



UNIVERSITY OF
WATERLOO



The CENTRE for EDUCATION in
MATHEMATICS and COMPUTING



2019
*Beaver
Computing
Challenge
(Grade 7 & 8)*





















*Questions,
Answers,
Explanations,
and
Connections*

Part A

Cloud Communication

Story

Smoke signals were used by different groups of ancient peoples to send messages. A very simple code using small and large smoke clouds is given below.

			
			
			
			
			
north	east	south	west

Messages are read from top to bottom. The following message contains an error. Either one small cloud should be a large cloud or one large cloud should be a small cloud.



Question

What is the correct message?

- (A) north
- (B) east
- (C) south
- (D) west

Answer

(A) north

Explanation of Answer

If you change the third cloud from large to small you will get the code for north. Therefore, Option A is correct. Option B is not correct because the first and the last clouds would have to be changed. Option C is not correct because the first, second, and fourth clouds would have to be changed. Option D is not correct because all except the first cloud would have to be changed.

Connection to Computer Science

If you design sequences of symbols to be used for communication, it is better to choose the sequences so that messages can be reconstructed even if some parts of the message are lost or damaged. This can be done by sending more symbols than necessary. The extra data can essentially be used in place of any missing or damaged data. In computer science, systems like this are called *error-correcting codes* and they are used widely. For example, redundant information is used when sending music in a digital form. This means that digital music can be played correctly even if some data is corrupt.

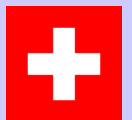
In this problem, only two clouds are needed to represent four different messages:



Using five clouds of smoke added extra information which allowed us to determine the correct meaning even when the message contained an error.

Country of Original Author

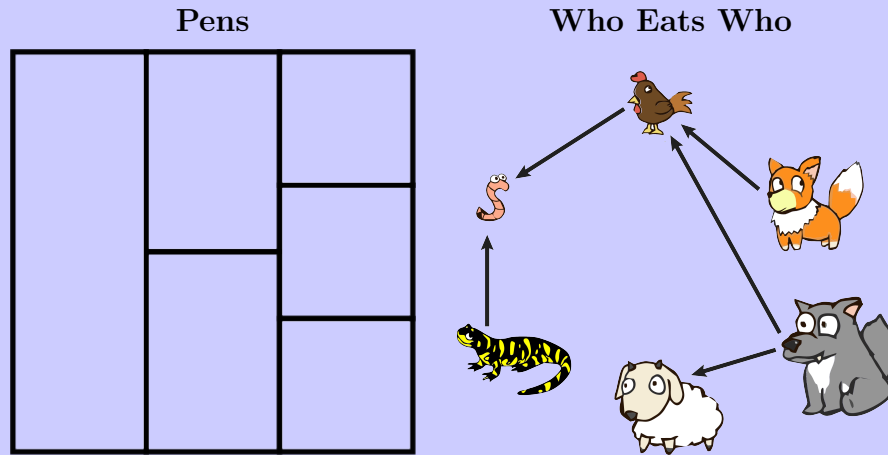
Switzerland



Koko's Animals

Story

Koko has six animals and needs to place each one in its own pen. Two animals cannot be placed in touching pens if one animal will eat the other. In the diagram shown, arrows point from an animal to all the other animals that it will eat.



For example, the wolf will eat the chicken, but the wolf will not eat the worm.

Question

Which of the following choices is **not** a good placement?

(A)

🐔	🐑	🐛
	🐕	🐺
	🐅	

(B)

	🐑	🐔
🐛		🐅
	🐕	🐺

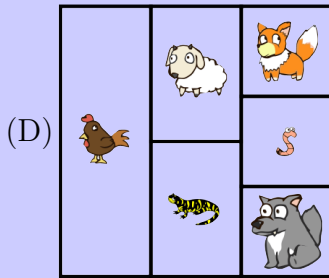
(C)

🐺	🐕	🐛
	🐑	🐔
🐅		

(D)

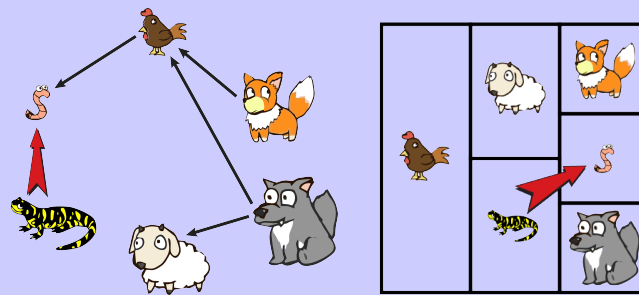
🐔	🐑	🐕
	🐛	🐺
🐅		

Answer



Explanation of Answer

Option D is not a good placement because the worm's pen is touching the salamander's pen and the salamander eats the worm.



For Options A, B, and C, consider the worm, chicken, and sheep. These are the three animals that can be eaten. If the worm's pen is not touching the salamander's or the chicken's, then the worm is safe. If the chicken's pen is not touching the wolf's or the fox's, then the chicken is safe. If the sheep's pen is not touching the wolf's, then the sheep is safe. Since the wolf, the fox, and the salamander cannot be eaten, they can be placed in any pen.

Connections to Computer Science

In this problem, you are asked to solve a *constraint satisfaction problem*. Constraints come from the fact that two animals cannot be placed in touching pens if one animal will eat the other. In general, a constraint satisfaction problem involves finding a solution that obeys a given set of rules. Other well-known examples are Sudoku puzzles and trying to colour a map such that two neighbouring regions are not the same colour. These problems are often very difficult to solve. Trying all the possibilities often takes too long so computer scientists look for more advanced *search algorithms* that will produce a solution.

Country of Original Author

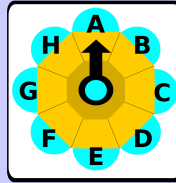
Indonesia



Safe

Story

A chef keeps secret recipes in a safe. It is unlocked using a circular knob with a pointer.



With the pointer at A to start, the chef unlocks the safe by turning the knob clockwise and counter-clockwise alternately as the password is spelled. For example, to enter the password BH, the chef:

turns one position clockwise	then turns two positions counter-clockwise

We represent passwords using numbers to indicate how far to turn and arrows to show the direction. For example, BH is represented by $1\curvearrowright 2\curvearrowleft$ which means turn one position clockwise and then two positions counter-clockwise.

To retrieve the secret recipes, the chef must enter the password CHEFDG.

Question

With the pointer starting at A, which of the following will unlock the safe?

- (A) $2\curvearrowright 3\curvearrowleft 4\curvearrowright 3\curvearrowleft 3\curvearrowright 3\curvearrowleft$
- (B) $2\curvearrowright 5\curvearrowleft 5\curvearrowright 1\curvearrowright 3\curvearrowright 3\curvearrowleft$
- (C) $2\curvearrowright 3\curvearrowleft 5\curvearrowright 7\curvearrowleft 6\curvearrowright 5\curvearrowleft$
- (D) $2\curvearrowright 1\curvearrowleft 4\curvearrowright 3\curvearrowleft 3\curvearrowright 2\curvearrowleft$

Answer

- (C) $2\curvearrowright 3\curvearrowleft 5\curvearrowright 7\curvearrowleft 6\curvearrowright 5\curvearrowleft$

Explanation of the Answer

Option C correctly tells the chef to move:

- two positions clockwise from A to C, then
- three positions counter-clockwise from C to H, then
- five positions clockwise from H to E, then
- seven positions counter-clockwise from E to F, then
- six positions clockwise from F to D, and finally
- five positions counter-clockwise from D to G.

Option A begins by incorrectly telling the chef to spell CHD. Option B begins by incorrectly telling the chef to spell CF. Option D begins by incorrectly telling the chef to spell CB.

Connections to Computer Science

When solving a problem it is often helpful to keep track of the current position or *state* of an object. In this problem, where the arrow is pointing is part of the state of the safe. The state of an object can also include a history of actions done on it. Here, as the knob is turned, we must remember which letters of the password have already been spelled.

Performing the same action on an object may not always have the same effect. Actions done on an object can have different effects based on its current state. For example, in this problem, turning five positions counter-clockwise after spelling “CHEFD” will open the lock. But turning it five positions counter-clockwise after spelling only “CHE” will not.

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

There are many other examples where the state is important. When following directions on a phone while walking, the important state will include your current location. When writing a computer program, a programmer has to decide what the state of the system they are working with is and write the program to correctly keep and update that state.

Country of Original Author

Taiwan



Mystery Beaver

Story

Beavers named NICOLE, KEVIN, KRISTINA, EDGAR, and KATEY play a game with their teacher. The teacher must discover the mystery beaver among them whose name satisfies these rules:

1. The name of the beaver has the letter G or I in it.
2. The name ends with a letter that no other name starts with.
3. The name starts with a letter that at least one other name starts with.

Question

Who is the mystery beaver?

- (A) KEVIN
- (B) EDGAR
- (C) KATEY
- (D) KRISTINA

Answer

(D) KRISTINA

Explanation of the Answer

KRISTINA has an I in it so Option D satisfies Rule 1. KRISTINA ends with an A and no other name starts with an A so Option D satisfies Rule 2. KEVIN and KATEY both begin with a K as does KRISTINA so Option D satisfies Rule 3. Therefore, Option D satisfies all three rules.

Option A is incorrect because KEVIN ends with an N and NICOLE starts with an N which violates Rule 2. Option B is incorrect because only EDGAR begins with an E which violates Rule 3. Option C is incorrect because KATEY does not contain a G or an I which violates Rule 1.

Connections to Computer Science

In this problem, you are asked to solve a *constraint satisfaction problem*. The constraints are given by the three rules. In general, a constraint satisfaction problem involves finding a solution that obeys a given set of rules. Other well-known examples are Sudoku puzzles and trying to colour a map such that two neighbouring regions are not the same colour. These problems are often very difficult to solve. Unlike in this problem with four multiple choice answers, trying all the possibilities often takes too long so computer scientists look for more advanced *search algorithms* that will produce a solution.

Country of Original Author

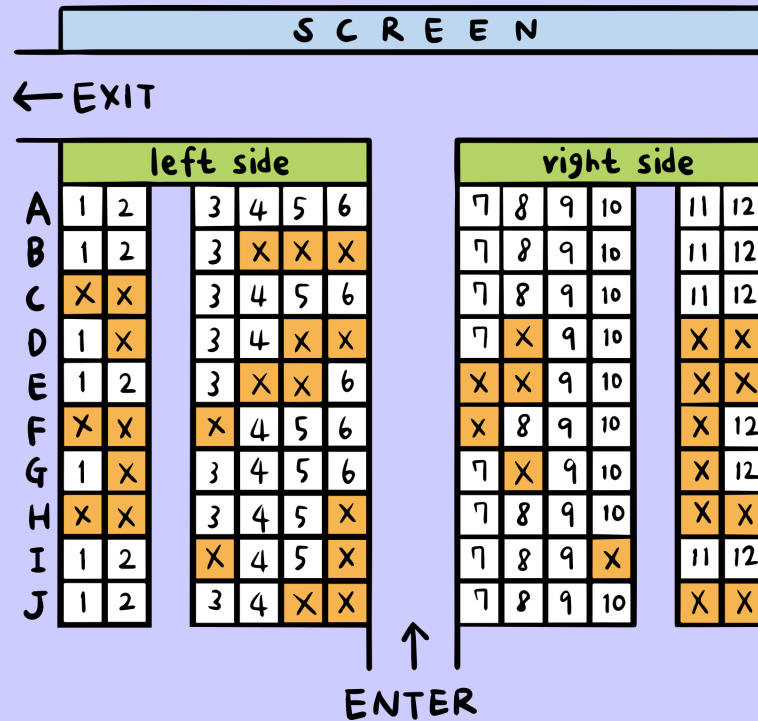
Cyprus



Movie Theatre Seats

Story

Three friends Alex, Bao, and Chiti are choosing seats in a movie theatre. The seats marked X can't be selected because someone else has already taken them.



Alex, Bao, and Chiti each say what will make them happy:

- Alex: "I want to sit on the right side."
- Bao: "The three of us must sit right beside each other without any seats or aisles between us."
- Chiti: "I don't like when the screen is too close! Let's not sit in the first three rows."

For example, If they choose seats G3, G4, and G5, then Alex will be unhappy. If they choose D7, D9, and D10, then Bao will be unhappy. If they choose A7, A8, and A9, then Chiti will be unhappy.

Question

In how many ways can the three friends choose seats so that they are all happy?

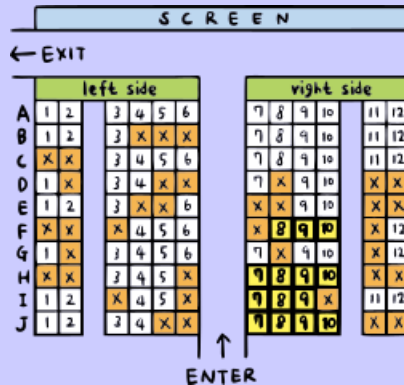
- (A) 3
- (B) 6
- (C) 7
- (D) 9

Answer

- (B) 6

Explanation of Answer

To make Alex happy, they must choose seats from column 7 to column 12. To also make Chiti happy, they can't choose seats in rows A, B, or C. Finally, to make Bao happy, they can only choose seats that are one of at least three consecutive untaken seats (not separated by an aisle). These possibilities are highlighted below.



Among these seats, they have the following six choices:

- F8, F9, and F10
- H7, H8, and H9
- H8, H9, and H10
- I7, I8, and I9
- J7, J8, and J9
- J8, J9, and J10

Connections to Computer Science

This problem is a *constraint satisfaction problem* much like the Mystery Beaver problem. It is also a nice example of how important *specification* is. The specification of a problem is how it is written and conveyed to the intended audience.

Care must be taken (and was taken in the design of this problem) to carefully describe what will make the three friends happy. When we think of problem solving, we often focus on finding the answer or verifying that the answer is correct. However, the important first step is to understand the problem. Computer scientists and mathematicians have developed specific language that allows problems to be described clearly and precisely. History has shown that this is difficult to do well. Many computer bugs and software failures have been caused by an unclear specification or a specification that did not meet the original *requirements* of the problem. Many computer scientists and software engineers specialize in requirements and/or specification and have developed tools to help with both.

Country of Original Author

South Korea



Part B

Cleaning the Lawn

Story

After an outdoor concert, a recycling robot is assigned to pick up the eight items left on the lawn. The robot starts on the left side of the lawn as shown.



The robot starts by identifying the closest item. The robot then moves to this item and picks it up. Then the robot identifies the remaining item that was closest to the item it just picked up, and moves to this new item. The robot continues this process until all of the items have been picked up.

Question

Which type of item will the robot pick up last?

- (A)  (B)  (C)  (D) 

Connections to Computer Science

The *algorithm* that the robot follows in this problem is based on distance. This makes it a *geometric problem*. Geometric problems rely heavily on geometry you learn in math class such as shapes and angles, as well as important computer science concepts.

Of course, this problem also involves a robot. The behaviour of a robot is controlled by a *computer program*, running on the computer that is built into the robot. A computer program consists of instructions written in a language that a computer can understand. Robots have cameras and sensors that enable them to identify objects in their environment and calculate distances to these objects. *Robot control programs* process the inputs of these sensors and enable robots to behave *autonomously* (i.e. without human interference).

Autonomous robots can be programmed to do jobs that are dangerous for humans. They can be helpful in scenarios such as exploring sites of catastrophes caused by weather or human conflict.

Country of Original Author

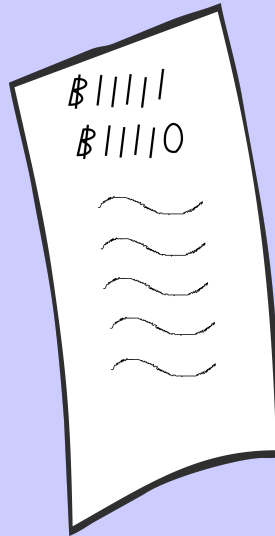
Germany



B-taro's Bills

Story

B-taro has exactly five bills worth the following amounts of money: B1, B10, B100, B1000, and B10000. He lists all the possible total amounts of money he can make from greatest to least.



For example, B11110 ranks 2nd on B-taro's list.

Question

Where does B11010 rank on B-taro's list?

- (A) 4th
- (B) 6th
- (C) 8th
- (D) 9th

Answer

(B) 6th

Explanation of Answer

To find the rank of B11010, we consider all possible amounts greater than B11000. This must include the B10000 bill, the B1000 bill, and at least one other bill because B-taro has exactly one of each type of bill. In order from greatest to least we get:

- 1st: B11111
- 2nd: B11110
- 3rd: B11101
- 4th: B11100
- 5th: B11011
- 6th: B11010
- 7th: B11001

Therefore, B11010 is the 6th greatest possible total amount.

Connections to Computer Science

The values of the bills in this problem are decimal numbers that we are used to working with in our math classes. That is, the *base* we are used to working with is ten. However, notice that each digit in the values of the bills in this problem happens to be either a 0 or a 1. This means that they can be viewed as *binary numbers*. Instead of using the digits 0 through 9, only the digits 0 and 1 are used. The base of this *binary number system* is two. For example, $6 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$ so we write or represent 6 as 110 in binary. The ranking of amounts in the solution is a list of decimal numbers but if we think of these as binary numbers, the ranking is the same.

Computers are based on the binary system. Why? The answer is that computer memory and instructions are based on *bits* – zero or one representing “off” or “on”. This actually corresponds to whether or not there is enough electrical current or positive charge at some point in hardware or memory.

Country of Original Author

Japan



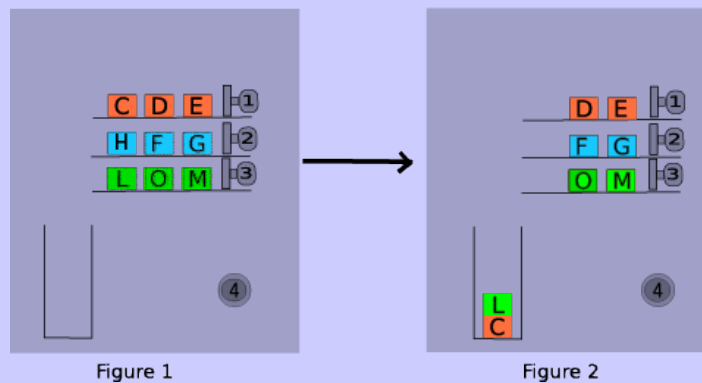
Packing Machine

Story

A machine, controlled by buttons 1, 2, 3, and 4, can drop a block from a shelf into a box or remove a block from the box.

Press	Result
1	leftmost block on the top shelf drops into box
2	leftmost block on the middle shelf drops into box
3	leftmost block on the bottom shelf drops into box
4	top block from the box is removed

Blocks that drop into the box always land on top of other blocks already in the box. For example, suppose we start as shown in Figure 1. After buttons 1, 2, 4, and 3 are pressed (in that order), we will end as in Figure 2. Notice that Block H has been removed from the box and is no longer in the picture.



Question

Suppose we start as shown in Figure 1 and then the buttons are pressed in the order: 3, 1, 1, 4, 4, 2, 3, 4. After this happens, which of the following blocks is the second from the top of the box?

- (A) Block H
- (B) Block O
- (C) Block C
- (D) Block L

Answer

(D) Block L

Explanation of Answer

Button 3 is pressed first which moves Block L into the box. Next, Button 1 is pressed twice which moves Blocks C and D into the box in that order. Next, Button 4 is pressed twice causing the machine to remove Blocks D and C. At this point the box only contains Block L.

Button 2 is pressed next which moves Block H into the box. Button 3 is pressed next which moves Block O into the box because Block L was removed from the bottom shelf earlier.

Finally, pressing Button 4 will cause the machine to remove Block O. This leaves blocks L and H in the box, so the second from the top is Block L.

Alternatively, notice that when Button 4 is pressed, the first block placed in the box (Block L) will never be removed. By counting the number of blocks that go into and out of the box, you will find that there are two blocks in the box after all buttons are pressed. Therefore the block on the bottom (Block L) is also the second block from the top after all of the buttons have been pressed.

Connections to Computer Science

In this problem, you are asked to keep track of which blocks are in the box as you consider what happens as buttons are pressed. The key property is that the last block put in the box, is the first one that must be taken out of the box the next time an attempt is made to start removing blocks from the box. We say that this follows the *last-in first-out* or *LIFO* principle. In computer science, a collection of items that follows this principle is called a *stack*. It is one fundamental way that a collection can be stored. In particular, a *computer program* will often be broken into smaller pieces called *subroutines*. The order in which these subroutines are executed is often determined using a stack.

Country of Original Author

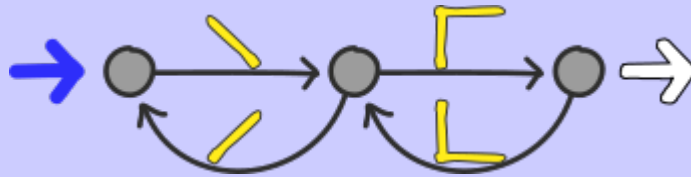
Japan



Making Stitches

Story

A sewing machine can make four different types of stitches. The rules that the machine follows are shown.



The machine starts a new line of stitches by following the thick blue arrow on the left.

Then the machine moves from circle to circle following in the direction of the arrows. Every time an arrow is followed, the machine makes the stitch shown on that arrow. If a circle has more than one arrow leading out of it, the user of the machine can choose to follow either one of the arrows.

The machine finishes by following the outlined arrow on the right.

Question

Which line of stitches **cannot** be made using the above rules?

- (A)
- (B)
- (C)
- (D)





Answer



Explanation of Answer

The stitch in Option D **cannot** be made.



The problem (shown above) is that a  stitch is followed by a  stitch. This is not possible because after a  stitch, the machine can only create a  stitch or finish the line of stitches.

The stitches in Options A, B, and C can all be made. We can see this by following arrows in the rules stitch by stitch. For example, Option A can be made by following arrows in the directions right, right, left, right, left, right, left, right.

Connections to Computer Science

The diagram presented in this problem is a *state diagram* of a *finite-state machine*. Finite state machines are used in computer science to describe systems which can be in one of a finite set of possible states at any given time and move from one state to another by means of a finite set of actions. The next state of the system depends only on the current state and the executed action. Usually, the *initial state* and the *final states* of the system are also specified.

More specifically, this problem is also a nice example of the *syntax validation process*. Before searching for logical errors in computer programs (this search is called *debugging*), the *syntax* of the program is checked. That is, a *compiler* or *interpreter* first has to check that the text of the program is a valid sequence of characters much like in this problem where we are checking whether or not a particular line of stitches could have been made using the given rules.

Country of Original Author

Lithuania



Classifier

Story

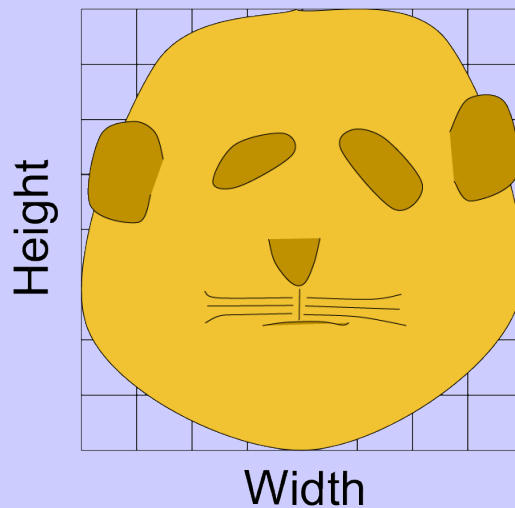
Given an image of an animal, a machine measures various parts of the animal: head, ears, and whiskers. The height of a part is the distance from its lowest point to its highest point. The width of a part is the distance from its leftmost point to its rightmost point.

These measurements are used to identify the animal based on the chart shown.

	Rabbit	Beaver	Bear	Cat
ear height	$\frac{1}{2}$ of head height	$\frac{1}{4}$ of head height	$\frac{1}{4}$ of head height	$\frac{1}{2}$ of head height
whiskers width	head width	$\frac{1}{2}$ of head width	$\frac{1}{2}$ of head width	head width
head width	$\frac{1}{2}$ of head height	$\frac{1}{2}$ of head height	head height	head height

Question

What type of animal does the machine identify the following image as?



- (A) Rabbit
- (B) Beaver
- (C) Bear
- (D) Cat

Answer

(C) Bear

Explanation of Answer

In this image, the head width is equal to that of the head height, the whiskers width is half of the head width, and the ear height is a quarter of the head height. These measurements match all the chart entries for a bear.

The image is not identified as a rabbit because it does not match any entries in the chart. The image is not identified as a beaver because the head width is equal to that of the head height. The image is not identified as a cat because the ear height and whiskers width do not match the chart.

Connection to Computer Science

We can imagine that the chart in this problem is produced by inspecting a large set of images of rabbits, beavers, bears, and cats. This would be an example of using *machine learning* to recognize images. The immense power in today's modern computers has made machine learning an increasingly important subdiscipline of *artificial intelligence*. This is because, to be effective, a machine learning *algorithm* typically needs to be trained on a large data set.

Another example of an application of machine learning is *autonomous driving*, where a machine learning algorithm is trained to recognize the lines and signs along a road as well as all possible objects encountered by a car.

Country of Original Author

Pakistan




















Part C

Wood Allergies

Story

For some beavers, eating some types of wood will make them sick. Ang is making appetizers out of wood for a party. Each appetizer is made from one type of wood and each appetizer is large enough to feed all the beavers. Ang has a list of the beavers attending the party, and the types of wood that they can eat without getting sick.

Name	Wood They Can Eat
Ang	 willow  oak  ash  maple
Baadi	 willow  oak  poplar
Chakor	 oak
Dalma	 ash  birch
Edgard	 willow  maple  birch
Filipa	 oak  ash
Georgios	 poplar  maple

Ang wants to make as few appetizers as possible. However, she wants to make sure that every beaver can eat at least one appetizer without getting sick.

Question

How many appetizers should Ang make?

- (A) 3
- (B) 4
- (C) 5
- (D) 6

Answer

(A) 3

Explanation of Answer

Ang must make an appetizer from oak for Chakor. This appetizer will also feed Ang, Baadi, and Filipa. There is no common wood that the three remaining beavers can safely eat, but there are appetizers that two of the three remaining beavers can eat. This means that three is the minimum number of appetizers that Ang should make.

One possible way Ang can make this minimum number of appetizers is to make the following three appetizers:



Connections to Computer Science

In this problem, you are asked to connect beavers with types of wood. A *graph* is set of *vertices* and *edges* that model connections like this. The beavers and types of wood are the vertices of the graph. Each combination of a beaver and type of wood that it can eat, is an edge of the graph. Since edges only connect beavers with wood and there are no connections between two beavers or between two types of wood, we call the graph *bipartite*. One *part* of the graph is the set of beavers and the other part consists of the types of wood.

Graphs have an incredible number of applications and are widely studied by computer scientists. The particular problem here is a special case of the *dominating set* problem. The goal is to find a set of vertices, W , in one part (specifically the types of wood) such that each vertex in the other part (specifically the beavers) is connected by some edge to at least one vertex in W . In fact, we want the smallest such set. Determining where to place cameras for security purposes and figuring out how many fire stations to place in a city to service everyone are two other examples of this type of problem.

This problem is known to be what we call *NP-complete*. Most computer scientists believe that efficient solutions to NP-complete problems do not exist. Even though this means it may not be worth looking for an efficient full solution to the minimal dominating set problem, finding a set that is small enough might be okay in practice.

Country of Original Author

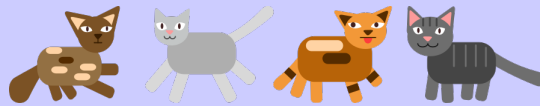
Slovenia



Swapping Cats

Story

Four cats stand in a line as shown.



The line-up can be changed by performing a *swap*. A swap is performed by exchanging the positions of any two cats in the line. Two cats do not need to be side-by-side for a swap to occur.

Question

If exactly two swaps occur, one after the other, then which of the following line-ups **cannot** be the result?

- (A) An orange tabby cat, a black cat with white stripes, a brown cat with dark spots, and a light grey cat.
- (B) A brown cat with dark spots, an orange tabby cat, a black cat with white stripes, and a light grey cat.
- (C) An orange tabby cat, a black cat with white stripes, a light grey cat, and a brown cat with dark spots.
- (D) A black cat with white stripes, a light grey cat, a brown cat with dark spots, and an orange tabby cat.

Answer



Explanation of Answer

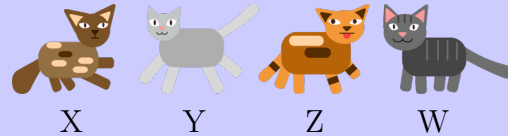
If the cats in the second and third positions swap, we get:



Next, swapping the cats in the third and fourth positions, results in Option B.

Option A can be achieved by swapping the cats in the first and third positions, and then the cats in the second and fourth positions. Option D can be achieved by swapping the cats in the first and third positions, and then the cats in the first and fourth positions. Therefore, the correct answer must be Option C. Let us verify that the line-up in Option C cannot be achieved with only two swaps.

Explanation 1: First, label the four cats as follows:



The initial line-up can be written as $XYZW$ and the final line-up in Option C can be written as $ZWYX$. There are six possibilities for the first swap and the resulting line-up is given for each below:

Positions swapped	Resulting line-up
First and Second	YXZW
First and Third	ZYXW
First and Fourth	WYZX
Second and Third	XZYW
Second and Fourth	XWZY
Third and Fourth	XYWZ

Notice that in each of these six possible resulting line-ups, there are at least three cats that are not in the correct position for the desired final line-up of $ZWYX$. Since only two cats can change positions during a second swap, it is impossible to produce the final result in Option C with only one more swap.

Explanation 2: First, notice that if we compare the initial line-up to the line-up in Option C, we see that all four cats have changed positions. This means that each cat must have participated in at least one swap. Assuming that only two swaps were performed, it must be the case that each cat participated in *exactly one* swap. Since each cat moved only once, we do not have to worry about the order of the swaps, just which cats were paired for the swaps. Since the cat in the first position must end up in the fourth position to produce Option C, the first and fourth cats must have been paired. However, since the cat in the fourth position must end up in the second position, the second and fourth cats must have been paired. Since the fourth cat cannot have been paired with two different cats, we cannot possibly achieve Option C with only two swaps.

Connections to Computer Science

Data is stored in a computer's memory. This includes both *internal memory* often referred to as *RAM* and *external memory* which might be a hard drive or flash drive, for example. Computers can access data in internal memory very quickly compared to external memory but external memory is much less expensive than internal memory. This means that modern computers usually have a lot more external memory but computer scientists try to find ways to use the internal memory as much as possible.

For this problem, the only operation we are concerned about is a *swap*. If we think of the line up of cats as data stored in the memory of a computer, then a swap involves changing the location of two pieces of data. If this data is in external memory, then aiming for as few swaps as possible is exactly what we need to do in order to perform the overall task as quickly as possible.

Swap operations are generally very common, but may not be as easy to accomplish as you think. Most computers can't simultaneously copy the value from A to B and the value from B to A . Instead, they must copy the value from A to a temporary location, then copy the value from B to A and finally copy the value from the temporary location to B .

When dealing with numbers, you can also swap two values without using a temporary location. It is tricky but uses simple arithmetic. Look at the following example and see if you can work out how it swaps the numerical value in A with the numerical value in B .

- Add the value in B to A , and store this in A . (No change is made to the value in B .)
- Subtract the value in B from A , and store this in B .
- Subtract the value in B from A , and store this in A .

Country of Original Author

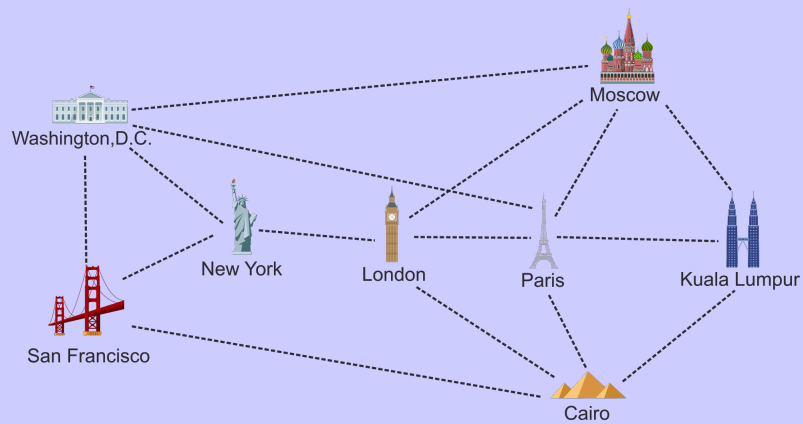
Canada



Cancelled Flights

Story

The dashed lines in the diagram represent all Bebras Air flights. Each flight operates in both directions. The airline is popular because its customers are able to fly between any two cities (possibly stopping in one or more cities in between).



The airline wants to cancel some flights but it still wants its customers to be able to fly between any two cities.

Question

What is the maximum number of flights that Bebras Air can cancel?

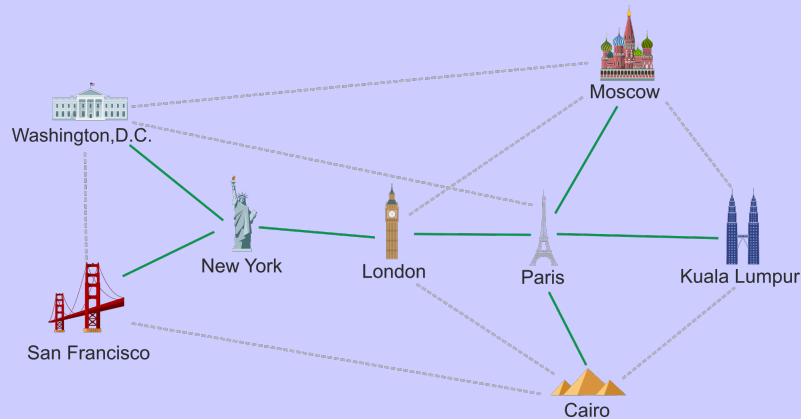
- (A) 6
- (B) 7
- (C) 8
- (D) 9

Answer

(C) 8

Explanation of Answer

The example below shows that it is possible to cancel 8 flights and still allow customers to fly between any two cities. Note that 7 flights are left.



We need to show that 8 cities cannot be connected in this manner using less than 7 flights. We will do so by showing that it cannot be done with 6 flights (and so cannot be done with fewer either).

Suppose that we have 6 flights arranged in such a way that customers are able to fly between any two of the 8 cities. Each of the 6 flights involves 2 cities and so if we list all of the cities involved in the flights we get $6 \times 2 = 12$ cities in total (with some repetitions). We know that each city must be a part of at least one flight and so each city appears in this list at least once. Since there are 8 cities, and our list contains 12 in total (with repetitions), some cities must appear only once in the list and hence cannot be involved in more than 1 flight.

Let City A be a city that is involved in exactly 1 flight. Remove this flight and City A itself from the picture. What we have left is 7 cities connected using 5 flights. Note that removing the single flight involving City A cannot break the connectivity of the remaining 7 cities. We can use similar reasoning to argue that at least one of these 7 cities, call it City B, is involved in exactly 1 flight. When we remove City B, and the accompanying flight, we are left with 6 cities connected using 4 flights. Similarly, we can produce 5 cities connected using 3 flights and then 4 cities connected using 2 flights and finally 3 cities connected using 1 flight. Now we have reached a clear problem. We cannot possibly have 3 cities connected using only 1 flight. This means that we could not have had the original scenario where we claimed to have 8 cities connected using only 6 flights.

Connections to Computer Science

There is a *graph* at the centre of this problem. A graph is used to model connections. It consists of a set of vertices and a set of edges each of which connects a pair of vertices. Here, the vertices represent cities and the edges represent flights. The diagram is a visual representation of the graph.

The goal is to remove as many edges (flights) as possible while leaving a *connected* graph. A connected graph is one for which there exists a *path* between any two vertices. In general, if a connected graph has n vertices, then there exists a set of only $n - 1$ edges, chosen from the “original” edges, for which the graph is still connected if all other edges are removed. The resulting graph with $n - 1$ edges is called a *minimum spanning tree*. If a graph with n vertices has less than $n - 1$ edges, it cannot be a connected graph.

Minimum spanning trees are important in applications such as transportation and water supply networks.

Country of Original Author

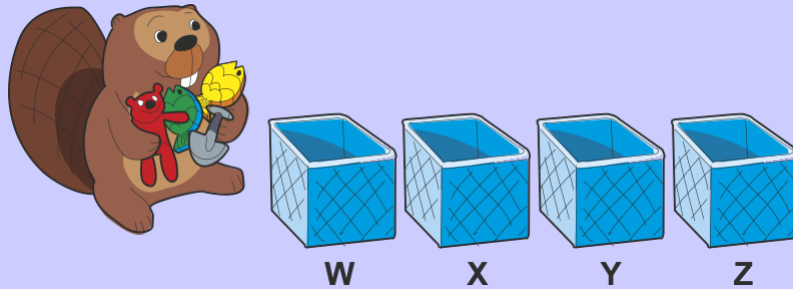
Belgium



Triple Trouble

Story

A beaver puts each of four toys into boxes labeled W, X, Y, and Z. Each box can hold any number of toys.



At least one of the three conditions in each row of the table shown is satisfied.

a toy is in X	no toy is in Y	no toy is in Z
a toy is in W	a toy is in X	no toy is in Z
no toy is in X	no toy is in Y	a toy is in Z
no toy is in W	no toy is in X	no toy is in Y
no toy is in X	a toy is in Y	no toy is in Z

Question

What is the minimum possible number of empty boxes?

- (A) 3
- (B) 2
- (C) 1
- (D) 0

Answer

(C) 1

Explanation of Answer

Notice that if there is a toy in each box, then no condition in the fourth row is satisfied. This means there must be at least one empty box. To show that it is possible to have exactly one empty box, consider placing a toy in boxes X, Y, and Z. (We could find this combination by trying the four possible combinations of three boxes.) Since a toy is in box X, the first two rows have a condition satisfied. Since a toy is in box Z, the third row has a condition satisfied. Since there is not a toy in box W, the fourth row has a condition satisfied. Finally, since there is a toy in box Y, the fifth row has a condition satisfied. We have shown that the minimum possible number of empty boxes is one.

Connections to Computer Science

Logical propositions are very common in computer science. This kind of problem in which one has to find a configuration that satisfies constraints is very common and, unfortunately, a problem that in general nobody knows how to solve in an efficient way.

This particular problem is known as the *Boolean satisfiability problem* (abbreviated as *SAT*). Three rules for each condition (*3-SAT*) are enough to make the problem very difficult to solve when the number of conditions is large. We had just five conditions, but our strategy required to look at them all each time we checked a configuration. Computer scientists know better strategies, but they are all quite inefficient; there are SAT competitions in which programmers try to write the fastest *SAT solver*. Since many different problems can be turned into an equivalent SAT one, SAT solvers can be useful in many situations.

Country of Original Author

Canada



Farmer's Report

Story

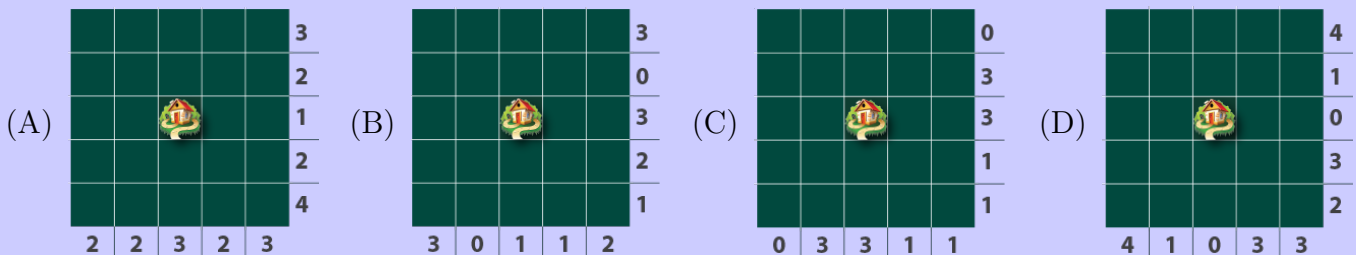
Farms are divided into square fields. There is always a farmhouse in the centre square. Every year, farmers must decide whether a field will grow wheat or grass. They must report the total number of wheat fields in each row and column. An example of a report is shown.



The totals given are accurate because there is one wheat field in each row, one wheat field in the left column, two wheat fields in the middle column, and no wheat fields in the right column.

Question

In each of the following reports, each dark green square, except the centre square containing the farmhouse, represents either a wheat field or a grass field. Which report could contain accurate totals?

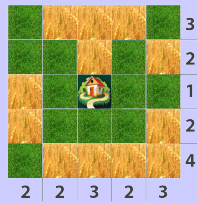


Answer



Explanation of Answer

The following configuration shows that Option A could be accurate.

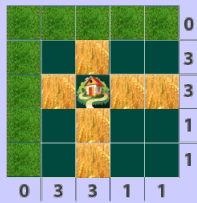


In Option B, the sum of the numbers at the bottom of the diagram is $3 + 0 + 1 + 1 + 2 = 7$ and the sum of the numbers to the right of the diagram is $3 + 0 + 3 + 2 + 1 = 9$. These must both equal the total number of wheat fields so Option B is inaccurate. Option D is inaccurate for a similar reason.

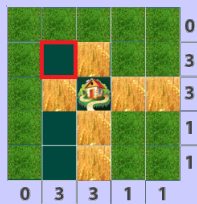
In Option C, the sums of rows and columns match, so we have to argue in a different way. The first row and the first column must have 0 wheat fields.



Thus there is only one way to cover the fields in the third row and in the third column by wheat.



Then, all others fields in columns 4 and 5 must be covered by grass.



Now, it is impossible to plant wheat on two more fields in the second row because only one more field (marked in red) is available. Therefore Option C is inaccurate.

Connections to Computer Science

The information about which fields grow grass and which fields grow wheat is fully captured by the contents of the squares in the two-dimensional diagrams. The totals for each row and column are additional summary information. However, calculating a total each time it is needed might take too much time. In general, this idea of *precomputation* can speed up computer operations. In fact, it is common for machines to store intermediate calculations by default. This is called *caching* and saves time if the machine is repeatedly asked to perform the same calculation. The downside is that these intermediate results consume memory making this a classic *trade-off* of time versus memory. For this reason, users can often find settings which control caching behaviour.

A second use of the totals is to check the validity of the diagrams. Digital data is not perfect. It can be lost or damaged. By storing both the two-dimensional information and the totals, we can check that they are consistent. Consistency will not guarantee that the data is correct but it will sometimes tell us when an error has occurred. This is related to the idea of *check digits* and *error-correcting codes*.

Country of Original Author

Cyprus

