

Bebras Australia Computational Thinking Challenge

2023 Solutions Guide Round 2



Primary School
Grades 3–6

bebras.edu.au

Bebras Australia Computational Thinking Challenge

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

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

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

The Bebras international community has now grown to 60 countries with over 3.9 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 May 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.

csiro.au/digital-careers

581

Australian schools participated in Round 1 2023



31,837

Australian students participated in Round 1 2023



3.29 million

Students participate worldwide



digital
careers



What is a Solutions Guide?

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

Within this Solutions Guide you will find all of the questions and tasks from Round 2 of the Bebras Australia Computational Thinking Challenge 2023. On each page above the question you will find the age group, level of difficulty, country of origin and key Computational Thinking skills.

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

Contents

What is a Solutions Guide?	3
What is Computational Thinking?	5
Computational Thinking skills alignment	6
Australian Digital Technologies curriculum key concepts	7
Digital Technologies key concepts alignment	8
Years 3+4	9
Visiting Iguazu Falls	10
Ski Village	11
Learn Traditional Art	12
Trail to Grandma	13
New Car	15
Golden Ticket	16
Strange Payment	18
Where Is Beaver?	20
Making A Necklace	22
Journey to the Hive	23
Password	25
Robot Beaver	26
Pool Party Lockers	28
Moving Around	30
Birthday Party	32
Years 5+6	34
Geography	35
Secret Message	37
The Bay of Beaver Programmers	38
Building Instruction	39
Colourful Tower	40
Pantry Map	42
Rank	44
Forest Party	46
Colour the Frog!	48
Robot Factory	51
Self-Driving Cars	53
Loop Kolam	55
Robot Beaver	56
Ski Trail Map	58
Ancient Runes	59

What is Computational Thinking?

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



DECOMPOSITION

Breaking down problems into smaller, easier parts.



PATTERN RECOGNITION

Using patterns in information to solve problems.



ABSTRACTION

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



MODELLING AND SIMULATION

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



ALGORITHMS

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



EVALUATION

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

More Computational Thinking resources

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

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



Computational Thinking skills alignment

2023 Round 2 Questions	Grade level	Decomposition	Pattern Recognition	Abstraction	Modelling & Simulation	Algorithms	Evaluation
Years 3+4							
Iguazu Falls	Easy						
Ski Village	Easy						
Learn Traditional Art	Easy						
Trail to Grandma	Easy						
New Car	Easy						
Golden Ticket	Medium						
Strange Payment B	Medium						
Where is Beaver?	Medium						
Making a Necklace	Medium						
Journey to the Hive	Medium						
Password	Hard						
Robot Beaver A	Hard						
Pool Party Lockers	Hard						
Moving Around	Hard						
Birthday Party	Hard						
Years 5+6							
Geography	Easy						
Secret Message	Easy						
The Bay of Beavers Programmers	Easy						
Building Instruction	Easy						
Colourful Tower	Easy						
Pantry Map	Medium						
Rank	Medium						
Forest Party	Medium						
Colour the Frog!	Medium						
Robot Factory	Medium						
Self Driving Cars	Hard						
Loop Kolam	Hard						
Robot Beaver B	Hard						
Ski Trail Map	Hard						
Ancient Runes	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

2023 Round 2 Questions	Grade level	Abstraction	Data Collection	Data Representation	Data Interpretation	Specification	Algorithms	Implementation	Digital Systems	Interactions	Impacts
Years 3+4											
Iguazu Falls	Easy										
Ski Village	Easy										
Learn Traditional Art	Easy										
Trail to Grandma	Easy										
New Car	Easy										
Golden Ticket	Medium										
Strange Payment B	Medium										
Where is Beaver?	Medium										
Making a Necklace	Medium										
Journey to the Hive	Medium										
Password	Hard										
Robot Beaver A	Hard										
Pool Party Lockers	Hard										
Moving Around	Hard										
Birthday Party	Hard										
Years 5+6											
Secret Message	Easy										
The Bay of Beavers Programmers	Easy										
Building Instruction	Easy										
Colourful Tower	Easy										
Pantry Map	Medium										
Rank	Medium										
Forest Party	Medium										
Colour the Frog!	Medium										
Robot Factory	Medium										
Self Driving Cars	Hard										
Loop Kolam	Hard										
Robot Beaver B	Hard										
Ski Trail Map	Hard										
Ancient Runes	Hard										

Bebras Challenge 2023 Round 2

Years 3+4

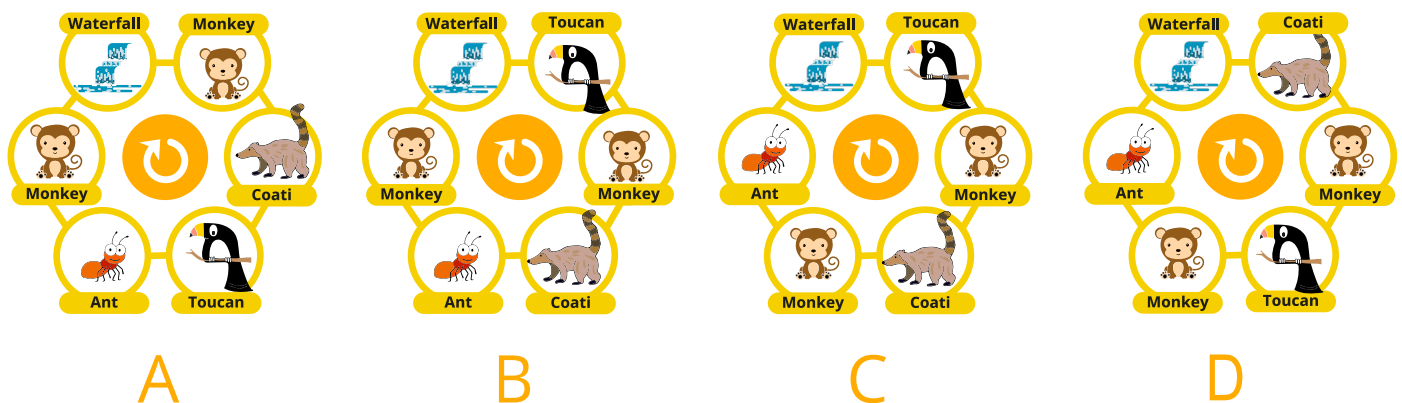


Visiting Iguazu Falls

Martha and her father visited the jungle on their summer vacation. Her father wanted to show Martha the beautiful waterfalls. On the way to the waterfalls, they saw several different animals in the following order: Toucan, Monkey, Coati, Monkey, and Ant. Finally, they reached the falls.

Question

Moving clockwise, as shown by the arrow in the centre of the images, choose the option that represents the order in which Martha saw the animals during the journey to the falls.



EXPLANATION

Answer

The correct answer is C.

Explanation

If we look for the first animal on the list (toucan) and follow the order of the animals moving clockwise, we see that option C has a monkey after the toucan and then a coati followed by another monkey and finally an ant.

Option A cannot be correct because it starts with a monkey.

Option B can't be correct because, despite following the sequence toucan, monkey, coati, the next animal is an ant instead of a monkey.

BACKGROUND INFORMATION

Algorithms are sequences of steps or instructions in an orderly manner, which must be followed to solve problems. In this question, the design of the algorithm shows us how the animals have been visualised during the journey to reach the falls.



Ski Village

Petra, Vid and Ante are little beavers who got lost in the ski village, where there are five huts. Every one of them remembers what some parts of their hut look like, enough to find it. They are shown in the table on the right.

Beaver	Number of windows	Hut colour	Door colour
Petra	4	green	black
Vid	4	not yellow	red
Ante	2	yellow	blue

Question

Help them find their huts based on what they remember.



Petra

Ante

Vid

EXPLANATION

Answer

Correct answer is shown on the picture on the right.



Explanation

Petra's hut - there are 2 green huts with 4 windows, but we have to find the one with black door

Vid's hut - there are 2 huts with 4 windows, but we have to find the one with red doors.

Ante's hut - there are 2 yellow huts with 2 windows, but we have to find the one with blue doors.

BACKGROUND INFORMATION

Processing information in different format is an essential part in informatics. In this task, students are expected to map a set of features presented in a table to an object presented in a visual format.



Learn Traditional Art

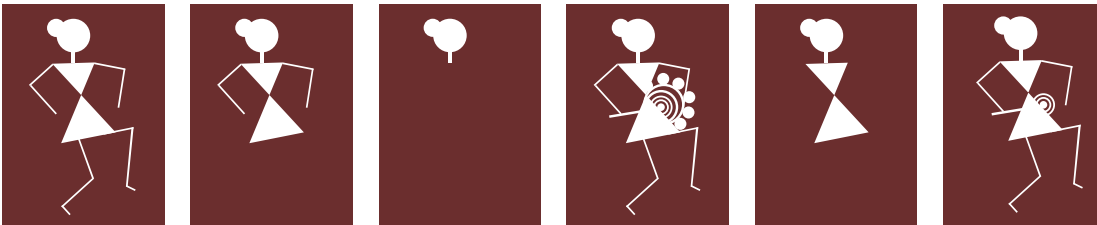
Deepa is Learning Warli painting. With the help of some cards, she wants to create the following picture:



She has to follow the step-by-step instructions shown on the cards, but she dropped them and now they are all mixed up!

Question

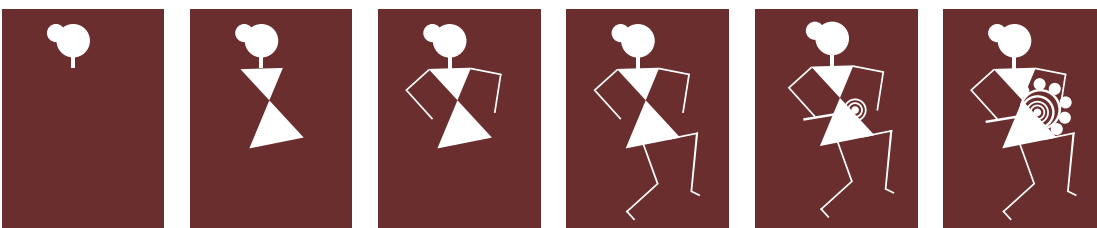
Put the cards back in the proper order.



EXPLANATION

Answer

The correct order of the cards is:



Explanation

Each card adds one more feature of the final picture.

BACKGROUND INFORMATION

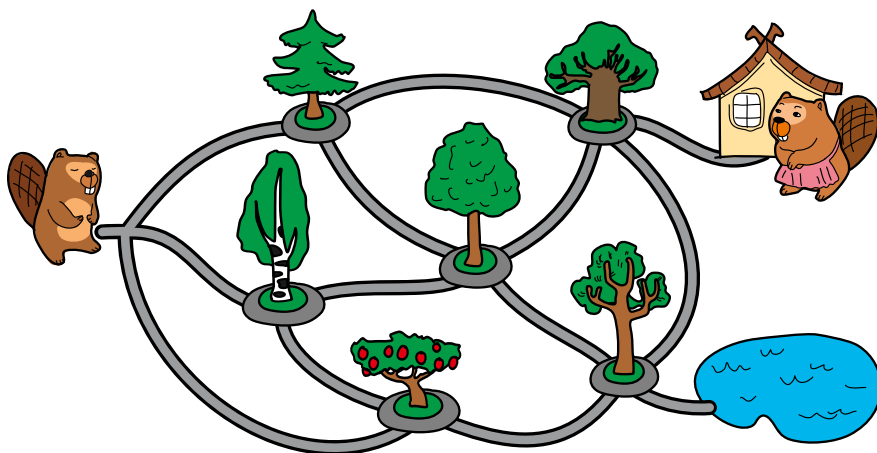
This task is a good illustration of the computational thinking skill pattern recognition, specifically finding similar or common elements.

To put the cards in order you have to compare pairs of cards to find the ones that are closest together in terms of pattern similarity. Then, understanding the rule for how one image is transformed into another (i.e. adding new white geometric shapes) you can order each pair.



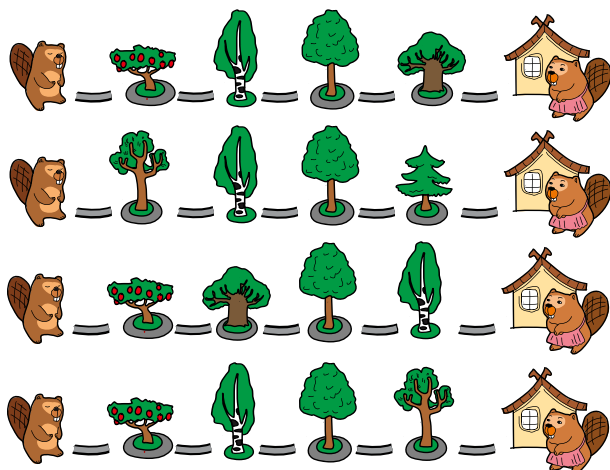
Trail to Grandma

A beaver is going to visit their grandma. There are several different trails to grandma's house.



Question

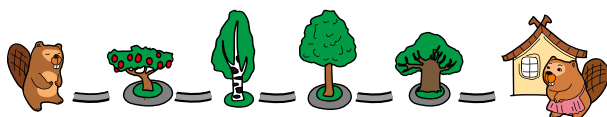
Which one of the trails below can lead the beaver to grandma's house?



EXPLANATION

Answer

There are several trails that lead to grandma's house. Of the presented options, there is only one correct answer:





Trail to Grandma - continued

Explanation

	Correct answer. The trail is marked by red.
	This tree cannot be the first tree passed by the beaver.
	This tree can be the first tree on a trip to grandma's house, but then it's not possible for a beaver to pass the following tree.
	This trail leads to a lake, not to grandma's house.

BACKGROUND INFORMATION

Motion planning, also path planning (also known as the navigation problem) is a computational problem to find a sequence of valid configurations that moves the object from the source to destination. The term is used in computational geometry, computer animation, robotics and computer games.



New Car

Christine wants a red, electric car. It needs to be able to drive at least 150km without needing to be recharged.

Shopper		Comet	
Colour: Red Range: 120km Fuel: Electric		Colour: Black Range: 250km Fuel: Electric	
AG12		IT Car	
Colour: Red Range: 200km Fuel: Electric		Colour: Red Range: 350km Fuel: Petrol	

Question

Which car should Christine purchase?

Shopper Comet AG12 IT Car

EXPLANATION

Answer

The AG12 fits all Christine's requirements.

Explanation

The Shopper cannot drive far enough on one charge. The Comet has the wrong colour.

The IT car is not an electric car.

BACKGROUND INFORMATION

The task involves statements (conditions/requirements) that must be evaluated (determined to be true or false) for a set of objects (cars). Conditions and their evaluation are an important part of programming and algorithmic thinking.

Conditions can be simple statements. However, more complex statements can be formed using logical operators such as AND, OR, NOT. This task uses the AND operator as both criteria need to be met to get the correct choice.



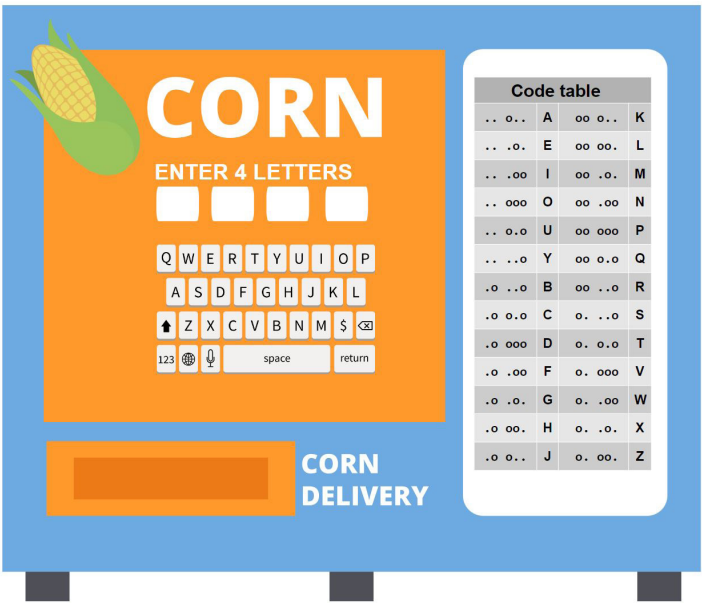
Golden Ticket

Maria, the armadillo, has a golden ticket that is worth four pieces of corn. Together with Juan, the capybara, they go to the vending machine to redeem it. On the way, they look at the ticket, but do not understand the code on it.



When they see the code table on the vending machine, everything becomes clear: They will have to figure out what four letters are encoded on their ticket.

Code table			
.. 0..	A	00 0..	K
.. .0.	E	00 00.	L
.. .00	I	00 .0.	M
.. 000	O	00 .00	N
.. 0.0	U	00	P
.. ..0	Y	00 0.0	Q
.0 ..0	B	00 ..0	R
.0 0.0	C	0. ..0	S
.0 000	D	0. 0.0	T
.0 .00	F	0. 000	V
.0 .0.	G	0. .00	W
.0 00.	H	0. .0.	X
.0 0..	J	0. 00.	Z



Question

What four letters should they enter to get their corn?

EXPLANATION

Answer

The code corresponds to letters R ST D.



Golden Ticket - continued

Explanation

Replacing each code on the ticket with a letter:

oo ..o	R
o. ..o	S
o. o.o	T
.o ooo	D

BACKGROUND INFORMATION

The code table that appears on the vending machine is a replica of Baudot's code. The Baudot's code is a binary code that uses crosses and dots. The Baudot code is a character encoding invented for telegraphy by Emile Baudot in the 1870s. It was used for teleprinter messages instead of the Morse Code.

In cryptography, this type of code table is called a *key*. A key can be used to *encrypt* a written message to a format that is impossible to read without the key. With the key, the message can be decoded back to a readable format. Encryption is used, for example, for digital signatures, verification to protect data privacy, Internet web browsing, and confidential communication such as credit card transactions and email.

Baudot's code was initially used by Clementina, which was the first computer used for scientific purposes to arrive in 1961 and was located in Pavilion I of the University of Buenos Aires. The instructions were given by rolls of perforated paper and the results were returned in the same way.



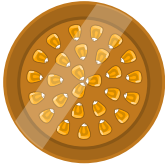
Strange Payment

Delilah wants to buy some rope in order to strengthen a dam. She can buy it at the local hardware store.

The clerk tells her that she has to pay 21 beavercoins for the rope.

Beavercoins are only available in the denominations of 1, 3, 9, and 27 beavercoins.

Today Delilah and the clerk each only have one coin of each denomination.

			
1	3	9	27

Question

How can Delilah pay the 21 beavercoins for rope?

She pays two 9 beavercoins and one 3 beavercoin.

She pays one 27 beavercoin and one 3 beavercoin, then gets back one 9 beavercoin.

She pays one 27 beavercoin and gets one 3 beavercoin and one 1 beavercoin back.

She cannot pay 21 in beavercoins.

EXPLANATION

Answer

The correct solution is that Delilah gives a 27 beavercoin and a 3 beavercoin to the clerk and receives a 9 beavercoin change. That way she pays $27 + 3 - 9 = 21$ beavercoins.

“She pays two 9 beavercoins and one 3 beavercoin” is incorrect as Delilah only has one 9 beavercoin to pay with.

“She pays one 27 beavercoin and gets one 3 beavercoin and one 1 beavercoin back” is incorrect as Delilah has paid 24 beavercoins in this scenario, not 21.

“She cannot pay 21 in beavercoins” is incorrect as the correct answer shows that it is indeed possible for Delilah to pay 21 beavercoins.

Continued on next page



Strange Payment - continued

There is no other possible solution for this problem. Each coin has three times the value as the next coin with lesser value. Delilah and the clerk only have one coin of each type. So starting from any value they may only add or subtract one coin's value each. In order to reach the value of the next coin, however, one of them would have at least two coins of the same type: only then one could give two (or more) coins to the other and receive (at most) one back. Because of this, each payment is unique and $27 + 3 - 9$ is the only solution.

BACKGROUND INFORMATION

The face value of the coins in this task is chosen on purpose. Each coin has three times the value as the next coin with lesser value. That results in face values of 1, 3, 9, 27, and so forth. These numbers are integer powers of 3: $3^0 = 1$, $3^1 = 3$, $3^2 = 9$, $3^3 = 27$, ... The situation of Delilah and the clerk is that she can either give one coin to the clerk (a payment of +1, +3, +9, +27), receive one from him (-1, -3, -9, -27), or not bother with a type of coin (+/- 0). This way any value from 0 to the sum of the coins available to Delilah (40 in this case) can be represented.

This type of setup is called a *numeral system*, in this case a *balanced ternary numeral system*. The numeral system is *ternary*, because it is based on the number 3. And it is called balanced, because not only the "digits" +Coin and 0 could be used, but also the "digit" -Coin. A payment can easily be represented by writing down for any type of coin whether Delilah gives one to the clerk, no coin of this type gets moved, or the clerk gives one to Delilah. A *place-value numeration* for this numeral system could use the digits 1, 0, and 1̄. As for the decimal system also, the last digit has the least value. So in the balanced ternary numeral system, Delilah would pay 11̄01 beavercoins.

The balanced ternary numeral system was actually used in 1958 in the Setun computer built in the Soviet Union by a team lead by Sergei Sobolev (1908–1989) and Nikolay Brusentsov (1925–2014). Although the balanced ternary numeral system has some technical advantages over the binary numeral system today virtually every computer uses, it isn't clear yet whether computers like that ever will be produced again.

Besides the historical use of the balanced ternary system in computers, this task shows that computer scientists sometimes have to combine complex theoretical concepts with practical applications in order to produce efficient computers. Today when boundaries of miniaturization are pushed further for instance when storing data using magnetism or when creating microprocessors, physics, chemistry, engineering, and mathematics have to be combined to create reliable products.



This question comes from
Japan

Years 3+4 Medium

Years 5+6

Years 7+8

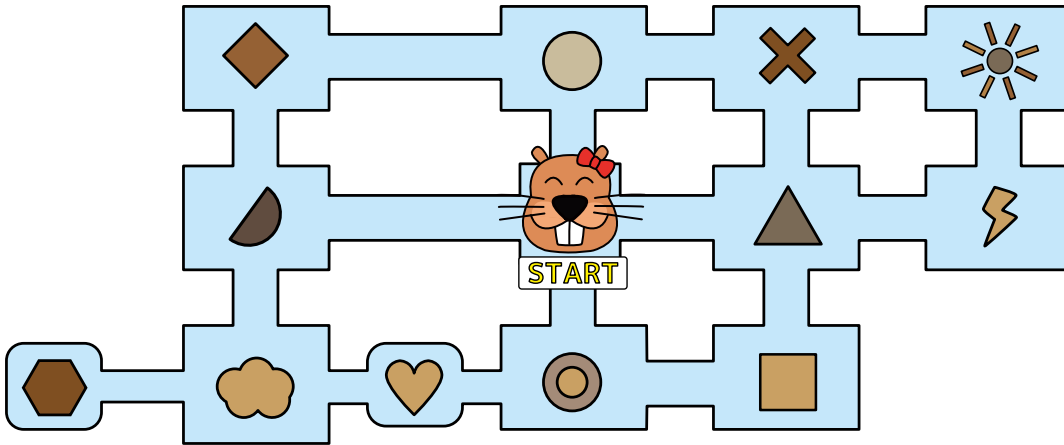
Years 9+10

Years 11+12

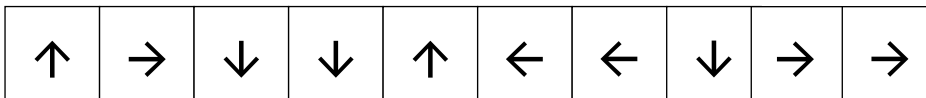


Where Is Beaver?

Bella wants to move from the START room to Bobby's room. Bella uses a map of the rooms. On the map, each room is marked by a symbol.



Bella moves by the following arrow sequence:



Each arrow tells Bella in which direction to move from one room to the next.

Question

Which symbol is Bobby's room marked by?



Continued on next page



This question comes from
Japan

Years 3+4 Medium

Years 5+6

Years 7+8

Years 9+10

Years 11+12



Where Is Beaver? - continued

EXPLANATION

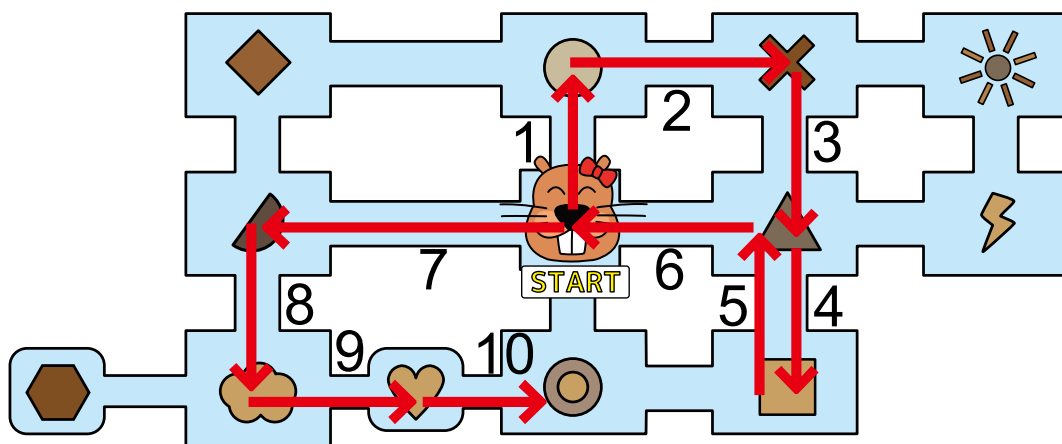
Answers

The correct answer is:



Explanation

Bella moves in the order of red arrows on the map.



BACKGROUND INFORMATION

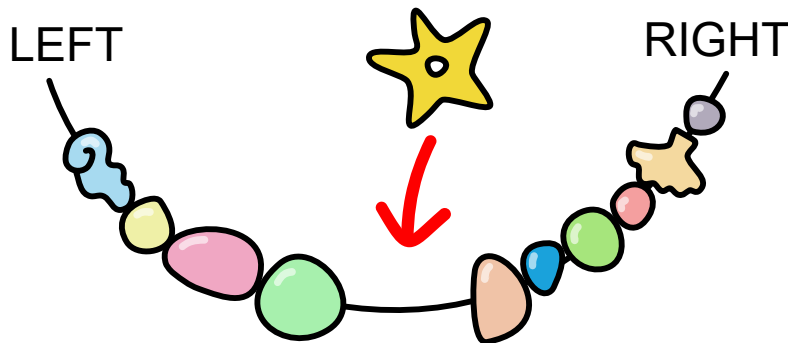
Each arrow in this Bebras task tells Bella what to do next, and the full sequence of arrows determines the way. This task asks programming basic concepts of sequence using arrows as commands and executing them in order from left to right.

Tracking movement of a pointer (Bella in this case) along its path is a very important thing in reverse engineering of a computer program to understand the path that this program follows. What are the symbols this pointer sees along this path, and where does it stop? It helps to understand what this computer program does.



Making A Necklace

Layla Likes to make necklaces with beads. When making the necklace, Layla uses one piece of string which cannot be cut. The necklace is created by adding or removing beads from the Left or right side.



Question

Layla wants to insert a star bead where the arrow points. She needs to remove the fewest number of beads. How many beads should Layla remove?

3 beads

4 beads

5 beads

6 beads

EXPLANATION

Answer

The correct answer is 4 beads.

Explanation

This necklace has no knots, so Layla can remove or insert beads from either side. If Layla removes the beads from the Left, Layla has to take out four beads to put a star-shaped bead where she wants. If Layla removes the beads from the right, six beads must be taken out.

BACKGROUND INFORMATION

An untied necklace string allows the beads to be removed or added on either side. And from either end of the necklace string, the Last bead have to be removed first. In informatics, this structure is called a double ended queue, as the elements can be removed or added from both sides.

If one end is tied, the beads can only be removed or added from the other side. In informatics, this structure is called a stack. Structures such as double ended queues and stacks are used in informatics to sequentially input, store and remove data.

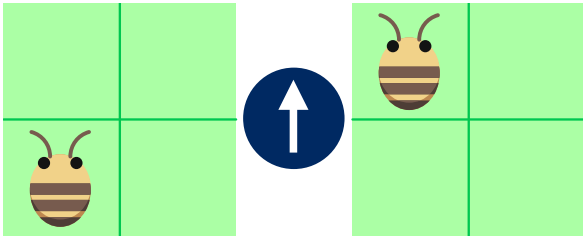
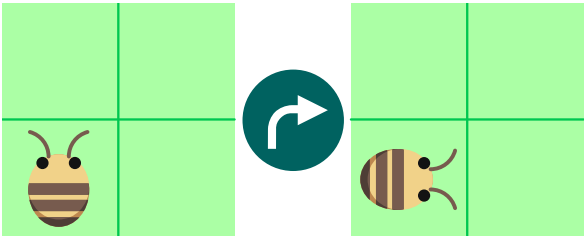


Journey to the Hive

The robotic bee knows these commands:

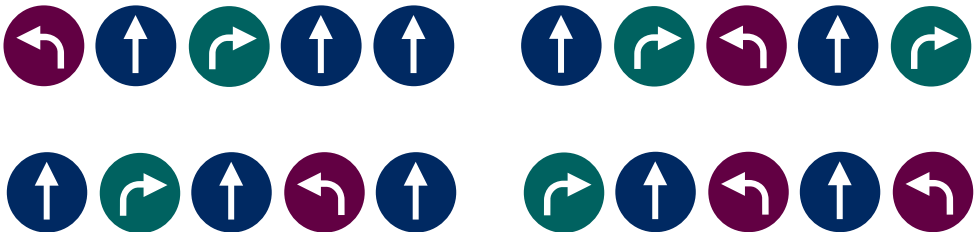
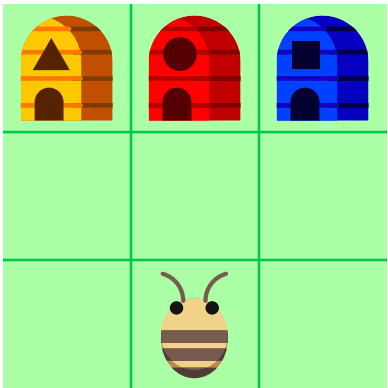
forward		(moves one square in its direction)
right		(turns right)
left		(turns left)

Examples



Question

Which commands will make the robotic bee go to the square with the yellow hive ?





Journey to the Hive - continued

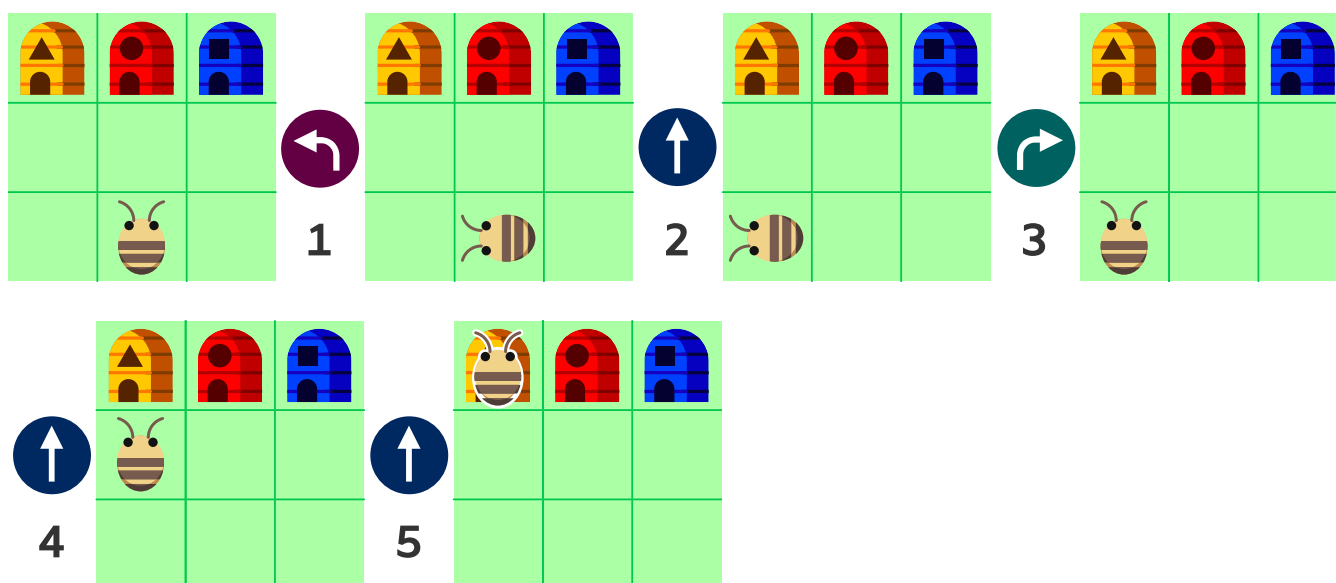
EXPLANATION

Answer



Explanation

This is what happens after each command:



The other answers are incorrect because:



will take the bee to the red hive.



will take the bee to the blue hive.



will take the bee to under the blue hive.

BACKGROUND INFORMATION

The robot bee's path is described by a sequence of arrows - commands that the robot runs one after another, changing its position. Running a sequence of commands is one of the basic ideas in programming. A different sequence of command arrows will take the robot bee on a different path. The sequence of commands is therefore important in a program.

The control of the bee in this problem is "relative". This means it is from the point of view of the bee. In absolute control, it means moving a box in a given direction, but in relative control, it means turning the bee in the appropriate direction without moving it.



Password

At Maggie's school, they have created the following rules about passwords:

- A valid password must have at least one lower case letter, at Least one upper case Letter, at Least one symbol, and at Least one digit.
- If you compare two valid passwords the Longer one is stronger.

The school suggests that you base your password on a song, by using some Letters from the song and then adding symbols or digits as needed. For instance, if the song is 'I have a dream' from the year 1979, you could come up with the password 'iHad_1979'.

Maggie wants to create a password based on the song 'Think about things' from the year 2020. She comes up with four possible passwords.

Question

Which of these four is the strongest valid password according to the school's rules?

!TAVOUTT2020!!

Th1nk,Thln9s2

thinkabout20!

2020TaT!

EXPLANATION

Answer

The correct answer is: Th1nk,Thln9s2

Explanation

It is the longest password that has a symbol, upper case letter, lower case letter, and digit.

BACKGROUND INFORMATION

It is becoming more and more important to create strong passwords. Too many people get hacked by not creating a strong password.

Sometimes passwords are used by computers when communicating directly. In this case the password is often called a 'key' or a 'secret'. Computers can easily remember very long strings of random characters, so the passwords get made more secure just by making them longer.

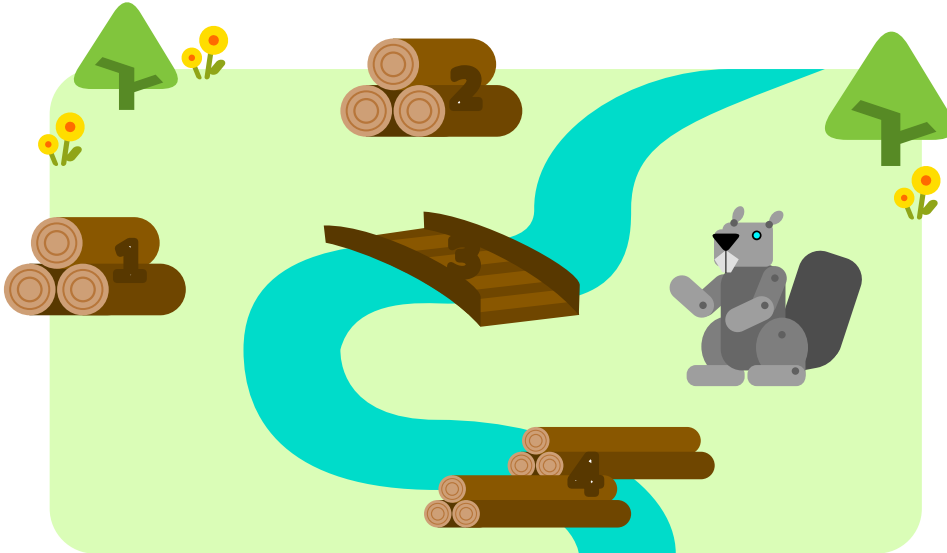
Of course the person must be able to remember the password as well. If you cannot remember the password what good does it do you! So the problem that you are trying to solve is how to create a password that is both very strong and easy to remember at the same time.

There are many different mechanisms for creating strong passwords. The mechanism from this task where you use a song is only one of these ways. Can you come up with other ways as well?



Robot Beaver

The beaver, Bob, made a robot that would build dams instead of him, he named it Robo. Robot Robo could only cross the river by going over the bridge and he can only carry one log at a time.



The robot Robo can perform these commands:

Tx – go to x, where x can be 1, 2, 3, or 4

G – grab,

D – drop.

Example

For example, for Robo to take the log from place number 1, and put it on the bridge (place number 3), Bob should write this program: T3 T1 G T3 D.

The beaver needs a program that would instruct Robo to take one log from each location (1 and 2) and place it in the dam it is building (place number 4).

Question

Which program is correct?

T3 T1 G T3 T4 D T3 T1 G T3 T4 D

T3 T1 G T3 T4 D T3 T2 D T3 T4 D

T3 T2 G T3 T4 D T3 T1 G T3 T4 D

T3 T1 G T3 D T3 T2 G T3 D T4 T3



Robot Beaver - continued

EXPLANATION

Answer

The correct answer is T3 T2 G T3 T4 D T3 T1 G T3 T4 D.

Explanation

In answer T3 T1 G T3 T4 D T3 T1 G T3 T4 D, robot Robo will take two logs from place 1 and none from place 2. There is no command T2.

In answer T3 T1 G T3 T4 D T3 T2 D T3 T4 D, robot Robo does not grab a log from place 2, but tries to drop a log there. The bold letter represents the wrong drop at place 2: T3 T1 G T3 T4 D T3 T2 D T3 T4 D.

In answer T3 T1 G T3 D T3 T2 G T3 D T4 T3, Robo will grab a trunk from place 1, and drop it to place 3. The bold letter represents the drop at place 3, instead at place 4: T3 T1 G T3 D T3 T2 G T3 D T4 T3.

BACKGROUND INFORMATION

An algorithm is a finite sequence of well-defined instructions, typically used to perform a computation or solve a class of specific problems. Algorithms resemble recipes. Recipes tell you how to accomplish a task by performing a number of steps.

Algorithms can be expressed in many kinds of notation, including natural languages, pseudocode, flowcharts, drakon-charts, programming languages or control tables. Natural language expressions of algorithms tend to be verbose and ambiguous, and are rarely used for complex or technical algorithms. Try to explain in words, using natural language, to your friend how to do some daily work, for example, how to brush your teeth, or how to spread pancake with cream. During the game the friend should „forget” how particular activities have worked so far and have the right to interpret each of your instructions in their own way. This game can be very interesting.

Pseudocode, flowcharts, drakon-charts and control tables are structured ways to express algorithms that avoid many of the ambiguities common in the statements based on natural language.

The word “algorithm” comes from the name of Arabic mathematician and writer Muhammad ibn Mūsā al-Khwārizmī, but currently the word is strongly associated with computer science.

In computer programming, debugging is the process of finding and resolving bugs (defects, errors, or problems that prevent correct operation) within computer programs, software, or systems.

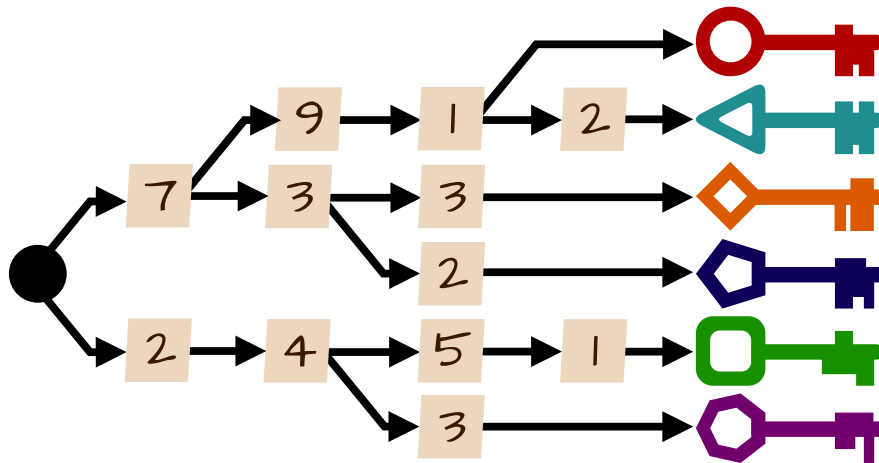


Pool Party Lockers

The Beavers are going to a pool party!

After putting their valuables in lockers, each beaver wrote down their locker number on a sheet of paper. Then they entrusted both the key and this sheet to the Old Beaver for safekeeping.

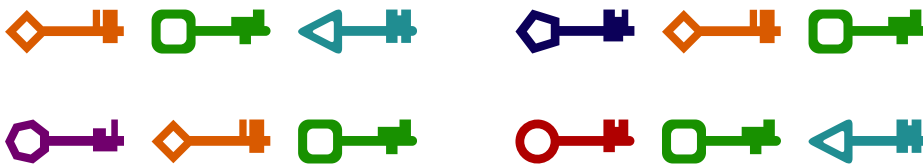
Old Beaver split the given locker numbers into digits and made the picture. For example, for the locker number 243 the Old Beaver draws the digits 2, 4, and 3 one by one and then connects the digits to the purple key (see at the bottom of the picture).



Later in the evening, the beavers found out that water had gotten on the sheet of paper and smudged some of the written digits.

Question

Knowing that the keys for locker numbers 73X, 24XX, 7X1X are arranged in order, which one of the following sets of keys is the correct set for these lockers?



EXPLANATION

Answer





Pool Party Lockers - continued

Explanation

The third locker number has 4 digits and starts with 7. Therefore, it can only be the  key. Similarly, the second key needs to be , as it starts with 2 and has 4 digits. The first key can be either  (733) or  (732) as both start with 73. Therefore the correct answer is   

An alternate correct answer could have been   , but this choice is not available.

Answer    is wrong since  corresponds to a 3 digit number.

Answer    is wrong because the number of , starts with digit 2 instead of 7.

Answer    is wrong because the number of  has the second digit 9 instead of 3.

BACKGROUND INFORMATION

The graph drawn by the Old Beaver represents so called Trie (comes from ReTrieval), a tree data structure used to store key-value pairs, similar to a hash table. The Trie is constructed as directed tree graph, where each edge is labeled with a letter. A value is obtained by following the graph letter by letter, just as in the presented example.

The advantage over hash tables, where a value can be obtained only with the exact match of the key, is that Trie allows searching using a prefix, and finding all the possible matches. This is very useful in auto-completion tasks. The structure can also be extended to suffix search in addition to prefixes.

Trie is used for storing dictionaries, especially due to the common root words and their derivation. This is applied to indexing task inside search engines. It can also be used for sorting, as its internal representation is similar to Radix Sort. Other usages include pattern recognition tasks, drawing close to automata theory.



This question comes from
Syria

Years 3+4 Hard

Years 5+6

Years 7+8

Years 9+10

Years 11+12



Moving Around

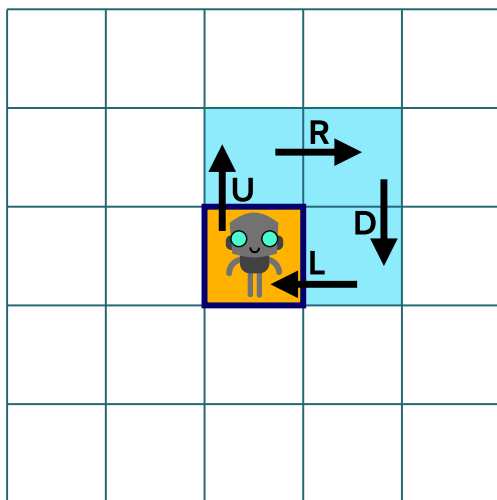
A robot moves on a grid, from field to field, and is controlled by these commands:

L - Left, R - Right, U - Up, D - Down

When executing a command sequence, the robot always starts in the middle field of the grid.

Example

See the command sequence URDL. This sequence will make the robot return to its starting field.



Question

Which of these command sequences will make the robot return to its starting field as well?

ULDDRRUR

DDRURUL

ULDDRRUL

LURRRDLD

EXPLANATION

Answer

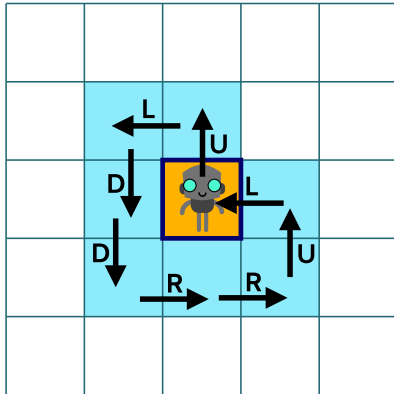
The correct answer is ULDDRRUL.

Continued on next page

Moving Around - continued

Explanation

This command sequence makes the robot move as shown and finally return to its starting field.



The other command sequences will not make the robot return to its starting position:

ULDDRRUR	DDRURUL	LURRRDL

A way to check a program would be to count the number of each of the commands; in order to get back to the starting position, the number of Rs should equal the number of Ls, and of course the same demand is valid for the number of Us and Ds.

BACKGROUND INFORMATION

Executing a command implies a state change for the robot, where the state is its position on the grid. Commands Land Rare reverse operators, they undo the previous state change. Getting back to the starting position means that the net result is no state change at all.

Debugging the presented programs here means checking if the robot will return at its starting position.

In computer programming and software development, debugging is the process of finding and resolving bugs (defects or problems that prevent correct operation) within computer programs, software, or systems.

Debugging tactics can involve interactive debugging, control flow analysis, unit testing, integration testing, log file analysis, monitoring at the application or system level, memory dumps, and profiling.

Many programming languages and software development tools also offer programs to aid in debugging, known as debuggers.



Birthday Party

Maxie the beaver is planning a birthday party. He makes a to-do list with tasks to be done before the party, shown below:

To-Do List	 Making sure how many beavers are coming	 Buying snacks	 Picking a date	 Estimating the cost	 Choosing a venue
Tasks to Be Done Beforehand		 	None	  	

Maxie realises that some tasks need to be completed before other tasks. For example, picking a date before confirming how many beavers are coming.

Question

What is a correct order to complete the tasks?



EXPLANATION

Answer

The correct answers are:

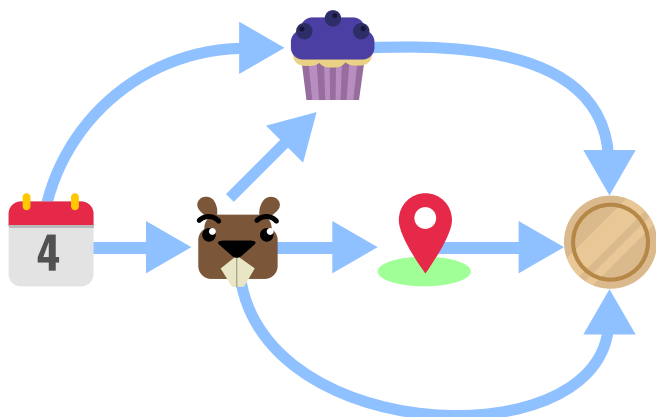




Birthday Party - continued

Explanation

If we take the information from the table and use arrows to represent the order of carrying out tasks, we will get the following picture:



In this picture, each circle represents a task, and each arrow represents which task should be done before the other. Based on the picture, we can see that there are only two possible solutions for this problem. Picking the date always has to be the first task to be done. Counting people needs to be the second task. Either buying the snacks or choosing a venue can be placed next. The last task will always be estimating the cost.

BACKGROUND INFORMATION

The algorithm for sorting these tasks is called a *topological sort*. Computers are equipped to sort items with numerical order. However, when items to be sorted do not possess any numerical characteristics, or when we want to sort items based on non-numerical characteristics, we have to let the computer know the sequence relationship between each item. We can often use a graph like this to represent the sequence.

In our daily life, topological sorts are applied in scheduling a series of jobs or tasks. For example, when playing online games, we have to pass a certain level before we can play the next level.

Bebras Challenge 2023 Round 2

Years 5+6



Geography

Poppy has access to a database of the United Nations. This database contains the following data about each country:

Country	Area [km ²]	Population [2015]	Density [1/km ²]	Life expectancy
	2,780,400	45,864,941	16	78
India	3,287,263	1,339,330,514	407	70
Tuvalu	26	11,448	440	68
Mexico	1,964,375	130,207,371	66	77
China	9,596,960	1,397,897,720	146	76
Germany	357,022	79,903,481	224	81
Thailand	513,120	69,480,520	135	77
Nauru	21	9,770	465	68
Japan	377,915	124,687,293	330	85
United States	9,833,517	334,998,398	34	80

(Note: this table is only illustrative, it is not accurate.)

Poppy wants to compare how many more people live in other countries than hers.

Question

What data column does Poppy need to use to find this out?

Population

Life expectancy and area

Density and life expectancy

Life expectancy

EXPLANATION

Answer

The correct answer is population.

Continued on next page



This question comes from
Czech Republic

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



Geography - continued

Explanation

We do not need any data other than the population of the two states to compare them.
It doesn't matter if we understand what the other answers mean.

BACKGROUND INFORMATION

There is a lot of data around us and computers can process that mass data and give us interesting results. We need to understand which data is necessary as input for data processing and which is not useful to lead to the right results.

This task is about abstraction because we need to find which data is not important for solving an applied task by computing.



This question comes from
Japan

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



Secret Message

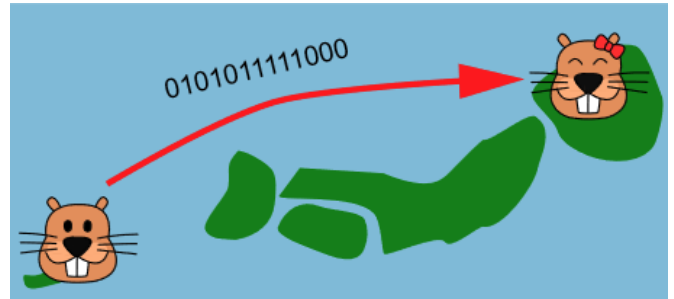
One beaver decided to send a message to a distant friend. They had arranged to only use the symbols 0 and 1 to send messages and had agreed to substitute a, b, e, and s with sequences of 0's and 1's as follows:

a=110, b=111, e=10, s=0

They had also agreed to write the symbols from left to right, as we normally do when writing words.

He sent the following message:

0101011111000



Question

Which is the correct translation of the sent message
into letters?

seebee

seebass

seebabs

seebees

EXPLANATION

Answer

The correct answer is seebass.

Explanation

In this problem, the correct answer can be found by analysing the sent message from left to right: as soon as a subsequence of 0's and 1's is found that has a corresponding letter, that subsequence should be substituted by the corresponding letter and the analysis should proceed from the next symbol.

So, the message in the task starts with 0, which corresponds to 's'; then we have 10, which corresponds to 'e'; then again 10, thus another 'e'; then 111, which corresponds to 'b'; then 110, which corresponds to 'a'; then 0, which corresponds to 's'; and finally again 0, which corresponds to 's'; thus yielding: seebass.

BACKGROUND INFORMATION

This task is about coding and decoding of information, something very common in informatics, in particular coding with sequences of 0's and 1's, which are the symbols used in computers to represent and process all information.

A coding system specifies how data in some representation system is represented in another representation system: the coding in this task specifies how some letters are represented as sequences of 0's and 1's. The coding system used here is called *Huffman code*, a so called *prefix code*, that has the property that it can be decoded from left to right unambiguously, and which assigns shorter codes to characters that occur more frequently.

Indeed in the correct answer, "see bass," 's' appears three times, 'e' twice, 'a' once, and 'b' once. Therefore, we assigned "0" to 's', "10" to 'e', "110" to 'a', and "111" to 'b'.



The Bay of Beaver Programmers

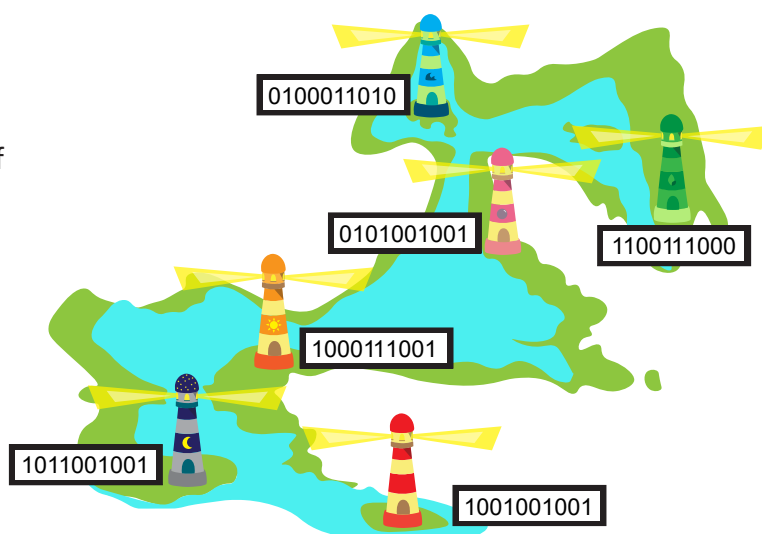
In the Bay of Beaver Programmers, the lights on each lighthouse are turned on and off in a unique way.

Each lighthouse has its own unique **code** describing the way in which it turns on and off the light so that sailors can easily determine which lighthouse they are observing.

For example, if a lighthouse has code “1011001001” it describes that during the first 10 seconds of working, each second light turns on and off as follows:

on, off, on, on, off, off, on, off, off, on.

Every 10 seconds, the light on the lighthouse turns on and off in the same way.



Question

If all lighthouses start working at the same time, in which second will all the lighthouses light up together for the first time?

EXPLANATION

Answer

The answer is 7.

Explanation

If we write the lighthouse's codes one below the other, it will be easy to notice that in all the codes there will be a “1” (“1” determines that the light will be on) in the seventh second (read digits from left to right):

```
0100011010
0101001001
1100111000
1000111001
1011001001
1001001001
```

BACKGROUND INFORMATION

In informatics, we often encode certain events or statuses with digits, letters, numbers written in a different numeral systems, etc. For example, if an event occurs, we can assign the appropriate value to the variable in our computer program in order to indicate that event happened or the status was reached. Later, by examining the current value of this variable in our program, the corresponding branches of the program will be executed.

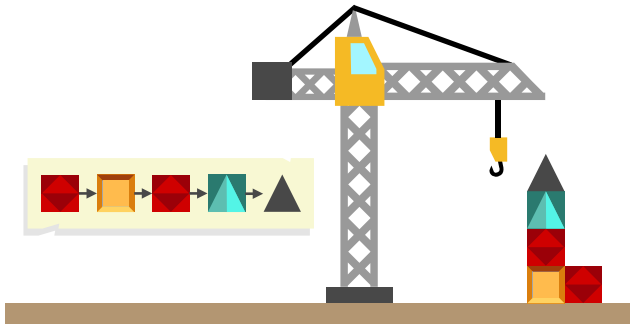


Building Instruction

In Beaverland, block houses are built by following instructions.

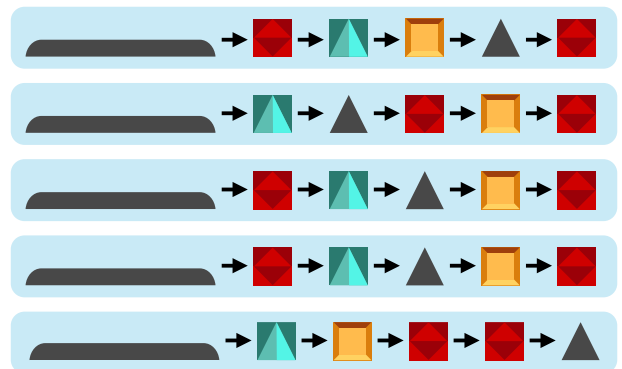
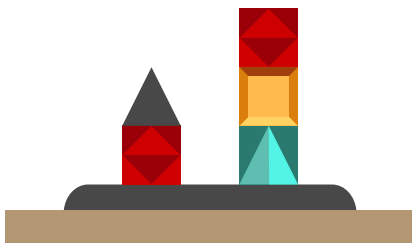
A crane takes each block in the order shown in the instructions. It then puts it on the base or on top of another block.

For example, the house shown in this image could be built following the instructions below:



Question

Which of the building instructions below *cannot* be used to build the house below?



EXPLANATION

Answer  →  →  →  →  → 

Explanation

In the house shown, the triangle block is placed on top of a red block, so at least one the red blocks must be placed before the triangle block. This is not the case with the instructions above. All other instructions can be followed to build the house shown.

BACKGROUND INFORMATION

Computers operate by following a list of instructions, called a *program*, that has been written to carry out a particular task. Programs are written in languages that have been specially designed to tell computers what to do with a limited set of instructions. Some languages are more suitable for some purposes than others.

Regardless of what language they use, programmers must become adept at specifying exactly what they want the computer to do. Unlike human beings, a computer will carry out instructions to the letter even if they are patently ridiculous!



This question comes from
Vietnam

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

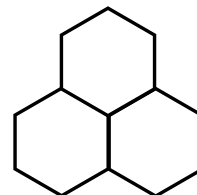


Colourful Tower

Sam has 30 hexagonal puzzle pieces,
10 of each of these three types:

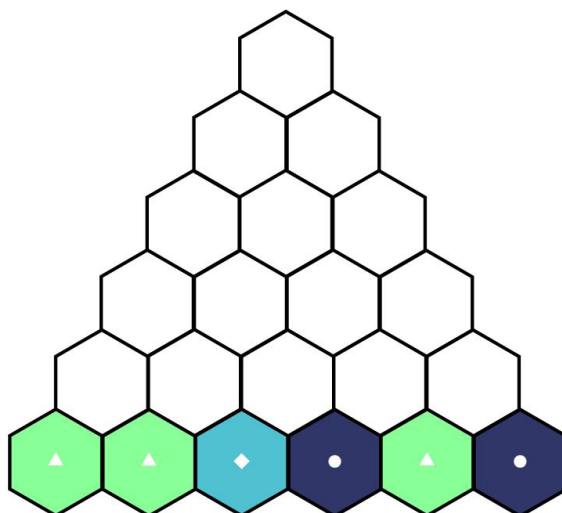


When Sam places three pieces in the following pattern, the three pieces must all be the same colour OR all different colours.

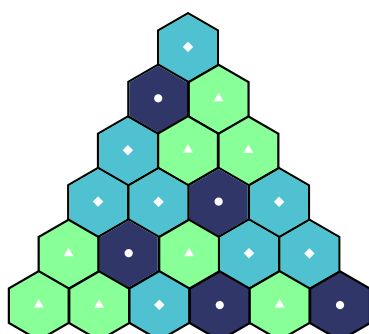


Question

Fill in all the white tiles with the correct colours.



EXPLANATION



Answer

Continued on next page



Colourful Tower - continued

Explanation

When we know the colours of two pieces beside each other, we know what the colour of the piece above them must be. For example, consider the bottom left unknown piece. Since the two pieces below it are green, it must also be green. On the other hand, the piece to its right must be blue because the two pieces below it are green and yellow. We can use this idea to work from the bottom to the top of the tower. There is only one possible answer.

Note that we can work in any order on each row of the tower.

BACKGROUND INFORMATION

We found it important to identify which of two cases apply (i.e. two pieces are the same colour or a different colour). Then we repeat this until all the colours of the tower are known. This sequence of steps forms an algorithm that can be used to solve the problem. If we wrote a computer program to implement the algorithm, identifying the cases and taking the right action in each case might correspond to a conditional statement. The repetition of this might correspond to a loop.

Designing and implementing algorithms is one of the most central and important tasks performed by computer scientists. For both of these activities, critical skills include identifying patterns (e.g. arrangements of three pieces and colours) and making careful use of logic (e.g. breaking into cases).



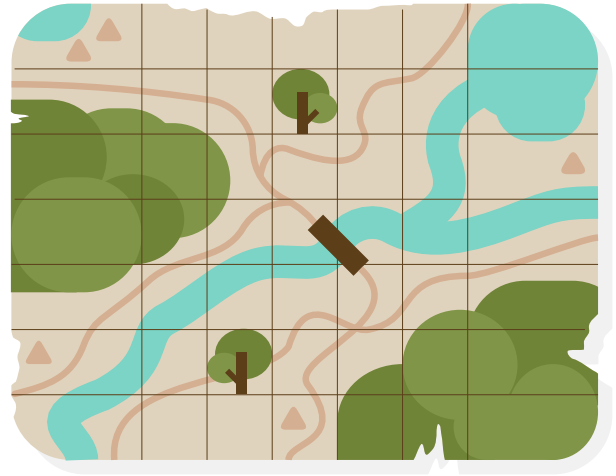
Pantry Map

Graeme found two good hiding places for his food.

He draws a map, divided into squares, to help him remember where his food is hidden. He wants to mark the squares, where his hiding places are, on his map with an "X". However, if his rival Adam finds the map, he would know where to look!

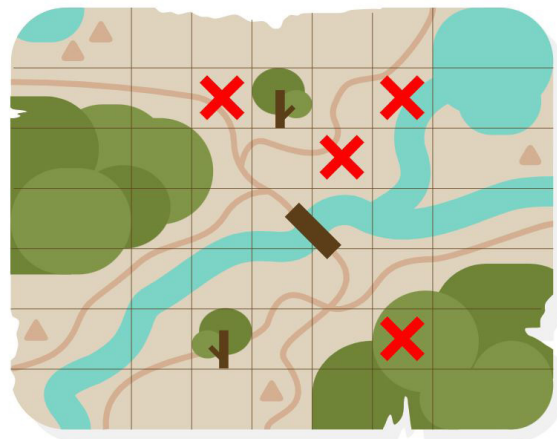
To confuse Adam, Graeme adds some more "X"s, in other squares of the map, making sure that the total number of "X"s in each row and each column is even. (Note: 0 is also an even number.)

Then Graeme erases the two "X"s showing where his hiding places are. The resulting map is shown below.



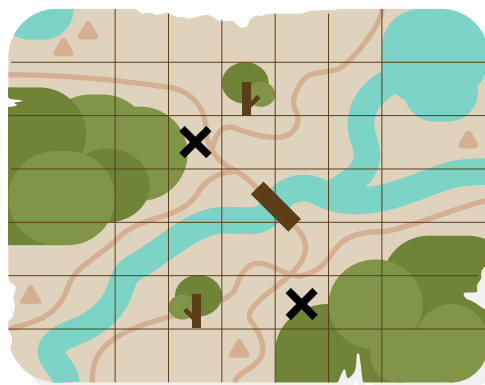
Question

In which two squares has Graeme hidden his food?



EXPLANATION

Answer





Pantry Map - continued

Explanation

To find them, we look at the original map and find that there are two rows and two columns where the number of “X”s is not even: Rows 2 and 5 (from the bottom), and columns 3 and 5 (from the left).

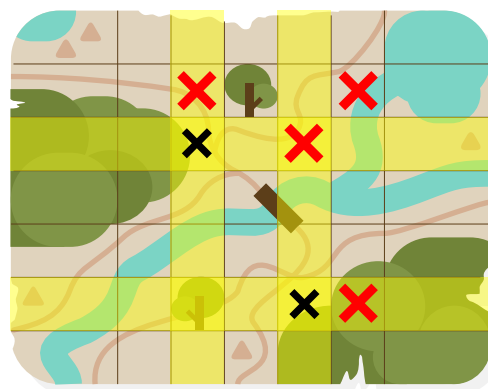
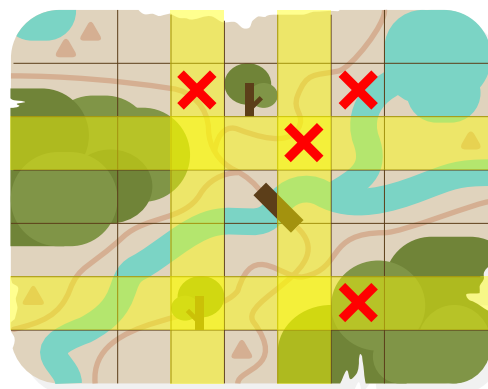
This shows the rows and columns involved in yellow:

The two erased “X”s showing the real hiding spots must have been in them! So we must find a way to add them back into these rows and columns to restore the even constraint.

There are 4 squares at the intersection of the yellow rows and columns. The upper red intersection square is marked with an “X” already: it cannot be a hiding place, since we know Graeme has erased the “X”s from the true hiding places. So in order to recover an even number of “X”s on that second row, we have to add it in the upper left yellow intersection square: we now know one hiding place.

The second hiding place can be determined as follows: Once we’ve found the first one, we cannot place the second one in the bottom left red intersection square, as this would result in 3 “X”s in the second column, and 3 is not even. The only choice left is the lower right intersection square, which gives the answer shown above.

This shows the map *before* Graeme erased “X”s, with an even number of “X”s in each row and column:



BACKGROUND INFORMATION

Graeme is using a trick related to the concept of *parity bit*, which is used a lot in computer science, as part of a larger set of techniques known as *error-detecting and error-correcting codes*. The idea is that whenever we have to store or transmit data as a series of bits, 1s and 0s, we add extra bits which help us detect if transmission or storage errors have occurred — typically, if a bit has flipped: for instance, if it was transmitted as 1 and erroneously received as a 0.

For example, if we use a simple error-detecting code, a parity bit would be added so that the number of 1’s was even. 0110101 would have a 0 added to become 01101010. (keeping the number of 1 bits even). If the 2nd bit was flipped and the message now transmitted 00101010, then this received message does not meet the even parity requirement (3 bits are 1’s). Note that if two bits were in error, this method may not detect a problem.

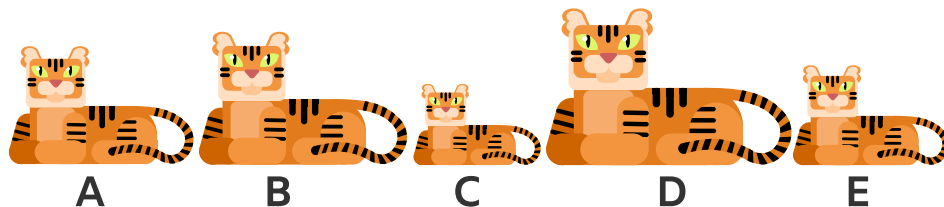


Rank

The Bebras Store has ordered tiger dolls to celebrate the Chinese tiger new year.

Jack put the dolls randomly on the shelf (as shown in the picture below). Mark thinks this looks too messy.

He asked Jack to rearrange the tiger dolls from shortest to tallest (from left to right). Jack can only pick up and switch two dolls at a time.



Question

What is the smallest number of times Jack has to switch two dolls to put them in the correct order?

3

4

5

6

EXPLANATION

Answer

The answer is 3.

Explanation

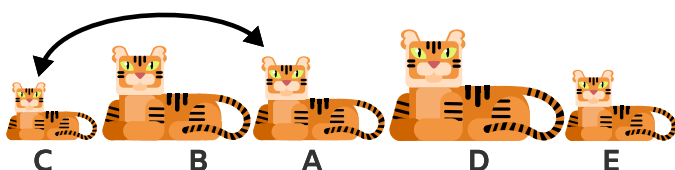
Firstly, Jack has to decide where he is going to start his sort order. (shortest first, tallest first, etc)

One example is starting with the shortest doll: Before each switch, Jack has to find the shortest doll from the unsorted tiger dolls, put it at the beginning of the unsorted sequence (its correct position) and repeat this until all of the tiger dolls are sorted correctly.

Starting positions:



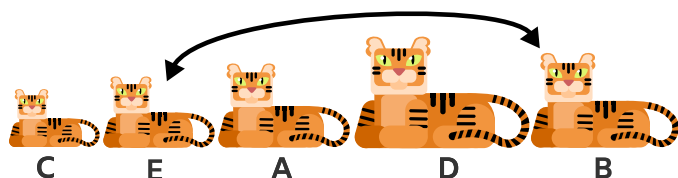
Switch 1: Doll C is the shortest and has to go to the front of the row, so he switches doll A with doll C.



Switch 2: Doll E is the second shortest, so he has to switch doll B with doll E to put it in its correct place right after doll C.

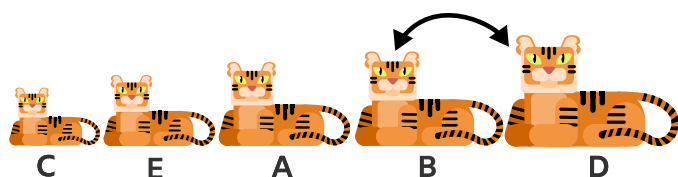


Rank - continued



Doll C and E are already in their correct position. From the three dolls left (A, D and B) doll A is the shortest doll and it is already in the correct position so he does not have to make a switch.

Switch 3: Dolls C, E and A are already sorted correctly, only D and B still have to be sorted. Doll B is the shortest doll of the two, so he switches doll D with doll B to put it in its correct place right after doll A.



All dolls are now sorted from shortest to tallest.

BACKGROUND INFORMATION

Selection sort might be one of the simplest sorting algorithms in computer science. How does it work? First, you are supposed to select the smallest (or largest) element from all the unsorted elements and place it at the beginning of the sequence. Second, among the remaining unsorted parts, you have to find the smallest (or largest) piece, and put it at the end of the sorted sequence. Repeat the second step until all of the elements are sorted, and finally you will get a low-to-high (or high-to-low) sequence.



Forest Party

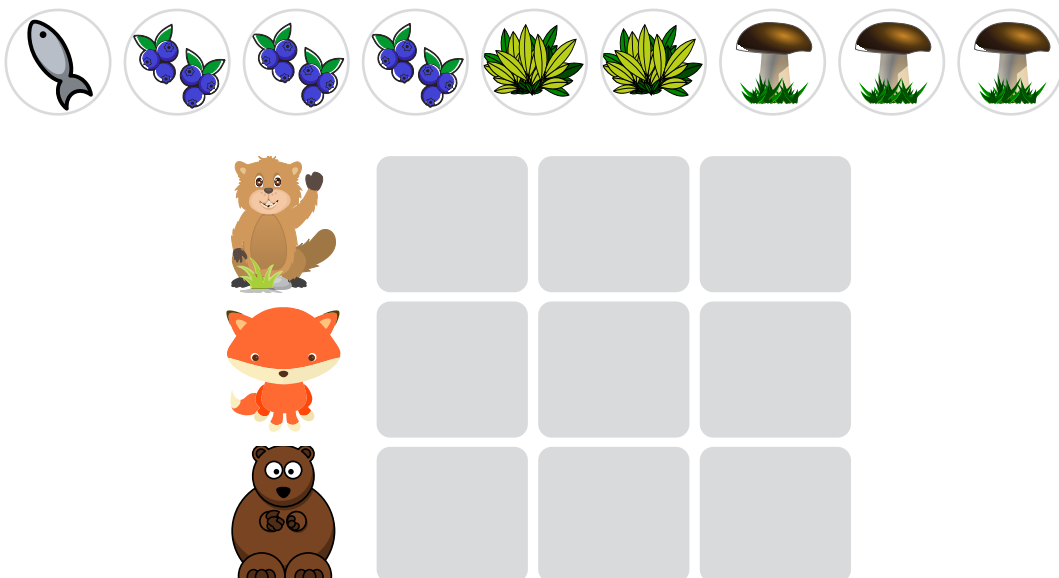
Betty Beaver, Fiona Fox, and Bobby Bear are having a party. The table shows which foods each friend can eat.

	 Leaves	 Fish	 Mushrooms	 Berries
 Betty Beaver	Yes	No	No	Yes
 Fiona Fox	No	Yes	No	Yes
 Bobby Bear	No	Yes	Yes	Yes

They have nine food portions at the party. Each friend should get three portions of food.

Question

Divide the food so that each friend gets three portions of food.



Continued on next page

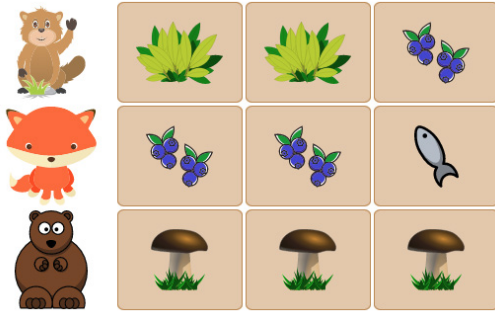


Forest Party - continued

EXPLANATION

Answer

The correct answer is:



Explanation

To solve this task, one could start with Betty Beaver since Betty is the only friend that eats leaves. This means that all leaves should go to her. The only other food she eats is berries so her third food has to be berries. Next, Bobby Bear is the only friend that eats mushrooms, so all three mushrooms should go to him. Fiona Fox will get what is left over: two berries and the fish.

There are the other way of solving this problem, for example one could start with Fiona Fox and observe that since Fiona only eats fish and berries, then she must get at least two portions of berries and her third portion will be either fish or berries. Betty Beaver is the only friend that eats leaves. This means that all leaves should go to her. The only other food she eats is berries so her third food has to be berries. Because there are no berries left, now we know that Fiona Fox's third food has to be fish. Bobby Bear will get what is left over: the three mushrooms.

BACKGROUND INFORMATION

Solving this task (a constraint satisfaction problem) requires the computational thinking skills of logic and decomposition.

Logic: This task shows an example of a logic property table and the challenge is how to read it. There is a clear answer, yes or no, for each animal and food item telling whether they eat that type of food or not. Also, each food portion can only be allocated to one friend, and this allocation has to satisfy the logic table. By allocating the easy food portions first, we can generate new knowledge that allows us to allocate others.

Decomposition: This task is an example of decomposition *breaking into parts*. Each food portion is allocated independently, although the order that we allocate the portions is important so that we find an allocation that satisfies the logical constraints of the task.



Colour the Frog!

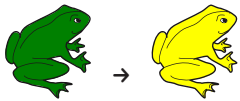
Lulu is learning to write programs that change the colour of images. She wants to turn her frog green.

She has written four different programs for changing the colour of the frog. Each program consists of a sequence of 'if-then-else' or 'if-then' statements. Each of the steps are carried out in order, one after the other. Each step will change a frog's colour once.

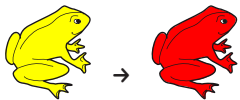
For example, for the following instruction:

IF the frog is **green** **THEN** colour it **yellow** **ELSE** colour it **red**

If the frog was green, this step would turn it yellow.



If the frog was yellow (or any other colour that isn't green), this step would turn it red.



She wants to be sure that the frog is green when the program is finished, regardless of the frog's colour beforehand.

Question

Three out of four of Lulu's programs will ensure the frog is ultimately coloured green. Which program, when finished running, does **not** ensure the frog will be green?

IF the frog is **red** **THEN** colour it **yellow** **ELSE** colour it **green**

IF the frog is **yellow** **THEN** colour it **red**

IF the frog is **NOT yellow** **THEN** colour it **green**

IF the frog is **red** **THEN** colour it **yellow**

IF the frog is **NOT red** **THEN** colour it **green**

IF the frog is **yellow** **THEN** colour it **red** **ELSE** colour it **green**

IF the frog is **yellow** **THEN** colour it **green**

IF the frog is **NOT yellow** **THEN** colour it **red**

IF the frog is **red** **THEN** colour it **green** **ELSE** colour it **yellow**

IF the frog is **yellow** **THEN** colour it **green** **ELSE** colour it **red**

IF the frog is **red** **THEN** colour it **green** **ELSE** colour it **yellow**

Continued on next page



Colour the Frog! - continued

EXPLANATION

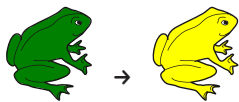
Answer

Explanation

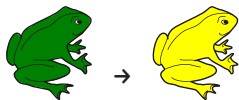
If at the beginning the frog is yellow, following the first **if-then-else** instruction colours it green

IF the frog is **yellow** THEN colour it **green** ELSE colour it **red**

IF the frog is **red** THEN colour it **green** ELSE colour it **yellow**



after which the condition “the frog is red” is false and it will be coloured yellow again.



If you run this program:

After the first **if-then-else** instruction, the frog is either yellow or green. After the second instruction it is either red or green. Therefore the last condition “the frog is not yellow” is true and it will therefore be coloured green.

IF the frog is **red** THEN colour it **yellow** ELSE colour it **green**

IF the frog is **yellow** THEN colour it **red**

IF the frog is **NOT yellow** THEN colour it **green**

If you run this program:

When the second instruction is done, the condition “the frog is not red” will be true, so the frog will be coloured green. After this, the third instruction (if-then-else) colours it green again. We don’t have to worry about a frog being yellow going into the third instruction because second instruction all yellow frogs have already been turned green.

IF the frog is **red** THEN colour it **yellow**

IF the frog is **NOT red** THEN colour it **green**

IF the frog is **yellow** THEN colour it **red** ELSE colour it **green**



Colour the Frog! - continued

If you run this program:

After the second instruction, the condition “the frog is not yellow” will be true, so the frog will be coloured red. Following this, the third instruction (if-then-else) colours it green.



BACKGROUND INFORMATION

Imperative programming is a way of providing instructions to a computer on how to deal with situations without demanding a particular outcome. This can be achieved by writing a sequence of *conditional statements*. These statements change a program’s *state*, and must be actioned in the given order.

A conditional statement has the form **if C then A else B** or the form **if C then A**.

It should be noted that the two scripts

if C then A else B

and

if C then A;

if not C then B

do not always have the same meaning.

In the second case (where the condition is evaluated twice), the actioning of instruction A could change the state of the system in such a way as to make condition C false. This would cause – immediately afterwards – the actioning of instruction B too. However, in the first case, one and only one of the two instructions is actioned.



Robot Factory

A factory in Beaver City makes robots that can do household chores. Robots have one head design, one body design, and one leg design. For each robot, there are 3 different heads, 4 different bodies and 3 different sets of legs to choose from.

An automatic machine builds the robots, choosing the parts (head, body, legs) according to these lists:

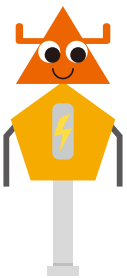
Head	Blue → 	Green → 	Orange 	(repeat)	
Body	Circle → 	Square → 	Triangle → 	Pentagon 	(repeat)
Legs	Two legs → 	Four legs → 	One leg 	(repeat)	

Each day the machine starts the same - the first robot is combined from a blue head, a circle body and two legs.

Last night there was a power outage at the factory that caused the machine to malfunction and change. The machine still makes the first robot as usual. But the machine now follows a new pattern. Instead of choosing the next part in each list:

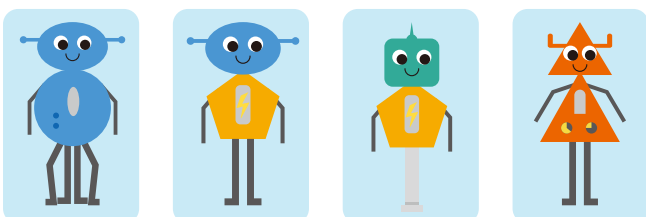
- For head and legs – the machine skips ahead one part in the original list.
- For body – the machine skips ahead two parts in the original list.

For example, the 2nd robot made looks like this:



Question

The workers at the factory realised the machine was broken and turned it off after the 10th robot. What did this 10th robot look like?



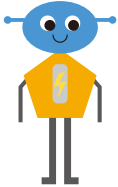


Robot Factory - continued

EXPLANATION

Answer

Correct answer is blue head, pentagon body, two legs.



Explanation

The first robot is still created the same – blue head, circle body and two legs.

Head: The machine was choosing every second head in the list. So today the new pattern was – blue → orange → green (repeats again). The 10th robot would have a blue head.

Body: The the machine was choosing every third body part in the list. So today the new pattern was – circle → pentagon → triangle → square (repeats again). The 10th robot would have a pentagon body.

Legs: The machine was choosing every second set of legs in the list. So today the pattern was – two legs → one leg → four legs (repeats again). The 10th robot would have two legs.

BACKGROUND INFORMATION

In computational thinking, pattern recognition and algorithms are important concepts. Pattern recognition is a method used in data analysis where similarities between objects or datasets are drawn out, allowing us to analyse trends or make predictions about future data. Algorithms are sets of rules or instructions that explain how a process should be done.

In this task, the machine in the factory follows a set pattern/algorithm when assembling robot body parts. A computer works in a similar manner – it follows set instructions to produce an outcome. If the rules are changed/pattern are changed, the end result will also change.

Like in the question, it is possible for algorithms or processes within a computer to break. While this can a problem that the user may only notice after a while, there are some steps that computers take to try and detect when these errors occur. One example is a *parity check*, where computers use part of a packet of data that verifies that the result has the right number of 1s and 0s in it, and nothing has changed during communication. Computers often use many error checking methods to make the chance of an error slipping through without notice very small.



Self-Driving Cars

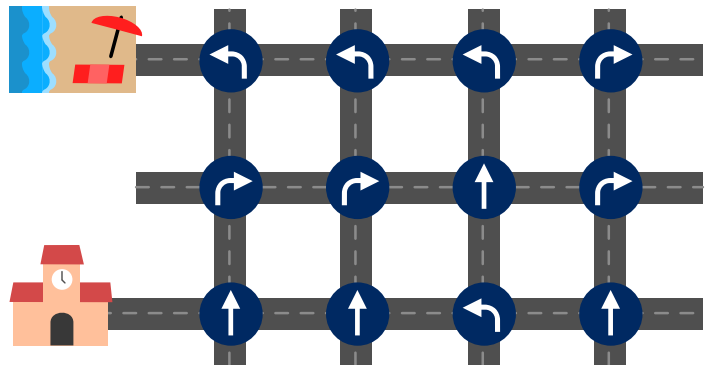
Some new models of self-driving cars are being tested. The cars can recognise traffic signs, but sometimes they fail to recognise other signs. When a car does not recognise a sign, it will ignore it and keep driving straight ahead. These are the signs used on the test track and their meanings:

- Drive straight ahead 
- Turn right 
- Turn left 

This is the map of the test track. Each sign always has the same meaning, whatever the direction the car is coming from.

Example:

For a car just leaving the school, the sign on the first crossroad means “drive straight ahead”, so that at the next crossroad it gets the same sign.



Question

Starting from the school, which of the following cars reaches the beach?

Car A fails to recognise the third sign it encounters.

Car B fails to recognise the fourth sign it encounters.

Car C fails to recognise the fifth sign it encounters.

Car D fails to recognise the sixth sign it encounters.

EXPLANATION

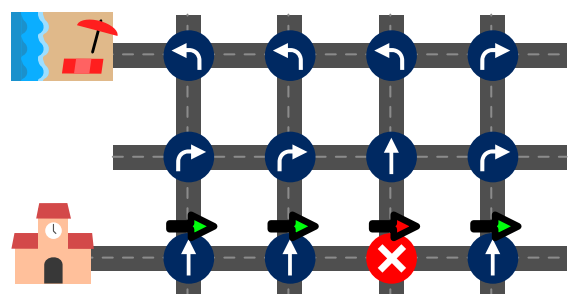
Answer

Car B.

Explanation

Only car B reaches the beach. This can be seen by tracing the movements of each car.

When leaving the school, car A first recognises the “drive straight ahead” sign and goes straight; it then recognises the “drive straight ahead” sign and goes straight; it fails to recognise the “turn left” sign and goes straight; finally, it recognises the “drive straight ahead” sign and drives off the map.



Continued on next page

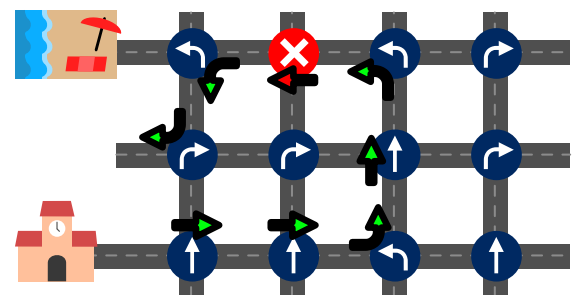
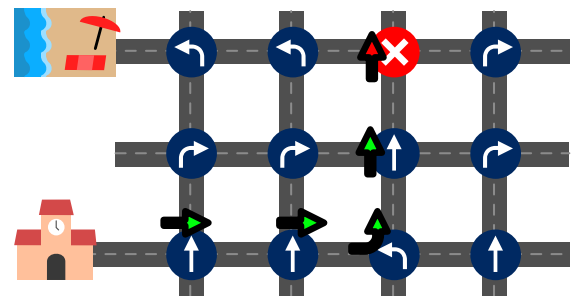
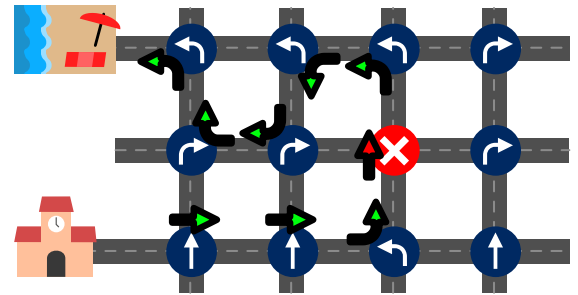


Self-Driving Cars - continued

When leaving the school, car B first recognises the “drive straight ahead” sign and goes straight; it then recognises the “drive straight ahead” sign and goes straight; it then recognises the “turn left” sign and turns left; it fails to recognise the “drive straight ahead” sign but goes straight ahead anyway; it then recognises the “turn left” sign and turns left; it then recognises the “turn left” sign and turns left; it then recognises the “turn right” sign and turns right; it then recognises the “turn right” sign and turns right; finally, it recognises the “turn left” sign and turns left to the beach.

When leaving the school, car C first recognises the “drive straight ahead” sign and goes straight; it then recognises the “drive straight ahead” sign and goes straight; it then recognises the “turn left” sign and turns left; it then recognises the “drive straight ahead” sign and goes straight; it then fails to recognise the “turn left” sign and drives off the map.

When leaving the school, car D first recognises the “drive straight ahead” sign and goes straight; it then recognises the “drive straight ahead” sign and goes straight; it then recognises the “turn left” sign and turns left; it then recognises the “drive straight ahead” sign and goes straight; it then recognises the “turn left” sign and turns left; it then fails to recognise the “turn left” sign and goes straight; it then recognises the “turn left” sign and turns left; it then recognises the “turn right” sign and turns right off the map.



BACKGROUND INFORMATION

Digital cameras “see” the world as a mosaic of tiny colored squares called pixels. Understanding what objects are on such an image is not an easy task for computers. The area of computer science developing solutions to this problem is called *computer vision*.

Deploying computer vision to real life is even more difficult than getting it to work on “clean” images. When a self-driving car is out on streets, its cameras may be blurred by rain or fog. Traffic signs may also be partially covered by snow or the car’s view of them may be blocked by tree branches or other cars, and so on.

After a self-driving car has recognised the surrounding objects, it needs to understand what the pedestrians and other cars around it are doing, which traffic rules apply in the situation, and only then can start planning its own actions.

Software engineers working on self-driving cars must test carefully the programs they have designed: they must imagine all the situations the car can meet and how the limitations of their design impact on the reliability of the operations. In the task the path a car is supposed to follow is fixed and marked by the sign, but the different cars have a different behavior on some of the signs: therefore their software can be different, or the environmental conditions (light, angle, etc.) in which they sense the signs are different, thus giving different results in the test.



Loop Kolam

Kolams are decorative patterns. You can draw Kolam on the floor with chalk.

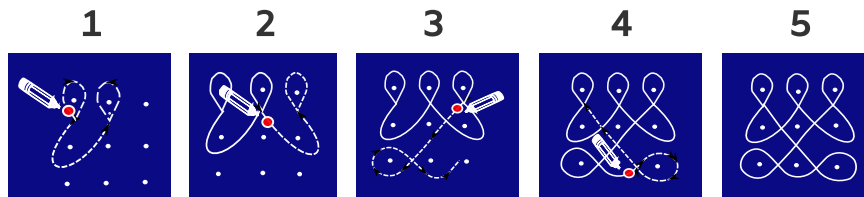
First you place dots and then you draw lines around them. You start and end at the same point by forming a continuous pattern, without lifting the chalk. Lines may cross each other, but you cannot draw sharp turns.



The turn of this line is allowed.

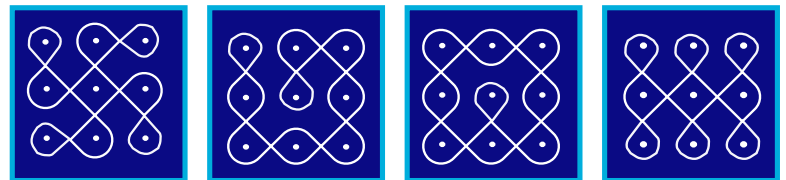


This turn would be a sharp turn, which is forbidden.



Question:

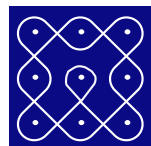
Which Kolam can you NOT draw this way?



EXPLANATION

Answer

The correct answer is:



Explanation

The design of the other options are continuous. Start from any point on the design and follow through the curves, you will reach back at starting point without lifting the chalk.

But in the correct answer you will notice that you have lifted your hand after completing the first pattern.

Observe the steps as shown in the images:



BACKGROUND INFORMATION

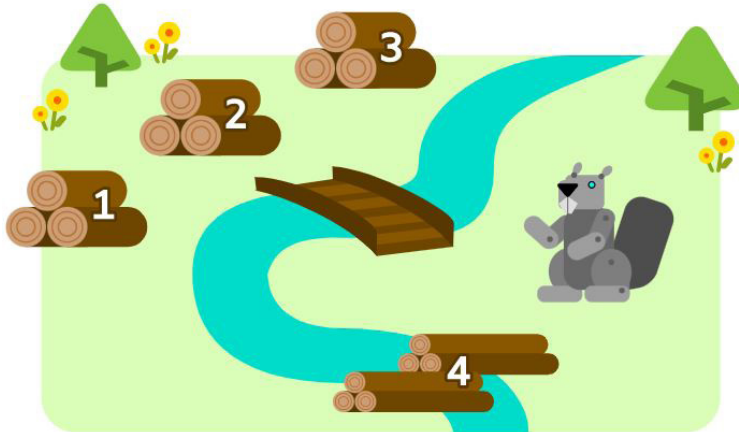
Kolams are intricate patterns found in towns and villages across India. How are these patterns remembered and reproduced? One theory is that they are generated by simple rules which are passed down from one generation to the next.

In computer science, one can write more complicated two-dimensional array grammars that generate pictures like kolams. Kolam can be called an Eulerian graph named after Leonhard Euler. Leonhard Euler, the genius Swiss Mathematician was a prolific contributor to several areas of Mathematics is considered as the inventor of the concept of a graph.



Robot Beaver

The beaver, Bob made a robot, Robo that would build dams instead of him. Robot Robo can carry only one log at a time.



Robo can perform these commands:

Tx – go to x, where x can be 1, 2, 3, or 4

G – grab,

D – drop.

For example, for Robo to take the log from place number 1, and put that log in the dam (place number 4), Bob should write this program:

T1 G T4 D

The beaver needs the program that would instruct Robo to take one log from each location (1, 2, and 3) and place it in the dam he is building (place number 4)

Question

Which program is correct?

T1 G T4 D T1 G T4 D T1 G T4 D

T3 G T4 D T2 G T4 D T1 G T4 D

T1 G T4 D T2 D T4 G T3 D T4 G

T1 G T4 D T2 G T4 G T3 G T4 D

EXPLANATION

Answer

The correct answer is T3 G T4 D T2 G T4 D T1 G T4 D.



Robot Beaver - continued

Explanation

In answer T1 G T4 D T1 G T4 D T1 G T4 D, robot Robo will take three logs from place 1, and none from places 2 and 3 there are no commands T2 and T3.

In answer T1 G T4 D T2 D T4 G T3 D T4 G, robot Robo tries to drop a log at place 2 instead of grabbing a log there, also it grab a log from the dam 4 and drop it on the place 3. Wrong commands are marked bold T1 G T4 D **T2 D T4 G T3 D T4 G**.

In answer T1 G T4 D T2 G T4 G T3 G T4 D, Robo does not drops a log at place 4, but tries to grab one. Wrong commands are marked bold: T1 G T4 D T2 G **T4 G T3 G T4 D**.

BACKGROUND INFORMATION

An **algorithm** is a finite sequence of well-defined instructions, typically used to perform a computation or solve a class of specific problems. Algorithms resemble recipes. Recipes tell you how to accomplish a task by performing a number of steps.

Algorithms can be expressed in many kinds of notation, including natural languages, pseudocode, flowcharts, drakon-charts, programming languages or control tables. Natural language expressions of algorithms tend to be verbose and ambiguous, and are rarely used for complex or technical algorithms. Try to explain in words, using natural language, to your friend how to do some daily work, for example, how to brush your teeth, or how to spread pancake with cream. During the game the friend should “forget” how patricular activities have worked so far and have the right to interpret each of your instructions in their own way. This game can be very interesting.

Pseudocode, flowcharts, drakon-charts and control tables are structured ways to express algorithms that avoid many of the ambiguities common in the statements based on natural language.

The word “algorithm” comes from the name of Arabic mathematician and writer Muḥammad ibn Mūsā al-Khwārizmī, but currently the word is strongly associated with computer science.

In computer programming, **debugging** is the process of finding and resolving bugs (defects, errors, or problems that prevent correct operation) within computer programs, software, or systems.

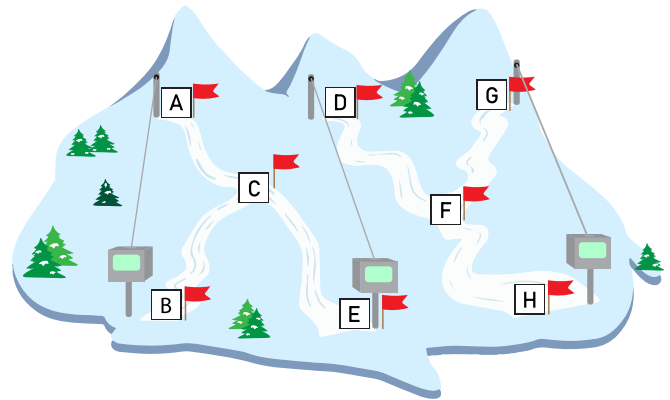


Ski Trail Map

The picture shows a map of ski trails and ski lifts in a ski resort.

Skiers use only ski lifts to go uphill and they ski downhill only on the official ski trails. There are six ski trails at the ski resort: A-C, C-B, C-E, D-F, F-H, and G-F.

A skier made a bet with his friend that he can find a way to use all the ski lifts and ski downhill on all the ski trails exactly once.



Question

From which location must the skier start to do so?

- ☐ A ☐ B ☐ C ☐ D ☐ E ☐ F ☐ G ☐ H

EXPLANATION

Answer

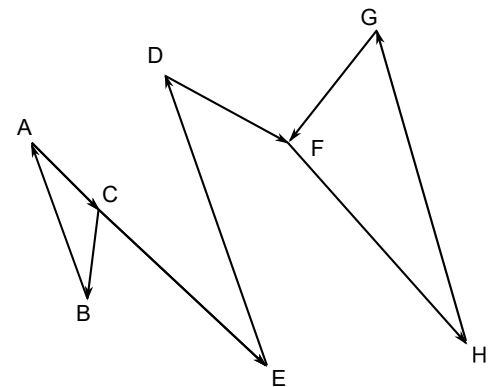
The correct answer is C.

Explanation

In order to go down all the ski trails exactly once and use each ski lift exactly once, the skier should pass the locations in this order: C, B, A, C, E, D, F, H, G, F.

The following picture represents the ski lifts and ski trails and the direction the skier can travel on them:

There is only one path to and from each location on the picture, except for C and F. At location C there are two paths out (to B and E), while at location F there are two paths in (from D and G).



As a skier cannot ski from location F to location D and from location F to location G (because he would have to go uphill by the ski trails), location F is not the right solution.

Location C remains the only possible solution. Following the arrows from C, you can trace the path through all the arrows in the picture, which means C is a correct answer.

BACKGROUND INFORMATION

A picture like in the answer Explanation is called a *graph*. A graph has *vertices* (the dots that represent the flags) and *edges* (the lines that represent the trails or the lifts). This particular graph is a *directed* graph, because it is important to know in which direction the skier can go from one vertex to another, indicated by the arrows on the edges.

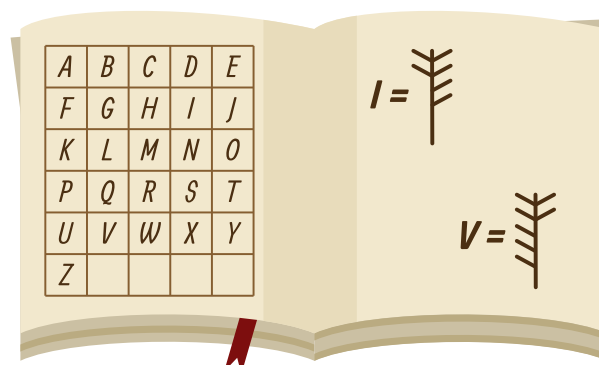
Graphs are used quite often in computer science to give an abstract representation of a problem. As you can see, such a graph often contains sufficient information to solve the problem.



Ancient Runes

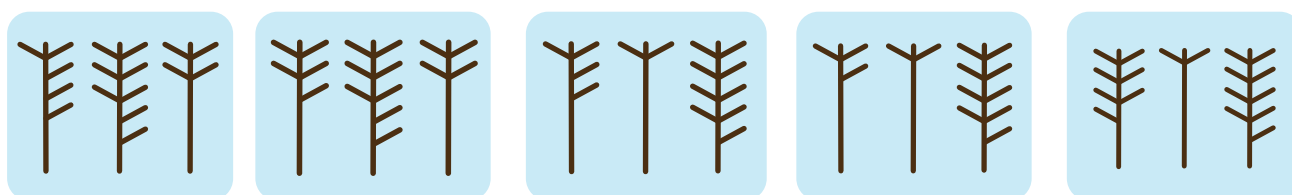
The ancient runes were partially decoded by Professor Challenger before his mysterious disappearance.

You have found the page below from his notebook.



Question

Which of the following can be decoded as “BAT”?



EXPLANATION

Answer



Explanation

The number of spines on the left represents the row number and the number of spines on the right represent the column number. The decoded letter lies in the intersection of the row and column (“I” has 2 on the left and 4 on the right, “V” has 5 on the left and 2 on the right).

Choice D spells “BAT”.

Choice A spells DOG, Choice B spells HOG, Choice C spells CAT, Choice E spells RAT.

BACKGROUND INFORMATION

The problem is an example of a coded (single worded) message. Cracking codes was arguably the birth of modern computing, as featured in movies about Alan Turing. He was at the forefront of cracking the Enigma code used by Germany in World War 2.

This code is simple but relies on you being given 2 clues (“I” and “V”) to work out the code.

The alphabet is given to you in what, in Computing terms, is called a two dimensional (“2D”) array. Accessing items in a 2D array is done in a program by using “indexes” which can take on different values for the row index and the column index.

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

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

Contact us

CSIRO Digital Careers

digitalcareers@csiro.au

csiro.au/education/Programs/Digital-Careers