UNIVERSITY OF WATERLOO

The CENTRE for EDUCATION in MATHEMATICS and COMPUTING

2022 Beaver Computing Challenge (Grades 7 & 8)

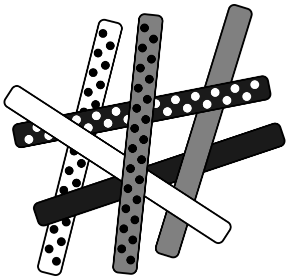
Questions, Answers, Explanations, and Connections

Part A

Pick Up Sticks

Story

Ana drops six sticks on a table as shown.

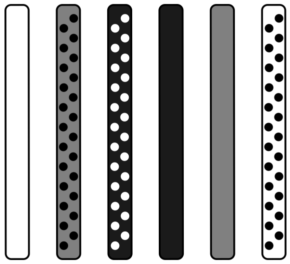


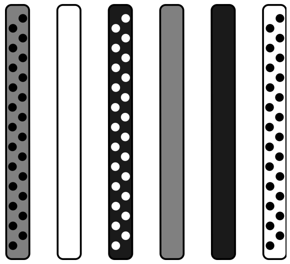
Then she picks all the sticks up according to the following rules:

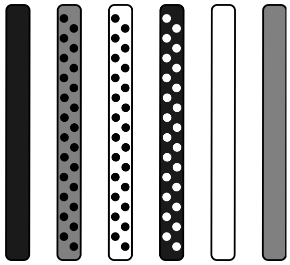
1. Pick up one stick at a time.
2. Only pick up a stick if no other stick is on top of it.

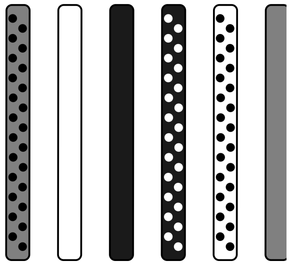
Question

In which order did Ana pick up the sticks?

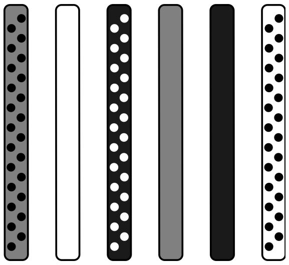
(A) 

(C) 

(B) 

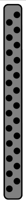
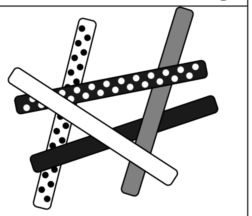
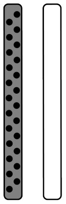
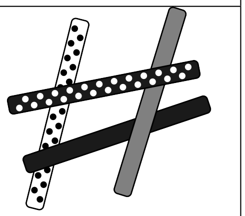
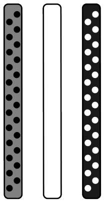
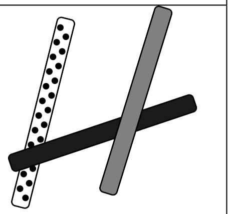
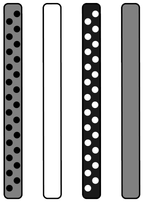
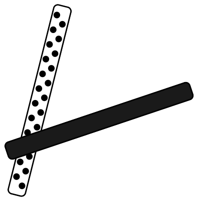
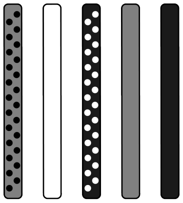
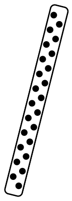
(D) 

Answer

(C) 

Explanation of Answer

According to the rules, Ana must pick up the stick on the top of the pile each time. Therefore, she must pick up the sticks as follows.

Sticks Removed	Sticks Remaining
	
	
	
	
	

Thus, Ana picked up the sticks in the order shown in Option C.

Connections to Computer Science

This task focusses on following operations in *sequential order*. That is, each stick must be removed from the top of the pile one at a time. After each operation, the *state* of the pile of sticks changes, and a new stick at the top.

The pile of sticks can be thought of as a simple *data structure* is 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.

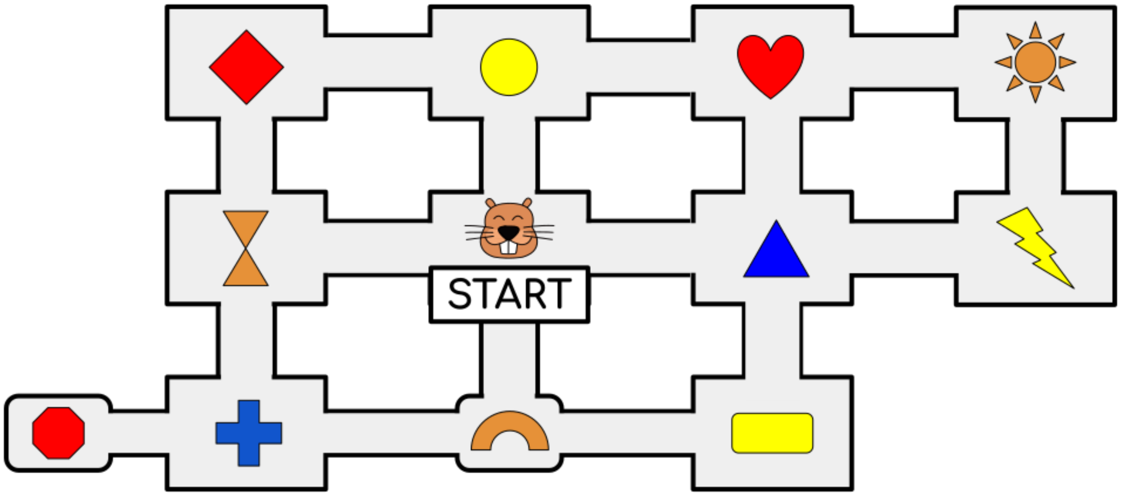
Country of Original Author

Brazil

Missing Library Book

Story

Bibako has left a book somewhere in their house. The picture below is a map of the house. They begin searching for the book in the room marked START. Each of the other rooms is marked with a picture.

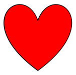


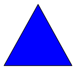
Bibako moves around the house from room to room until they find the book. Their movements, in order, are represented by the sequence of arrows below. Each arrow represents a move from Bibako's current location to an adjacent room indicated by the direction of the arrow.

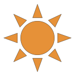
↓	←	←	→	↑	↑	→	↓	→	→

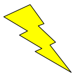
Question

Which picture corresponds to the room where the book is found?

(A) 

(B) 

(C) 

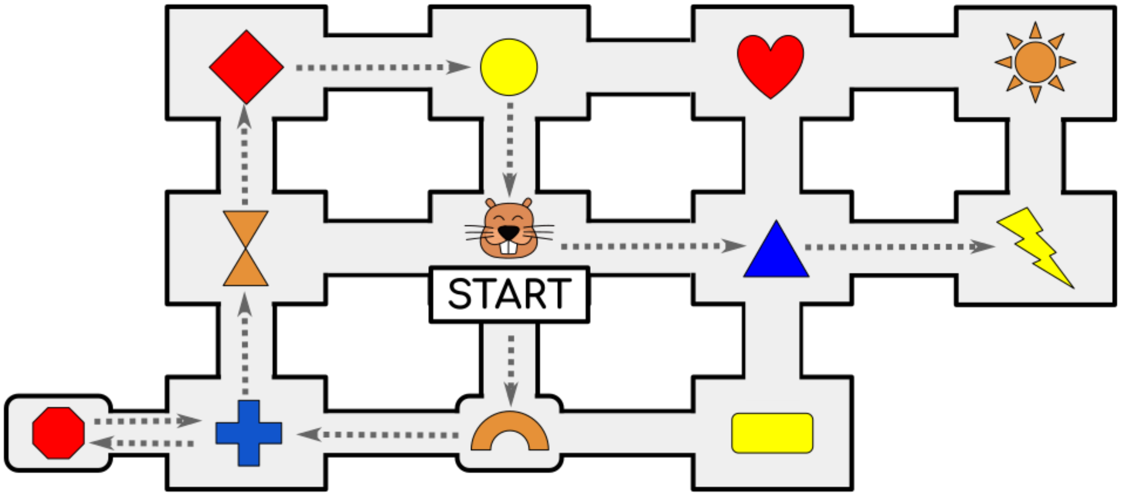
(D) 

Answer

(D) ⚡

Explanation of Answer

The picture below shows Bibako's movements.



A lightning bolt marks the room where Bibako ends and finds the book.

Connections to Computer Science

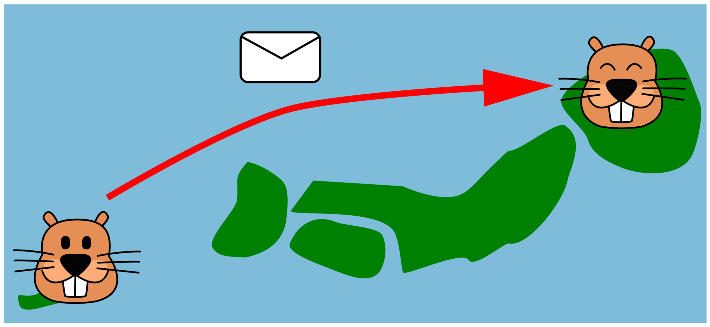
Each arrow in this task indicates what position to move to next, and the sequence of all arrows can be thought of as the *algorithm* that is to be followed. An algorithm is a sequence of operations, and this task is concerned with *tracing* that algorithm: that is, remembering the *state*, which in this task is the position that Bibako is in after each step. The ability to trace a program and understand what the state is after each step is a crucial skill in computer science, since it allows program developers to reason about their program before it is fully implemented and executed.

Country of Original Author

Japan

Secret Message

Story



Beavers send messages to each other using sequences of only 0s and 1s. In this system, each letter of the alphabet has its own code. Here are some letters and their codes:

A → 110	B → 111	E → 10	S → 0

The letters of a message are each replaced by their code and the result is 0101011111000.

Question

What was the message?

(A) SEEBASE

(B) SEEBASS

(C) SEEBESS

(D) SEEBEES

Answer

(B) SEEBASS

Explanation of Answer

Since only the code for S begins with 0, the first letter of the message must be S. Since 1 is not the code for any letter and only the code for E begins with 10, the second letter must be E. We can continue analysing the remaining 0s and 1s of the result from left to right. As soon as we see a subsequence of 1s and 0s that is the code for a letter, we know that must be the next letter of the message. The end result is this:

S	E	E	B	A	S	S
0	10	10	111	110	0	0

It is important to notice that this approach might not work for all sets of codes. For example, if 11 was the code for a fifth letter, say U, then we would not know if 110 came from the message US or the message A. In general, our solution works because there is not a code which begins with the code for another letter followed by more 0s and 1s.

Connections to Computer Science

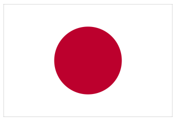
This task focusses on *encoding*. Encoding is the process of applying a code to data in order to represent the data in a different but equivalent way. Specifically, each letter is *encoded* as a sequence of 0's and 1's, and the focus of the task is to *decode* a particular sequence of 0's and 1's back into the original letters.

Some examples of converting letters to *binary encodings* (composed of 0's and 1's) are *Morse code*, which was used predominantly in the age of the telegraph; *ASCII*, which encodes simple text characters; and *Unicode*, which deals with characters in any language as well as emojis.

The particular code used in this task is a *prefix-free code*: no code for a letter is the prefix (or starting sequence) of any other code for a different letter. This allows the sequence to be decoded unambigously.

Country of Original Author

Japan

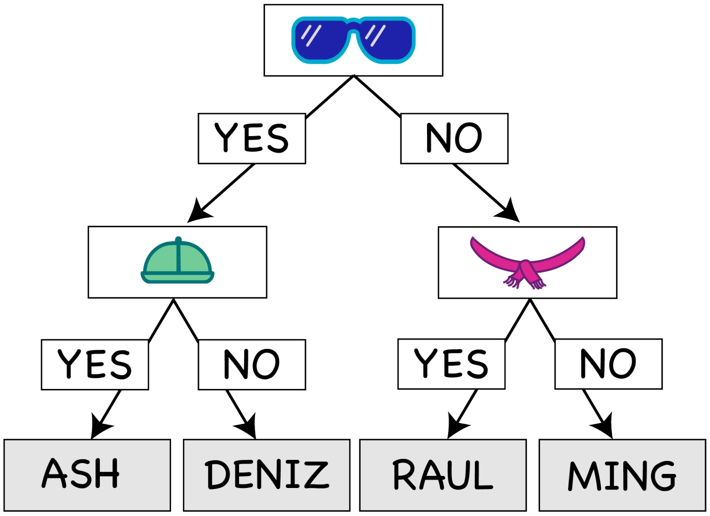


Remembering Faces

Story

Talia is very forgetful, so she has created a system to help her remember the names of her four group members.

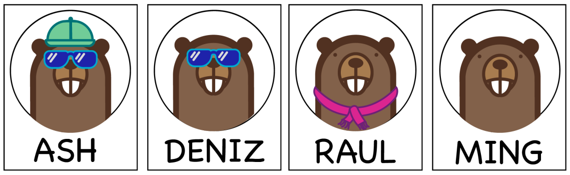
If a group member is wearing sunglasses, Talia checks to see if they are wearing a hat. If they are wearing a hat, then it is Ash, otherwise it is Deniz. If the group member is not wearing sunglasses, Talia checks to see if they are wearing a scarf. If they are wearing a scarf, then it is Raul, otherwise it is Ming.



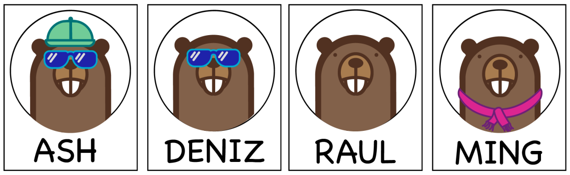
Question

Which of the following correctly matches names with faces?

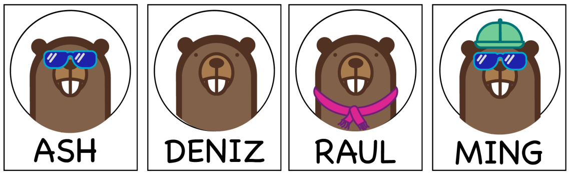
(A)



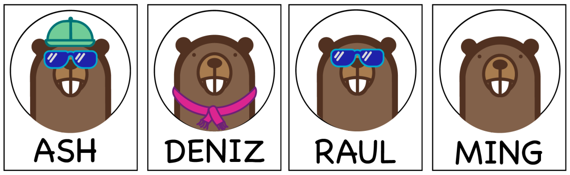
(B)



(C)

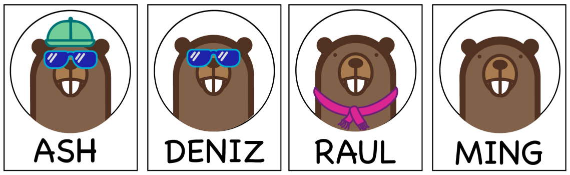


(D)



Answer

(A)



Explanation of Answer

Talia's system tells us that

- Ash is wearing sunglasses and a hat,
- Deniz is wearing sunglasses and is not wearing a hat,
- Raul is not wearing sunglasses and is wearing a scarf, and
- Ming is not wearing sunglasses and is not wearing a scarf.

These four things are true for Option A and not the other three options.

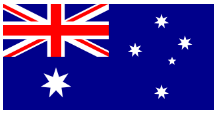
Connections to Computer Science

The picture used this problem is called a *decision tree*. A decision tree is composed of *nodes*. The *root node* is where we start the decision process. In this task, this root node is the sunglasses. Some nodes have a set of *binary* decision branches: these are the results of questions being answered either "yes" or "no". The other nodes are *leaf* nodes at the bottom of the tree and they represent the final *outcome* based on the decision path.

Decision trees are used in computer science in *artificial intelligence*, specifically in *machine learning*. For example, when a program is being designed to distinguish between various images, such as "dog", "fish", or "traffic light", various features such as "rectangular shape", "two eyes", "fins" are used as decisions. Each feature has a "yes" or "no" outcome, and with repeated *training*, the program will develop enough distinguishing features to correctly *classify* an image.

Country of Original Author

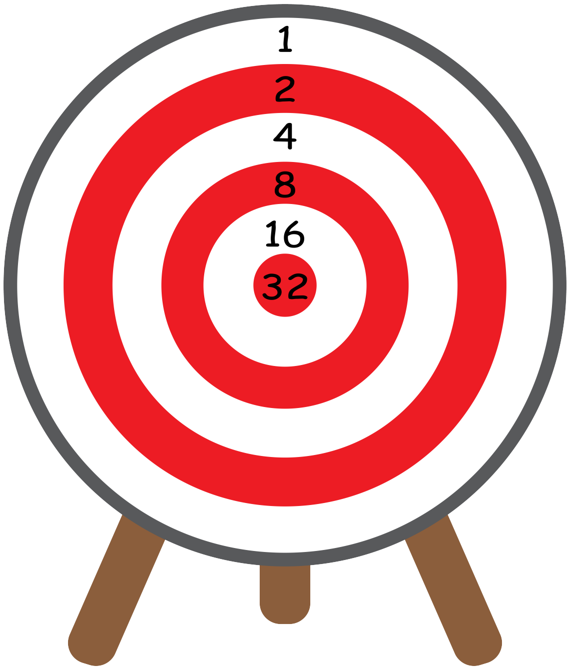
Australia



Target Practice

Story

Maryam is shooting arrows at the target shown. The number in a ring indicates how many points Maryam scores if an arrow hits that ring.



Maryam shoots three arrows and they all hit *different* rings.

Question

Which total score is **not** possible for Maryam to obtain?

(A) 56 points

(B) 21 points

(C) 10 points

(D) 7 points

Answer

(C) 10 points

Explanation of Answer

A total score of 56 points can be obtained by hitting the rings worth 8, 16, and 32 points.

A total score of 21 points can be obtained by hitting the rings worth 1, 4, and 16 points.

A total score of 7 points can be obtained by hitting the rings worth 1, 2, and 4 points.

To obtain a total score of 10 points, the rings worth 16 and 32 points are not helpful. That is, arrows can only hit rings worth 1, 2, 4 or 8 points. If the arrow does not hit the ring worth 8 points, then the score is $1 + 2 + 4 = 7$ which is not equal to 10. If the arrow does hit the ring worth 8 points, then the only possible scores are $1+2+8 = 11$, $1+4+8 = 13$, and $2+4+8 = 14$, none of which are equal to 10.

Therefore, it is not possible for Maryam to obtain a total score of 10 points.

Connections to Computer Science

We often think of a number as being the sum of multiples of powers of 10. That is, some number of 1s, plus some number of 10s, plus some number of 100s, etc. This is the *decimal number system* where the number 10 is the *base* of the system.

In this task, the point values of the rings double as you move towards the centre of the target: each point value is a power of 2. Maryam's total score is calculated as the sum of these powers of 2. When the number 2 is the base of a system, the system is known as the *binary number system.*

A number system based on the number 10 is typically easier for humans to work with (since we have 10 fingers or "digits"), but computers generally operate using the binary number system. This is because much of what a computer does is all determined by two possible states of electricity: the electrical signal is either "off" or "on", represented either as 0 or 1.

Country of Original Author

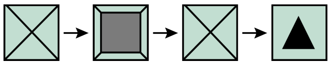
Canada

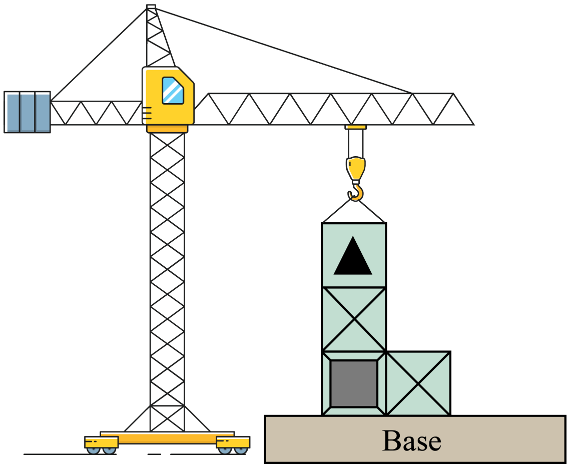
Part B

Building Instruction

Story

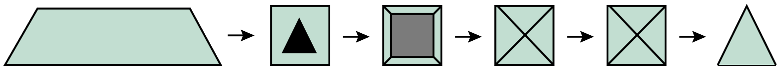
In Beaverland, houses are built out of different blocks. A crane takes blocks one by one in a given order. It puts each block either on the base or on top of another block.

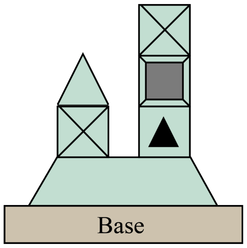
For example, using the block sequence , the following block house can be built.

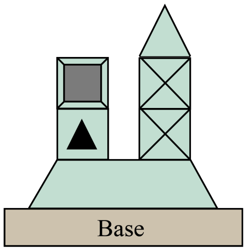


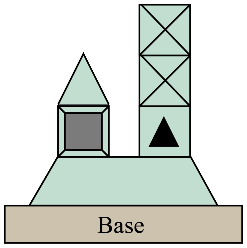
Question

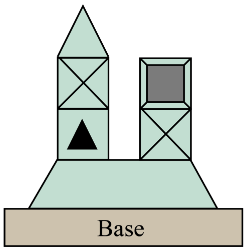
Which house **cannot** be built out of the following block sequence?



(A) 

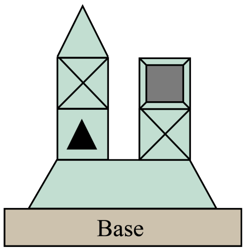
(B) 

(C) 

(D) 

Answer

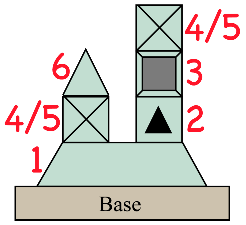
(D)



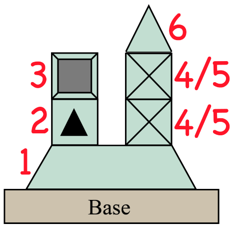
Explanation of Answer

The blocks in each house have been numbered from 1 to 6, based on their order in the given sequence.

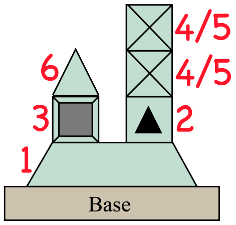
(A)



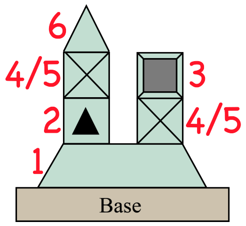
(B)



(C)



(D)



In Options A, B, and C, it is possible for the numbers to increase as you move upwards, indicating that the house could have been built out of the given block sequence.

In Option D, block number 3 is on top of block number 4 (or 5), which tells us it could not have been built out of the given block sequence.

Connections to Computer Science

This task combines together three ideas from computer science: *sequences of instructions*, *stacks*, and *state*.

This task focusses on a sequence of instructions, each instruction being the placement of one block. Any *algorithm* is a sequence of instructions, and algorithms are one of the key concepts in computer science.

This task also focusses on stacks. A stack is a data structure that uses the *last in, first out*, or *LIFO* strategy: the most recent item placed into the stack is the first item to be removed. In this task, each tower that is built is a stack, and blocks can only be placed on top of, but not underneath, existing blocks.

Finally, this task also focusses on state. The state is the placement position of the blocks. Knowing what the state is, and how it changes at each step of the algorithm is a crucial skill in *tracing* and *debugging* programs. For this task, tracing involves determining all possible locations for the current block to be placed, or starting from one of the five possible answers and tracing the removal of blocks in the reverse order that they were placed.

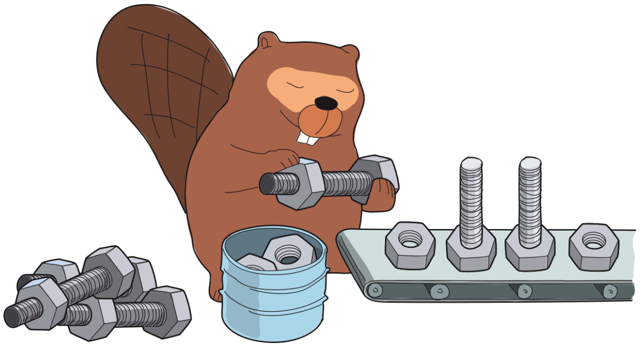
Country of Original Author

Vietnam

Nuts and Bolts

Story

At the Beaver Construction Factory, Lana works on the nuts and bolts assembly line.



Her job description is as follows:

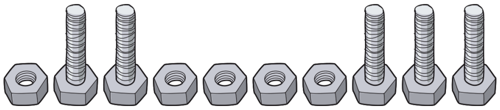
- Lana stands at one end of a long conveyor belt, which contains a line of nuts and bolts.
- Lana's job is to take each part, either a nut or a bolt, off of the conveyor belt.
- If Lana takes a nut from the conveyor belt, she puts it in the bucket beside her.
- If Lana takes a bolt from the conveyor belt, she takes a nut from the bucket beside her, attaches the nut and bolt together, and adds this to a pile of assembled parts.

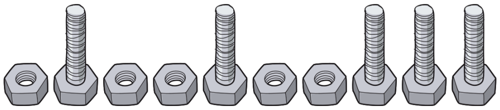
However, things can go wrong for Lana in two different ways:

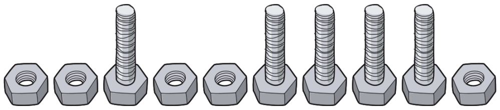
1. Lana takes a bolt from the conveyor belt, and there is no nut in the bucket to attach it to.
2. There are no more bolts on the conveyor belt, and there are still nuts in the bucket.

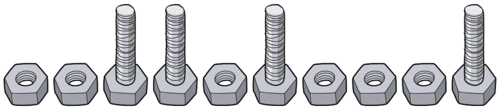
Question

Which sequence of nuts and bolts, when processed from left-to-right, will **not** cause things to go wrong for Lana?

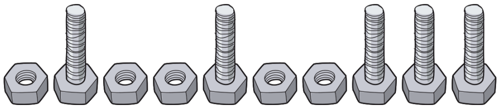
(A) 

(C) 

(B) 

(D) 

Answer

(C) 

Explanation of Answer

For Option C, we can check that things don't go wrong by keeping track of the state of the bucket and conveyor belt from left-to-right using N to represent a nut and B to represent a bolt.

Bucket	**Conveyor Belt**
empty	N B N N B N N B B B
N	B N N B N N B B B
empty	N N B N N B B B
N	N B N N B B B
N N	B N N B B B
N	N N B B B
N N	N B B B
N N N	B B B
N N	B B
N	B
empty	empty

Option A will go wrong after N B B, since there will be no nut in the bucket when the second bolt is encountered.

Option B will go wrong after N N B N N B B B B, since there will be no nut in the bucket when the fifth bolt is encountered. (Notice that there are only four nuts before this fifth bolt.)

Option D will go wrong after the entire sequence is processed, since there will be two nuts left in the bucket. (Notice that there are six nuts and four bolts in total.)

Connections to Computer Science

This task highlights the use of a *push-down automata* (PDA). A PDA is a way of describing a *algorithm* that relies on the current *state*, but also has an unlimited amount of memory in the form of a *stack*. In this task, the state is either having a nut or having a bolt on the conveyor belt, and the stack is the bucket which holds the nuts.

A PDA can be used to recognize, or *parse context-free languages*. To recognize or parse a language means to determine if a given sequence of symbols belongs to the language. In this case, we can think of the nuts and bolts as a representation of *balanced parentheses*, where N=(and B=). That is, balanced parentheses are valid arrangements of parentheses in arithmetic expressions. Examples of a sequence of parentheses which are not balanced are (((() or ())(. Detecting balanced parentheses is important in *compilers*, since many *programming languages* rely on parentheses to indicate *nested scopes* as well as arithmetic expressions.

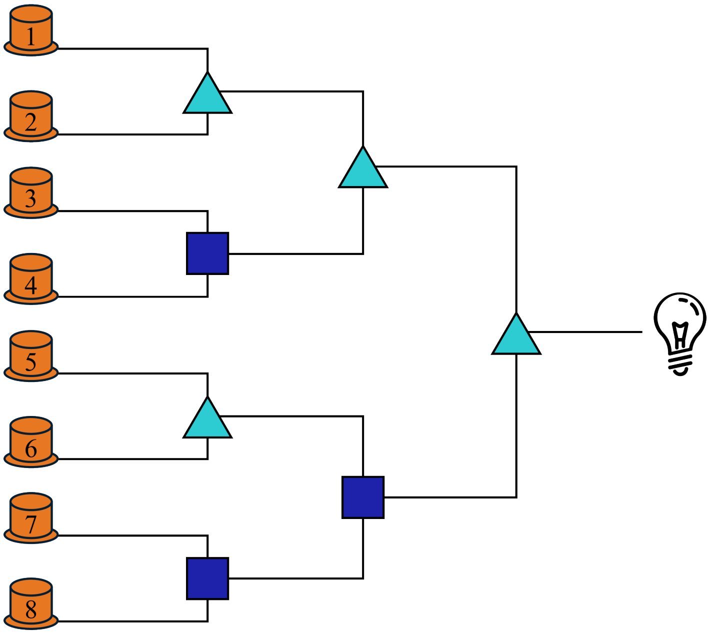
Country of Original Author

Canada

Lights On

Story

Beaver Sofia and her friends are playing an arcade game called “Lights On”. The game has 8 buttons for the beavers to stand on. Standing on a button will send a signal down a wire. These wires pass through some triangle or square boxes and eventually lead to a light bulb, as shown.



A triangle box will send on a signal if it receives signals from *both* incoming wires.

A square box will send on a signal if it receives a signal from *exactly one* of the incoming wires (in other words, from one of the incoming wires but not the other incoming wire).

The beavers win the game if they can turn the light on.

Question

Which of the following combinations of buttons could the beavers stand on in order to win the game?

(A) 2, 3, 4, and 8

(B) 1, 2, 5, and 6

(C) 1, 2, 3, 5, 6, and 7

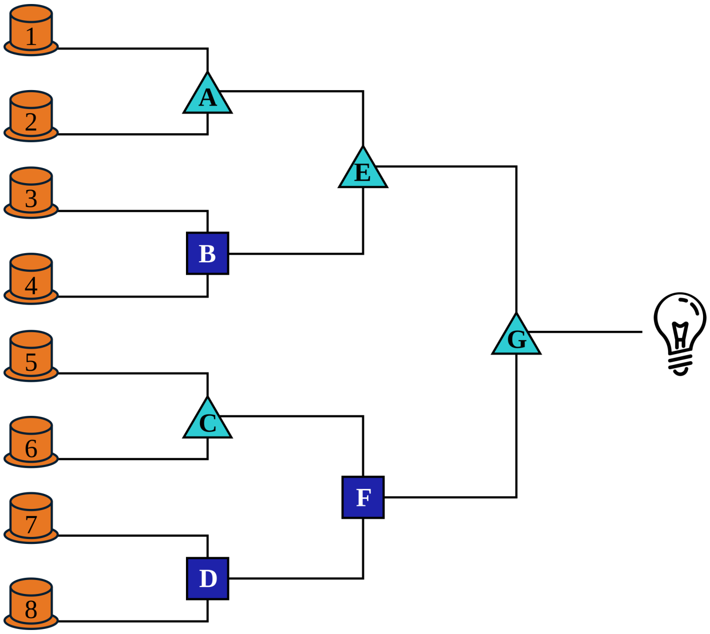
(D) 1, 2, 4, and 8

Answer

(D) 1, 2, 4, and 8

Explanation of Answer

We will label the triangle and square boxes as shown.



A good approach to this question is to work backwards. The light bulb is connected to the wire coming from triangle G. In order for the light bulb to be on, triangle G must receive a signal from both incoming wires. These wires are connected to triangle E and square F.

In order for triangle E to send on a signal, it must receive a signal from both incoming wires. Therefore, triangle A and square B must both send on a signal. In order for this to happen, both buttons 1 and 2 must be pressed, and exactly one of buttons 3 and 4 must be pressed. This eliminates Option A because button 1 is not pressed. It also eliminates Option B because neither button 3 nor 4 is pressed.

In order for square F to send on a signal, it must receive a signal from exactly one of the incoming wires. Therefore, either triangle C sends on a signal, or square D sends on a signal, but not both. If triangle C sends on a signal, then both buttons 5 and 6 must be pressed. If square D sends on a signal, then exactly one of buttons 7 or 8 must be pressed. This eliminates Option C because both triangle C and square D would send on signals.

Option D satisfies these constraints, so is the only option that will turn the light on.

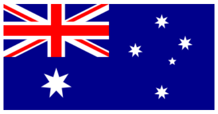
Connections to Computer Science

In this task, each button can either be ON or OFF, which computer scientists would indicate with the *binary digits* 1 or 0, or alternatively, by the *boolean values* “true” or “false”.

Computer circuits, including those used in the hardware components such as the *central processing unit (CPU)*, use *logical gates* to alter boolean values in order to do computation. The triangles used in this task are examples of *AND gates*: so long as both input values are true, then the output value will be true; otherwise, the output will be false. The squares used in this task are examples of *XOR gates*, meaning “exclusive-or”. If exactly one input value to an XOR gate is true, then the output will be true; otherwise, the output will be false.

Country of Original Author

Australia



Embroidery Machine

Story

Benoit has a machine which can embroider two kinds of stitches: + and ×. The machine can also move the fabric by the width of one stitch so that stitches are embroidered from left to right.

If the two kinds of stitches are embroidered (in either order) without moving the fabric, the result is: ✳.

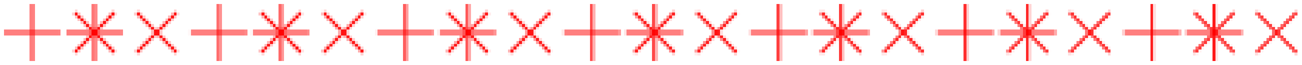
The machine can be programmed using the characters +, x and `>` in some sequence.

- `+` means to embroider +,
- `x` means to embroider ×, and
- `>` means to move the fabric by the width of one stitch.

The machine repeats the entered program some number of times. For example, the program

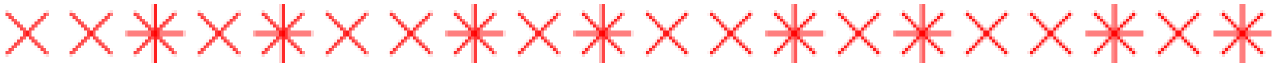
```
+ > + x > x >
```

can cause the machine to embroider:



Question

What program can cause the machine to embroider the following?



(A) `x > x > x + > x > + x > x > x >`

(B) `x > x > x + > x > x >`

(C) `x > x > x + > x > + x >`

(D) `+ x > + x > x > + x > x >`

Answer

(C) `x > x > x + > x > + x >`

Explanation of Answer

The correct answer can be found by noticing that ✕ ✕ ✳ ✕ ✳ is a repeating pattern.

Option C causes the machine to move the fabric five times which is correct for this pattern. (It should not move before the first stitch but it has to move after the last stitch so that it is in the correct place when the pattern repeats.) The table below shows how the program in Option C also causes the machine to correctly embroider this pattern between movements.

Code	Stitch Embroidered
`x`	✕
`x`	✕
`x +`	✳
`x`	✕
`+ x`	✳

We can check that Option A is a program that causes the machine to embroider the pattern

✕ ✕ ✳ ✕ ✳ ✕ ✕

which is incorrect because if this pattern is repeated, it will result in four ✕ stitches in a row.

Option B corresponds to a pattern with this same problem:

✕ ✕ ✳ ✕ ✕.

Option D is incorrect because it causes the program to begin by embroidering two ✳ in a row.

Connections to Computer Science

This task illustrates one way to write an *algorithm*. An algorithm is a sequence of instructions that must be followed in a *sequential order*. Changing the order of the sequence may cause the algorithm to perform very differently: for example, switching the placement of the > would change the entire pattern.

This task is also connected to one historical foundation of computer science: the *Jacquard machine*. The Jacquard machine was used to control a weaving loom, so that fabrics with various patterns could be woven in a specified way. The Jacquard machine was controlled by a sequence of wooden *punched cards*, which *encoded* the pattern of each row of the design. These punched cards formed the basis for how Charles Babbage described his *Analytical Engine*, and how large *mainframe* computers developed in the 1950s used paper punched cards to describe algorithms.

Country of Original Author

Slovakia

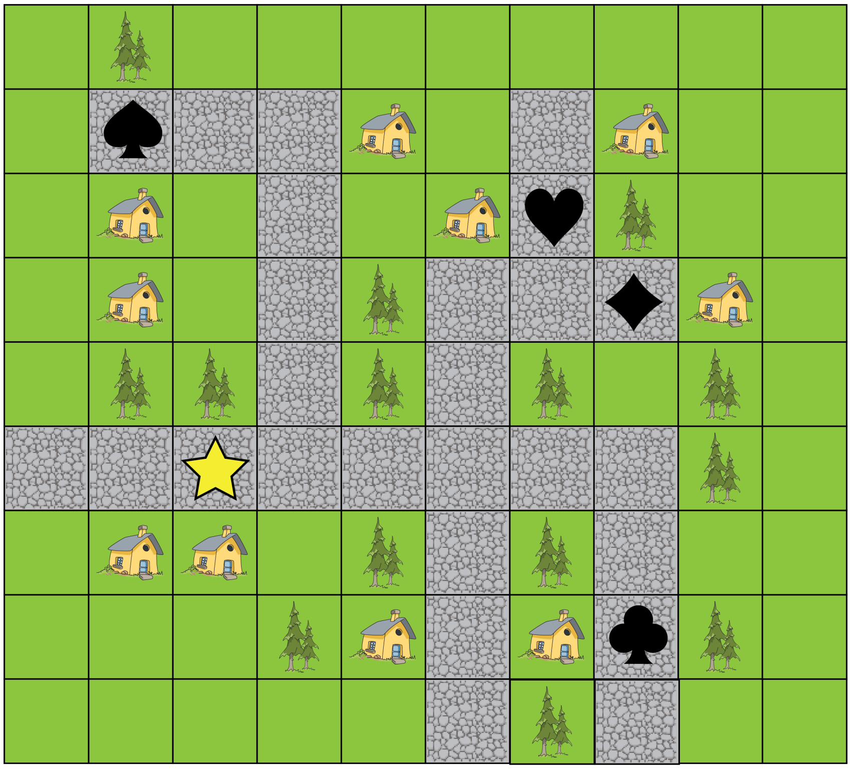
Talking Tiles

Story

Terry lives in a terrific town that is built on top of talking tiles. Each tile contains either a house, trees, grass, or stone.

Terry is visually impaired and so he uses the talking tiles to help him navigate. When he steps onto a tile, the talking tile tells him what is found in the four squares surrounding him in the following order: north (↑), east (→), south (↓), and west (←).

Below is a map of the town. Terry is currently standing on the stone tile marked with a star. When he stepped onto this tile, the talking tile said “trees stone house stone”.



Terry decides to go for a walk around town and hears the following as he walks:

- “stone stone grass stone”
- “trees stone trees stone”
- “stone stone stone stone”
- “stone trees stone trees”
- “house stone stone trees”
- “stone stone trees stone”
- “stone trees stone house”

Terry only walks north, east, south and west.

Question

Where is Terry now?

(A) ♥ (B) ♦ (C) ♣ (D) ♠

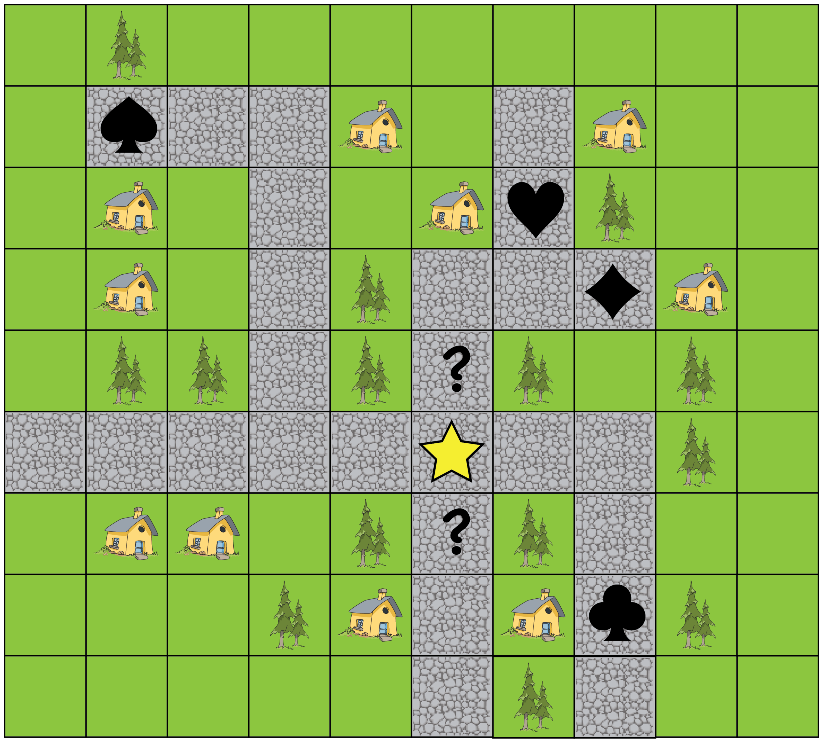
Answer

(A) ♥

Explanation of Answer

Since the first thing Terry hears after he begins his walk around town is “stone stone grass stone” we know that he must have stepped east. Since he next hears “trees stone trees stone” he must have stepped east again. His third step is also east since he next hears “stone stone stone stone”.

The fourth thing Terry hears is “stone trees stone trees”. He could have stepped either north or south.



He next hears “house stone stone trees”. If he had previously stepped south, there would be no adjacent tile he could move to in order to hear this. So he must have previously stepped north, and then moved to the adjacent tile to the north.

The second last thing Terry hears is “stone stone trees stone”. This means he must have stepped east. He then must have stepped north in order to hear “stone trees stone house”. Therefore, Terry is now on the tile marked with a ♥.

Connections to Computer Science

This task highlights how a *log* of transactions can be used to *recover* information to determine what the *state* of a system should be in.

We can think of a log as a list of important information that was true for a system at various points in time. In this task, the log is composed of the description of each tile, based on what tile Terry is currently occupying. The *state* is the location of Terry. The initial state is given by the star, and based on each log entry, it is possible to determine the next location or state.

Many computer systems use this methodology: both general *file systems* and *database systems* use logs to either recover the current state if there is a failure, or to possiblly *rollback* transactions to get to an earlier state.

Country of Original Author

Germany

Part C

Signal Lamps

Story

At a small seaside cafe, Bombom takes food orders and Lala does all the cooking. To send food orders to Lala, Bombom uses two signal lamps; one shaped like a leaf, and the other shaped like a flower. At first the cafe sold only coffee and tea and they assigned each item the following codes.

Item	**Code**
coffee	🍃 🍃
tea	🍃 🍃 ✿

This caused a problem, however, because Lala starts to prepare each drink as soon as she receives a complete code, instead of waiting to see if Bombom is finished. So whenever Bombom turns on the leaf lamp twice, Lala does not wait to see if another signal is sent. Instead, she starts to prepare a coffee, since the complete code for coffee (two leaves) has been sent.

They expanded their menu and assigned new codes to avoid any problems like this. The new menu and codes are as follows.

Item	**Code**
coffee	🍃 🍃
tea	🍃 ✿ 🍃 ✿
juice	✿ 🍃
sandwich	🍃 ✿ ✿
muffin	✿ ✿ ✿

Question

If they decide to add pie to the menu, which new code can be used without causing any problems?

(A) ✿ ✿ 🍃

(C) 🍃 ✿ 🍃

(B) 🍃 🍃 ✿

(D) ✿ ✿ ✿ 🍃

Answer

(A) ✿ ✿ 🍃

Explanation of Answer

Option A can be used because it does not begin with a complete code for another item, nor is it the beginning of the code for another item.

The other three Options will all cause a problem.

Option B will cause a problem because the first two symbols are the code for coffee. So Lala will interpret all pie orders as coffee orders.

Option C will cause a problem because the code is the first three symbols in the code for tea. So Lala will interpret all tea orders as pie orders.

Option D will cause a problem because the first three symbols are the code for muffin. So Lala will interpret all pie orders as muffin orders.

Connections to Computer Science

When information is transmitted between computers, either through wires or through the air wirelessly, the information is sent as sequences of *signals*. Each signal can be viewed as describing one of two possibilities: either *on* or *off*. In this problem, each menu item corresponds to a particular *binary sequence*, where instead of sending on/off signals, the leaf and flower signals are used.

The specific way by which information is translated into signals is called a *binary code*. In this problem, a *variable length code* was used: the code length of a menu item could be 2, 3, or 4 *bits*. Most computers use *fixed length codes* to encode information: for example, the most common encoding of textual information is (or is based on) *ASCII*, where there are 8 signals for each different character.

It is crucial that the receiver of a coded message can translate the signals back to the original message without making mistakes. Mistakes could include the translation does not match the original message or the translation could be interpreted in two or more ways. In other words, codes must be designed carefully, and ensure that they are *uniquely decodable codes.*

A special kind of uniquely decodable code is the *prefix-free code*. These codes have the property that no code is the prefix (or starting sequence) of any other code. The expanded menu which has all items is a prefix-free code, and adding pie to that menu must ensure that the the code for pie is not a prefix of the code for any existing menu item, nor is any existing menu item a prefix of the code for pie. If that condition is not met, then a particular sequence could be decoded in multiple ways, and thus the wrong menu items would be prepared.

Prefix-free codes tend to be quite short and they are easily decoded. They are not only used for communication purposes but also are used in several *compression algorithms*.

Country of Original Author

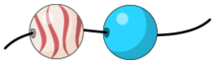
Indonesia

Beach Necklaces

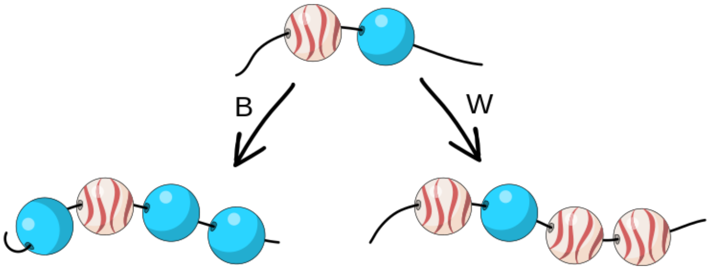
Story

Bashir makes necklaces using wavy beads and blue beads. He always makes them as follows.

1. Place one wavy bead and one blue bead on a string with the wavy bead to the left of the blue bead.



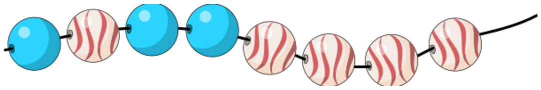
2. Do one of the following two actions.
   - *Action B*: Add a blue bead to both ends of the string.
   - *Action W*: Add two wavy beads to the rightmost end of the string.

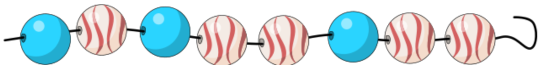


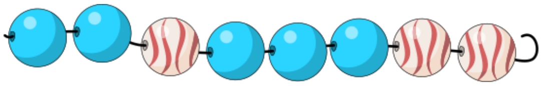
3. Repeat step 2 until the necklace is complete.

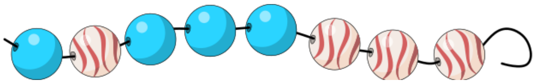
Question

Which necklace below **cannot** be made by Bashir?

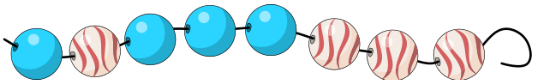
(A) 

(C) 

(B) 

(D) 

Answer

(D) 

Explanation of Answer

Each necklace begins with one wavy bead and one blue bead to its right.

The necklace in Option A can be made by taking *Action B*, then *Action W* and then *Action W* again.

The necklace in Option B can be made by taking *Action B*, then *Action B* again and then *Action W*.

The necklace in Option C can be made by taking *Action W*, then *Action B* and then *Action W*.

Bashir cannot have made the necklace in Option D. One way to see this is to notice that there is only one place on that necklace with a wavy bead to the left of a blue bead. If Bashir had made the necklace in Option D, this must have been where he started. These two beads are surrounded by blue beads which means the only possible first action taken by Bashir is *Action B*. At that point, *Action B* would produce another blue bead on the left side of the necklace and *Action W* would produce a wavy bead where there is a blue bead. Either way, Bashir cannot make the necklace in Option D.

Another clever way to rule out Option D is to notice that Bashir always starts with one blue bead and when he adds more blue beads, he always does so two at a time. This means that there must be an odd number of blue beads. For the same reason, there must always been an odd number of wavy beads. In Option D, there are four blue beads and four wavy beads. Since four is an even number, this is impossible.

Connections to Computer Science

In this task, beads could only be added or removed to the ends of the necklace string. It was not possible to add or remove a bead to the middle directly.

The necklace is similar to a *data structure* used in computer science called a *double-ended queue* or *deque*.

One common use of a deque is to store web browser's history: there are "back" and "forward" buttons in most web browsers that behave in a similar way to the ends of the necklace. Other applications of deques include scheduling print jobs and verifying the validity of math expressions such as $((1 + 2) * (3 + 4))$. Checking for matching parentheses can be done in a way that is quite similar to how the validity of the necklaces was verified.

Country of Original Author

Slovakia

Beautiful Gems

Story

Troy has a collection of gems. He ranks his gems from most beautiful to least beautiful.

Sarah knows what gems are in Troy's collection, but she does not know how he has ranked them.

Sarah has a strategy to find out which gem is most beautiful according to Troy:

- Sarah chooses four of Troy's gems and asks Troy: "Out of this group of four, which gem is the most beautiful?"
- Sarah chooses a new set of four gems and asks her question again.
- Then she chooses a third set of four gems and asks her question for the last time.

When Sarah chooses her second and third set of four gems, she may sometimes include gems she has chosen before. Troy answers each question honestly and Sarah remembers all of Troy's answers.

Sarah's strategy allows her to find the most beautiful gem according to Troy, regardless of his ranking.

Question

What is the largest possible number of gems in Troy's collection?

(A) 8

(B) 10

(C) 11

(D) 12

Answer

(B) 10

Explanation of Answer

The gem that Troy identifies as the most beautiful in a group of four could be most beautiful in the collection overall. If Sarah selects different gems in her first two questions, then she has two possible candidates for the most beautiful gem overall. With 10 gems, if Sarah's third group of four gems includes the top picks from the first two groups and the remaining two gems that have not been selected yet, one of these four gems must be the most beautiful in the whole collection.

We have shown that there is a strategy of Sarah's that works if there are 10 gems. (Note that there are other correct strategies that work for 10 gems also.)

If Troy has at least 11 gems, we consider Sarah's first two questions.

- If at least one gem is part of both questions, then at least four gems will be unconsidered after the first two questions. In this case, Sarah must ask about these four unconsidered gems because if she doesn't, an unconsidered gem could be the most beautiful gem according to Troy. On the other hand, she won't have any information about the most beautiful gem according to Troy among the last four unconsidered gems compared to the other 7 gems. In this case, her strategy cannot determine which gem is the most beautiful according to Troy.
- If there is no gem that is part of both questions, then candidates for the most beautiful gem according to Troy include the two answers to these questions and the (at least) three remaining unconsidered gems. This is a total of (at least) five possibilities for the most beautiful gem according to Troy and Sarah cannot obtain enough information with only one question remaining.

We have shown that there is not a strategy for Sarah that works if there are more than 10 gems.

Connections to Computer Science

To answer this task, we need to find the best strategy that will always work for Sarah. The strategy can be described as an *algorithm*, which is a sequence of steps that solves the problem. Not only do we need to find an algorithm, but need to find an *optimal algorithm* that solves the problem.

When writing an algorithm, programmers would like the algorithm to be both *efficient* and *optimal*. An efficient algorithm is one that uses the least amount of time to find a solution. An optimal algorithm produces the best possible output value. This task involves finding both an efficient solution, since Sarah only has three sets of gems that she can pick, and an optimal solution, since Sarah must find the most beautiful gem, and not just any of the top 3 beautiful gems, for instance.

The *route finding problem*, which is concerned with finding the shortest route between two cities, given roads that connect various cities, has many possible algorithms. One inefficent but optimal algorithm is to try every possible ordering of the roads to take. This algorithm will take an extremely long time when there are thousands of cities, but it will always given an optimal solution. There is also an algorithm that is efficient but not optimal (which is referred to as *suboptimal*), which is to just pick the shortest road when moving from city to city. This algorithm will find an answer quickly, but the answer may not be the shortest possible route. Finally, there are efficient and optimal solutions such as *breadth-first search*.

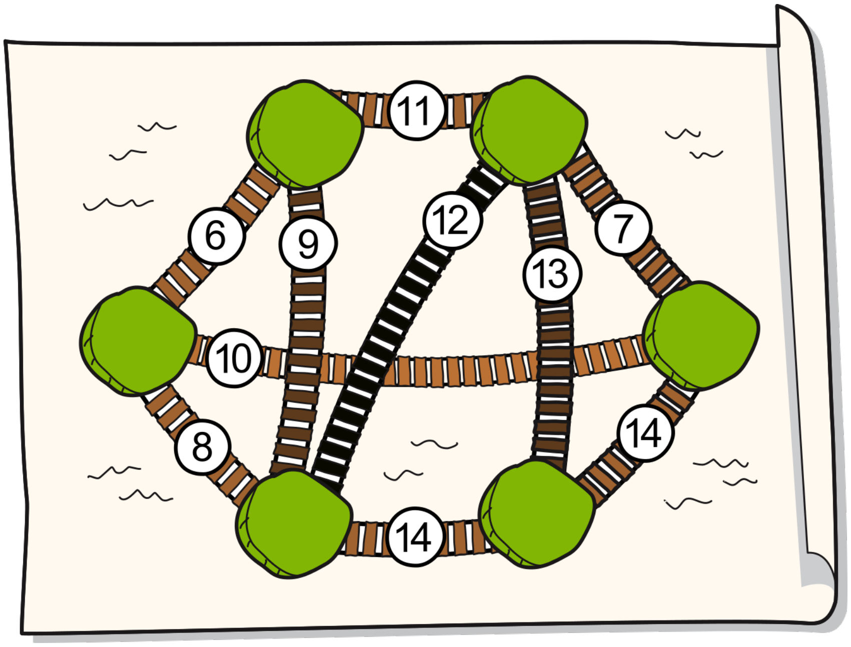
Country of Original Author

Canada

Connected Islands

Story

A community spread across six islands must connect all the islands together by building bridges. A map of all the possible bridges and the cost of building each bridge is shown.



The community needs it to be possible to travel from any island to any other island either directly or indirectly.

Note that when you are on a bridge, you can only leave the bridge where it ends at an island. For example, you cannot hop between the bridges that cost 10 and 13 even though they overlap.

Question

What is the lowest possible total amount that the community can spend on bridges?

(A) 46

(B) 45

(C) 44

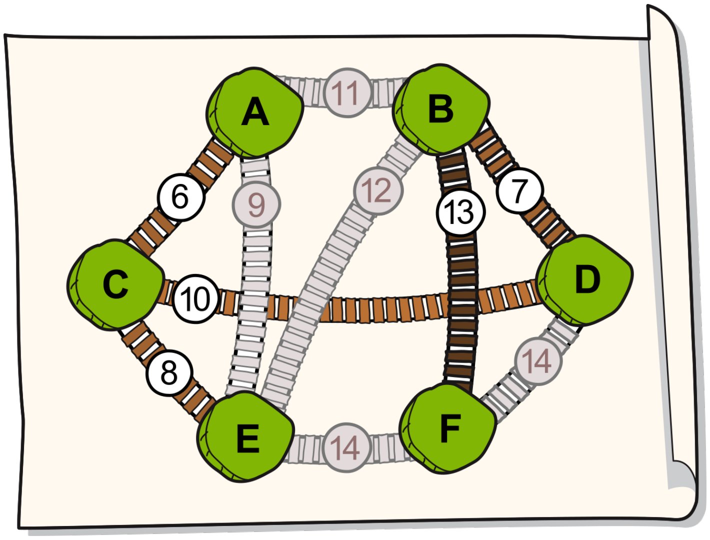
(D) 40

Answer

(C) 44

Explanation of Answer

The image shows five bridges that the community can build. The total cost is $6+7+8+10+13 = 44$. We can check that it would be possible to use only these bridges to travel from any island to any other island either directly or indirectly.



We should convince ourselves that the community must spend a total amount of at least 44. The islands have been labelled to help us do this.

First, we try to understand why it is not possible to build fewer than five bridges.

Note that there must be at least one bridge from each island. Therefore, if there were only three bridges, there would be three pairs of islands with no way to travel between islands in different pairs (e.g. if bridges were built between islands A and C, islands B and D, and islands E and F).

Even if we had one more bridge for a total of four, it would not be possible to use these bridges to travel from any island to any other island. Specifically, with only four bridges, we would be left with either a disconnected pair of islands (e.g. if bridges were built between islands A and C, islands B and D, islands D and F, and islands E and F), or there would be two groups of three islands with no way to travel between the groups (e.g. if bridges were built between islands A and C, islands C and E, islands B and D, and islands D and F). You might want to draw some pictures to convince yourself of these conclusions.

We must use a bridge of cost 13 or 14 because no cheaper bridge can be used to connect island F. Adding in the four cheapest bridges, the total cost must then be at least $6+7+8+9+13 = 43$, but this choice of bridges would leave islands A, C and E disconnected from islands B, D and F. Therefore, the community must spend a total amount of more than 43. That is, the community must spend a total amount that is at least 44.

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 potential bridges that could be built. The diagram is a visual representation of the graph.

The goal of this task keep as few edges (bridges) 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.

One algorithm to find the minimum spanning tree in a graph is *Kruskal's algorithm*, which repeatedly selects the lowest-cost edge so long as there is no *cycle* (a sequence of edges that connect a vertex to itself) introduced.

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

Country of Original Author

Lithuania

Mysteria

Story

The inhabitants of Mysteria can be wizards, fairies, potions, or dragons. They always arrange themselves in a row and can undergo four types of magical transformations:

1. A wizard can turn into a fairy

   🧙 → 🧚

2. A wizard can turn into a wizard (on the left) and a fairy beside them (on the right)

   🧙 → 🧙 🧚

3. A fairy can turn into a potion (on the left) and a dragon (on the right)

   🧚 → 🧪 🐉

4. A fairy can turn into a wizard, with a potion beside them (on the left) and a dragon beside them (on the right)

   🧚 → 🧪 🧙 🐉

These magical transformations can happen any number of times, in any order. That is, any wizard and any fairy in Mysteria can transform at any time.

Question

Starting with the single wizard, which state of Mysteria is **not** possible to obtain?

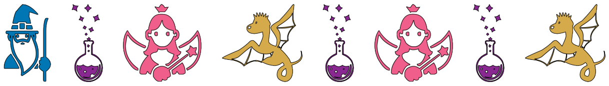
(A) 🧪 🧙 🧪 🐉 🐉

(B) 🧙 🧪 🧚 🐉 🧪 🧚 🧪 🐉

(C) 🧚 🧪 🐉 🧪 🧙 🐉

(D) 🧪 🐉 🧪 🐉

Answer

(B) 

Explanation of Answer

Option A can be obtained by applying transformations 1, 4, 2, and 3 in that order.

Option C can be obtained by applying transformations 2, 2, 3 (to the leftmost fairy), 4, and 1 (to the leftmost wizard) in that order.

Option D can be obtained by applying transformations 2, 1, 3 (to the leftmost fairy), and 3 in that order.

One quick way to see that Option B is not possible is to notice that the transformation rules always create a potion and a dragon at the same time. Therefore, the number of potions in Mysteria will always equal the number of dragons, which is not the case in Option B.

Connections to Computer Science

The magical transformations can be thought of as a set of rules used to generate sequences of objects in Mysteria.

In computer science, a *context-free grammar* is one tool that can be used to describe rules that generate patterns. Context-free grammars can describe languages (both formal and natural). By repeatedly applying the rules of the grammar, *words* (or *strings*) that form the language are created.

Context-free grammars are used for verifying the correctness of the structure, or *syntax* of computer programs. For example, a programming language may allow statements to be *nested* within other statements, such as *if-then statements* inside of *loops* which are within another if-then statements. Context-free grammars are used for *parsing* of programs, which is the process of detecting *syntax errors*. In this task, you were asked to determine which of the given words was not part of the Mysteria language, which is equivalent to finding which words had syntax errors.

Country of Original Author

Canada