UNIVERSITY OF WATERLOO

The CENTRE for EDUCATION in MATHEMATICS and COMPUTING

2018 Beaver Computing Challenge (Grade 7 & 8)

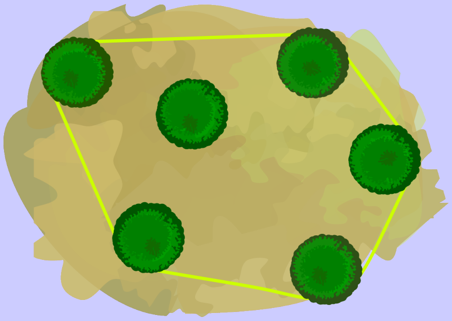
Questions, Answers, Explanations, and Connections

Part A

Roped Trees

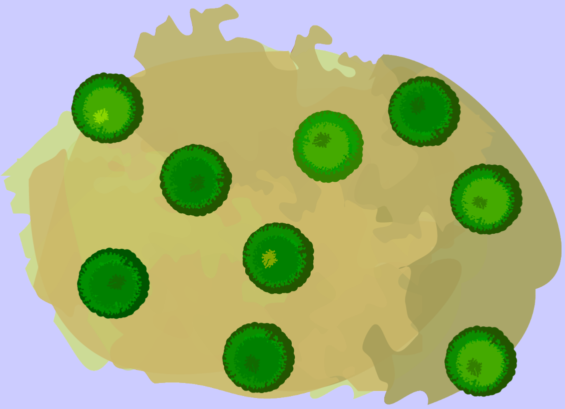
Story

Joni Beaver uses rope to mark groups of trees. The rope forms a very tight loop so that each tree either touches the rope or is entirely inside the loop. Below is an example where the rope touches exactly 5 trees when viewed from above.



Question

How many trees will the rope touch if the trees are arranged as follows (when viewed from above)?



(A) 4

(B) 5

(C) 6

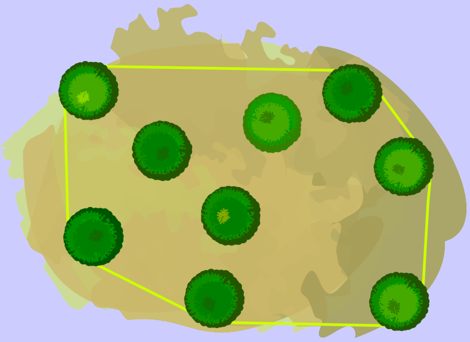
(D) 7

Answer

(C) 6

Explanation of Answer

We can count the trees touching the rope in the picture below.



Connections to Computer Science

This problem is known as finding the *convex hull* of a set of points. One way to define this is the polygon with the smallest area which contains all the points in the set. The word *convex* means extending outward or curving out, and any two points within a convex shape, when drawn on paper, can be connected with a straight line which is inside of the shape. The word *hull* here is used to describe the “shell” or “encasement”, as in the hull of a ship or the hull of a seed, such as corn.

Finding the convex hull of a set of points is very useful:

- pattern recognition: Is there a face in an image?
- processing written text: Is that handwritten character the letter B?
- geographic information systems: What is the size of a floodplain or river system?
- packaging: What is the least amount of material needed to wrap an object?

Country of Original Author

Canada

Rotation Game

Story

Beavers play a simple game. The game always begins with this starting position:

Red	Green
Blue	Yellow

From this starting position, rotation instructions are followed. All the rotations are clockwise and one quarter of a complete turn. The possible instructions are:

- 1R – meaning rotate the squares one time,
- 2R – meaning rotate the squares two times,
- 3R – meaning rotate the squares three times.

For example, if the first instruction is 2R, the top-left square will be Yellow as shown below.

Yellow	Blue
Green	Red

Question

From the starting position, what colours will the top-left square be after each of the instructions 1R, 2R, 2R, and 3R are followed in order?

(A) Red Green Blue Green Yellow

(B) Red Blue Green Blue Red

(C) Red Blue Yellow Red Green

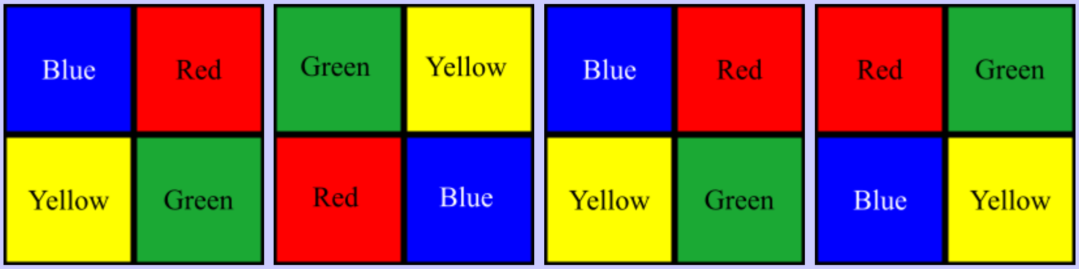
(D) Red Red Yellow Red Blue

Answer

(B) Red Blue Green Blue Red

Explanation of Answer

The top-left square is Red for the starting position and the results after each rotation are shown below.



Connections to Computer Science

This problem involves following a series of steps and keeping track of the current *state*. The state could be what the current orientation looks like, or only what is in the top-left corner (which can be used to determine the position of the other three colours). The ordered sequence of instructions is an *algorithm*. The ability to execute algorithms and remember state is the defining property of a computer.

This problem also involves simplifying and applying reasoning, since a rotation by two 2R rotations, or 1R plus 3R rotations (for a total of 4R rotations) will yield the game in the same orientation as it started.

Country of Original Author

Canada

Beaver Graffiti

Story

Beaver graffiti consists of three different symbols: 🐟, 🌸, and 🍃.
Sequences of symbols are built using two steps:

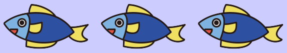
1. One of the symbols is drawn once or twice.
2. One of the symbols is is drawn once to the left of the current sequence and once to the right of the current sequence.

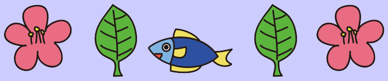
Step 1 happens first and exactly one time. Step 2 may happen any number of times. Here are five examples:

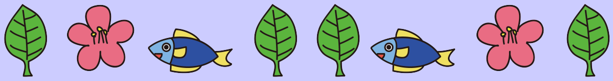
- 🐟
- 🌸
- 🍃 🍃
- 🌸 🐟 🌸
- 🍃 🐟 🐟 🍃

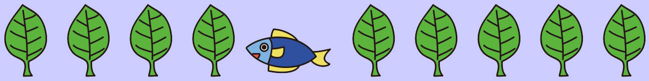
Question

Which of the following is **not** an example of beaver graffiti?

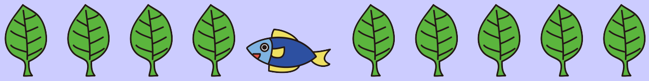
(A) 

(B) 

(C) 

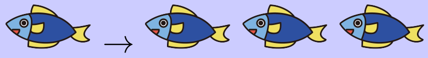
(D) 

Answer

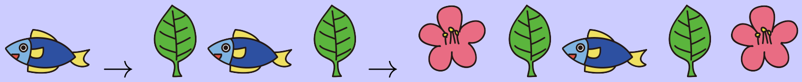
(D) 

Explanation of Answer

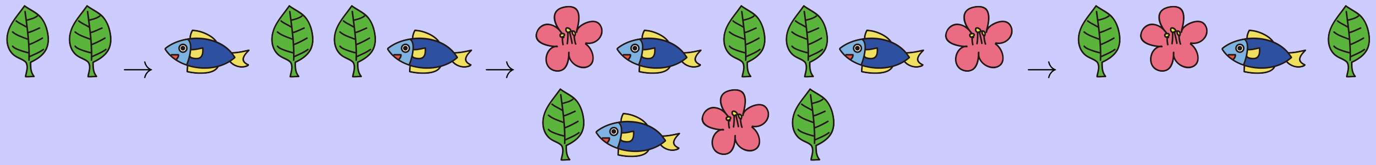
Option A can be built as follows:



Option B can be built as follows:



Option C can be built as follows:



If it was possible to build Option D, then Step 1 must involve the fish because there is only one fish. Then, each time Step 2 happens, the same number of leaves must be drawn but there are four leaves to the left of the fish and five to the right of the fish. Thus Option D cannot be built.

Connections to Computer Science

A *palindrome* is a word, phrase, number, or other sequence of characters which reads the same backwards as forwards, such as noon or racecar. Palindromes play a role in our understanding of DNA and human genetics studied by *computational biologists*.

More generally, in computer science, *formal languages* and *automata theory* provide ways to express the rules that generate sequences of characters (called *words*). They also provide *models of computation* to help us understand how complex a language of words is.

In this problem, graffiti is a word and the task is to recognize palindromes. This problem is possible to solve for relatively short words. However, even today's most powerful computers would require an infinite amount of memory to solve this problem for words that are arbitrarily long.

Country of Original Author

South Korea

Computer Science Museum

Story

A museum has received statues of five famous computer scientists. However, there is only room to display one statue at a time. They must decide the order in which the statues will be displayed. They come up with the following rules:

- Turing before Berners-Lee
- Turing before Hopper
- Hopper before Lovelace
- Hopper before Gates
- Lovelace before Gates

Question

What is one order in which the statues could be displayed?

(A) Turing, Hopper, Lovelace, Gates, Berners-Lee

(B) Turing, Berners-Lee, Hopper, Gates, Lovelace

(C) Turing, Gates, Hopper, Lovelace, Berners-Lee

(D) Turing, Berners-Lee, Lovelace, Hopper, Gates

Answer

(A) Turing, Hopper, Lovelace, Gates, Berners-Lee

Explanation of Answer

Answer A follows all of the rules.

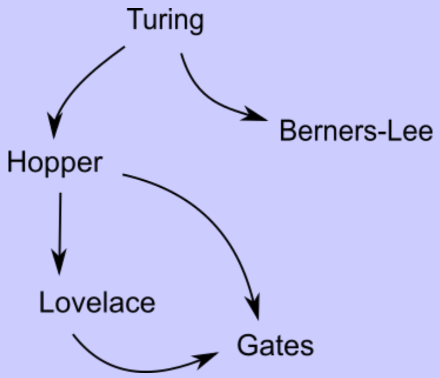
Answer B breaks the rule “Lovelace before Gates” so it is not correct.

Answer C breaks the rules “Hopper before Gates” and “Lovelace before Gates” so it is not correct.

Answer D breaks the rule “Hopper before Lovelace” so it is not correct.

Connections to Computer Science

The rules can be represented by arrows as illustrated below.



This diagram is called a *directed acyclic graph* (or DAG for short). A DAG is commonly used to describe relationships in computer science, such as for scheduling, parallel processing, and data compression.

A *conflict-free ordering* is an ordering of the statues (in a line) with all arrows pointing forward. This is called a *topological ordering* of the corresponding DAG. Returning to one of our applications, you might be able to imagine how such an ordering assists in scheduling events where one event must occur before another.

Country of Original Author

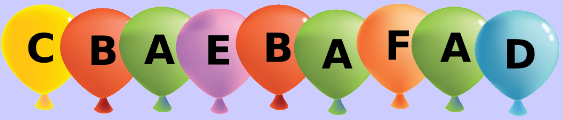
Belgium

Balloons

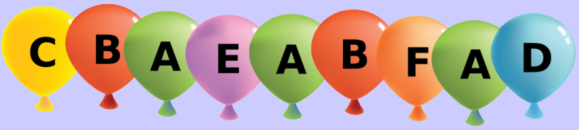
Story

Mark goes to a birthday party. A room at the party is decorated with balloons in rows:

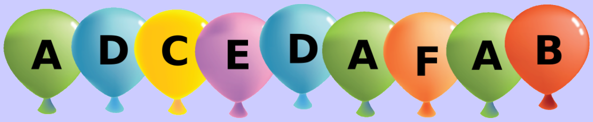
Row 1:



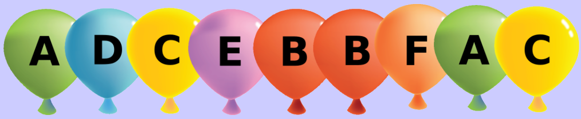
Row 2:



Row 3:



Row 4:



Mark can't see colours clearly. For him, yellow (C) looks the same as green (A), and blue (D) looks the same as red (B).

Question

Which two rows of balloons look the same to Mark?

(A) Row 1 and Row 4

(B) Row 2 and Row 4

(C) Row 1 and Row 2

(D) Row 1 and Row 3

Answer

(D) Row 1 and Row 3

Explanation of Answer

If we write out the symbols we get:

Row 1:	CBAEBAFAD
Row 2:	CBAEABFAD
Row 3:	ADCEDAFAB
Row 4:	ADCEBBFAC

Then we substitute each C with A because these look the same to Mark:

Row 1:	ABAEBAFAD
Row 2:	ABAEABFAD
Row 3:	ADAEDAFAB
Row 4:	ADAEBBFAA

We also substitute each D with B because these look the same to Mark:

Row 1:	ABAEBAFAB
Row 2:	ABAEABFAB
Row 3:	ABAEBAFAB
Row 4:	ABAEBBFAA

Now, we can easily see that Row 1 and Row 3 look the same to Mark.

Connections to Computer Science

Lists of objects are one of the simplest *data structures*. Computer scientists use data structures to organize data so that common tasks involving the data can be performed *efficiently*.

A very common task involving lists is to simply compare them. To compare lists of the same length, you have to compare the corresponding objects in each list. When computer scientists write programs to compare complicated objects, such as newspaper stories or photographs of people, some differences have to be ignored, and some different things have to be treated as “the same.” In this task, we are given lists of balloons and two simple rules for what “the same” means. Ignoring irrelevant differences and looking for common elements is the process of *abstraction*, which computer scientists use to write algorithms that work on a variety of inputs.

Country of Original Author

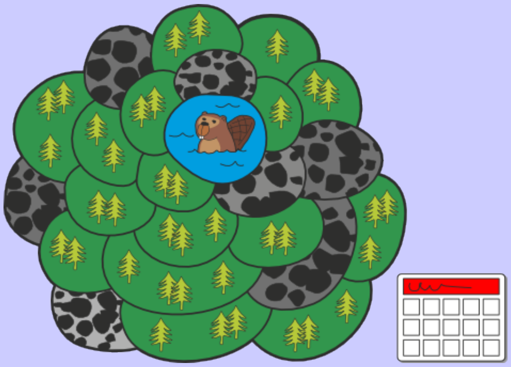
Ireland

Part B

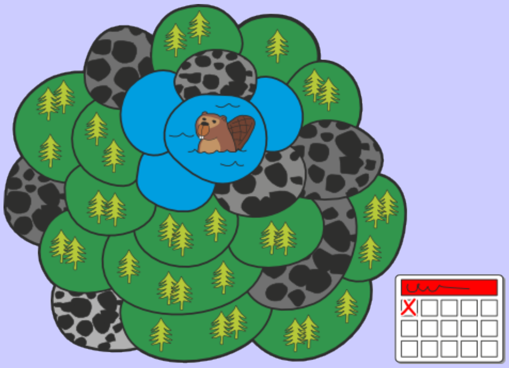
Beaver Lake

Story

Beavers live in a valley surrounded by mountains. In the valley, there is a lake. The lake is surrounded by fields with either trees or stones.



Every day, beavers flood all those fields with trees that are next to the lake or flooded fields. Fields with stones are not flooded.



For example, after one day, three fields will be flooded, as shown above.

Question

After how many days in total will all the fields with trees be flooded?

(A) 4 days

(B) 5 days

(C) 6 days

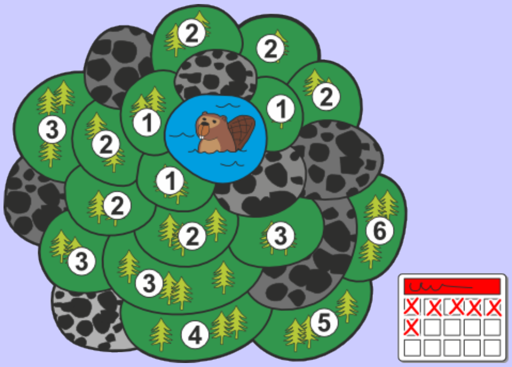
(D) 7 days

Answer

(C) 6 days

Explanation of Answer

Numbers in the picture below show flooded fields after the corresponding number of days.



Fields next to the lake are marked by 1. Then all unmarked fields with trees next to a field marked with 1 are marked with 2. This process is continued until all fields with trees are marked. Numbers in the picture below show on which day the field becomes flooded.

As we can see above, the last field with trees will be flooded after 6 days.

Connections to Computer Science

Computer scientists study different types of *algorithms*. An algorithm is a sequence of steps that solves a problem. This task demonstrates the *wavefront algorithm* which is used to scan an area divided into cells. You can think of this algorithm as taking a picture once per second when a stone is dropped in a still pool of water.

This scanning uses a *breadth-first search (BFS)* approach. A BFS algorithm begins at a cell called the *root*. Then, at each step, every cell next to previously visited cells (sometimes called the *fringe*) are visited. In this task, the cells correspond to fields and visiting a cell corresponds to flooding a field. When done carefully, BFS visits every cell in an area to be scanned exactly once. This is done in an organized way. In fact, as seen in the solution, cells can be numbered as they are visited. These numbers describe how far each cell is from the root. That is, BFS algorithms can be used to find the *shortest paths* from the root to all the other cells.

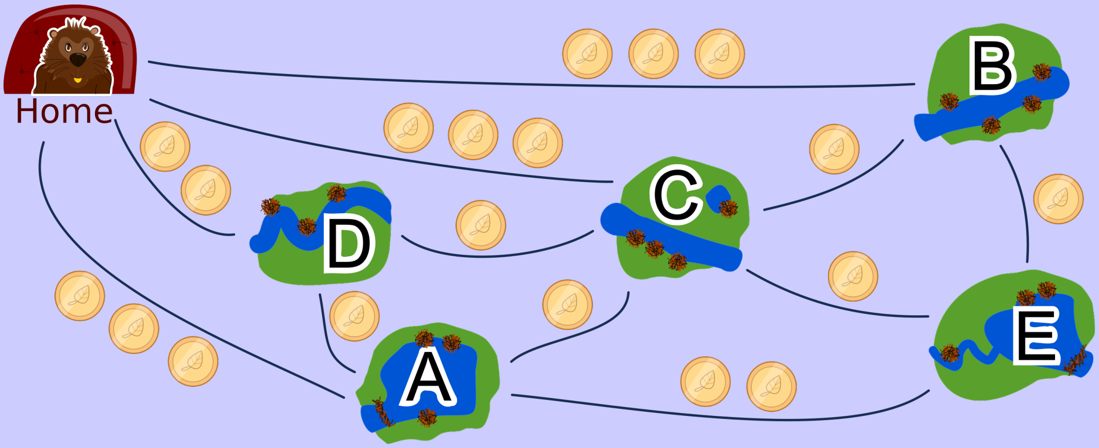
Country of Original Author

Lithuania

Visiting Friends

Story

Livia wants to visit all of her friends in five villages using public transportation. She visits them in one journey, without visiting a village more than once, and she returns home at the end of her journey. The number of coins it costs to travel on the direct route between each pair of villages is shown below.



For example, it would cost Livia a total of 11 coins to visit villages in the order:

Home → B → E → A → D → C → Home.

Question

What is the least possible total cost for Livia's journey?

(A) 7 coins

(B) 8 coins

(C) 9 coins

(D) 10 coins

Answer

(C) 9 coins

Explanation of Answer

There are two journeys with a total cost of 9 coins:

Home → B → E → C → A → D → Home
Home → D → A → C → E → B → Home

Notice that these are the “reverse” of each other.
No journey costs less. One way to determine this is to calculate the total cost of all possible journeys. A better way is to notice that Livia’s journey must begin and end with direct routes costing a total of $2 + 3 = 5$ or $3 + 3 = 6$ coins. Either way, exactly four other direct routes must be used in between, each of which costs at least one coin. So the total cost is at least $5 + 4 = 9$ coins.

Connections to Computer Science

A fundamental computer science problem is to search for the best solution to a problem. Sometimes, an *optimal* solution can be found. That is what you are asked to do in this problem which is a small instance of the famous *Traveling Salesman Problem* (TSP).

Although an immense amount of research has gone into solving TSP, nobody has found a general algorithm to *efficiently* find an optimal solution. Instead, the most efficient techniques we know use *approximation algorithms* which do their best to get close to an optimal answer. These algorithms rely on *heuristics* which use a relatively simple rule of thumb to make a “best guess” along the way.

Country of Original Author

Switzerland

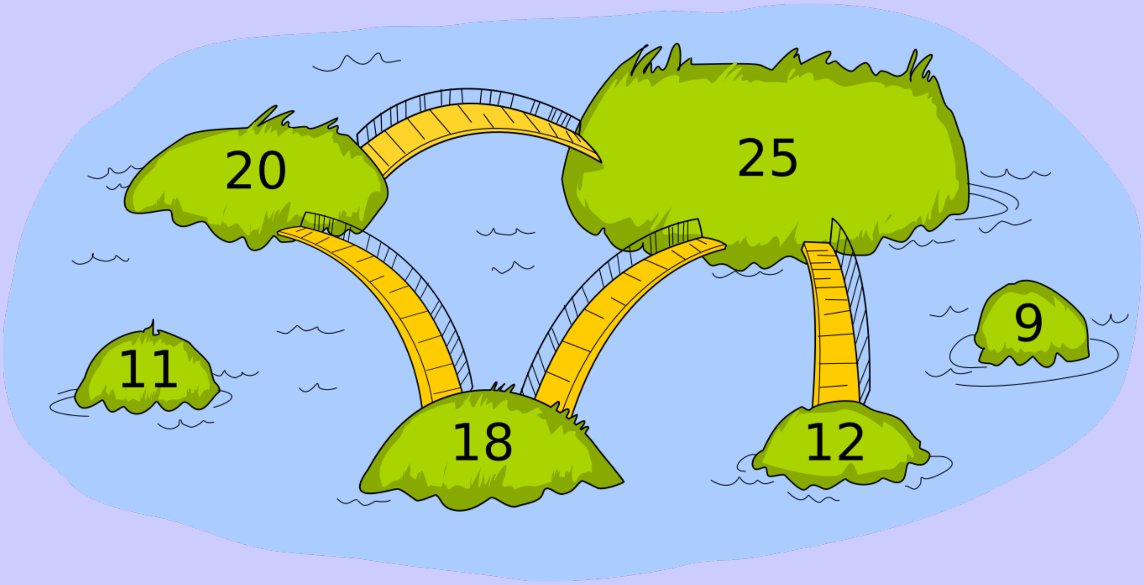
Connect the Islands

Story

People of Kastoria use only one rule to decide where bridges are to be built:

> They choose one number called the *bridge number*. If the sum of the populations of two islands is greater than the bridge number, a bridge is built between the islands. Otherwise, a bridge is not built between the two islands.

The six islands of Kastoria and their populations are shown below. The bridges built using the above rule are also shown.



Question

What bridge number was chosen?

(A) 34

(B) 35

(C) 36

(D) 37

Answer

(C) 36

Explanation of Answer

The chosen number cannot be 34 or 35 because $11 + 25 = 36$ and there is not a bridge between the islands with populations of 11 and 25.
The chosen number cannot be 37 because $12 + 25 = 37$ and there is a bridge between the islands with populations of 25 and 12.
For each pair of islands connected by a bridge, the sum of their populations is greater than 36. That is, $18 + 20 = 38$, $18 + 25 = 43$, $20 + 25 = 45$ and $12 + 25 = 37$.
Therefore, the chosen number is 36.

Connections to Computer Science

Islands and bridges are classic examples of *graphs* which represent connection between objects. In a graph, the objects/islands are called *vertices* and the connections/bridges are called *edges*.

A graph with n vertices might have much more than n to edges. In this case, it may be possible and beneficial to define the edges without listing all of them. These sorts of graphs are called *implicit graphs*, as opposed to *explicit graphs* where each edge is stored or described. One such implicit graph occurs when it is possible to specify a *threshold* above which an edge is present, as we have done for the islands. In that case, we can save a lot of computer memory. Graphs for which this is possible are called *threshold graphs*.

Country of Original Author

Italy 🇮🇹

Outdoor Soccer

Story

Mr. Castor is planning for his class to play soccer outside in the schoolyard. Several issues must be considered:

- The students can only play soccer on a sunny day.
- An outdoor activity is only allowed when the wind speed is lower than 20 km/h.
- The schoolyard cannot be used if another class has already reserved it on that day.

Mr. Castor has the following related information:

Weather Forecast

	Monday	Tuesday	Wednesday	Thursday	Friday
Weather Condition	☀	☂	☂	☀	☀
Wind Speed	5 km/h	24 km/h	13 km/h	7 km/h	40 km/h

Schoolyard Reservations

	Monday	Tuesday	Wednesday	Thursday	Friday
Class	Ms. Garcia	-	-	-	-

Question

What day should Mr. Castor plan to play soccer on?

(A) Tuesday

(B) Wednesday

(C) Thursday

(D) Friday

Answer

(C) Thursday

Explanation of Answer

The issues to be considered are summarized in the following table.

	Monday	Tuesday	Wednesday	Thursday	Friday
Weather Condition	OK	X	X	OK	OK
Wind Speed	OK	X	OK	OK	X
Availability of Schoolyard	X	OK	OK	OK	OK

Since all three conditions must be met, Mr. Castor should plan to play soccer on Thursday.

Connections to Computer Science

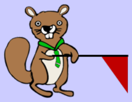
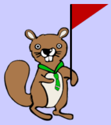
A *database* is a large amount of information stored in an organized manner for a long time. Databases store information gathered through most websites or apps, whether that is social media information or purchases made through websites. The data stored in these databases are often considered very valuable (e.g., customer data that indicates purchasing preferences). In order to get value out of data, one needs to be able to identify and extract data from a database according to provided criteria. This is called *data retrieval*. This problem is a small example of data retrieval: we wish to find data that matches certain criteria, such as weather conditions and availability.

Country of Original Author

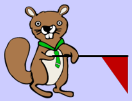
South Korea

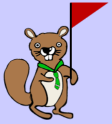
Flag Codes

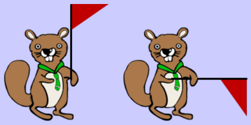
Story

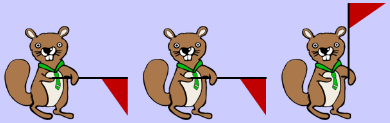
Beavers communicate by holding flags held horizontally  or vertically  .

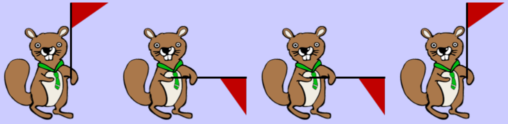
Five different letters (P, Q, R, S, and T) can be sent by using the following codes:

P: 

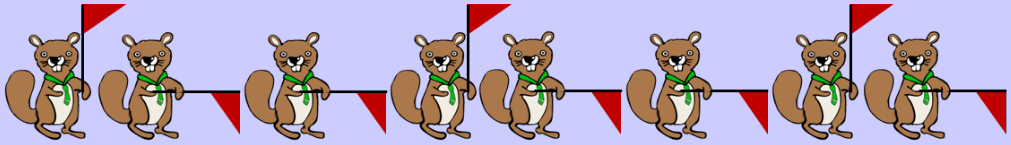
Q: 

R: 

S: 

T: 

Sequences of letters are sent using the codes for each letter in the sequence in order. Beaver Adanma sends this:



Question

What sequence of letters could Adanma be sending?

(A) QPPTP

(B) TSQ

(C) RPQSR

(D) RPSP

Answer

(A) QPPTP

Explanation of Answer

We use H to represent the horizontal position of a flag and V to represent the vertical position. For example, the sequence of flag positions: vertical, horizontal, horizontal, vertical, horizontal, horizontal, vertical, horizontal, is represented by VHHVHHVH.

In the given code, P is encoded as H, the letter Q is encoded as V, the letter R is encoded as VH, the letter S is encoded as HHV, and the letter T is encoded as VHHV.

So Adanma sent VHHVHHVH and we can use the code to see that this is how Option D would be sent. The other three options given something different:

(B) TSQ → VHHVHHVV
(C) RPQR → VHHVHHVVH
(D) RPSP → VHHHHVH

Connections to Computer Science

Sending messages using flag positions is an old practice in naval communication called the *semaphore* system of communication. To send messages in this way, the code used must be *unambiguous*, which means that a code can have exactly one interpretation. One way to make a code unambiguous is to ensure it is *prefix-free*. This means that no code of a letter is a prefix of the code of some other letter.

In this problem, the codes are not prefix-free. Can you see why?

Today, ships send messages using modern technology. We probably all send some form of digital message to another person or website many times a day. We write in a *natural language* such as English but some code is needed to store the message on a computer. Two common encoding of information are *ASCII*, which deals with simple text characters, and *Unicode*, which deals with text, accented letters and emojis. How to encode and decode information falls under a field of research called *information theory* which includes the study of *data compression* (how to store data using less memory) and *cryptography* (how to store or send information securely).

Country of Original Author

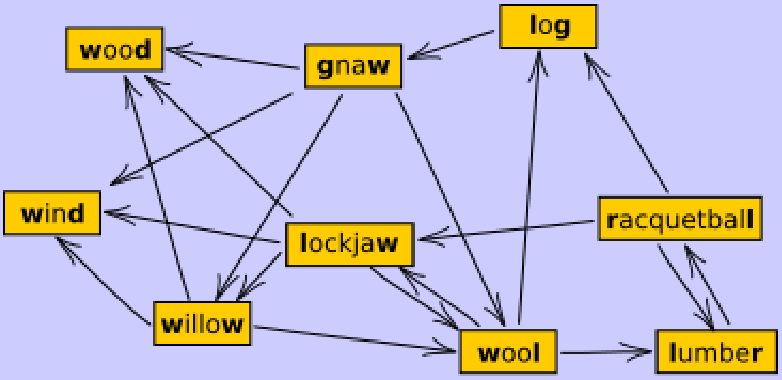
Taiwan

Part C

Longest Word Chain

Story

Beavers play a word chain game. One beaver starts by saying a word. The other beaver must say a different word which begins with the last letter of the previous word. Then the first beaver says another word (which was not said yet) using this same rule, and so on. If a beaver is unable to say a new word, that beaver loses the game. These beavers do not know many words. In fact, they can draw their entire vocabulary like this:



Notice that an arrow out of a word points at the next possible word(s) that can be said.

Question

What is the largest possible number of words that can be said in one game?

(A) 6

(B) 7

(C) 8

(D) 9

Answer

(C) 8

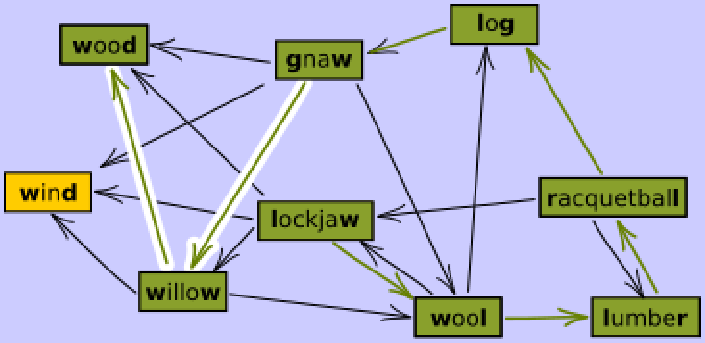
Explanation of Answer

The beavers can use at most 8 words in one game. One example is:

lockjaw-wool-lumber-racquetball-log-gnaw-willow-wood

(Can you find another game of the same length?)

To be sure that 8 is the largest possible number of words, we have to convince ourselves that it is not possible to use all 9 words. Consider the words *wood* and *wind*. There is no word beginning with *d*, so if either of these words is said, it must be the last word of the game. Since there cannot be two words that are said last, it is not possible to use the entire vocabulary of 9 words.



Connections to Computer Science

In this problem, the drawing is a *graph*. We call the words *vertices* and we call the arrows *edges*. You are asked to find the *length of the longest path* in the graph.

Graphs appear almost anytime there are objects connected together in some way. Sometimes the vertices and edges actually look like the drawing in a graph. For example, we might think of a map where the vertices as cities and the edges as roads between the cities. Other times, the graph is more hidden as in this problem. Did you notice that all you need to solve this problem is the list of words themselves? You are probably glad that a picture was provided. This illustrates one way in which graphs are a powerful tool.

Finding the longest path in a graph is very well-studied. It is related to the famous *Traveling Salesman Problem* and other *optimization problems* problems involving paths which often appear in industrial problems.

Country of Original Author

Czech Republic

Timetabling

Story

Bebras Tech offers the following evening classes:

- Computing (C),
- Geography (G),
- Language (L),
- Math (M), and
- Science (S).

Three beavers would like to sign up for these courses:

- Xavier wants to take C, L, and M;
- Yvette wants to take C, G, and S;
- Zoey wants to take L, M, and S.

Bebras Tech wants to squeeze these courses into as few evenings as possible such that:

- each course is offered on exactly one evening, and
- beavers can take at most one course per evening.

Question

What is the least number of evenings needed for Bebras Tech to schedule these courses?

(A) 2

(B) 3

(C) 4

(D) 5

Answer

(C) 4

Explanation of Answer

The following schedule shows that four evenings can be used.

Evening 1: Computing (taken by Xavier and Yvette)
Evening 2: Language (taken by Xavier and Zoey) and Geography (taken by Yvette)
Evening 3: Science (taken by Yvette and Zoey)
Evening 4: Math (taken by Xavier and Zoey)

It is not possible to use only three evenings. To see this, consider Computing, Language, Science, and Math. There are six (unordered) pairs of these subjects and for each pair, at least one beaver wants to take both subjects in the pair:

- Xavier wants to take Computing and Language.
- Yvette wants to take Computing and Science.
- Xavier wants to take Computing and Math.
- Zoey wants to take Language and Science.
- Xavier and Zoey want to take Language and Math.
- Zoey wants to take Science and Math.

This means that these four subjects have to be offered on different evenings, so at least four evenings are required. This is just one possible solution: can you find the other(s)?

Connections to Computer Science

Solving small scenarios like this can be achieved by careful observation and logical thinking. When this problem is scaled up to hundreds of students and classes and more time slots it becomes much more difficult to solve without an algorithm. One such algorithm is that of *graph colouring* which is a technique used to solve a number of difficult problems in *graph theory*.

Graph colouring has many practical applications and can even be used to create and solve Sudoku puzzles. Unfortunately, there is no known *efficient* algorithm to do graph colouring: if there are hundreds of nodes, even the fastest computers would take thousands of years to solve it.

Country of Original Author

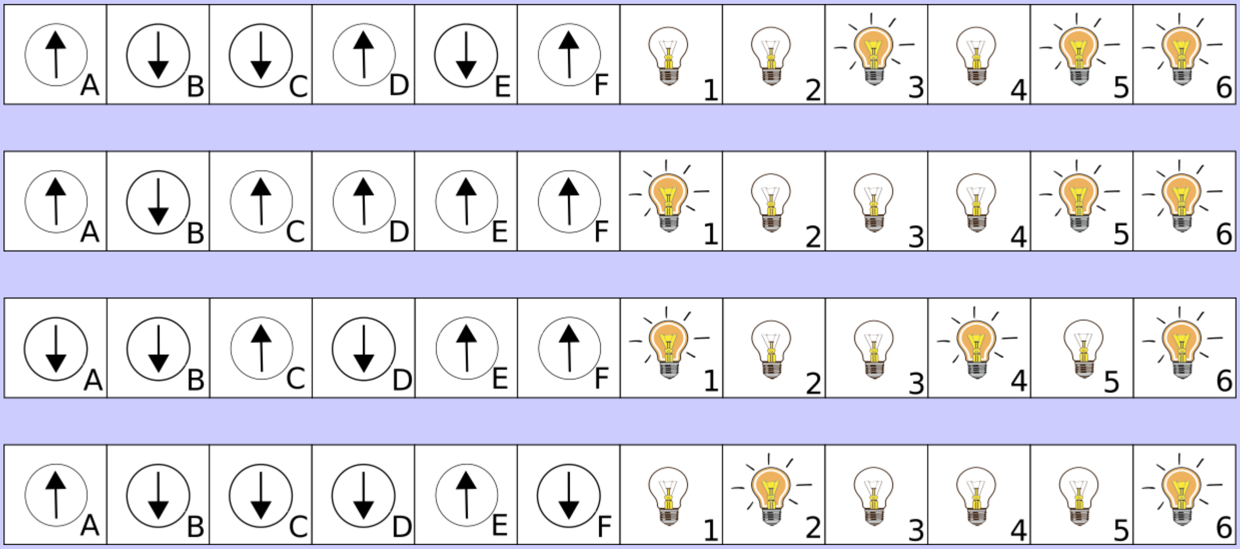
United Kingdom

Bulbs

Story

An amateur electrician connected 6 bulbs (numbered 1, 2, 3, 4, 5, and 6) to 6 switches (labelled A, B, C, D, E, and F). Each switch operates exactly one bulb but nobody knows which one. Each switch can be either up or down, but we don't know which position corresponds to the bulb being on and which position corresponds to the bulb being off. To make matters worse, this could be different for different switches.

Four experiments were conducted to determine which switch is connected to which bulb. The results of these experiments including the position of the switches and on/off status of the bulbs are shown below.



Question

Which switch is connected to which bulb?

(A) C → 1, E → 2, D → 3, A → 4, F → 5, B → 6

(B) C → 1, F → 2, E → 3, A → 4, D → 5, B → 6

(C) C → 1, F → 2, D → 3, E → 4, A → 5, B → 6

(D) C → 1, F → 2, B → 3, A → 4, D → 5, B → 6

Answer

(B) C → 1, F → 2, E → 3, A → 4, D → 5, B → 6

Explanation of Answer

Between the first and second picture, the only switches in different positions are C and E and the only bulbs with a different status are 1 and 3. This means that the correct correspondence is either C → 1 and E → 3, or E → 1 and C → 3. This is our first observation. Now consider the third and fourth picture. In these pictures, the status of bulb 1 changes but the status of bulb 3 is unchanged. Moreover, C is in a different position in these two pictures but E is not. Together with our first observation, this means we know that the correct correspondence is C → 1 and E → 3.

Using the same idea and looking at the second and third picture, we know that the correct correspondence must be A → 4 and D → 5, or A → 5 and D → 4. Looking at the third and fourth picture, of these two possibilities, we know the correct one must be A → 4 and D → 5 .

The remaining unknown connections involve switches B and F and bulbs 2 and 6. We look at the third and fourth picture again. The status of bulb 2 changes between these pictures but not the status of bulb 6. The position of switch B is the same in these two pictures but the position of switch F is not. Thus the correct correspondence must be B → 6 and F → 2.

In summary, we get A → 4, B → 6, C → 1, D → 5, E → 3, F → 2.

Connections to Computer Science

When working with computer code or larger computer systems, it is important to understand how it works so that you can find and fix errors or possibly modify the code/system to add additional features. As programs become more complicated, and involve more and more decisions or cases to consider, modifying or finding errors becomes increasingly difficult.

Debugging, which is the process of finding the cause of *bugs* or *defects* in code which create undesired behavior, is an important skill needed for computer scientists. One technique in debugging is to run small controlled experiments on various *inputs* to determine if the *output* matches what is expected. In this problem, the input is the setting of the switches and the output is the bulbs which are on or off. In this problem, we could determine connection between switches and bulbs using only four tests, rather than having to run as many as 64 ($= 2^6$) tests.

Country of Original Author

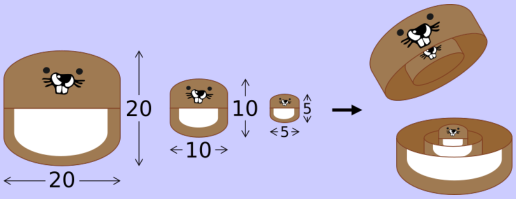
Hungary

Nesting Dolls

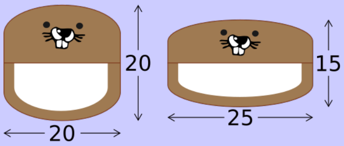
Story

Wooden toy dolls have different widths and heights. They are hollow and can be separated into two parts. This means that a doll can be nested inside any other doll that is both wider and higher.

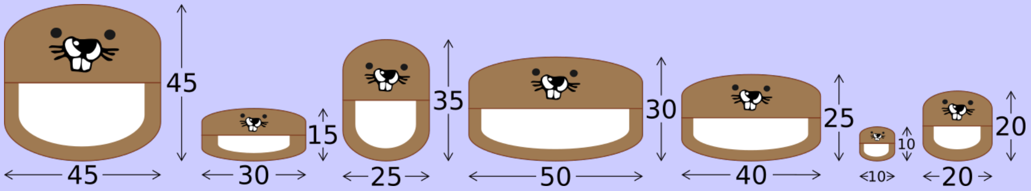
For example, a doll with width 5 and height 5 fits inside a doll with width 10 and height 10, which in turn fits inside a doll with width 20 and height 20. After this, only one doll is visible.



On the other hand, a doll with width 20 and height 20 cannot fit inside a doll with width 25 and height 15. Also, a doll with width 25 and height 15 cannot fit inside a doll with width 20 and height 20. So, if these are the only two dolls, they will both always be visible.



Ian has the following collection of dolls and starts fitting them inside each other.



Question

What is the fewest possible number of dolls that are visible after Ian is done?

(A) 1

(B) 2

(C) 3

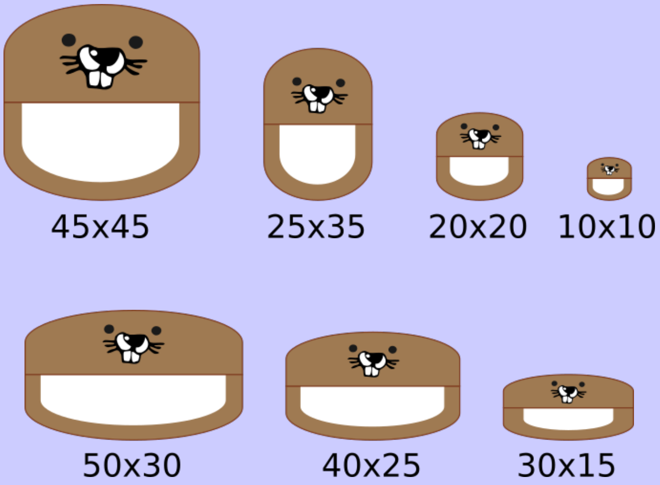
(D) 4

Answer

(B) 2

Explanation of Answer

The dolls in the collection can be nested in two groups leaving the largest doll in each group unhidden:



It is impossible to be left with only one unhidden doll because neither of the two largest dolls in these groups can fit in the other.

Connections to Computer Science

This problem can be solved using a *greedy strategy*. For this problem, we can process the dolls in sorted order according to width or height and each time we try to nest a doll within a doll, we find the closest match possible in either dimension.

More generally, a greedy strategy involves making the best choice at each stage in the hopes of finding the best solution. In this way, it makes one greedy choice after another, reducing the given problem into a smaller one.

However, not every problem can be solved optimally using greedy strategy: for example, *knapsack problems* usually do not have optimal solutions that can be found using a greedy strategy. Nevertheless, the greedy strategy is still useful because it is easy to describe and implement and often gives a good approximation to the optimal solution.

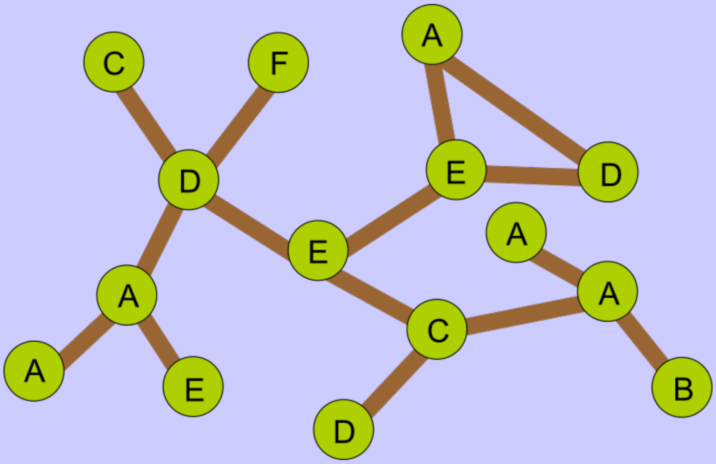
Country of Original Author

Taiwan

Park Walk

Story

A map of a park is shown below. Green circles represent trees and brown lines represent paths. Trees are labelled with letters but note that some letters are used to label more than one tree.



Two families walk in the park along paths from tree to tree. Each time they visit a different tree, they write down the letter of that tree, even if they have visited the tree before.

- The Bea family writes down B A A A C E D E E D A.
- The Ver family writes down F D C D A E A D E D A.

Both families started walking at the same time and the amount of time it took them to walk along any path from one tree to another was constant.

Question

How many times did the two families meet at a tree?

(A) one time

(B) two times

(C) three times

(D) never

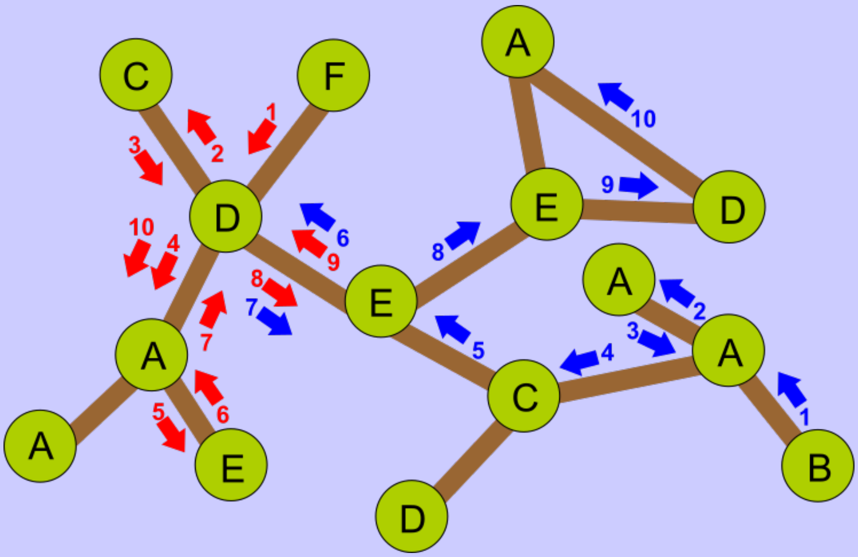
Answer

(D) never

Explanation of Answer

It is not enough to find the same letter at the same position (i.e. same time) during the families' walks, because the same letter can represent a different tree. For example, both families ended their walk at a tree labeled A, but if we follow both walks step by step then we can determine that the families actually ended at different trees.

Following each step of each family is possible because the neighbours of each tree (the trees that are connected to it by a path) are always labeled by different letters. We also know where each family started because only one tree is labelled B and only one tree is labelled F. Knowing this, below we illustrate the paths of the Bea family (in blue) and Ver family (in red).



Observe that only two trees were visited by both families (with labels D and E), but the numbers on the paths show that they were not visited at the same time.

Connections to Computer Science

This problem involves *graphs*: in this problem, the trees correspond to *vertices* and the paths connecting trees correspond to *edges*. Graphs are used to represent a variety of scenarios involving connections between objects: social networks, road systems, electrical grids are three such scenarios.

Another interesting point about this task is the *representation* of the walks in the park. Despite the fact that some of the trees (vertices) are marked by the same letter, the walks that start either from B or F can be *unambiguously* described by the sequence of letters along the walk. It means that one sequence of letters describes only one walk. It is because the *neighbours* of each tree (neighbours of tree X are the trees that are directly connected to X by a path) are always labeled by different letters. So if we know where we are at some point of the walk and we see the next letter in the walk's representation then there in no doubt (or ambiguity) about which tree should we visit next.

This problem also involves keeping track of the *state* in a *representation*. The task involves remembering the state of what tree each families is at, and then changing the state to be the corresponding next tree for each family, and determining whether there is a state where they are at the same tree.

Country of Original Author

Slovakia