



UNIVERSITY OF
WATERLOO



The CENTRE for EDUCATION in
MATHEMATICS and COMPUTING



2023
*Beaver
Computing
Challenge*
(*Grades 7 & 8*)

*Questions,
Answers,
Explanations,
and
Connections*

Part A

Lakes

Story

Six lakes are connected by rivers. Water always flows from a lake with a greater height above sea level to a lake with a lesser height above sea level.

An online search provides the following information about these lakes:

Lake	pH Level	Height Above Sea Level (m)	Estimated Fish Population
Atlyn	6.5	320	32000
Clare	7.1	740	1500
Doffin	6.8	490	65000
Kazba	7.2	673	4200
Mus	6.5	973	22100
Soul	6.2	382	43000

Question

In what order does water flow between these six lakes?

- (A) Atlyn, Clare, Doffin, Kazba, Mus, Soul
- (B) Mus, Clare, Soul, Atlyn, Doffin, Kazba
- (C) Kazba, Clare, Doffin, Mus, Atlyn, Soul
- (D) Mus, Clare, Kazba, Doffin, Soul, Atlyn

Answer

(D) Mus, Clare, Kazba, Doffin, Soul, Atlyn

Explanation of Answer

If we sort the table by the third column in descending order, we will have:

Name	pH Level	Height Above Sea Level (m)	Estimated Fish Population
Mus	6.5	973	22100
Clare	7.1	740	1500
Kazba	7.2	673	4200
Doffin	6.8	490	65000
Soul	6.2	382	43000
Atlyn	6.5	320	32000

Reading the names from the top row to the bottom row gives the order in which water flows between the lakes. Note that the other columns do not have any information that is necessary to solve the problem.

Connections to Computer Science

There is a tremendous amount of data that is stored in a *structured* way, whether that is in a database or in a spreadsheet. Quite often, the data has some *key* field, which distinguishes between different data elements: in this task, the name is the key field. Attached to each key field are other associated *values*: in this task, the pH level, fish population, and height above sea level are values associated with each key.

One benefit of structured data is that it can be manipulated in various ways. In particular, it may be useful to *sort* data in a particular way based on a particular values, as was done in this task. In other situations, it may be important to *filter* certain data to consider the data which falls between certain values, such as finding only those lakes with a pH level in a certain range.

Country of Original Author

Canada



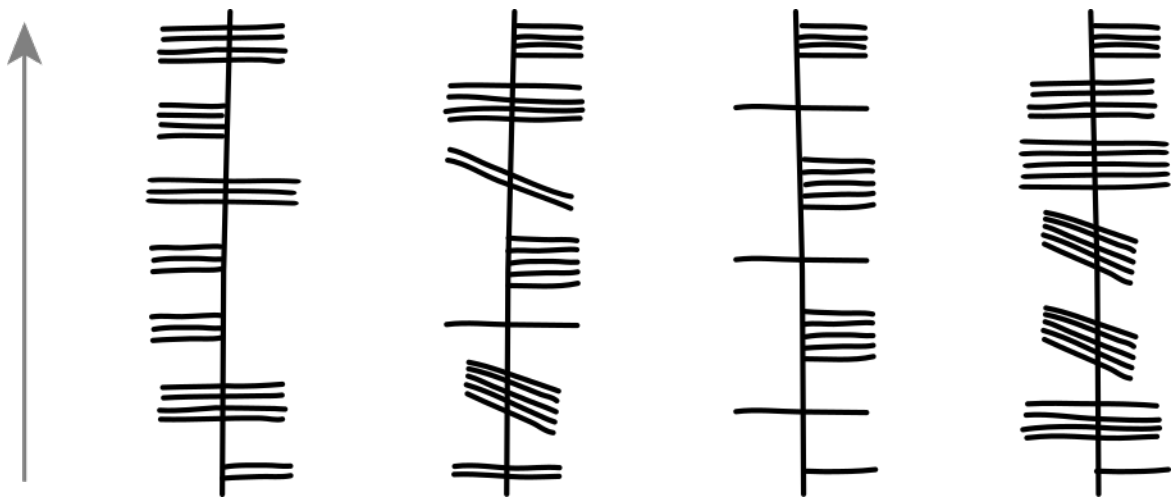
Ogham Code

Story

Ogham is a medieval alphabet used to write words vertically along a pillar.

Each letter is represented by a group of lines that always touches or crosses the pillar in the same way. Groups (letters) are separated by big gaps and arranged upwards from the bottom of the pillar to the top of the pillar.

Eabha writes four words using the Ogham alphabet as shown. The words are BANANAS, BERRIES, LETTUCE, and ORANGES, but we do not know which word corresponds to which image.



Question

From left to right, what is the order in which Eabha has written these words?

- (A) LETTUCE, ORANGES, BANANAS, BERRIES
- (B) ORANGES, LETTUCE, BANANAS, BERRIES
- (C) BERRIES, BANANAS, LETTUCE, ORANGES
- (D) LETTUCE, ORANGES, BERRIES, BANANAS

Answer

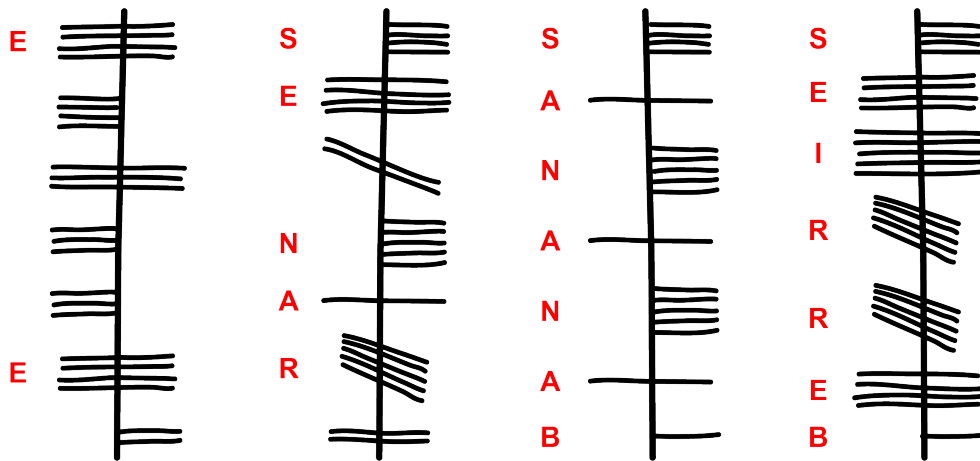
(A) LETTUCE, ORANGES, BANANAS, BERRIES

Explanation of Answer

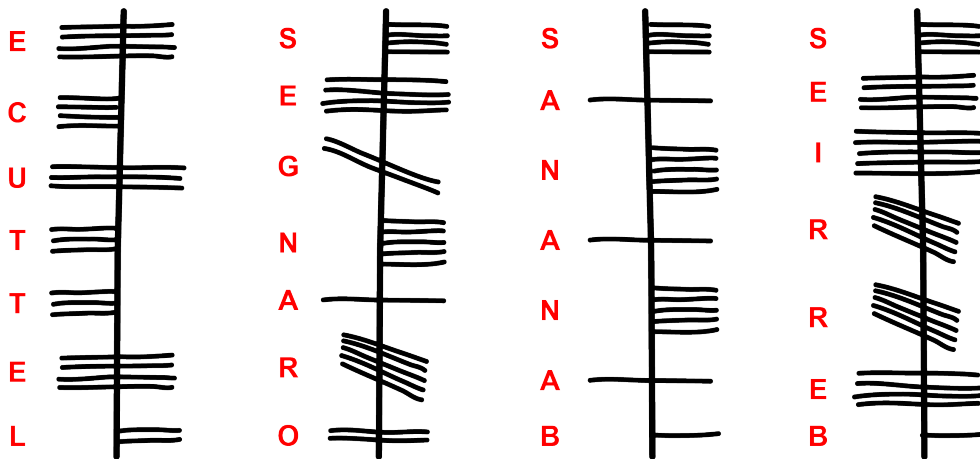
We will number the words in the image 1 to 4 from left to right.

Word 3 and Word 4 begin with the same letter, so Word 3 and Word 4 must be BANANAS and BERRIES in some order. The 2nd, 4th, and 6th letters of Word 3 are all the same, so this makes Word 3 BANANAS and Word 4 BERRIES.

This teaches us the code for the letters in those two words, which we can use to fill in some of the letters from Word 1 and Word 2 as well:



Now it becomes clear that Word 2 must be ORANGES and Word 1 must be LETTUCE:



Therefore, from left to right, Eabha wrote the words in the order LETTUCE, ORANGES, BANANAS, BERRIES.

Connections to Computer Science

This task focusses on the idea of *encoding* information in a way to keep the information secret. One method of keeping data secret so that unauthorized persons cannot access it is *encryption*, which is the key area of study in *cryptography*. Cryptography began as early as 3500 years ago, with simple methods of encryption such as replacing each letter in the message with a different letter, which is known as a *substitution cipher*. Simple substitution ciphers can be easily *decrypted* by using *frequency analysis*: some letters, such as “E” and “T” appear much more frequently than other letters, such as “Q” or “Z” in English text. Therefore, someone reading a long encrypted message can try to guess that the most frequently occurring encrypted letters are probably “E” or “T” and begin to decrypt the message. The decryption in this task is based on the common occurring letters between words.

An alternative method to a direct substitution cipher is to use *rotation-based* ciphers, which typically were mechanical devices with various rotors that had sequences of letters, and with each letter, the rotor would move some amount. The most well-known of these sorts of devices is the *Enigma machine* used to transmit messages during World War II. Similar to this machine in this task, each letter rotates a different amount, such that the sequence of letters “PA” and “SA” (51-31 and 61-21) have the letter “A” encoded with two different values. This type of encryption method is much more difficult to decrypt without knowing the original starting point of the arrow, and is much less vulnerable to frequency analysis.

Many computer systems, such as *databases* and *web browsers*, rely on strong encryption methods to prevent unauthorized access to information.

Country of Original Author


Ireland

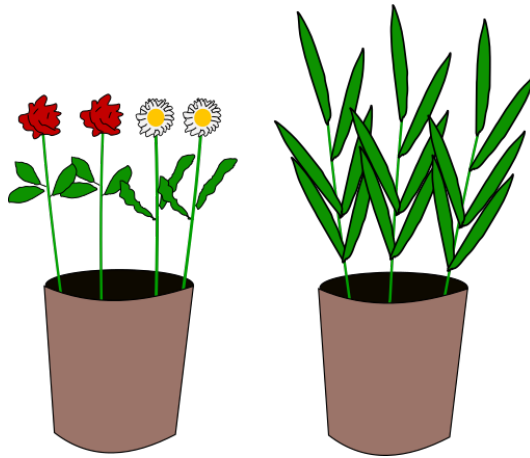


Flower Shop

Story

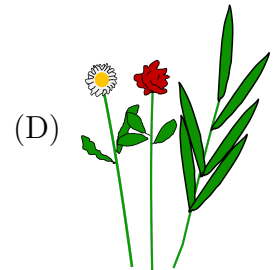
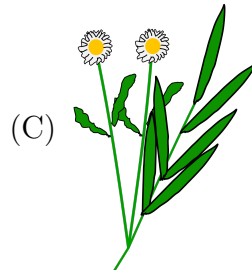
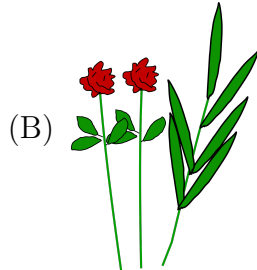
A florist makes bouquets by following these three steps in order:

1. Pick one flower from the bucket shown on the left to start the bouquet.
2. If the flower picked is a daisy , add a second daisy to the bouquet.
3. Complete the bouquet by adding at least one branch of leaves from the bucket shown on the right.

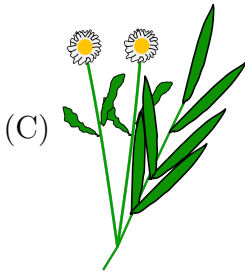


Question

Which bouquet might the florist make?

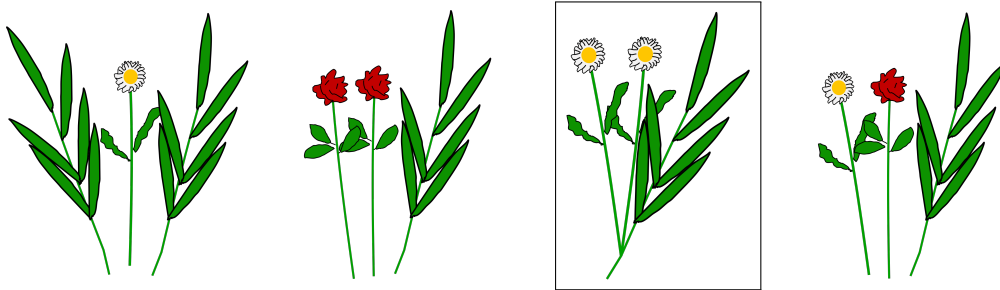


Answer



Explanation of Answer

When the first two steps are considered together, we can infer that bouquets will either have exactly 1 rose (and no daisies) or exactly 2 daisies (and no roses). This eliminates all but one of the bouquets.



Since this bouquet also has at least one branch of leaves, it also satisfies step 3. This bouquet might be made by the flower shop.

Connections to Computer Science

The instructions to compose a bouquet are an example of an *algorithm*. Formally, an algorithm is a finite sequence of instructions to accomplish some goal. Every *computer program* is a description of an algorithm.

In this task, there are two main types of instructions that are common in *programming languages*. The first type of instruction is a *conditional* instruction, where an instruction happens only if certain conditions are met. For this task, the second step to “pick another daisy” happens *if* there was a daisy picked already. Conditional statements are often represented in programming languages using *if-then-else* statements, which allow certain instructions to happen only if a given decision/question has a *true* value.

The second type of instruction in this task is the *repetition* instruction, indicating that some instructions should be repeated, usually while a certain condition is met. The third step for the florist to “pick a branch from the second bucket”, is repeated at least once, but possibly several times. In computer programs, repetition is often represented with a *loop* instruction.

Country of Original Author

Germany



Magic Tree

Story

Bain the Beaver has a magical tree growing near their home.

Whenever a bird 🐦 lands on it, the tree sprouts 2 apples.

Whenever a squirrel 🐿️ climbs up it, the tree drops 1 apple (if it has any).

Bain has also noticed that whenever a snake 🐍 visits the tree, all of the apples instantly disappear!

One morning Bain notes that the magical tree contains 25 apples. Bain then spends the rest of the day drawing pictures of all the animals that come to the tree. The drawings, in order, are:



Question

How many apples are on the tree at the end of the day?

- (A) 3
- (B) 7
- (C) 17
- (D) 31

Answer

(B) 7




Explanation of Answer

The answer is Option B. There are 7 apples on the tree at the end of the day.

Since all of the apples instantly disappear whenever a snake visits the tree, we can ignore everything that happens before the arrival of a snake. After the last snake, four birds land on the tree which means the tree sprouts $4 \times 2 = 8$ apples. Then one squirrel climbs up the tree which causes one apple to drop, leaving $8 - 1 = 7$ apples.

Connections to Computer Science

This task introduces the ideas behind two fundamental programming concepts.

The first is the idea of a *variable*. A variable is used to store information that a computer program needs. The value of a variable can change depending on what the rest of the program's instructions are. In this task, the number of apples on the tree is a variable and its value can increase () , decrease () , or reset () .

In order to decide how to change the value of a variable, a computer program needs the ability to make decisions. This is the second fundamental programming concept and is referred to as *selection*. Decision making is accomplished by using special instructions called *conditional statements* that allow selection from different possible outcomes. Conditional statements commonly take the form “**if** this is true, **then** do this instruction”. In this task, one conditional statement would be “if a bird lands on the tree, then increase the number of apples by 2.” A good exercise is to try to find other conditional statements in this task.

Country of Original Author

Canada

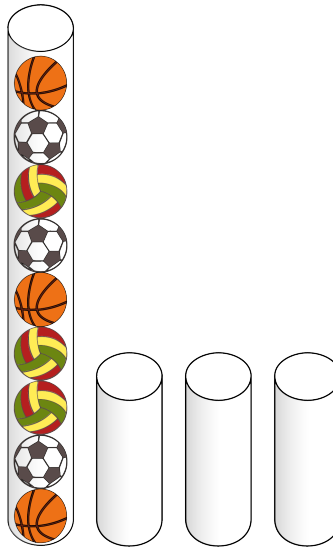


Equipment Sorting

Story

An equipment tube contains basketballs 🏀, soccer balls ⚽, and volleyballs 🏐.

Sarah reorganizes the equipment by moving the balls one by one. That is, Sarah takes each ball out of the top of the equipment tube and then drops it into the top of one of the smaller tubes. After moving each ball once, each type of ball is stored together in a smaller tube.



All the balls are the same size and each smaller tube holds three balls.

Question

Which of the following statements is **true**?

- (A) Sarah fills the tube of soccer balls first.
- (B) Sarah fills the tube of volleyballs last.
- (C) Sarah fills the tube of soccer balls after filling the tube of basketballs.
- (D) Sarah fills the tube of volleyballs before filling the tube of basketballs.

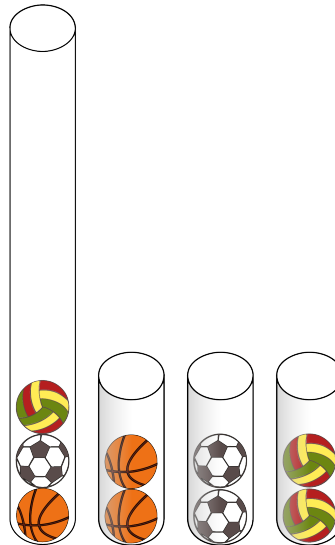
Answer

(D) Sarah fills the tube of volleyballs before filling the tube of basketballs.

Explanation of Answer

Since Sarah moves the balls, one by one, out of the top of the equipment tube, she will move the balls in order from top to bottom. A smaller tube will be filled when she moves the third ball of any type out of the top of the equipment tube.

After moving the first six balls, the tubes will look like this:



All the smaller tubes are now one ball away from being full. By examining the balls still in the equipment tube from top to bottom we can determine the order in which the smaller tubes will be filled: volleyballs first, soccer balls next, and basketballs last.

This means only the statement “Sarah fills the tube of volleyballs before filling the tube of basketballs” is true.

Connections to Computer Science

In this task, the balls are moved between tubes using two operations; adding an ball to a tube and removing the top-most ball from a tube. An individual tube can be thought of as a simple *data structure* used in computer science, called a *stack*. A stack follows the *last in, first out* or *LIFO* rule: the item that is placed in last is the first to be removed. Stacks are used frequently in computer science. One such usage is keeping track of operations that can be “undone”, such as edits to a document, browsing history in a browser, or searching through a maze by way of *backtracking* if a dead end is reached.

Country of Original Author

Saudi Arabia



Part B

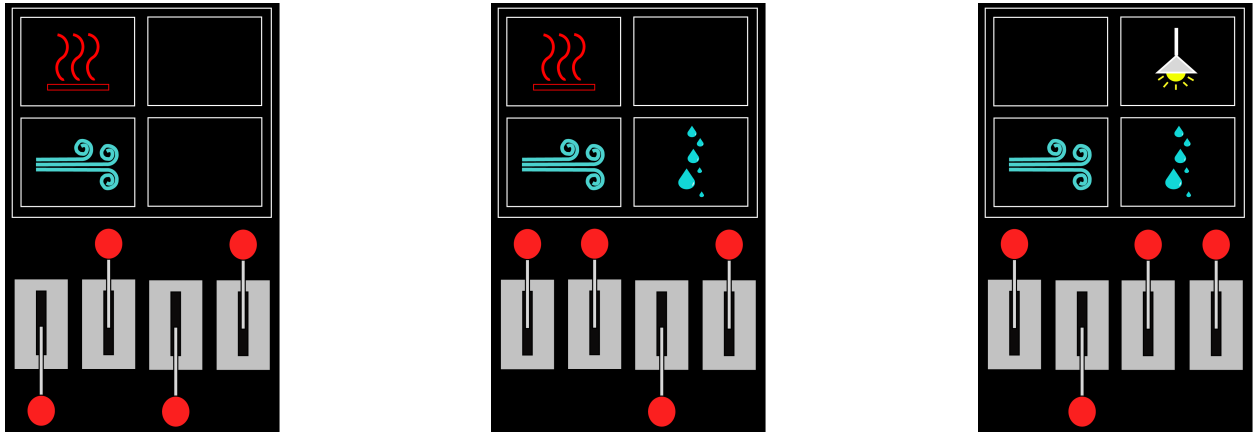
Levers

Story

Jacinta's new helicopter has a control panel with four levers that each control a different system.

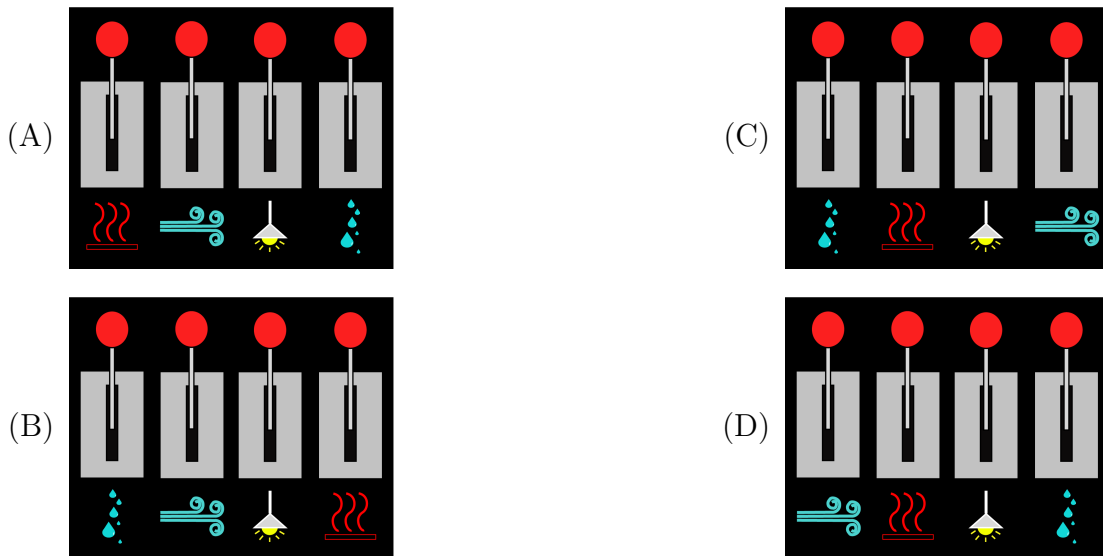
The labels on the levers are missing, and Jacinta was told that the system was wired in a confusing way so she doesn't know which lever controls which system. All she knows is that putting a lever up turns one system and its indicator light on, and putting a lever down turns one system and its indicator light off.

The image shows which indicator lights are on for three different configurations of the levers.

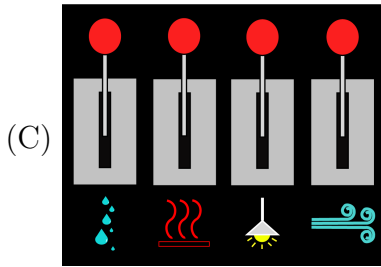


Question

Which of the following correctly matches each lever to the system that it controls?



Answer



Explanation of Answer

We will refer to the systems as heat , ventilation , lights , and humidity .

First we will number the levers 1, 2, 3, and 4, starting from the left. In the first image, levers 2 and 4 are up, and the heat and ventilation are on. Thus, levers 2 and 4 must control the heat and ventilation, in some order.

In the second image, levers 1, 2, and 4 are up, and the heat, ventilation, and humidity are on. Since we already know that levers 2 and 4 control the heat and ventilation, we can conclude that lever 1 controls the humidity.

In the third image, levers 1, 3, and 4 are up, and the ventilation, light, and humidity are on. Since we already know that levers 2 and 4 control the heat and ventilation, we can conclude that lever 4 must control the ventilation. It follows that lever 2 must control the heat. We also know that lever 1 controls the humidity, so we can conclude that lever 3 must control the lights.

Connections to Computer Science

This problem focusses on keeping track of the arrangement of four levers and the systems they control. Since a system is either on or off, we can keep track of each system using a *binary number*. A binary number is either the number 0 or 1, which we can think of as “off” or “on”, which in this task is indicated by a lever being either “down” or “up”.

We can then think about the *state* of all systems as being a list of four binary numbers. For example, if we list the systems in the order (heat, lights, ventilation, humidity), then the state (0, 1, 0, 1) would indicate that heat and ventilation are off, and that lights and humidity are on. In total, there would be $2^4 = 16$ different states for these systems.

This problem then involves determining the state of the systems based on the arrangement of the levers. This operation of updating a state is one of the fundamental operations of the *central processing unit (CPU)*: given the current state and some input, what new state should the computer be in? Every mouse click or key press changes the state of a computer, and the state is stored on the computer as a sequence of binary numbers.

Country of Original Author

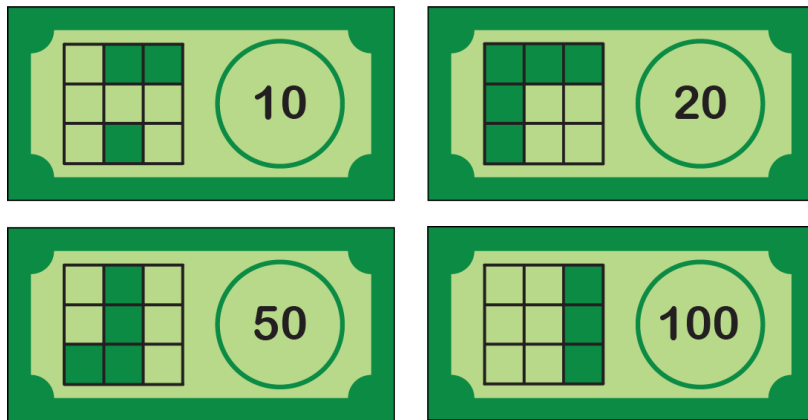
Perú



Vending Machine

Story

In Beaverland, paper money comes in four different values: 10, 20, 50, and 100. These values are written on the paper money and they are also encoded using a grid of nine squares that are either blank or filled in, as shown.

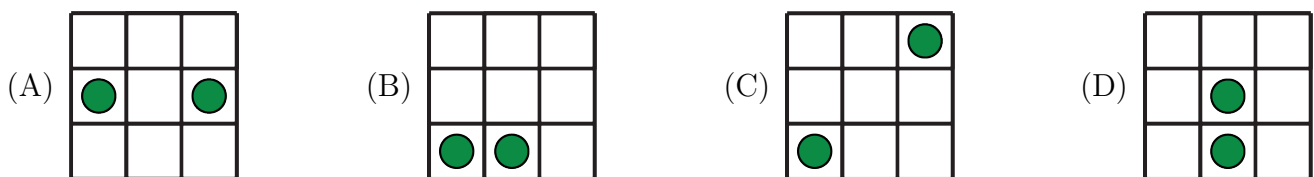


When paper money is inserted into a vending machine, the machine scans all nine grid squares and determines whether each square is blank or filled in. This is how the vending machine identifies the value of the paper money. For example, if the vending machine determines that only the squares in the rightmost column are filled in, it identifies the paper money as having a value of 100.

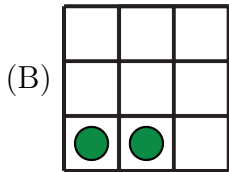
Sylvia has noticed that the vending machine doesn't need to scan all nine squares. It could correctly identify the value of the paper money by scanning only two squares.

Question

Which two squares could Sylvia reprogram the vending machine to scan (ignoring all others) in order to identify the value of the paper money?



Answer



Explanation of Answer

We will number the squares from 1 to 9 as shown.

1	2	3
4	5	6
7	8	9

If the vending machine scans only squares 4 and 6 and finds them both blank, it will not be able to tell the difference between the paper money with values 10 and 50.

If the vending machine scans only squares 3 and 7 and finds square 3 filled in and square 7 blank, it will not be able to tell the difference between the paper money with values 10 and 100.

If the vending machine scans only squares 5 and 8 and finds them both blank, it will not be able to tell the difference between the paper money with values 20 and 100.

However, if the vending machine scans only squares 7 and 8 it will be able to successfully identify the value of the paper money as follows:

Square 7	Square 8	Paper Money Amount
blank	filled	10
filled	blank	20
filled	filled	50
blank	blank	100

Connections to Computer Science

Sylvia is learning to identify the value of money by studying the squares on the money and looking for distinguishing characteristics. Computers can also be taught to do this, using a process called *machine learning*. As part of machine learning, a computer is presented with a collection of data, and then it searches for patterns within this data to draw conclusions.

Detecting recurring patterns in given data is an important skill in computer science, generally referred to as *pattern recognition*. For example, in *image recognition*, pattern recognition techniques assist in identifying objects in images, classifying them into different categories, and even recognizing faces. In *natural language processing*, pattern recognition enables computers to understand the structure of sentences, identify topics, and even generate human-like text.

Country of Original Author

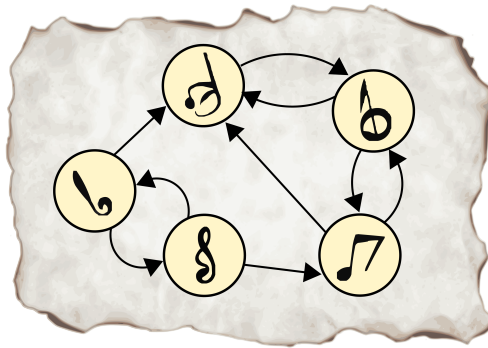
South Korea



Restoring Music

Story

Juno has found an ancient diagram that describes how to compose pieces of music using just five types of notes. Any note can be selected as the first note, but a note can only be selected next if there is an arrow pointing to it from the previous note.











Juno has also found the sheet containing a piece of music which was composed using the diagram above. Two notes are missing due to a hole torn in the sheet as shown below.



Question




Juno would like to restore this sheet of music. Which two notes must be missing?




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

Answer





(B) 

Explanation of Answer

There are two arrows pointing *away* from the note  which means there are two possibilities for the first missing note:  or .

There are two arrows pointing *to* the note  which means there are two possibilities for the second missing note:  or .





Since there is no arrow from  to itself, and no arrow from  to , the first missing note cannot be  and it must be .




Since there is no arrow from  to itself, but there *is* an arrow from  to , the second missing note must be .

Therefore, the two missing notes are .

Connections to Computer Science

The diagram describing how to compose music is an example of a *directed graph*. Each note is represented as a *vertex* of the directed graph. Each (*directed*) *edge* will go from one note to the next possible note.

The reason we say the graph is “directed” is because the order of the notes is only valid in one direction. For example, there is an edge from  to , but not the other way around. We would call  a directed *neighbour* of .

In this task, we want to find a *directed path* between the two  notes. One way to find this path is to perform a *breadth-first search*, where all of the neighbours of the  note are determined, and then repeatedly determine the neighbours of each of those neighbours, until the second  note is found.

The idea of searching for a path in a directed graph has many applications, such as mapping out a driving route, determining how to send information through the internet, and determining recommendations for who you may want to connect with on social media platforms.

Country of Original Author

South Korea



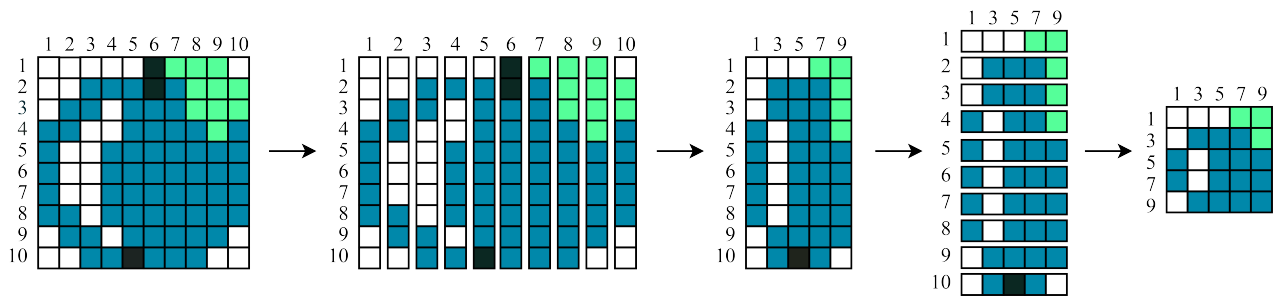
Snail Compress

Story

A beaver has a special technique to shrink images.

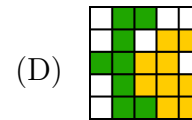
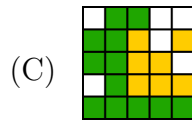
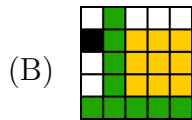
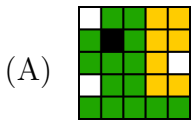
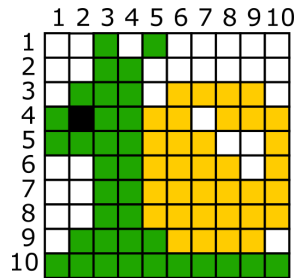
First, they cut the original image into 10 equally-sized vertical strips. Then, they remove the even-numbered strips and assemble the odd-numbered strips to create a new image.

Next, they cut the new image into 10 equally-sized horizontal strips. Then, they remove the even-numbered strips and assemble the odd-numbered strips to create a complete shrunken image. Here is an example:



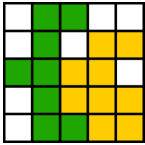
Question

If the beaver uses this technique to shrink the image below, what is the complete shrunken image?



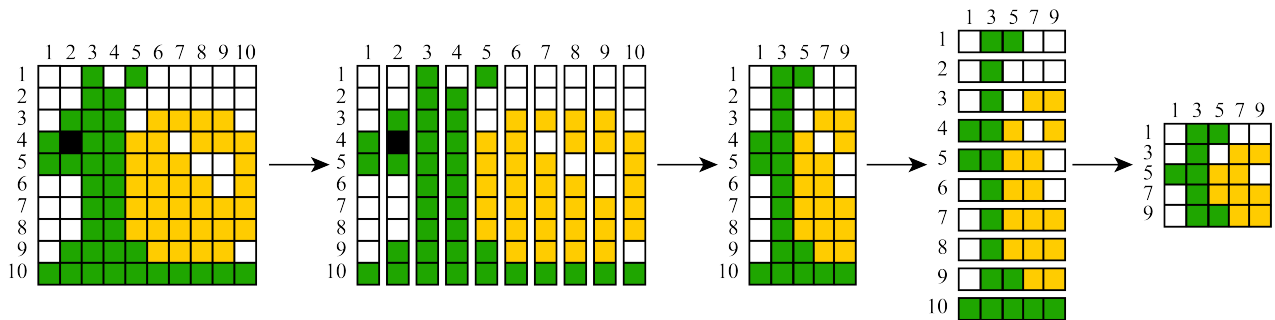
Answer

(D)



Explanation of Answer

The steps the beaver uses are illustrated below.



Alternatively, you can solve this multiple-choice question by considering the four options. Since the dark “eye” of the snail is in an even-numbered horizontal strip, it will be removed. This eliminates Options A and B. Similarly, the bottom horizontal strip (numbered 10) is the only one with more than four green cells. Since it is an even-numbered strip, it will be removed. This eliminates Option C because it contains five green cells in its bottom row. Therefore, of the options, Option D must be correct.

Connections to Computer Science

This task is focussed on *image compression*, which is the process of reducing the size of an image file by eliminating or minimizing some of the information in the image while still maintaining sufficient image quality for use. Reducing the size of an image can save both storage space and improve the time needed to transfer the image between devices, such as between two phones.

In this task, the total amount of information stored is only 25% of the original amount of information. However, some of the details of the image are removed. This type of compression is an example of a *lossy compression method* that works well for certain types of images that don’t have much variation.

There are also *lossless compression methods*, which allow the original information to be stored in a smaller size completely, without any loss of information. Two well-known examples of lossless compression are ZIP files and PNG files.

Country of Original Author

Vietnam

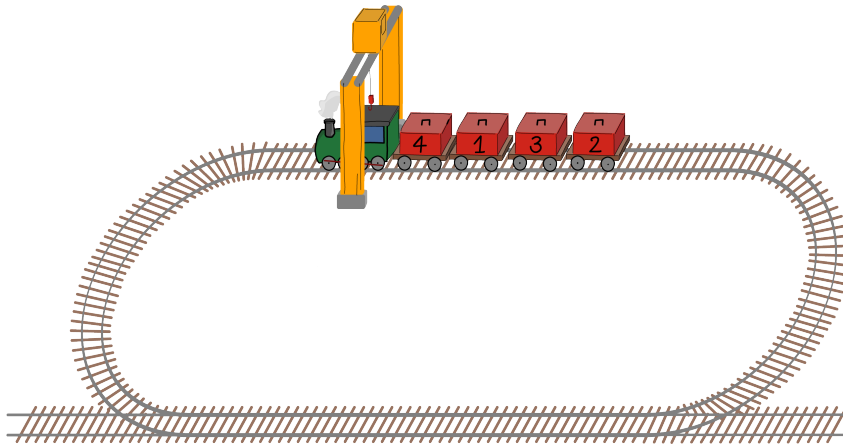


Unloading

Story

Freight trains consist of an engine followed by wagons, each holding a numbered box. The boxes must be unloaded in increasing order, starting from box 1. To unload a box, its wagon must be positioned directly below the crane.

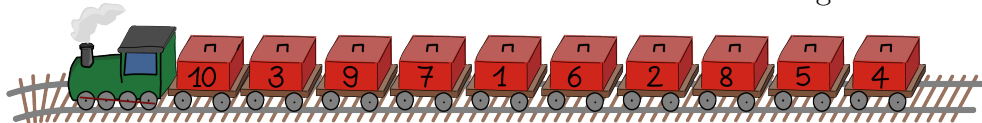
The crane is in a fixed position and trains can only move forward on a loop. Usually, this means that several rounds are needed to unload all the boxes. Each round begins with the engine directly under the crane.



In the example shown, the boxes have to be unloaded in the order 1, 2, 3, 4 and three rounds are needed to do this. In the first round of unloading, the train moves forward to skip box 4, unload box 1, skip box 3, and unload box 2. The train then goes around the track until the engine is under the crane again. In the second round of unloading, the train moves forward to skip box 4, skip the empty wagon, and unload box 3. The train then has to come back for a third round in order to unload box 4.

Question

How many rounds will be needed to unload all the boxes from the following train?



- (A) 5
- (B) 6
- (C) 7
- (D) 8

Answer

(C) 7

Explanation of Answer

We can simulate the unloading of the boxes in the required order 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 in as few rounds as possible. This just means that a box is unloaded if it is directly below the crane and all lower-numbered boxes have already been unloaded. Carefully following this procedure, we can determine that boxes are unloaded as shown in the following table.

Round	Boxes Unloaded
1	1,2
2	3,4
3	5
4	6
5	7,8
6	9
7	10

Alternatively, one might discover a general rule that determines exactly how many extra rounds (rounds after the first round) are needed. For each number in the sequence 1, 2, \dots , 10, if the next number in the sequence comes to the left of it on the train, an extra round is needed. For instance, if 3 appears to the left of 2, then 3 will be skipped to unload 2 and so an extra round is needed to bring 3 under the crane. For this question, there are six such pairs that are out of order on the train: (2,3), (4,5), (5,6), (6,7), (8,9), and (9, 10). Therefore, six extra rounds are needed, for a total of seven rounds.

Connections to Computer Science

This task has the concept of counting *inversions* as the key idea. Generally, an inversion in a list of numbers is where a number, say 6, has the next largest number, 7, written earlier in the list. In the context of this task, an inversion is where box K has box $K + 1$ somewhere to the left of it on the train. For each such inversion, an extra round is needed. If we count the number of inversions, we get the answer.

Counting inversions with respect to a desired sequence has many applications in computer science. For some *sorting algorithms*, such as *insertion sort*, the number of inversions tells us how many swaps we need to sort a given sequence. Consider another example from online shopping, where two customers rank the same set of items in order of preference. Then, the number of inversions in their rankings tells us how aligned their tastes are, and this information can be used by online retailers to identify “similar” customers to make product recommendations.

Country of Original Author

India



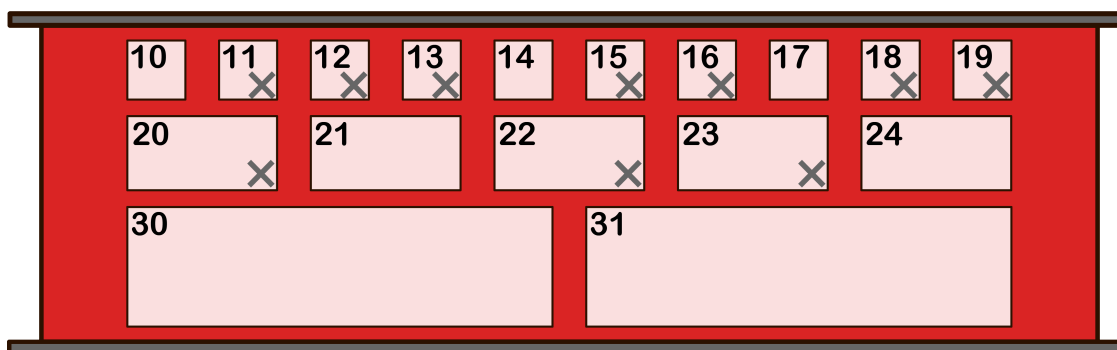
Part C

Lockers

Story

When packages arrive at the post office they are placed in lockers to await pick up. The top row of lockers can only hold small packages. The middle row of lockers can hold small or medium packages. The bottom row of lockers can hold packages of any size. Each locker can only hold one package at a time.

The following image shows what the lockers at the post office currently look like. Lockers marked with an X are holding a package.



When a new package arrives, it is placed in the lowest-numbered available locker in which it can fit. When a customer arrives to pick up a package from a locker, the locker becomes available again.

The post office has opened for the day and the following five events occur in this order:

- Four small packages arrive.
- The packages in lockers 11 and 19 are picked up.
- Two medium packages arrive.
- The packages in lockers 20 and 21 are picked up.
- Two small packages arrive.

Question

Then one more small package arrives. In which locker is it placed?

- (A) 20
- (B) 19
- (C) 24
- (D) 17

Answer

(A) 20

Explanation of Answer

We can keep track of the available lockers before and after each of the five events.

Event	Notes	Available Lockers
Post office opens		10, 14, 17, 21, 24, 30, 31
4 small packages arrive	They occupy lockers 10, 14, 17, 21	24, 30, 31
Pickup from lockers 11 and 19	This frees lockers 11, 19	11, 19, 24, 30, 31
2 medium packages arrive	They occupy lockers 24, 30	11, 19, 31
Pickup from lockers 20 and 21	This frees lockers 20, 21	11, 19, 20, 21, 31
2 small packages arrive	They occupy lockers 11, 19	20, 21, 31

The small package that arrives next is placed in the lowest-numbered available locker, which is locker number 20.

Connections to Computer Science

The key computer science concept outlined in this task is *memory allocation strategies*. Memory allocation strategies are used by the *operating system*, which executes various *applications* or *programs* and assigns each application/program an amount of memory in *random access memory (RAM)*. One such strategy is the *best-fit* strategy, where the smallest available block of memory that is large enough for the program/application is the one that is chosen. The best-fit strategy is what is used in this task: the lowest-numbered empty locker that is large enough to hold the package is chosen.

There are other allocation strategies, such as *first-first*, where the first available slot is chosen, and *worst-fit*, where the largest available slot is chosen. It is worth noting that for **any** allocation strategy, there is a sequence of allocations and deallocations that will cause an *out of memory* error, even though there is enough available memory. For example, in this task, if 7 small packages arrive, then all lockers will be allocated. Then, even if all the lockers from 10 to 19 are emptied, there is no room for a medium sized package. When this happens, we say that memory has been *fragmented*. In operating systems, fragmentation can be resolved by *paging* blocks of allocated memory into smaller units, or by *compacting* blocks to free up memory.

Country of Original Author

Iceland



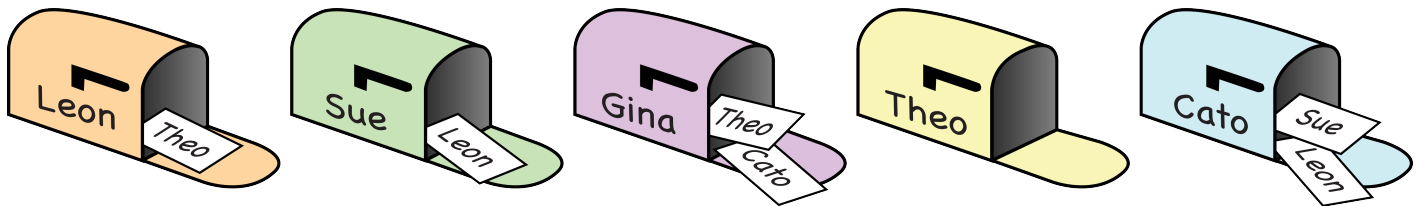
Delivering Mail

Story

When beavers want to send mail to each other they leave the mail in their own mailbox. Percy, the mail-delivery beaver, then does two things at each mailbox:

- He collects any mail in the mailbox and puts it in his bag.
- He delivers all the mail for that beaver from his bag by putting it in their mailbox.

At the start of Percy's shift, his bag is empty and the beavers' mailboxes contain the following mail:



At the end of Percy's shift, all the mail has been delivered. Percy also notes that he only had to visit each mailbox exactly once (which is unusual for Percy).

Question

In which of the following orders must Percy have visited the mailboxes?

- (A) Gina \rightarrow Cato \rightarrow Leon \rightarrow Sue \rightarrow Theo
- (B) Gina \rightarrow Sue \rightarrow Cato \rightarrow Theo \rightarrow Leon
- (C) Gina \rightarrow Cato \rightarrow Sue \rightarrow Leon \rightarrow Theo
- (D) Cato \rightarrow Gina \rightarrow Sue \rightarrow Leon \rightarrow Theo

Answer

(C) Gina \rightarrow Cato \rightarrow Sue \rightarrow Leon \rightarrow Theo

Explanation of Answer

Since Percy begins his shift with an empty bag and visits each mailbox exactly once, it follows that when Percy visits a beaver's mailbox, no other mailboxes can contain mail for that beaver.

The first mailbox visited must belong to a beaver who is not receiving any mail. This can only be Gina.

Now the remaining mailboxes contain mail for Theo, Leon, and Sue. Since Cato is the only remaining beaver without mail for them in a mailbox, Percy must have visited Cato next.

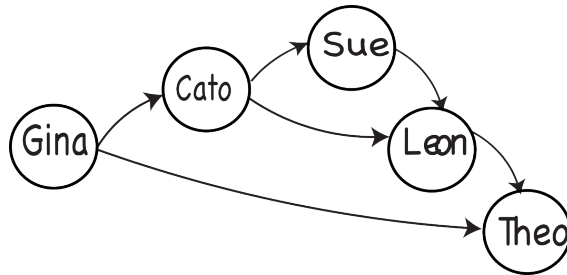
Now Leon, Sue, and Theo's mailboxes are left. Sue's mailbox has mail for Leon, and Leon's mailbox has mail for Theo. Therefore, Percy must have visited these three mailboxes in the order Sue, Leon, and Theo.

Therefore, Percy must have visited the mailboxes in the order:

Gina \rightarrow Cato \rightarrow Sue \rightarrow Leon \rightarrow Theo

Connections to Computer Science

The mail that is initially in the mailboxes before Percy starts delivering the mail provides information about the order in which Percy can visit the mailboxes. We can put this information together to create the following diagram called a *graph*. A graph consists of *nodes*, which in this diagram are the circles with names, and (*directed*) *edges* between the nodes, indicated as arrows from one node to another. In this graph, an arrow from node A to node B means person A wants to send mail to person B.



This diagram is called a *directed acyclic graph*, or DAG for short. A graph is “acyclic” because it doesn’t contain any *cycles*: that is, there is no path starting and ending at the same node. A DAG is commonly used to describe relationships in computer science, such as for *scheduling*, *parallel processing*, and *data compression*. This task asks to find the order in which Percy must visit the mailboxes. This is called a *topological ordering* of the corresponding DAG, where the names are arranged in a line with all arrows pointing forward.

A good exercise is to show that there is only one topological ordering for this DAG. That is, show why Percy *must* visit the mailboxes in the order given by Option (C).

DAGs are used everyday to solve task scheduling problems. For example, cooking usually requires multiple tasks. We make a mental list of the order of the tasks: some tasks cannot start until others are completed, while others can start at any time.

Country of Original Author

Canada



Logic Treasure

Story

An island contains three treasure chests: one by the mountains, one under the palm tree, and one on the beach. At the start of the day all three treasure chests were empty. Then, at some point during the day, Pirate Beaverbeard filled one of the chests with gold.



Three treasure hunters explored the island. One of them did all of their exploring *before* Beaverbeard filled a chest with gold. The other two treasure hunters did all of their exploring *after* Beaverbeard filled a chest with gold. None of the treasure hunters ever found the gold; they only discovered empty chests, as shown below.

Treasure Hunter	Empty Chest(s) Discovered
Alice	on the beach
Bob	on the beach / under the palm tree
Clark	by the mountains / under the palm tree

Question

Which treasure chest was the gold in?

- (A) The chest under the palm tree
- (B) The chest on the beach
- (C) The chest by the mountains
- (D) The chest cannot be determined

Answer

(C) The chest by the mountains

Explanation of Answer

Since there are three possible places for the gold to be, we can explore each possibility to see which one satisfies the details of the story.

If the gold was in the chest under the palm tree, then both Bob and Clark must have explored the island before Pirate Beaverbeard arrived. The story tells us that only one treasure hunter explored the island before, so this is a contradiction. The chest under the palm tree cannot contain the gold.

If the gold was in the chest on the beach, then both Alice and Bob explored the island before Pirate Beaverbeard arrived. This again contradicts the story. The chest on the beach cannot contain the gold.

If the gold was in the chest by the mountains, then only Clark needs to have explored the island before Pirate Beaverbeard arrived. Assuming that Alice and Bob explored the island after, the story is satisfied.

Therefore, Pirate Beaverbeard filled the chest by the mountains with gold.

Connections to Computer Science

This task involves *logical reasoning*. Specifically, reasoning about the three chests using logical expressions involving *AND*, *OR*, or *NOT*.

Each observation can be thought of as a *NOT* expression: that is, the chests examined by the treasure hunters are known to *NOT* contain treasure when they opened the chests.

The information gathered by each person can be connected together with an *AND*. For example, Clark states that there is no treasure in the chest by the mountains *AND* no treasure in the chest under the palm tree.

We know that one of the chests will have gold at some point. In other words, either the gold is in the chest on the beach *OR* in the chest under the palm tree *OR* in the chest by the mountains.

Logic plays a key role in computer science in a huge variety of areas: databases, programming languages, artificial intelligence, hardware and software design and verification, etc. Logic is one of the fundamental concepts that underlies *computational thinking*: being able to take information, model it logically, and make *logical conclusions* from that model.

Country of Original Author

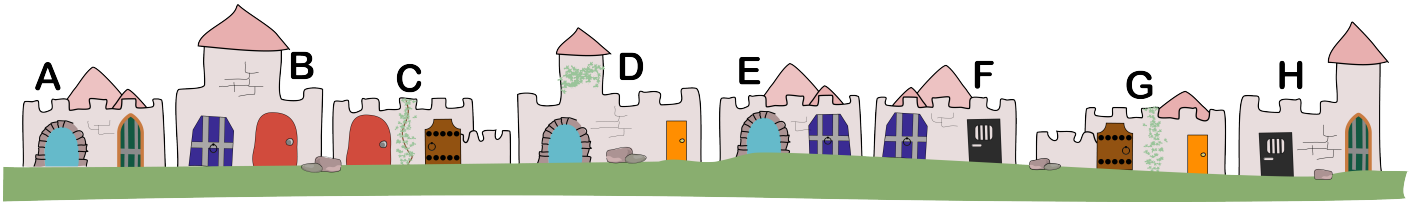
Uruguay



Magical Doors

Story



There are eight buildings, labelled A through H, along a road as shown below.



The only way to travel between the buildings is by using magical doors. There are seven different types of doors:



Each building has two different doors. When you exit a building through one of its doors, you can then enter any of the other buildings that have a door of the same type.

For example, if you exit building A via the leftmost door , then you can enter either building D or building E, and if you exit building A via the rightmost door , then you will enter building H.

Question

If you passed through the fewest buildings possible starting in building A and ending in building C, how many **types of doors** did you travel through?

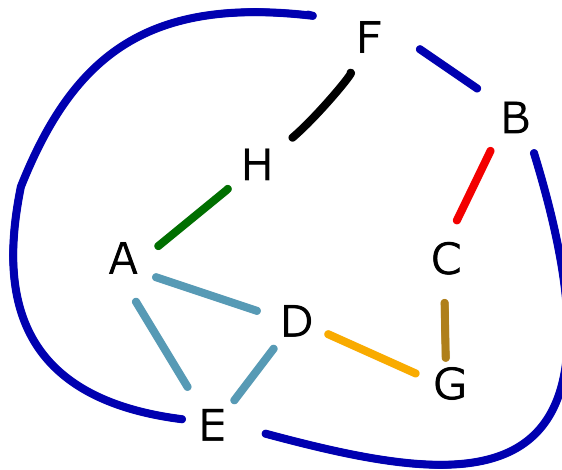
- (A) 2
- (B) 3
- (C) 4
- (D) 5

Answer

(B) 3

Explanation of Answer

The following diagram can help us solve and visualize this problem. It consists only of the labels of the buildings and lines connecting two buildings exactly when they have a door of the same type. Notice how the colours of these lines match the colour of the corresponding door.



Now it is easier to see that the only buildings you can reach from building A using one door are buildings H, D and E. From there, the only new buildings you can reach using a total of two types of doors are buildings F, G and B. This means that you cannot travel from building A to C using fewer than three types of doors. However, you can travel from building A to building C using a total of three types of doors. One way to do this is to travel from building A to building D and then to building G before using a third type of door to enter building C.

Connections to Computer Science

As shown in the Explanation of Answer, this task involves finding the *shortest path* in a *graph*. Graphs are composed of a set of *nodes* and a set of *edges* between nodes. In this task, the nodes are the buildings, and the edges connect two buildings which have a door of the same type.

One major application of graphs is to represent maps for driving: each street address is a node and roads are edges. When driving, finding the shortest path is the most common problem to solve. There are several *graph algorithms* which can help us find the shortest path. Two examples of shortest path algorithms are *Dijkstra's algorithm* (used in common GPS systems) and the *Warshall-Floyd algorithm* (used in Google Maps).

Country of Original Author

Japan

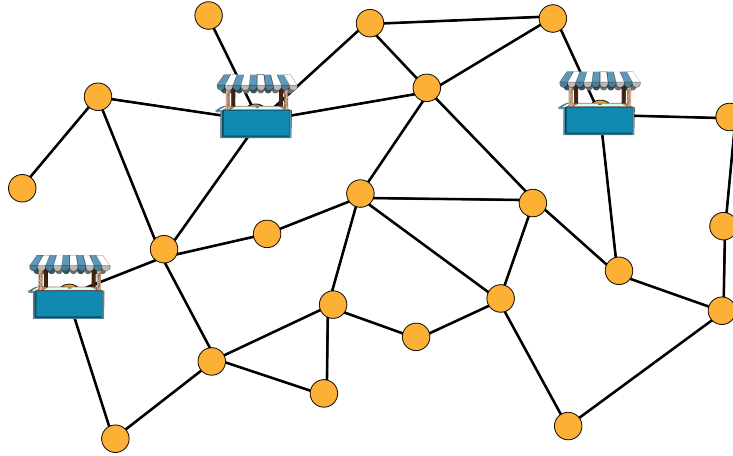


Juice Carts

Story

To celebrate National Juice Day, three juice carts have been placed in a huge park. The juice carts will distribute free juice.

The image below shows a map of the park. The juice cart icons represent points of interest where a juice cart has been placed, circles represent other points of interest, and the lines represent paths.



It takes 5 minutes to walk between any two points of interest which are directly connected by a path. There are some points of interest from which you have to walk more than ten minutes in order to reach a juice cart. This is considered too far to walk for free juice. Therefore, a point of interest must be chosen for another juice cart. After this fourth juice cart is placed, it must be possible to reach a juice cart from any point of interest in at most 10 minutes.

Question

How many different points of interest can be chosen for the location of the fourth juice cart?

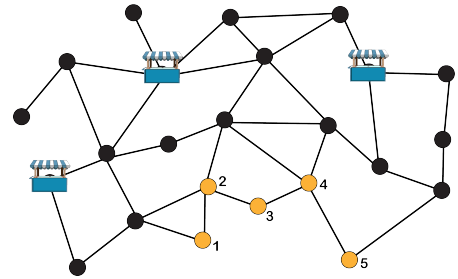
- (A) 1
- (B) 2
- (C) 3
- (D) 4

Answer

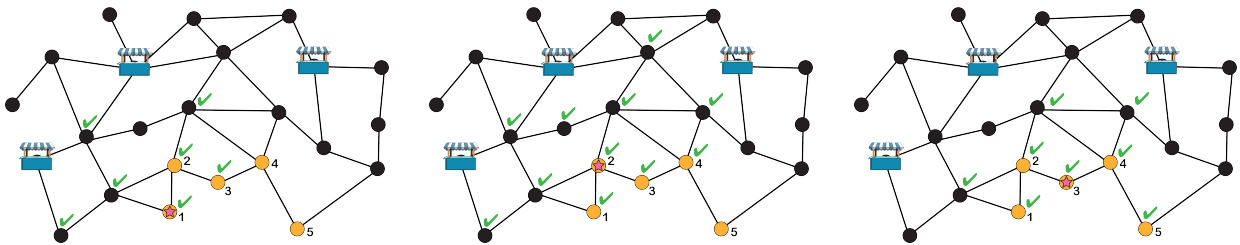
(B) 2

Explanation of Answer

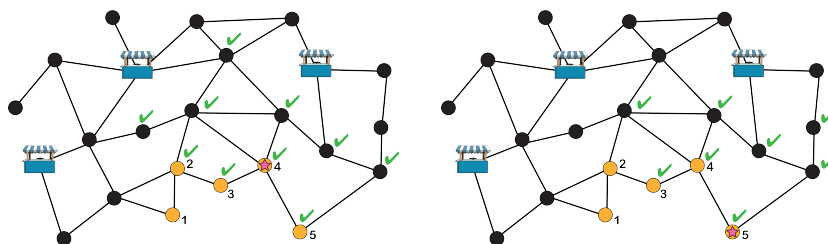
The points of interest where you can currently get free juice by walking at most ten minutes have been coloured black. The five points of interest where you must walk more than ten minutes are numbered. The placement of another juice cart must make it possible to get free juice by walking at most ten minutes from these five points of interest.



To get free juice from point of interest 1, a juice cart must be placed either on point of interest 1, or five or ten minutes away. The first picture below shows all these places using checkmarks. The second picture shows all the places where a juice cart could go in order to get free juice from point of interest 2. The third picture shows all the places where a juice cart could go in order to get free juice from point of interest 3.

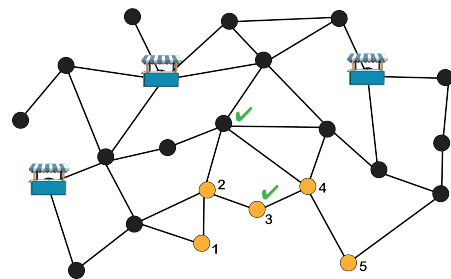


The first picture below shows all the places where a juice cart could go in order to get free juice from point of interest 4. The second picture shows all the places where a juice cart could go in order to get free juice from point of interest 5.



There are two points of interest that have checkmarks in all five of the pictures. These represent the places where a juice cart could go so that it is possible to reach a juice cart from any point of interest in at most 10 minutes.

Therefore, there are two different choices for the placement of the fourth juice cart.



Connections to Computer Science

This task deals with *graphs*. A graph is composed of a set of *nodes* and a set of *edges* connecting pairs of nodes. Graphs are used in many different contexts: for example, streets in maps, relationships between people, states in a closed system, or words organized hierarchically.

Some properties of graphs are relatively easy to determine, such as whether a graph is *connected*. Other properties of graphs can be more difficult to determine. In this task, we are trying to determine a variant of a *dominating set* of nodes. A set of nodes is dominating if every node in the graph is in the dominating set or is connected by an edge to a node in the dominating set. In this task, we are looking for a *2-distance dominating set*. A 2-distance dominating set has the property that every node is at most distance two away from a node in the set. That is, a node is either in the set, connected by an edge to a node in the set, or connected by an edge to a node which is connected by another edge to a node in the set. Finding the size of the smallest dominating set is a very difficult problem: it is in a category of problems called *NP-complete*, for which it is believed there is no efficient algorithm to solve the problem.

Country of Original Author

Brazil

