in which we build our solution step by step using the previous steps is called dynamic programming. This approach is widely used to solve a lot of modern problems. You can apply dynamic programming only in the case where each next step depends only on the current situation and not on the way we reached it.



Bee Hive

A beekeeper has a hive with bees. He wants to place the hive so that the sum of distances from the hive to every flower is minimum.

The field with flowers is represented in the following grid, with rows from 1 to 9 and columns from A to I.



The bees only fly horizontally and vertically on this field, so the distance between two tiles is the sum of the horizontal distance and the vertical distance.

For example, the distance between C4 and D7 is 4 (3 tiles vertically plus 1 tile horizontally).

Question

Where should the beekeeper place the hive to that the sum of the distance from the hive to every flower is minimum?

(Possible locations are marked with  on the map.)

D5

C7

E5

A9

Answer

The answer is D5.

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

Bebras Challenge 2020 Round 1

Years 9+10



Ancient Message

Lena discovered an ancient table of letters and symbols. This will help her decode the ancient message she found last year.

Lena now knows that letters have symbols! Each letter symbol is a combination of its column and row symbols.

Example

The letter H is made like this:

	I	II	III	IIII	○	⦿	⦿	⦿	⦿
⦿	A	B	C	D	E	F	G	H	I
⦿	J	K	L	M	N	O	P	Q	R
⦿	S	T	U	V	W	X	Y	Z	

	I	II	III	IIII	○	⦿	⦿	⦿	⦿
⦿	A	B	C	D	E	F	G	H	I
⦿	J	K	L	M	N	O	P	Q	R
⦿	S	T	U	V	W	X	Y	Z	

$$\text{⦿} + \text{⦿} = \text{⦿}$$

Lena can now decode this message:



Question

What is the message?

LOVEWATER

SLEEPDAYS

LOVEMYSUN

CAREFORME

Answer

LOVEWATER

At first, we check whether the length of the message gives us a hint. But all solutions are of the same length. Next we decode the first letter and find out that it is an "L". This tells us that it cannot be SLEEPDAYS nor CAREFORME. However answers LOVEWATER and LOVEMYSUN are still possible.

Then we look at the differences between LOVEWATER and LOVEMYSUN. On the fifth letter they start to differ. So we decode the fifth sign, and find that it is a "W". Therefore we know that solution LOVEWATER is correct.

To be certain that LOVEWATER is the correct solution, we decode another sign, for example the last sign. Here we find that it is a "R". This fits the expected solution.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

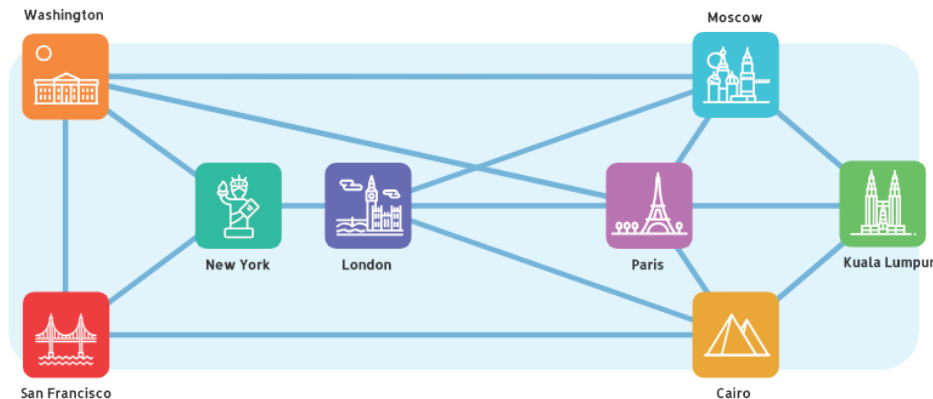
Data security is a big issue in society today. One of the methods to protect data from unauthorized people is to use secret ciphering. Cryptology started some 3500 years ago and one of the first methods was to replace each letter by another letter. In this example, new signs are created for the letters of the known alphabet, in such a way that one can easily remember the cipher system.

In this task, if there was no table to describe the encoding, and we were just given the symbols, the decoding of pictures into words would be much more difficult. Cryptanalysts, who try to break codes, use techniques like frequency analysis and pattern recognition in order to determine what encoded symbols should be.



Greener Flight Routes

Bebras International Airlines has a lot of flight routes connecting several major cities as shown:



In order to reduce CO2 emissions, the airline wants to cancel some of the flight routes without stopping customers from being able to fly to any city.

Example

If the flight route between San Francisco and Washington, D.C. is cancelled, customers could still fly from San Francisco to New York and then from New York to Washington, D.C.

Question

For the flight routes shown above, what is the maximum number of routes the airline can cancel?

6

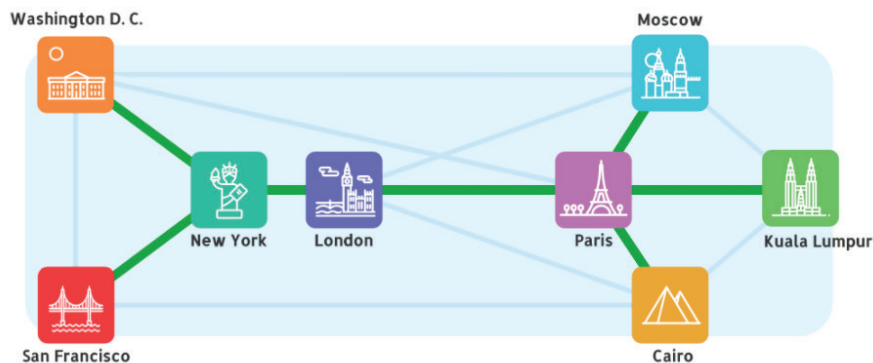
7

8

9

Answer

The example below, shows that it is possible to drop 8 routes and still allow customers to fly to any city.



Notice that when we drop 8 routes, we are left with $15 - 8 = 7$ routes. Why can't we drop more routes? If there were only 2 cities, it is clear that we would need 1 route. If there were 3 cities, then we would need 2 routes. This pattern continues and when there are 8 cities, we need 7 routes.

To be sure that pattern is correct, we need to convince ourselves that if you drop 9 routes (or more), then a customer will not always be able to fly to any city. If 9 routes (or more) are dropped, then there will be only 6 routes (or less). What goes wrong if there are only 6 routes? If there are only 6 routes, since there are 8 cities, then either there is a city not on any route or a city on exactly one route. If there is a city on exactly one route, then we can remove this city and route and be left with 5 routes and 7 cities. Continuing this argument, in all cases, we will be left with either a city not on a route or eventually no routes between 2 cities. Therefore, if there are only 6 routes, then a customer will not always be able to fly to any city.

Continued on next page



Greener Flight Routes - contin.

It's Computational Thinking

Computational Thinking Skills: Modelling and Simulation, Algorithms, Evaluation

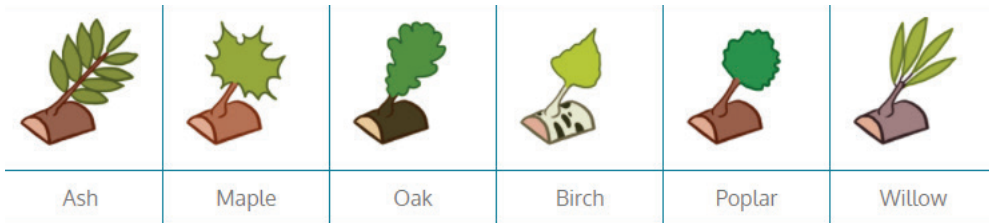
Concepts: Abstraction, Data Interpretation, Specification, Algorithms, Interactions, Impacts

The airline routes can be depicted as a graph. Cities are nodes in the graph and flight routes are edges. The problem of dropping maximum number of routes, while ensuring that all cities are connected, is an application of the Minimum Spanning Tree (MST) problem in Computer Science. The Minimum Spanning Tree problem consists of extracting a tree that connects all the nodes of a graph. For more information. Computer scientists use Minimum Spanning Tree algorithms to solve problems in laying telecommunications networks, transportation networks or water supply networks.



Wood Allergies

Some beavers are allergic to some types of wood and get really sick if they eat them. George is making wood snacks for a party and wants to make sure that everyone will have something to eat without getting sick. Each snack is made from one type of wood:



Name	Wood
Ann	Willow, Oak, Ash, Maple
Benjamin	Willow, Oak, Poplar
Cecil	Oak
Danny	Ash, Birch
Emma	Willow, Maple, Birch
Fred	Oak, Ash
George	Poplar, Maple

The beavers are happy to share snacks.

George has a list of the beavers attending the party and the types of wood that they can eat without getting sick.

To save time, George does not want to have to make snacks out of all six different types of wood, if possible.

Question

What are the fewest types of wood George can use to make snacks so that everyone can eat without getting sick?

1

2

3

4

5

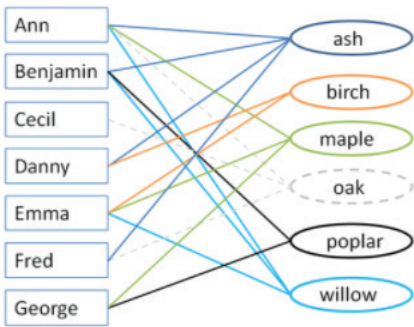
6

Answer

3

George will obviously have to make a dish from Oak for Cecil. This will also make Ann, Benjamin and Fred happy. There is no common dish for the remaining three beavers, so George will have to prepare at least two more. This will fortunately be enough. For example, George could make Oak, Ash and Maple.

This task is a classic example of a bipartite graph. Here, the first partition is the set of beavers, and the second partition is types of wood.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms, Evaluation
Concepts: Abstraction, Data Interpretation

This task is known in algorithm theory as the railway optimisation problem, in which the input is a schedule of trains and their stops and the goal is to find a set of train stations as small as possible so that every train visits at least one of the chosen stations.

Continued on next page



This question comes from
Slovenia

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



Wood Allergies - continued

It's Computational Thinking - continued

The problem is also known as a Dominating Set cover problem. Imagine six sets, each representing a different dish. Every set contains the beavers who like that dish. The task is to select the smallest number of sets (dishes) such that the union of these sets (all beavers who like any of the dishes represented by the chosen sets) contains all elements (beavers). In other words, the task is to cover all elements with the smallest number of sets.

The set cover problem is known as one of the hardest problems in computer science, a so-called NP-complete task. For these kinds of tasks, we do not know of any efficient algorithms that can be used to solve them. The only general solution is to try all different possibilities and even with a rather small number of sets the number of all possible combinations can be extremely large. With 240 dishes the number of different combinations are already about the same as the number of atoms in the entire universe!

Why have we been able to solve this task then? First, it was rather small. Second, it was made easier because Cecil had limited options. Finally, we used some simple logic that worked for this particular data, but wouldn't necessarily help us in general.



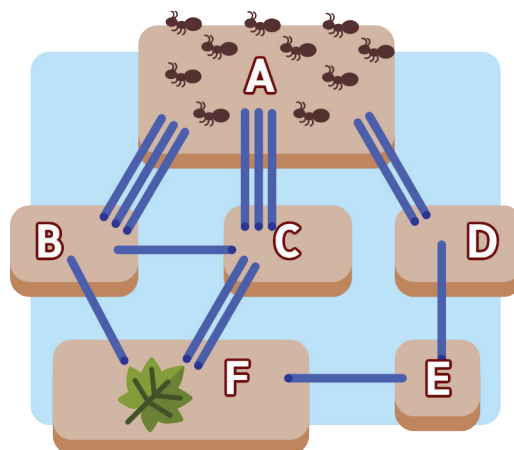
Ants in a Pond

Ten ants are on Stone A.

The ants want to get to the food on Stone F.

Only one ant can walk on a straw at a time.

It takes 1 minute for an ant to walk from one stone to another.



Question

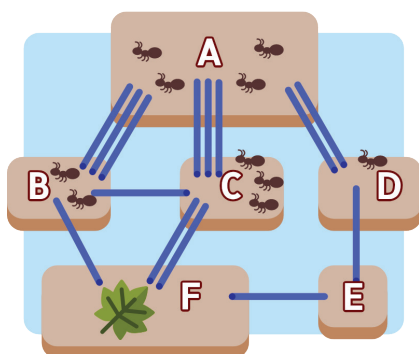
What is the maximum number of ants that can reach Stone F in 3 minutes?

Answer

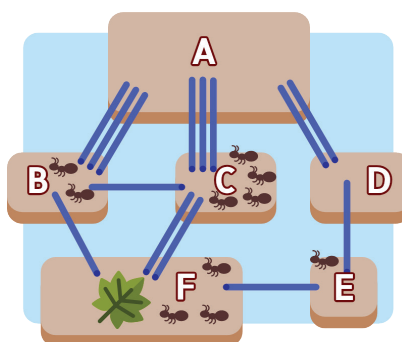
Correct answer is 7.

A possible situation after each minute is presented in the images below.

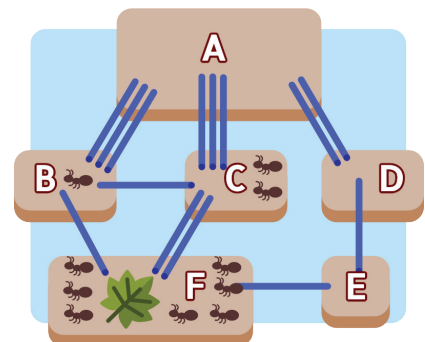
After one minute:



After two minutes:



After three minutes:



It's Computational Thinking

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

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

The goal here is to optimise the flow of the ants through the network so that as many ants as possible arrive at the food in 3 minutes time. This is called an optimization problem.

Ants that are not aware of the structure of the network they travel will not be able to find the best solution. But an observer, that can see the networks structure, can find an optimal strategy.

In this task, we assume that the ants are aware of the structure of the network, resulting in them choosing to move along a specific pathway.

Continued on next page



Ants in a Pond - continued

It's Computational Thinking - continued

Graphs are abstract data structures that are used to model networks and there are many existing algorithms to optimize the flow under specific circumstances. Alternatively, without making use of a pre-existing algorithm we can assert that:

- It makes no sense to send more than one ant via the stones D-E
- It makes no sense to send more than two ants via the stones A-B
- The straw between B-C does not contribute to the flow and can be ignored
- The limiting factor of the flow are the straws B-F and C-F

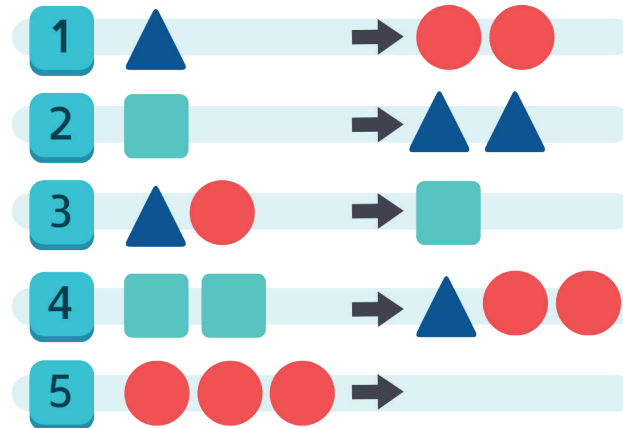
In our case, these considerations enabled us to find the optimal solution.

Shapes

Here is an initial sequence of 2 squares:



The buttons numbered 1 to 5 below replace the first instance of the given sequence as shown:



So, if button 1 is pushed, the first triangle, in any given sequence, is replaced with two circles. When button 5 is pushed, the first instance of three circles is replaced with nothing (that is, they are removed).

Question

Change the initial sequence of 2 squares by clicking on the buttons until you make the following sequence of 3 circles:



Answer

There are many possible solutions.

e.g. Press the buttons in the order: 4, 3, 2, 3, 2, 1, 1, 5, 1.

The idea is based on the observation that there is no button that creates an odd number of circles. Therefore we have the strategy to produce 6 circles and to collapse three of them with button 5.

How to create 6 circles? This can easily be done by button 2 if we already have 3 triangles. Now we have the intermediate goal to produce 3 triangles. This can be achieved by clicking buttons in this order: 4, 3, 2 and 3.

It's Computational Thinking

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

Concepts: Abstraction, Specification, Algorithms

The buttons of this task represent the rules of a formal language. A formal language consists of all the words that are produced according to precise rules. Each formal language also has a starting word, in our task, the two squares. Each rule produces from already existing (produced) words further words of the language.

The importance of formal languages in Informatics lies in the fact that programming languages are formal languages and that their syntax (grammar) can be exactly defined by these kinds of rules. Additional to the syntax is the semantics (the effects) of the elements of a programming language. These also have to be defined.





Seating Arrangement

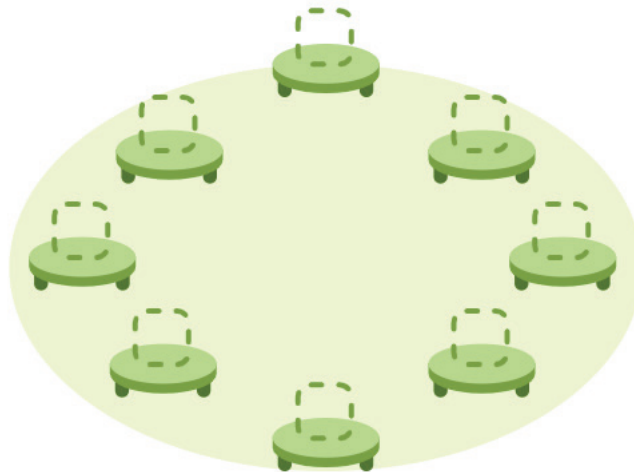
Eight friends are sitting in a circle. They are all facing inwards. We know the following facts about where they are sitting:

1. Alice is sitting directly opposite of David.
2. Henry is sitting between Greta and Eugene.
3. Franny is not next to Alice or David.
4. There is one person between Greta and Claire.
5. Eugene is sitting immediately to David's left.

Question

Place the friends in their correct places in the circle by dragging the letter next to their name to their chair.

There may be multiple correct solutions; you only need to find one.

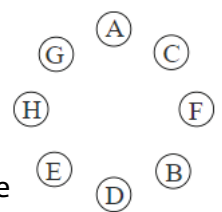


A	Alice
B	Bruce
C	Claire
D	David
E	Eugene
F	Franny
G	Greta
H	Henry

Answer

The following answer assumes that Alice is placed in the top seat. There are in fact 8 different answers, as an answer can be found corresponding to Alice being in any of the seats.

- Fact 1, Alice sits directly opposite to David, enables us to seat David.
- Now, Fact 5, Eugene is beside David, on David's left, enables us to seat Eugene.
- At this stage, Fact 2, Henry sits between Greta and Eugene, tells us where Henry sits.
- Knowing where Henry sits we now can place Greta.
- With five friends placed, Fact 3, Franny is not beside Alice or David, leaves only one space for Franny.
- Fact 4, there is one person between Greta and Claire, now tells us where Claire sits.
- Finally there is only one seat and one friend to place so we can place Bruce.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Algorithms, Evaluation

Concepts: Data Interpretation, Specification, Algorithms, Implementation

Logic is about following rules, understanding ordering and negation. For example, the key logic operator used in this question is negation, which is used whenever we say NOT. So, the fact that states that 'Franny is not beside Alice or David' can be stated differently as 'Franny not beside (Alice or David)' which is the same as '(Franny not beside Alice) and (Franny not beside David)'.

The negation operator, not, changes the logical expression of not (A or B) to an equivalent (not A) and (not B) which is an example of De Morgan's law. Knowing how logical expressions can be rewritten, combined, or simplified is a very useful skill in computer science.



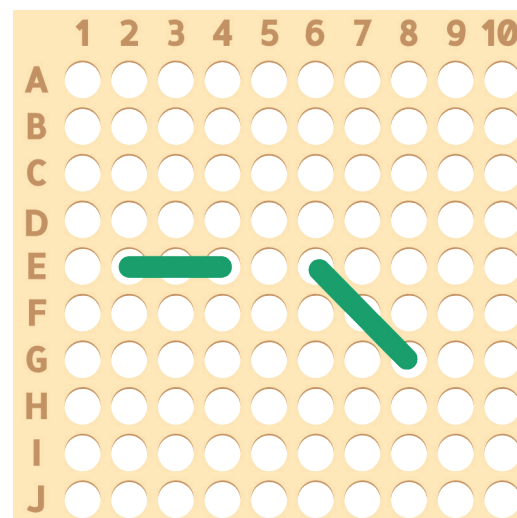
Embroidery Machine

A beaver wants to make an embroidery pattern by programming instructions for his embroidery machine.

The program for the instructions uses the command OUT(cc)-IN(dd), where cc and dd indicate the position of the needle in the grid.

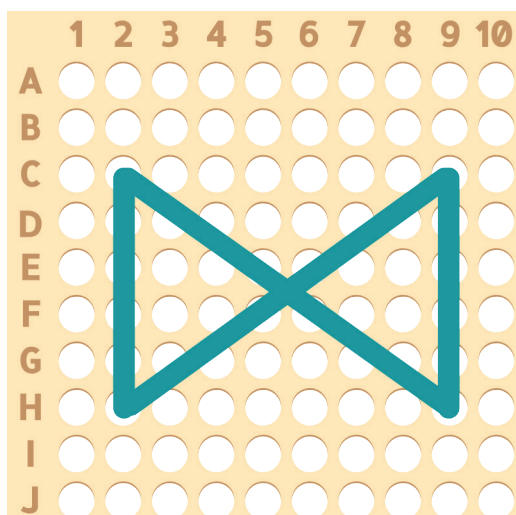
Example

OUT(E6)-IN(G8) moves the needle to the E6 position and pulls the cotton thread through the fabric from behind. The needle then moves to the G8 position and pushes the thread from the front through to the back. The following two commands create a pattern like the one below: OUT(E6)-IN(G8);OUT(E2)-IN(E4)



Question

Which set of commands would create a pattern like the one shown below?



OUT(H2)-IN(C2);OUT(H9)-IN(C9);OUT(C9)-IN(C2);OUT(H9)-IN(C2)

OUT(C2)-IN(H9);OUT(H2)-IN(C9);OUT(C2)-IN(H2);OUT(C9)-IN(H9)

OUT(H9)-IN(C9);OUT(H9)-IN(H2);OUT(C2)-IN(H2);OUT(C9)-IN(H2)

OUT(C2)-IN(C9);OUT(H2)-IN(H9);OUT(C2)-IN(H2);OUT(C9)-IN(H9)

Answer

OUT(C2)-IN(H9);OUT(H2)-IN(C9);OUT(C2)-IN(H2);OUT(C9)-IN(H9)

To create the ribbon pattern we need to run 4 commands, one for each of the four lines in random order. The commands are:

OUT(C2)-IN(H9) or OUT(H9)-IN(C2)

OUT(H2)-IN(C9) or OUT(C9)-IN(H2)

OUT(C2)-IN(H2) or OUT(H2)-IN(C2)

OUT(C9)-IN(H9) or OUT(H9)-IN(C9)

The choice that contains all four of the commands is the answer given above.

OUT(H2)-IN(C2);OUT(H9)-IN(C9);OUT(C9)-IN(C2);OUT(H9)-IN(C2) is wrong because it contains the command OUT(C9)-IN(C2) which creates a line that is not needed. Also the command OUT(H2)-IN(C9) or OUT(C9)-IN(H2) is missing.

OUT(H9)-IN(C9);OUT(H9)-IN(H2);OUT(C2)-IN(H2);OUT(C9)-IN(H2) is wrong because it contains the command OUT(H9)-IN(H2) which creates a line that is not needed. Also the command OUT(C9)-IN(H9) or OUT(H9)-IN(C9) is missing.



This question comes from the
Republic of Korea

Years 3+4

Years 5+6

Years 7+8

Years 9+10 Medium

Years 11+12 Easy



Embroidery Machine - contin.

Answer - continued

OUT(C2)-IN(C9);OUT(H2)-IN(H9);OUT(C2)-IN(H2);OUT(C9)-IN(H9) is wrong because it contains the commands OUT(C2)-IN(C9) and OUT(H2)-IN(H9) which create two lines that are not needed. Also the commands OUT(C2)-IN(H9) or OUT(H9)-IN(C2), and OUT(H2)-IN(C9) or OUT(C9)-IN(H2) are missing.

It's Computational Thinking

Computational Thinking Skills: Algorithms, Evaluation

Concepts: Data Interpretation, Specification, Algorithms

An algorithm describes the steps we follow to complete a task. Algorithms are common in computer science, but outside computer science algorithms may play a role in solving everyday life problems.

This task is an example of how an algorithm can be used to create a pattern by an embroidery machine.



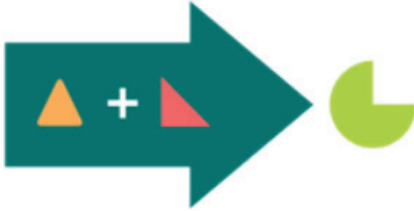
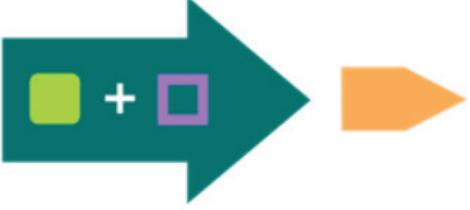
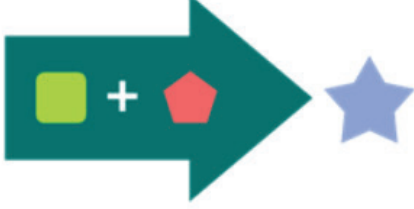
Scientists in the Lab

Three scientists are working in a laboratory with the following four chemicals.



They are trying to make a new chemical .

Each individual can carry out two reactions each, as shown in the table below.
There is only room in the laboratory for ONE scientist per day to carry out ONE reaction.

Microbiologist Maria (M):		
Physicist Peta (P):		
Dendrologist Dev (D):		

Question

Which sequence would NOT result in the production of chemical  ?

MPDPMD

PPMMDD

MPDMPD

PDPMMD

Answer

The correct answer is M P D M P D !

Since it is his second process that creates the final product, Dendrologist Dev (D) must be the last one in the sequence. To be able to create the final product using his second operation, he needs to have:





Scientists in the Lab - contin.

Answer - continued

To produce the blue circle, Microbiologist Maria (M) needs to have access to  and has to go to the lab after Physicist Peta (P). M also needs to perform her first operation before her second operation. To perform his first operation, Dendrologist Dev (D) needs Physicist Peta (P) to perform her first operation.

In the third answer, the first P in the sequence can only create one of the chemicals needed to perform the next two operations. One of them must fail. The order M P D M P D will not successfully create the final product.

It's Computational Thinking

Computational Thinking Skills: Pattern Recognition, Algorithms, Evaluation

Concepts: Data Representation, Data Interpretation, Specification, Algorithms

In Computer Science and in programming, we have processes where we use inputs and create outputs. It is important to understand that we cannot start a process if the input provided by the output of a previous process is missing. The programmer has to organise the subtasks to ensure that needed outputs from earlier processes are available.

A good programmer must be able to queue processes appropriately.

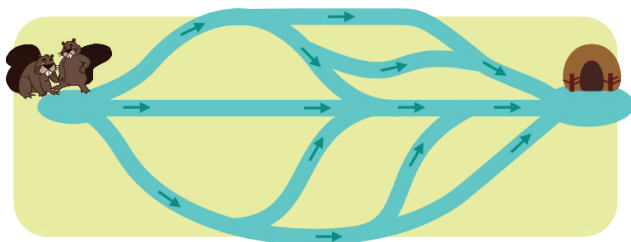
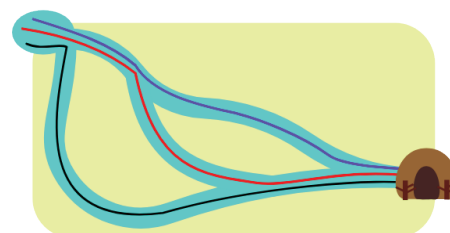


Many Beavers

Beavers are very ambitious: each beaver must have its own path to their home but are too lazy to swim upstream.

For example, in the river system pictured in the image below, we have at most 3 different paths from start to finish.

So, we can have at most 3 beavers living on this river.



There is a beaver couple living on a different river system shown on the picture below.

The beaver couple wants to have children and expand their family.

Question

What is the maximum number of kids they can have if each family member must have its own path to their home?

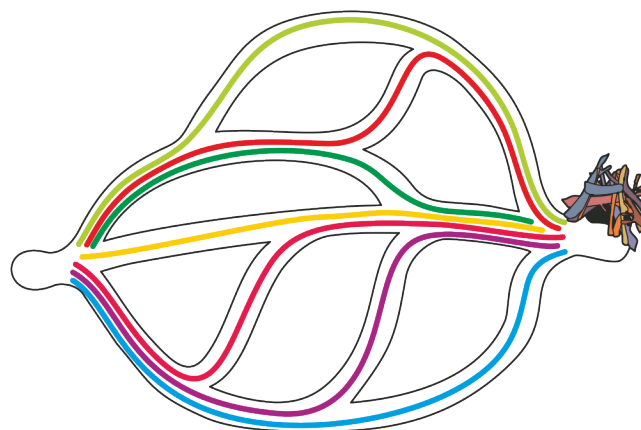
Answer

5

There are 7 different paths that lead to their home, 2 of which are reserved for the parents. We can count these paths in two different ways. One way is to look for all the paths in the picture and draw them in different colours.

A second way is to use a dynamic programming method. We look for all the junctions in the picture. We count how many paths lead to each junction and write the appropriate number on it.

To do it, we go from left to right and sum up the numbers on the junctions that are directly connected to the current junction by incoming arrows.



It's Computational Thinking

Computational Thinking Skills: Decomposition

Concepts: Data Representation, Specification, Algorithms, Implementation

Different tasks require different data representation. One of the ways is to use graphs. The water system in the picture can be represented as a directed graph. The edges represent the river sections and the nodes represent the junctions.

A directed graph is a convenient data structure that can help us find whether we can move from one node to another. We can write the length of each river section to the corresponding edge and get a weighted graph. We use weighted graphs to find the optimal path between starting and ending position.

The systematic approach in which we build our solution step by step using the previous steps is called dynamic programming. This approach is widely used to solve a lot of modern problems. You can apply dynamic programming only in the case where each next step depends only on the current situation and not on the way we reached it.



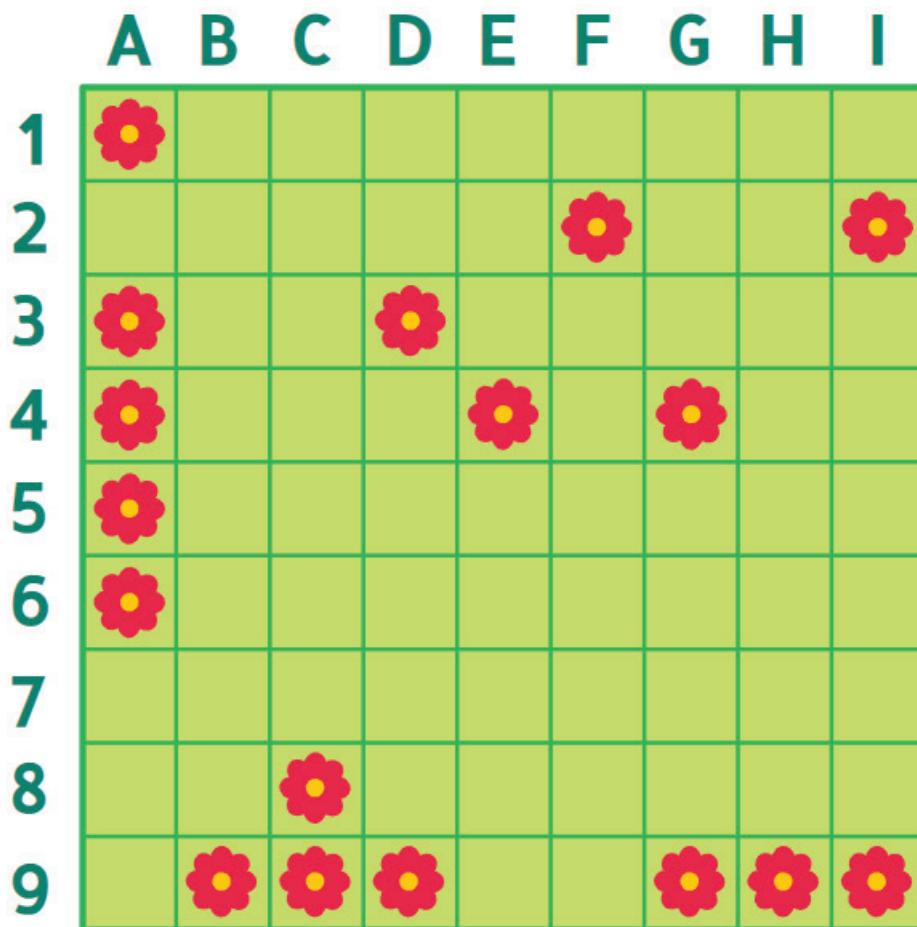
Bee Hive

A beekeeper has a hive with bees. He wants to place the hive so that the sum of distances from the hive to every flower is minimum.

The field with flowers is represented in the following grid, with rows from 1 to 9 and columns from A to I.

The bees only fly horizontally and vertically on this field, so the distance between two tiles is the sum of the horizontal distance and the vertical distance.

For example, the distance between C4 and D7 is 4 (3 tiles vertically plus 1 tile horizontally).



Question

Where should the beekeeper place the hive so that the sum of the distance from the hive to every flower is minimum?

Answer

The answer is D5.

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms



Knight's Moves

On a TV-show in Germany, contestants have to solve many interesting puzzles. One of these problems is to find a word in a grid-like the one below, in which all the letters are different.

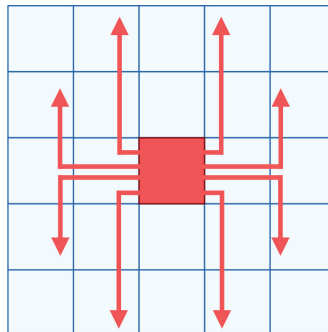
L	E	H
C		D
R	I	N

The contestants choose one of the eight letters as a starting point. Then they have to create a word of exactly eight letters by performing seven consecutive knight's moves.

In a knight's move, one can:

- go one step up or down, followed by two steps left or right, or
- go one step left or right, followed by two steps up or down, and
- a letter in the grid has to be visited exactly once, no more, no less.

The grid below shows all possible knight's moves possible from the red cell in the centre.



Example

Starting with the letter C follow the moves shown to find the word CHILDREN.

Beaver Viola wonders how difficult it would be to make a computer program which shows all possible eight-letter combinations that can be found in this manner.



Question

How many eight-letter combinations can be found in this way, visiting all the letters in the grid exactly once and assuming that all the letters in the grid are different?

Answer

16

Each letter in the scheme is connected with exactly two other letters. Starting at a random letter there are two possibilities to proceed to the next letter. This second letter is also connected with two other letters. But it is not allowed to visit a letter twice, so there is only one possible move to proceed. In this way we can carry on. So starting at a letter there are two ways to create an eight letter combination. As there are 8 possible starting points, $8 \times 2 = 16$ eight letter combinations can be composed.

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

When solving problems for which there is no obvious or elegant strategy available, one last resort is using 'brute force'. A computer is programmed to inspect all possible solutions and to select the solution(s) that work with the problem. When considering applying brute force to solve a problem one should at least estimate how many possible solutions have to be inspected.



Encoded Treasure Map



The King of the Beavers has hidden his treasure in a country of 7 provinces as shown in the map.

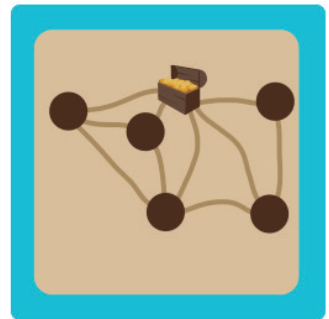
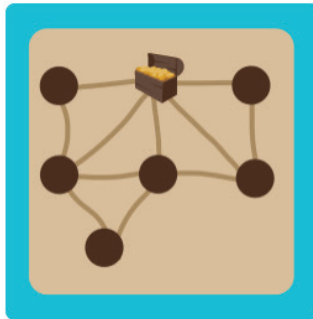
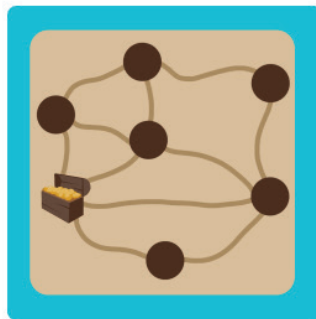
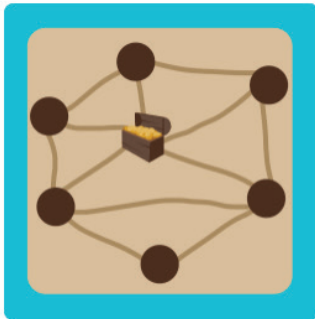
However, the king decides he needs to create an encoded map and then destroy the unencoded map above.

On his encoded map, circles represent provinces and two circles are connected by a line if the corresponding provinces border each other.

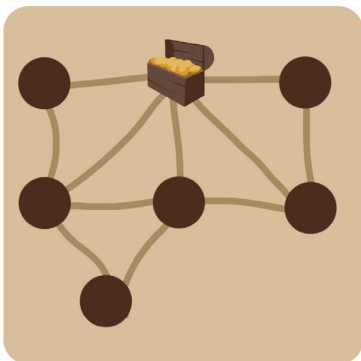
To confuse thieves even more, the king also made three false encoded maps.

Question

Which is the real map?



Answer



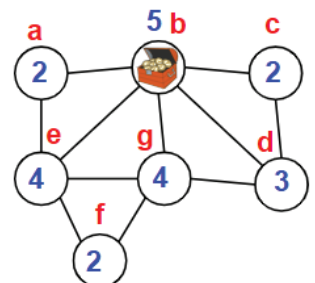
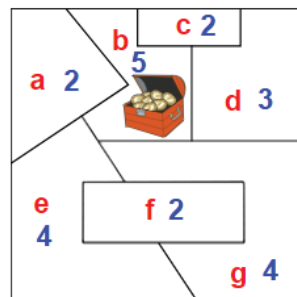
The map shows 7 provinces. Therefore one map can be ruled out straight away because there are just 6 circles in it.

The province with the treasure has five neighbours. We can therefore rule out the map that contains no province with five neighbours.

To decide between the remaining maps we need to match the circles (denoting provinces in the encoded map) with areas on the original map.

But counting the number of neighbours can help us. To be sure, we should still match the circles with countries. And it is actually possible like this:

Letters in red denote provinces. Numbers in blue denote the number of neighbours.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

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

This task shows an example how graphs can be used to represent reality. A graph has vertices (circles) and edges (lines) between them. Edges denote a relation between nodes. In this case the vertices denote provinces and edges denote the relation "is the neighbour of". Graphs are a powerful tool when we need to describe clearly the relations between some entities (provinces, in this case) of the real world. Mathematicians and computer scientists developed a lot of useful algorithms that can be applied to graphs (a simple example in this task is counting the vertices).



Willows and Poplars

Berty wants to find out about two types of trees: willow and poplar. To do this, he uses a search engine that uses two special symbols which allow for more complex search queries:

... # ... Combines the queries on both sides of the #. It only shows pages that satisfy BOTH queries.

[...] Reverses the query between the brackets. It only shows pages that do NOT satisfy the query.



Example

Query	Meaning
willow	Shows pages about willows.
[willow]	Shows pages NOT about willows.
willow # poplar	Shows pages about BOTH willows AND poplars.
[[poplar]]	Shows pages about poplars.
[willow # poplar]	Shows pages that are NOT about BOTH willows and poplars.
willow # [poplar]	Shows pages about willows BUT NOT about poplars.

Question

Which one of the following queries will find all the pages that are about willows, or about poplars, or about both willows and poplars?

[willow] # [poplar]

[[willow] # poplar] # [willow # [poplar]]

[[willow] # [poplar]]

[[willow] # [poplar]] # [willow # poplar]

Answer

[[willow] # [poplar]]

To see why this is correct, let's first have a look at the simpler alternative answer: '[willow] # [poplar]' This query asks to find all pages that satisfy both '[willow]' and '[poplar]', in other words, all pages that do NOT contain information about willows and at the same time do NOT contain information about poplars - in short: pages that mention neither willows nor poplars.

Continued on next page



Willows and Poplars - contin.

Answer - continued

This is certainly not correct, in fact this is the complete opposite of what Berty is looking for. But 'opposite' is what '['...']' means. So putting answer A between square brackets is precisely what we look for - and that happens to be: `[[willow] # [poplar]]`

`[[willow] # [poplar]] # [willow # poplar]` starts with the correct answer but adds an extra requirement: `'[willow # poplar]'`. The query `'[willow # poplar]'` indicates that pages that contain information on both willows and poplars should NOT be reported. In other words, it shows what Berty is looking for, except pages that have information on both willows and poplars. So, this query will lead to pages that have information on either willows or poplars, but not on both.

To understand what `[[willow] # poplar] # [willow # [poplar]]` searches for, we split it into two parts. The first part `'[[willow] # poplar]'` indicates that we do not want pages which are about poplars and not about willows, in other words, if the page is about poplars it must also be about willows. Similarly, the second part `'[willow # [poplar]]'` tells that if the page contains information about willows it must also contain information about poplars. Combining both requirements yields pages that must be about both willows and poplars or about neither of them.

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms, Digital Systems

Like the imaginary Search App, our modern search engines allow queries to be combined in several ways - using NOT, AND or OR. What Berty really wants, is to search for willows OR poplars, but unfortunately there is no symbol for 'OR' in his Search App.

This task shows that 'OR' can be expressed by using a combination of 'NOT's and 'AND's, in particular $a \text{ OR } b = \text{NOT}(\text{NOT}(a) \text{ AND } \text{NOT}(b))$ Logicians know this as one of De Morgan's laws.

You use many logical expressions when programming. Knowing some basic laws such as De Morgan's laws makes it easier to write and understand complex programs.

We have to be careful about the word 'OR' because it can have different meanings. One of the wrong answers - `[[willow] # [poplar]] # [willow # poplar]` - returns what we call the exclusive OR (or XOR): websites about either willows or poplars, but not about both. This is the OR we often use in daily life. If someone asks you whether you want to eat fries or pizza, you usually do not answer that you want to eat both.

`[[willow] # [poplar]]` returns the inclusive OR: pages about willows or about poplars or about both.

A final note:

You may find it strange that the Search App uses a symbol '#' for AND - maybe you expected the word 'AND' or the symbol '&' - and square brackets for NOT. For the computer this makes no difference, it just does what it is programmed to do. Usually programmers will choose conventions that are easy for a human to understand - to avoid confusion - but the computer could not care less: to the computer '#' is just like any other symbol.

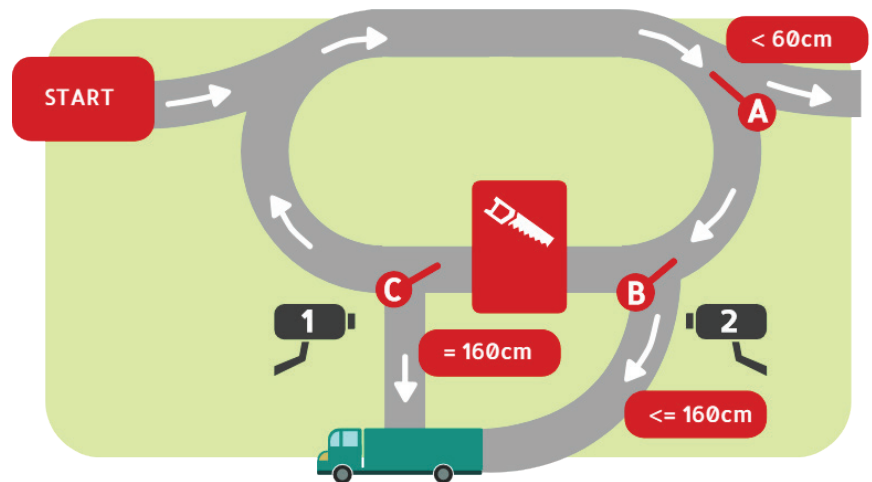


Sawmill

Logs of various lengths arrive at a sawmill. Only logs that are between 60 cm to 160 cm long are sent to the semi-trailer truck. The sawmill is shown below:

Key Sawmill features

- The logs arrive at the START and move around the mill on several conveyor belts.
- Switch A sends away logs that are shorter than 60 cm. Other logs continue around.
- Switch B sends logs that are shorter than or equal to 160 cm to the truck. Other logs continue around.
- The Saw cuts the logs into two logs, one of which is 160 cm long. All logs continue around.
- Switch C sends logs that are 160 cm long to the semi-trailer truck. Other logs continue around.
- Sensor 1 and sensor 2 count the number of logs sent to the truck.



Question

Three logs of different sizes (60cm, 140cm and 360cm) are sent to the sawmill. When no more logs are being sent to the saw, how many logs have the sensors counted?

Sensor 1: 1 log, Sensor 2: 3 logs

Sensor 1: 3 logs, Sensor 2: 1 log

Sensor 1: 2 logs, Sensor 2: 2 logs

Sensor 1: 0 logs, Sensor 2: 4 logs

Answer

Sensor 1: 2 logs, Sensor 2: 2 logs

Since no logs are physically joined together, the order of the logs does not matter. For the 60 cm log, it passes Switch A. Switch B sends it to the truck. Sensor 2 counts one log. For the 140 cm log, it passes Switch A. Switch B sends it to the truck. Sensor 2 counts one log. For the 360 cm log, it passes Switch A and Switch B. The saw cuts the log into two logs of lengths 160cm and 200cm. The 160 cm log gets sent to the truck by switch C and Sensor 1 counts one log. The 200 cm log is allowed to continue by switch C and merges into the stream of logs. The 200cm log now passes switches A and B. The saw cuts off a 160m log and Switch C sends it to the truck, Sensor 1 counts one log. The 40 cm log left continues past switch C but at Switch A it is sent away for being too short. In all, Sensors A and B will have counted 2 logs each.

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

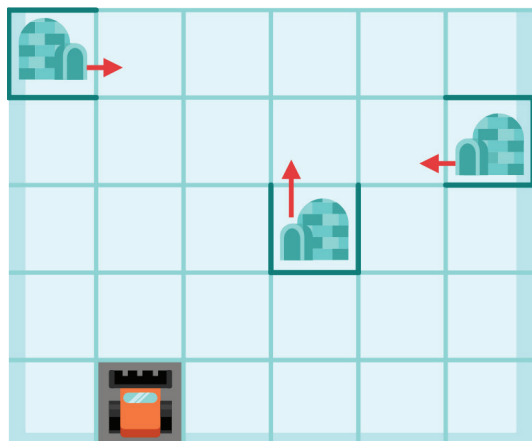
The logs travelling down the stream can be abstracted as a sequence of integers. This task then becomes a direct application of Reactive Programming, where a stream of integers (the lengths of the logs) is processed with several reactive operators (e.g. merge, filter, transform) and some measures are done with scan. Shortly after the Start is a merge operation, merging two streams. Switches A and B can be regarded as filter operations, each splitting a stream into two based on a threshold value. The saw together with Switch C can be regarded as a transform operation, transforming integers into pairs of integers. The Sensors can be regarded as scan operations, if we imagine they store a count of the logs passing by them.





Rescue Mission

After a snowstorm, three Inuit want their igloos connected to the main road system again. A robotic snowplough is given the task to clear a path to do this. The robot then needs to return to its starting position.



It takes 1 hour to move from one square to another if there is no snow in either square.

It takes 2 hours to move from one square to another if snow needs to be cleared.

It takes no time for the robot to turn around in a cleared square.

The robot does not need to go to an igloo, just clear the square in front of its entrance so the people can walk out.

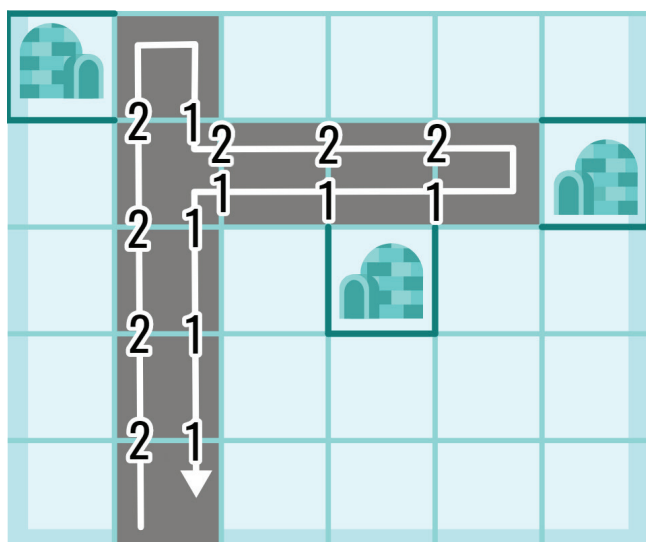
Question

What is the minimum time the robot needs to accomplish the task?

Answer

21 hours

Since it would take less time to re-enter a square that is already cleared, it would be optimal if we can reuse paths as much as possible. This gives us the idea that the cleared path must be as close to the igloos as possible. Keeping this in mind, we can build a path the one in the figure below. How to get the time for the path? Adding up the numbers in the above diagram produces the correct answer.



Can we do better than 21 hours?

As removing snow takes more time than simply crossing a square, we would like to minimise the removing of snow. In the figure above, all the cleared paths are reused to reduce time. Another way to reduce time would be to take some shortcuts and not continue on cleared paths, but there are none that takes less time than following the cleared paths in this case.

Therefore, the only way to reduce time now is to reduce the number of squares that the robot has to clear or pass through which is not possible in this figure.

It's Computational Thinking

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

Concepts: Abstraction, Specification, Algorithms

This optimisation task is a special case of the Steiner tree problem – finding graph that connects all nodes with minimum costs. Although it looks similar to the minimum spanning tree problem, where you have to find a graph that connects all nodes with minimum costs, but for the minimum spanning tree problem you have the constraint that you may only use existing edges ... in this case there are no edges between the nodes, the snowplough creates them. This task is also a bit special because first entry to a square “costs” more time than a second or further entry. We don’t know of an effective algorithm for solutions to such problems. Finding optimal solutions for bigger area would take very long time even when using powerful computers.

Bebras Challenge 2020 Round 1

Years 11+12



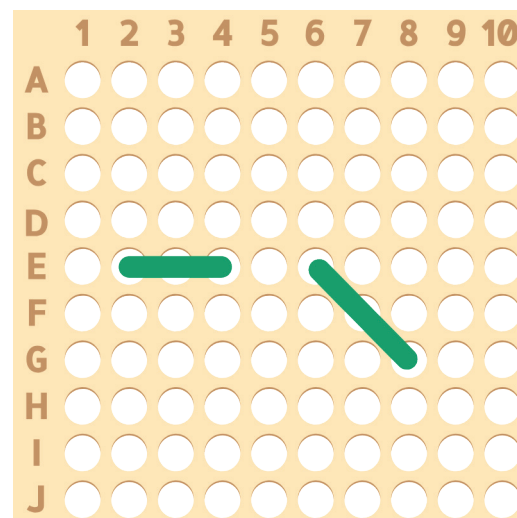
Embroidery Machine

A beaver wants to make an embroidery pattern by programming instructions for his embroidery machine.

The program for the instructions uses the command OUT(cc)-IN(dd), where cc and dd indicate the position of the needle in the grid.

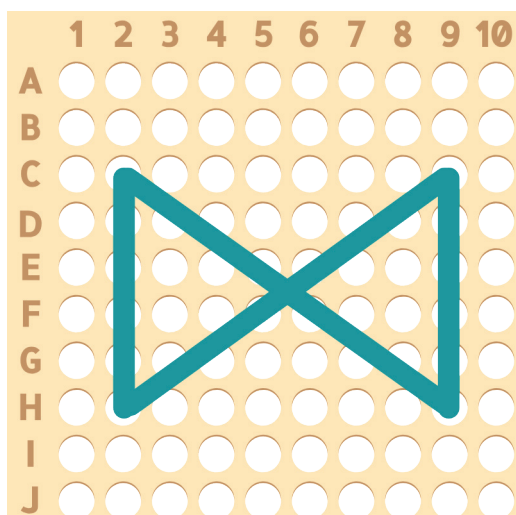
Example

OUT(E6)-IN(G8) moves the needle to the E6 position and pulls the cotton thread through the fabric from behind. The needle then moves to the G8 position and pushes the thread from the front through to the back. The following two commands create a pattern like the one below: OUT(E6)-IN(G8);OUT(E2)-IN(E4)



Question

Which set of commands would create a pattern like the one shown below?



OUT(H2)-IN(C2);OUT(H9)-IN(C9);OUT(C9)-IN(C2);OUT(H9)-IN(C2)

OUT(C2)-IN(H9);OUT(H2)-IN(C9);OUT(C2)-IN(H2);OUT(C9)-IN(H9)

OUT(H9)-IN(C9);OUT(H9)-IN(H2);OUT(C2)-IN(H2);OUT(C9)-IN(H2)

OUT(C2)-IN(C9);OUT(H2)-IN(H9);OUT(C2)-IN(H2);OUT(C9)-IN(H9)

Answer

OUT(C2)-IN(H9);OUT(H2)-IN(C9);OUT(C2)-IN(H2);OUT(C9)-IN(H9)

To create the ribbon pattern we need to run 4 commands, one for each of the four lines in random order. The commands are:

OUT(C2)-IN(H9) or OUT(H9)-IN(C2)

OUT(H2)-IN(C9) or OUT(C9)-IN(H2)

OUT(C2)-IN(H2) or OUT(H2)-IN(C2)

OUT(C9)-IN(H9) or OUT(H9)-IN(C9)

The choice that contains all four of the commands is the answer given above.

OUT(H2)-IN(C2);OUT(H9)-IN(C9);OUT(C9)-IN(C2);OUT(H9)-IN(C2) is wrong because it contains the command OUT(C9)-IN(C2) which creates a line that is not needed. Also the command OUT(H2)-IN(C9) or OUT(C9)-IN(H2) is missing.

OUT(H9)-IN(C9);OUT(H9)-IN(H2);OUT(C2)-IN(H2);OUT(C9)-IN(H2) is wrong because it contains the command OUT(H9)-IN(H2) which creates a line that is not needed. Also the command OUT(C9)-IN(H9) or OUT(H9)-IN(C9) is missing.



Embroidery Machine - contin.

Answer - continued

OUT(C2)-IN(C9);OUT(H2)-IN(H9);OUT(C2)-IN(H2);OUT(C9)-IN(H9) is wrong because it contains the commands OUT(C2)-IN(C9) and OUT(H2)-IN(H9) which create two lines that are not needed. Also the commands OUT(C2)-IN(H9) or OUT(H9)-IN(C2), and OUT(H2)-IN(C9) or OUT(C9)-IN(H2) are missing.

It's Computational Thinking

Computational Thinking Skills: Algorithms, Evaluation

Concepts: Data Interpretation, Specification, Algorithms

An algorithm describes the steps we follow to complete a task. Algorithms are common in computer science, but outside computer science algorithms may play a role in solving everyday life problems.

This task is an example of how an algorithm can be used to create a pattern by an embroidery machine.



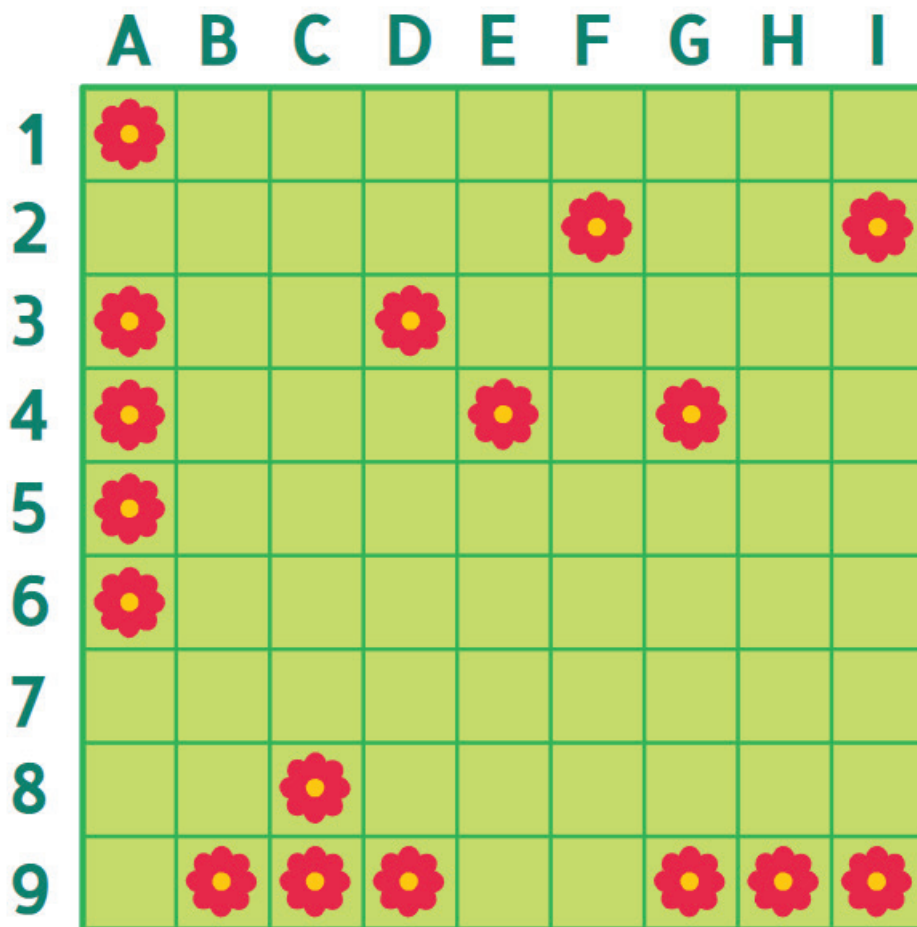
Bee Hive

A beekeeper has a hive with bees. He wants to place the hive so that the sum of distances from the hive to every flower is minimum.

The field with flowers is represented in the following grid, with rows from 1 to 9 and columns from A to I.

The bees only fly horizontally and vertically on this field, so the distance between two tiles is the sum of the horizontal distance and the vertical distance.

For example, the distance between C4 and D7 is 4 (3 tiles vertically plus 1 tile horizontally).



Question

Where should the beekeeper place the hive so that the sum of the distance from the hive to every flower is minimum?

Answer

The answer is D5.

It's Computational Thinking

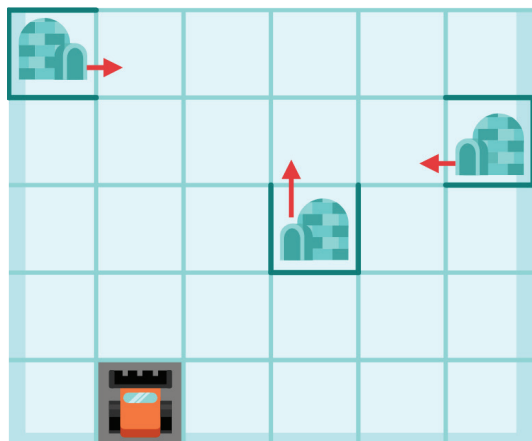
Computational Thinking Skills: Decomposition, Pattern Recognition, Abstraction, Algorithms

Concepts: Abstraction, Data Interpretation, Specification, Algorithms



Rescue Mission

After a snowstorm, three Inuit want their igloos connected to the main road system again. A robotic snowplough is given the task to clear a path to do this. The robot then needs to return to its starting position.



It takes 1 hour to move from one square to another if there is no snow in either square.

It takes 2 hours to move from one square to another if snow needs to be cleared.

It takes no time for the robot to turn around in a cleared square.

The robot does not need to go to an igloo, just clear the square in front of its entrance so the people can walk out.

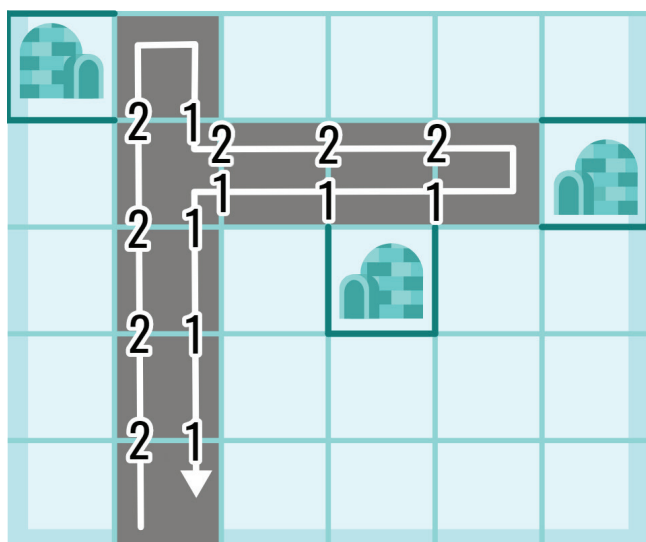
Question

What is the minimum time the robot needs to accomplish the task?

Answer

21 hours

Since it would take less time to re-enter a square that is already cleared, it would be optimal if we can reuse paths as much as possible. This gives us the idea that the cleared path must be as close to the igloos as possible. Keeping this in mind, we can build a path the one in the figure below. How to get the time for the path? Adding up the numbers in the above diagram produces the correct answer.



Can we do better than 21 hours?

As removing snow takes more time than simply crossing a square, we would like to minimise the removing of snow. In the figure above, all the cleared paths are reused to reduce time. Another way to reduce time would be to take some shortcuts and not continue on cleared paths, but there are none that takes less time than following the cleared paths in this case.

Therefore, the only way to reduce time now is to reduce the number of squares that the robot has to clear or pass through which is not possible in this figure.

It's Computational Thinking

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

Concepts: Abstraction, Specification, Algorithms

This optimisation task is a special case of the Steiner tree problem – finding graph that connects all nodes with minimum costs. Although it looks similar to the minimum spanning tree problem, where you have to find a graph that connects all nodes with minimum costs, but for the minimum spanning tree problem you have the constraint that you may only use existing edges ... in this case there are no edges between the nodes, the snowplough creates them. This task is also a bit special because first entry to a square “costs” more time than a second or further entry. We don’t know of an effective algorithm for solutions to such problems. Finding optimal solutions for bigger area would take very long time even when using powerful computers.



Drawing Triplets

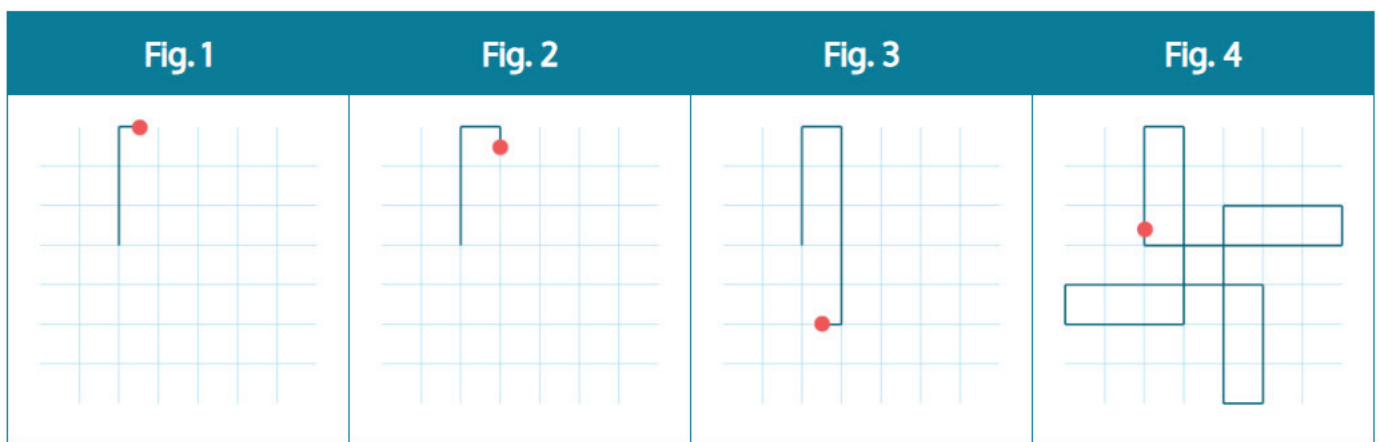
A robot makes a pattern by drawing a line while it moves on a square grid. Each pattern is represented by a set of three numbers.

Example

3, 1, 5 represents Fig. 4 because it means:

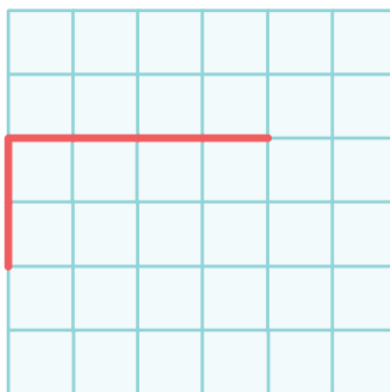
- Move forward 3 squares, then turn right (Fig. 1)
- Move forward 1 square, then turn right (Fig. 2)
- Move forward 5 squares, then turn right (Fig. 3)

The robot repeats the three moves forever.

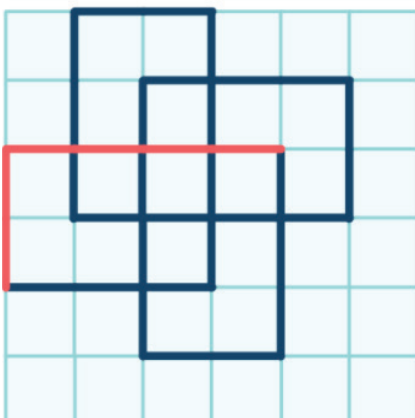


Question

Click on the grid lines to draw the pattern that is represented by 2,4,3



Answer



This is just a case of following the algorithm carefully. Starting from the point where the red line and black lines meet we go:

2 lines north, turn right, 4 lines forward, turn right, 3 lines forward, turn right, 2 lines forward, turn right, 3 lines forward, turn right...

until the black line meets the red.

In fact the program keeps running but it traces the same path.

Continued on next page



Drawing Triplets - continued

It's Computational Thinking

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

Concepts: Specification, Algorithms, Implementation

Computer Program Execution in computer and software engineering is the process by which a computer executes the instructions of a computer program. Each instruction of a program is a description of a specific action to be carried out in order for a specific problem to be solved. As instructions of a program, and therefore the actions they describe, are being carried out by an executing machine, specific effects are produced in accordance to the instructions being executed.

The solution of the task involves understanding the instructions and execution of a given algorithm which is an essential part of computer programming. You have to be able to read and understand the instructions and execute them step by step to see what is happening. This is a very important skill for every programmer and is used in the process of computer program debugging.



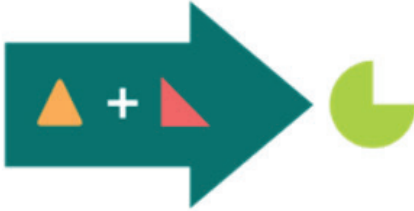
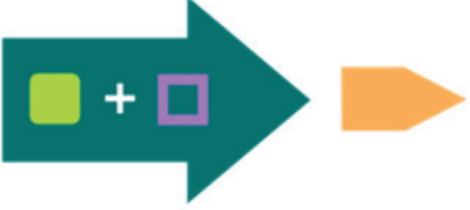
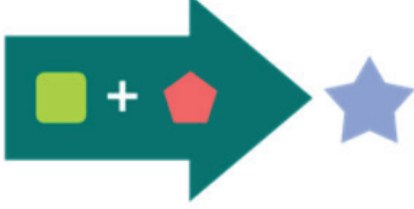
Scientists in the Lab

Three scientists are working in a laboratory with the following four chemicals.



They are trying to make a new chemical .

Each individual can carry out two reactions each, as shown in the table below.
There is only room in the laboratory for ONE scientist per day to carry out ONE reaction.

Microbiologist Maria (M):		
Physicist Peta (P):		
Dendrologist Dev (D):		

Question

Which sequence would NOT result in the production of chemical  ?

MPDPMD

PPMMDD

MPDMPD

PDPMMD

Answer

The correct answer is M P D M P D !

Since it is his second process that creates the final product, Dendrologist Dev (D) must be the last one in the sequence. To be able to create the final product using his second operation, he needs to have:





Scientists in the Lab - contin.

Answer - continued

To produce the blue circle, Microbiologist Maria (M) needs to have access to  and has to go to the lab after Physicist Peta (P). M also needs to perform her first operation before her second operation. To perform his first operation, Dendrologist Dev (D) needs Physicist Peta (P) to perform her first operation.

In the third answer, the first P in the sequence can only create one of the chemicals needed to perform the next two operations. One of them must fail. The order M P D M P D will not successfully create the final product.

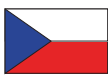
It's Computational Thinking

Computational Thinking Skills: Pattern Recognition, Algorithms, Evaluation

Concepts: Data Representation, Data Interpretation, Specification, Algorithms

In Computer Science and in programming, we have processes where we use inputs and create outputs. It is important to understand that we cannot start a process if the input provided by the output of a previous process is missing. The programmer has to organise the subtasks to ensure that needed outputs from earlier processes are available.

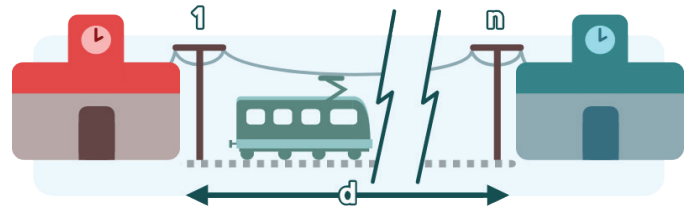
A good programmer must be able to queue processes appropriately.



Railway Electrification

The railway line between two stations is to be electrified. The masts must be placed at regular intervals, the first one at the 1st station and the last one at the 2nd station.

The mast-placing robot works according to the program below. The distance between stations is d meters; the robot must place n masts.



Question

Drag in the correct bits of code, on the right, into the spaces in the program below so that the mast-placing robot can complete its task correctly.

define

deploy the masts

drive to the warehouse

load masts

arrive at the 1-st station

place one mast

repeat

move by meters

place one mast

↺

n

d

1

2

n+1

d+1

n-1

d-1

n/2

d/2

n+d

n-d

d-n

d*n

n/d

d/n

n/(d+1)

d/(n+1)

n/(d-1)

d/(n-1)

Answer

The correct answer is (from top to bottom) n , $n-1$, $d/(n-1)$

If robot must place n masts, it must load n masts. After placing one mast, $n-1$ masts are left. So the robot must repeat $n-1$ times moving and placing one mast. Then we must divide the distance d by $(n-1)$ to know how much the robot has to move to place the next mast. Thus $(n-1) * [d/(n-1)] = d$.

It's Computational Thinking

Computational Thinking Skills: Modelling and Simulation, Algorithms, Evaluation

Concepts: Algorithms, Implementation, Digital Systems, Impacts

The piece of code in this task is a procedure. In this procedure we use variables d and n . A variable is considered as a name for a value, e.g. some number. These variables d and n are defined previously in the main program and used in this procedure. These are called global variables because they can be accessed in different procedures of the same program.

The task is about loops too. A loop is a sequence of instructions executed a number of times. In this task it is very important to know how many times the loop must be executed. In our task the loop must be executed $n-1$ times to complete the electrification of the railway line.

Procedures, sequences of program instructions, are powerful programming tools that perform a specific task. Values are passed to and from procedures through variables. The same procedure can work on different tasks by passing different values in the variables.

define

deploy the masts

drive to the warehouse

load

n

 masts

arrive at the 1-st station

place one mast

repeat

n-1

move by

d/(n-1)

 meters

place one mast

↺



Willows and Poplars

Berty wants to find out about two types of trees: willow and poplar. To do this, he uses a search engine that uses two special symbols which allow for more complex search queries:

... # ... Combines the queries on both sides of the #. It only shows pages that satisfy BOTH queries.

[...] Reverses the query between the brackets. It only shows pages that do NOT satisfy the query.



Example

Query	Meaning
willow	Shows pages about willows.
[willow]	Shows pages NOT about willows.
willow # poplar	Shows pages about BOTH willows AND poplars.
[[poplar]]	Shows pages about poplars.
[willow # poplar]	Shows pages that are NOT about BOTH willows and poplars.
willow # [poplar]	Shows pages about willows BUT NOT about poplars.

Question

Which one of the following queries will find all the pages that are about willows, or about poplars, or about both willows and poplars?

[willow] # [poplar]

[[willow] # poplar] # [willow # [poplar]]

[[willow] # [poplar]]

[[willow] # [poplar]] # [willow # poplar]

Answer

[[willow] # [poplar]]

To see why this is correct, let's first have a look at the simpler alternative answer: '[willow] # [poplar]'. This query asks to find all pages that satisfy both '[willow]' and '[poplar]', in other words, all pages that do NOT contain information about willows and at the same time do NOT contain information about poplars - in short: pages that mention neither willows nor poplars.

Continued on next page



Willows and Poplars - contin.

Answer - continued

This is certainly not correct, in fact this is the complete opposite of what Bertie is looking for. But 'opposite' is what '[...]' means. So putting answer A between square brackets is precisely what we look for - and that happens to be: `[[willow] # [poplar]]`

`[[willow] # [poplar]] # [willow # poplar]` starts with the correct answer but adds an extra requirement: `'[willow # poplar]'`. The query `'[willow # poplar]'` indicates that pages that contain information on both willows and poplars should NOT be reported. In other words, it shows what Bertie is looking for, except pages that have information on both willows and poplars. So, this query will lead to pages that have information on either willows or poplars, but not on both.

To understand what `[[willow] # poplar] # [willow # [poplar]]` searches for, we split it into two parts. The first part `'[[willow] # poplar]'` indicates that we do not want pages which are about poplars and not about willows, in other words, if the page is about poplars it must also be about willows. Similarly, the second part `'[willow # [poplar]]'` tells that if the page contains information about willows it must also contain information about poplars. Combining both requirements yields pages that must be about both willows and poplars or about neither of them.

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms, Digital Systems

Like the imaginary Search App, our modern search engines allow queries to be combined in several ways - using NOT, AND or OR. What Bertie really wants, is to search for willows OR poplars, but unfortunately there is no symbol for 'OR' in his Search App.

This task shows that 'OR' can be expressed by using a combination of 'NOT's and 'AND's, in particular $a \text{ OR } b = \text{NOT}(\text{NOT}(a) \text{ AND } \text{NOT}(b))$ Logicians know this as one of De Morgan's laws.

You use many logical expressions when programming. Knowing some basic laws such as De Morgan's laws makes it easier to write and understand complex programs.

We have to be careful about the word 'OR' because it can have different meanings. One of the wrong answers - `[[willow] # [poplar]] # [willow # poplar]` - returns what we call the exclusive OR (or XOR): websites about either willows or poplars, but not about both. This is the OR we often use in daily life. If someone asks you whether you want to eat fries or pizza, you usually do not answer that you want to eat both.

`[[willow] # [poplar]]` returns the inclusive OR: pages about willows or about poplars or about both.

A final note:

You may find it strange that the Search App uses a symbol '#' for AND - maybe you expected the word 'AND' or the symbol '&' - and square brackets for NOT. For the computer this makes no difference, it just does what it is programmed to do. Usually programmers will choose conventions that are easy for a human to understand - to avoid confusion - but the computer could not care less: to the computer '#' is just like any other symbol.



Encoded Treasure Map



The King of the Beavers has hidden his treasure in a country of 7 provinces as shown in the map.

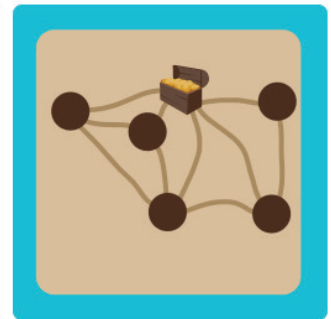
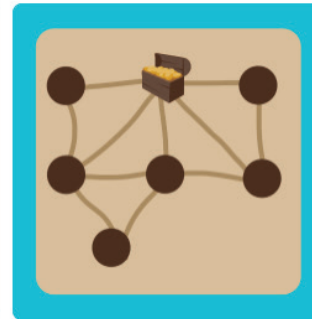
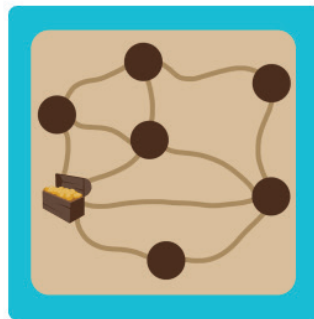
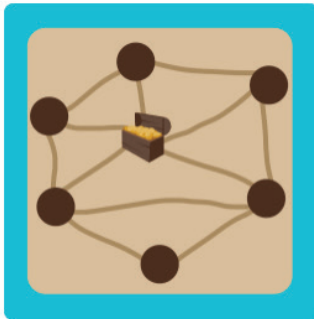
However, the king decides he needs to create an encoded map and then destroy the unencoded map above.

On his encoded map, circles represent provinces and two circles are connected by a line if the corresponding provinces border each other.

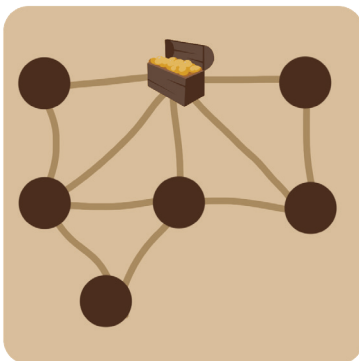
To confuse thieves even more, the king also made three false encoded maps.

Question

Which is the real map?



Answer



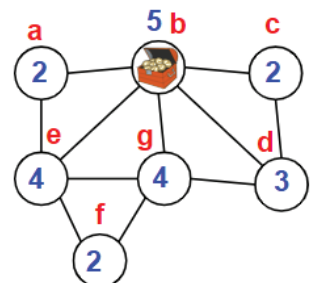
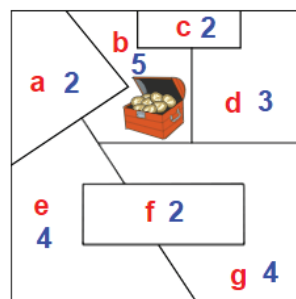
The map shows 7 provinces. Therefore one map can be ruled out straight away because there are just 6 circles in it.

The province with the treasure has five neighbours. We can therefore rule out the map that contains no province with five neighbours.

To decide between the remaining maps we need to match the circles (denoting provinces in the encoded map) with areas on the original map.

But counting the number of neighbours can help us. To be sure, we should still match the circles with countries. And it is actually possible like this:

Letters in red denote provinces. Numbers in blue denote the number of neighbours.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

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

This task shows an example how graphs can be used to represent reality. A graph has vertices (circles) and edges (lines) between them. Edges denote a relation between nodes. In this case the vertices denote provinces and edges denote the relation "is the neighbour of". Graphs are a powerful tool when we need to describe clearly the relations between some entities (provinces, in this case) of the real world. Mathematicians and computer scientists developed a lot of useful algorithms that can be applied to graphs (a simple example in this task is counting the vertices).



Way To Go Home

Tara the Beaver wants to go home. She can just move up or move to the right on the following map.

On a square there are either monsters or coins. Tara collects all the coins along the path that she follows. When she walks on a square with monsters, she needs to give each monster a coin. If she does not have enough coins, she will need to give it to them later when she has enough.

Question

Which route should Tara take to make sure she ends up with the most coins?

Answer

The correct answer is the path below. The correct answer is 10.

One way to solve the task is by making a matrix. A positive value shows number of coins that Tara can get when she is in the cell. A negative value means that she has to give coins to the monster.

We label the rows 1 to 5 from bottom to top and the columns 1 to 5 from left to right. In the beginning, (row 1, column 1), Tara does not have any coins. If she goes up, she gets 3 coins. If she goes to the right, she gets 2 coins. For the cell in (row 2, column 2), she gets $3-1=2$ coins if she comes from the left and $2-1=1$ coin if she comes from below. She chooses the larger value, so we record that she reaches (row 2, column 2) from the left, (row 2, column 1). In this way, row by row, for each cell, we compute coins possible by coming from the left or from below and record the maximum.

The table below shows these maximum values. The arrow indicates the direction from which the maximum value is derived. The yellow values indicate the best path for Tara to reach home.



$3+1=4$	→	$4-2=2$	→	$3+3=6$		$7+3=10$	→	10
↑		↑				↑		
$5-2=3$	→	$3+1=4$		$5-2=3$		$8-1=7$	→	$7+2=9$
↑				↑		↑		↑
$3+2=5$	→	$5-2=3$	→	$3+2=5$	→	$5+3=8$	→	$8-1=7$
↑								
$0+3=3$	→	$3-1=2$		$4-1=3$	→	$3-1=2$		$4+1=5$
↑				↑				↑
0	→	$0+2=2$	→	$2+2=4$	→	$4+2=2$	→	$2+2=4$

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms, Evaluation

Concepts: Abstraction, Specification, Algorithms

To solve the task, we have to calculate the best way of reaching each cell. There are many possible paths to a cell, and exploring all of them would be very tedious. We realize that each cell can only be reached from two directions, left and below, so we can use the values calculated for those cells to derive the value for the current cell. In this way, we can systematically compute the values for all cells, row by row from bottom to top, or column by column from left to right. This is an example of dynamic programming, an important problem-solving approach in algorithms.



Warehouses

There are 42 hedgehogs that store their bedding in 5 warehouses. The first hedgehog stores her bedding in the first warehouse, the second in the second warehouse, the sixth one again in the first warehouse and so on.

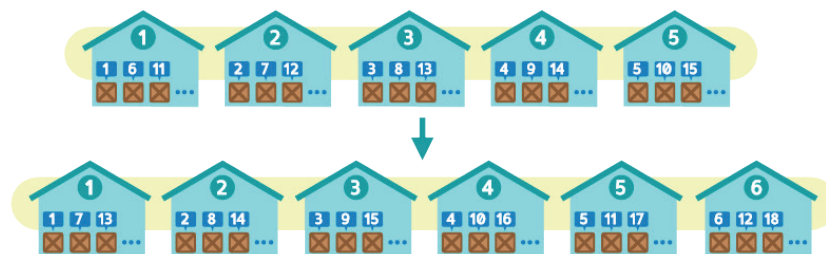
One day the hedgehogs build a new warehouse, warehouse number 6. They decide to move their bedding between warehouses to make their distribution simple again: the first hedgehog stores her bedding in the first warehouse, the second in the second warehouse, the seventh one again in the first warehouse and so on.

Question

How many hedgehogs did not need to move their bedding between warehouses?

Example

The first hedgehog always stores bedding in the first warehouse, she does not need to not move her bedding. But the sixth hedgehog needs to move his bedding from the first warehouse to the new warehouse number 6.



Answer

10

Hedgehogs numbered 1, 2, 3, 4, 5 and 31, 32, 33, 34, 35 do not need to move.

To see this we can, first, simplify the numbering. Let hedgehogs and warehouses be numbered from 0. So we have hedgehogs from 0 to 41. A hedgehog numbered x should not move if $x \bmod 5 = x \bmod 6$. This is possible if $30 \mid x - r$, where $r = x \bmod 5 = x \bmod 6$. So, this is only possible for $x = 0 + 0, 1, 2, 3, 4$ and $x = 30 + 0, 1, 2, 3, 4$.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

This is a task about distributed storage of resources: in order to retrieve them one can use a strategy called hashing.

In this approach each resource is associated to a hash key that can be computed efficiently; resources are then retrieved by computing the hash key and searching only among the ones with the same hashing keys. For example, if resources are persons with their name, a hashing strategy can use the first letter of the name as the hash key: if one has to retrieve the person "Mattia", the hash key will be "M"; to find it you can search only among the smaller group of persons with same hash key (for example "Mattia", "Mathilde", and "Marcello").

In the task the hash key is computed by using the mod function (the integer remainder of division), but when the number of warehouses increased the key changed. Good hash functions with the property of not changing the key when something in the structure of the data store changes are said to have the consistent hashing property.

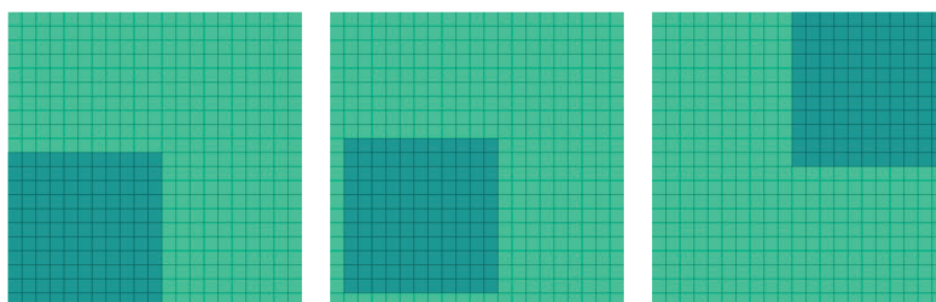


Video Compression

A computer image is a rectangular grid of coloured squares, called pixels. A video is a sequence of images, called frames, each slightly different from the previous one. The simplest way to store a video is to store all the pixels in each frame.

A more efficient way is to store the entire first frame and then only store those pixels that change from the current frame to the next one. In the picture below, the 10x10 dark coloured square moves from the lower-left corner to the upper right corner of the light coloured 20x20 field, moving one pixel horizontally and vertically in each frame.

This takes 11 frames. If we store this video in the simple format, it will require $(20 \times 20) \times 11 = 4400$ pixels.



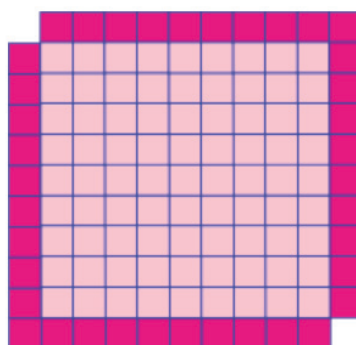
Question

How many pixels will you need to save these 11 frames in the more efficient format described above?

Answer

780

We need remember $20 \times 20 = 400$ pixels for the first frame and 38 changing pixels for each of the next frames.



It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms, Evaluation

Concepts: Abstraction, Data Representation, Specification, Algorithms, Digital Systems

This task is devoted to the idea of lossless compression, when new frames are considered as slices of a three-dimensional array and the change of the picture on the screen occurs only in those pixels that differ in adjacent slices-frames.

We should keep in mind that in the algorithm presented we have actually to remember not only the colour of changed pixel, but its coordinates as well. If all those parameters are considered as one-byte information, we need three times as many bytes than in the answer for all frames, except the first one which is $400 + 380 \times 3 = 1540$ bytes. However, it is still better than remembering each frame.



Every time Stephen uses the  rule, a single bracelet is eventually produced.

Stephen made these two bracelets following his rules:



Stephen made three of the four bracelets shown below using the same rules. Which bracelet did he NOT make?



It is not too difficult to make a similar picture for the first and second bracelets if you want to.

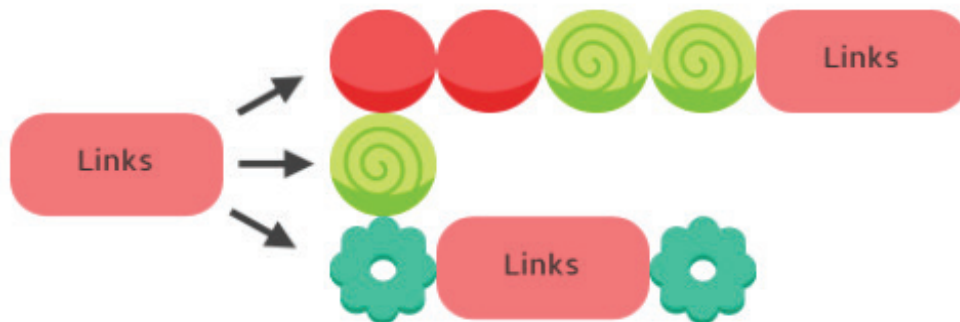
Continued on next page



Friendship Bracelets - contin.

Answer - continued

To prove that C can never be made using the rules, we look at the 'links'-rule, the 'swirled'-rule and the 'flowers'-rule. Together these can be combined into a new rule, which does not change the kind of bracelet we may produce:



In other words, if we ignore the flower beads, then every bracelet consists of the pattern  repeated a number of times (possibly only once, or not at all) followed by a single .

Three of the bracelets can be clearly seen to follow this pattern, but the third bracelet does not - there are three light blue 'swirly' links in the middle of the bracelet, and that is not allowed.

It's Computational Thinking

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

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

The set of rules given in this task, could be used by a computer to check whether a bracelet could have been made by Stephen. Similar rules are used by computers to check whether programmers made any typing mistakes in their programs, or to check whether what you type into a web form is correct. For example, the following rules could be used to tell the computer what is a valid number:

- number unsigned or + unsigned or - unsigned
- unsigned digit or unsigned digit
- digit 0 or 1 or 2 or 3 or 4 or 5 or 6 or 7 or 8 or 9

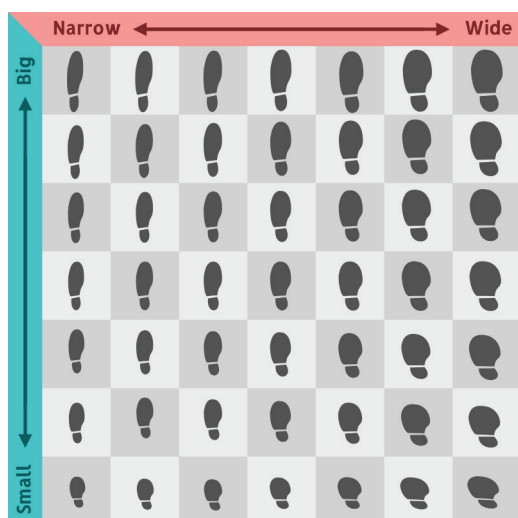
These types of sets of rules, in computer science, are called grammars, or context free grammars.



Shoe Shopping

Beaver wants to buy a pair of shoes. All the shoes have a different size and width. Beaver will have to try on shoes until he finds a pair which is the right size and width.

Whenever he tries on a shoe, he can feel both 1) if it is too wide, the right width, or too narrow and 2) if it is too big, the right size, or too small. The shoes in the store are arranged in increasing order of size as well as width as shown below:



Beaver uses a method that guarantees that he will know which shoe size fits him in 'n' tries, even if he hasn't tried the shoe that fits yet.

Question

What is the smallest possible value of 'n'?

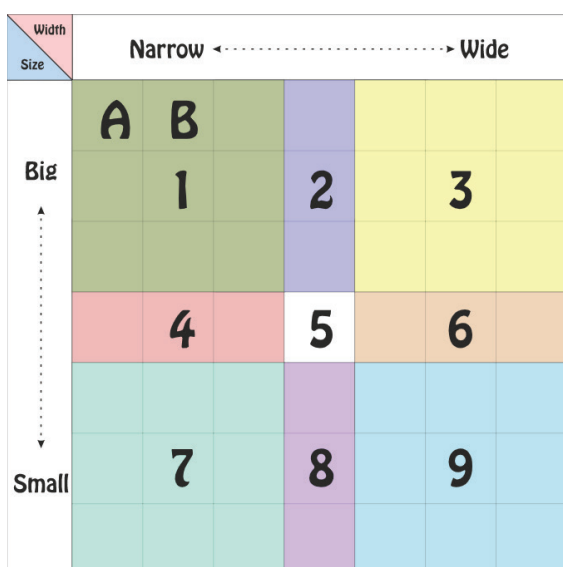
Answer

The answer is 2.

The beaver can get lucky and find the shoe in the first attempt. However, he can find the right fit after trying on 2 shoes. First, he can start with the shoe in the centre. The shoe will either be the right size, smaller, bigger and right width, narrow or wide

for the beaver. According to the fit, the beaver will know that the shoe that will fit him will be in one of the following nine coloured zones.

- If the shoe fits him, he has found the right pair of shoes.
- If the shoe is smaller and wider, he would try shoes in zone 1.
- If the shoe is smaller but the right width, he would try shoes in zone 2.
- If the shoe is bigger and narrow, he would try shoes in zone 9 and so on.



Let us suppose the shoe the beaver tried is smaller and wider for him. So he has to try for bigger and narrower shoes, which will be in zone 1. He now tries the shoe in the centre of zone 1.

- If the shoe fits him, he has found the right pair of shoes.
- If the shoe is still smaller and wider for him, the shoe in position A will be the right fit.
- If the shoe is smaller but the right width, the shoe in position B will be the right fit.

As we can see, the beaver will have to try at the most 2 shoes to find the right fit. If he starts in any other position, he will have to try on more shoes.

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

In the above problem, the shoes on display in the shoe store are arranged in increasing order of size and width along multiple dimensions. Such an arrangement is called sorting. Binary search algorithms are used on sorted data to find an item very quickly.

Continued on next page





This question comes from the
Republic of Korea

Years 3+4

Years 5+6

Years 7+8

Years 9+10

Years 11+12 Hard



Shoe Shopping - continued

It's Computational Thinking - continued

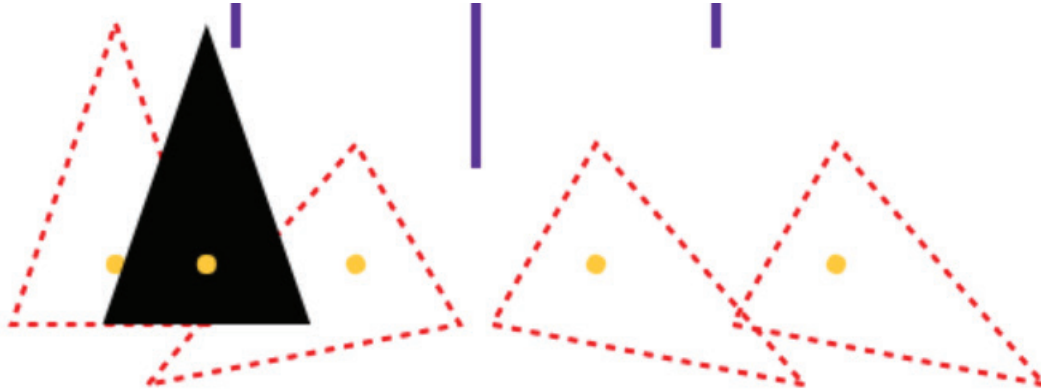
Binary searching reduces the search space into half every time to get to correct answer in fewer attempts. Using a binary search algorithm when playing the game “guess a number between 1 and 100” can be done with very few guesses if this search algorithm is used.

The Shoe Shopping problem is about binary searching that finds a desired value in a two-dimensional direction.



Rotate Rotate Rotate

A triangle moves on a conveyor belt. The triangle has a pivot. Two vertices are closer to the pivot, and one vertex is farther away from the pivot. The placement of the pivot is chosen so that when the triangle is rotated 120°, one of the vertices is always directly above the pivot.



There are arms above the conveyor belt. When the triangle passes under an arm, it may rotate 120° anticlockwise. There are two types of arms:

- A long arm which always rotates the triangle
- A short arm which rotates the triangle only if the vertex farthest from the centre is at the top

Bob is designing a process that uses triangles just like those shown above. He wants triangles that may arrive in various orientations to be arranged into a common orientation before he designs the next process. Bob asks you to find a sequence of arms that will be able to accept triangles in any orientation and rotate these triangles so that they always end up in the same position, ready for the next process. Bob does not mind what the final position is as long as it is always the same for any triangle entering the system.

Question

Which of the following sequences of arms achieves Bob's objective?

Short, Long, Short, Long

Long, Short, Long, Short

Long, Short, Short, Long

Short, Long, Long, Short

Answer

The correct answer is: Short, Long, Long, Short

It's Computational Thinking

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

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

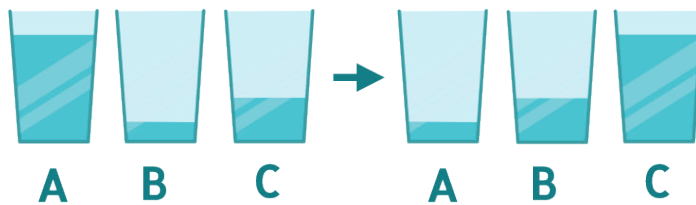
It has three states, the furthest vertex from the pivot is: UP, LEFT or RIGHT. There are also two transition operators: the short hand and the long hand. From each state the triangle can jump to another state via a transition. In this task we have to find the sequence of transitions which always results in a common end state, regardless of the starting state. We are thus being asked to produce a reset sequence. This is quite a common task in Computer Science.



Water Pouring

Three glasses A, B and C contain different quantities of water and are placed in a row. None of the glasses are initially full. The glasses do not have any scale marked on them, however, comparing the quantity of water in the glasses is possible.

Use one or more of the permitted operations, described below, to transform the state of the glasses as shown in the picture, without the use of extra glasses or swapping the glasses' positions.



After the operations, glass A should contain the quantity of water that was initially in glass B, glass B should contain the quantity of water that was initially in glass C, and glass C should contain the quantity of water that was initially in glass A.

Permitted Operations

(note that not all operations are always possible to carry out)

- Emptying: take a glass and pour all of its water into another glass
- Equalizing: take a glass and pour as much of its water into another glass so that the other glass contains exactly as much water as the third glass
- Filling: take a glass and pour as much of its water into another glass until that glass is full

Question

Which of the following statements is correct?

The desired outcome can be achieved without the 'emptying' operation

The desired outcome can be achieved without the 'equalizing' operation

The desired outcome can be achieved without the 'filling' operation

The desired outcome cannot be achieved using these 3 operations

Answer

The desired outcome cannot be achieved using these 3 operations.

Notice that after performing any positive number of the three operations, one of the following has to hold:

- One of the buckets is empty
- Two of the buckets have the same quantity of water
- One of the buckets is full

The required final state has none of these properties, therefore the transformation is impossible.

It's Computational Thinking

Computational Thinking Skills: Decomposition, Abstraction, Algorithms

Concepts: Abstraction, Data Interpretation, Specification, Algorithms

The three operations in the task are examples of information-destroying operations. One way to tell if information is preserved after an operation is if the operation can be reversed.

The initial state can be represented as three different numbers greater than 0 and less than 1. If you carry out the emptying operation, one of these numbers turns into zero. If you carry out the equalizing operation two of the numbers turn into the same.

Continued on next page



Water Pouring - continued

It's Computational Thinking - continued

If you carry out the filling operation, one of the numbers turns to the fixed value of 1. In each case it is impossible to reverse the operation in order to retrieve the initial numbers.

Computer programmers must think carefully about the effects of the operations in their programs to make sure important information is preserved.

A well-known example where information must be preserved in programming is swapping the values of two variables. Suppose variable a contains the value 0.5 and variable b contains the value 0.2 and our aim is to swap the values. A naive and wrong algorithm is:

```
a := b;  
b := a;
```

Afterwards both a and b contain the value 0.2 and the initial value 0.5 of variable a is lost. The programmer may introduce an intermediate variable c to store the initial value of a :

```
c := b;  
b := a;  
a := c;
```

Alternatively, the programmer may rewrite the algorithm in this way:

```
a := a - b;  
b := b + a;  
a := b - a;
```

In this case the difference between the initial values is preserved and this can be used to recalculate the initial values in the second and third program line.

In the task, each of the permitted operations causes loss of information. Intermediate variables are furthermore excluded by the absence of extra buckets.

Quantum computing is another area of computer science where information-preserving operations are important. In quantum computing, all operations must be reversible to be considered valid operations.

We would like to thank the International Bebras Committee and community for their ongoing support, resources and collaborative efforts.

Special thanks to Eljakim Schrijvers, Alieke Stijf and Dave Oostendorp for their support and technical expertise.

This Solutions Guide was adapted by Allira Crowe, Tom Grubb and Mel Phelps. Created by Hannah Piper and edited by Ruwan Devasurendra.

Contact us

CSIRO Digital Careers

digitalcareers@csiro.au

digitalcareers.csiro.au