



UNIVERSITY OF  
**WATERLOO**



The CENTRE for EDUCATION in  
MATHEMATICS and COMPUTING



2017  
*Beaver  
Computing  
Challenge  
(Grade 5 & 6)*

*Questions,  
Answers,  
Explanations,  
and  
Connections*

# Part A

# Bird House

## Story

A parent takes their child to the store to buy a bird house. The child says “I would like a bird house with two windows and a heart decoration.”



HOUSE 1



HOUSE 2



HOUSE 3



HOUSE 4

## Question

Which of the bird houses matches this description?

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

### Answer

(C) House 3

### Explanation of Answer

Answer A is not correct because House 1 has only one window and 2 hearts.

Answer B is not correct because House 2 has two windows, but no heart.

Answer D is not correct because House 4 has only one window.

### Connections to Computer Science

*Abstraction* is the process of extracting the most important or defining features from a problem. These extracted features provide the key information about the problem, and allow the problem to be viewed only based on those features, rather than other features. This can help by highlighting the similarities and differences in aspects of a problem. This task focuses on identifying the objects with certain features (such as the number of windows or type of decoration) and ignoring other features (such as colour).

### Country of Original Author

Romania



## Risk

### Story

Darren's computer is connected to the Internet but does not have any antivirus or firewall software. None of the accounts on his computer are protected by a password.



### Question

Which computers are at risk because of this?

- (A) only Darren's own computer
- (B) only the computers in the same room as Darren's computer
- (C) only the computers in the same country as Darren
- (D) all computers in the world which are connected to the Internet and set up like Darren's

### Answer

(D) all computers in the world which are connected to the Internet and set up like Darren's

### Explanation of Answer

Due to Darren's reckless behaviour, his own computer may get infected by malicious software. However, infected computers are often used as a platform for attacking other computers. This means that all computers connected to the Internet without proper protection, such as firewalls and antivirus software are at risk because of Darren's actions.

### Connections to Computer Science

Owners of other computers can reduce some of the risks by following good practices. For example, installing *security updates* and running *antivirus software* reduces the risk of malware spreading to other computers. Unfortunately, this does not eliminate all risks. *Malware*, which is software designed to do malicious things, if it is on Darren's computer, may be used to send spam or overload network connections (these are called *DDoS attacks*), and there is very little other computer users can do to protect themselves against these larger scale attacks.

*Cybersecurity* (or *Information Technology security*) helps protect computer systems from theft or damage to the hardware, software or the information on them, as well as from disruption or misdirection of the services they provide.

### Country of Original Author

Estonia

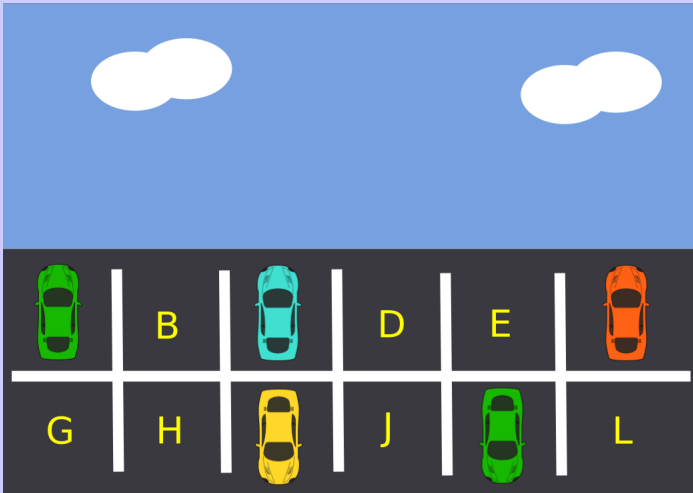


# Parking Lot

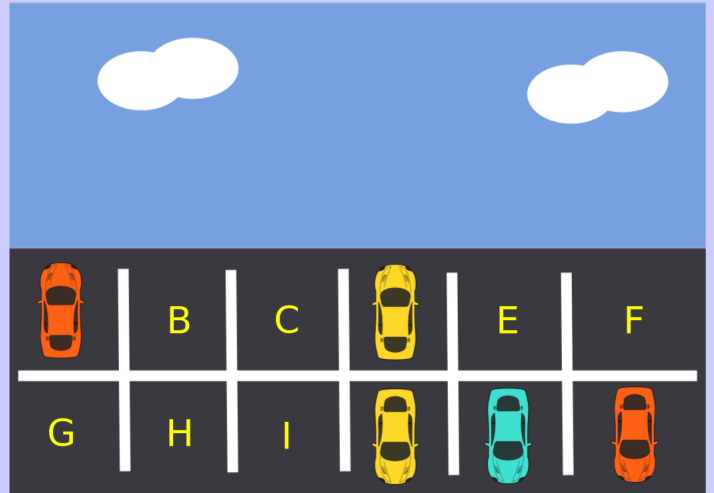
## Story

There are 12 spaces for cars in a parking lot. The pictures below show which spaces were used on Monday and which spaces were used on Tuesday.

Parking lot on Monday



Parking lot on Tuesday



## Question

How many parking spaces were empty on both Monday and Tuesday?

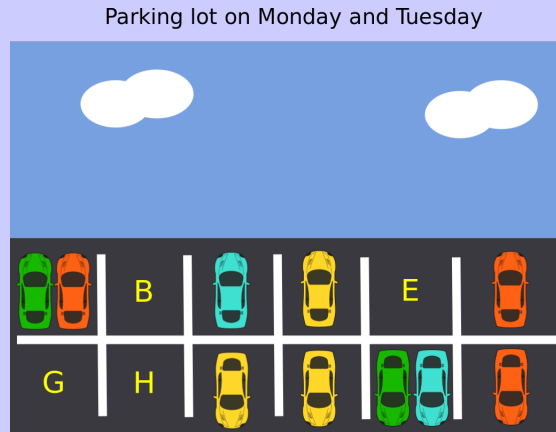
- (A) 3
- (B) 4
- (C) 5
- (D) 6

### Answer

(B) 4

### Explanation of Answer

We can see which spaces were used by placing the cars from both days on the parking lot at the same time.



Then we can simply count empty spaces to determine that four spaces were empty on both Monday and Tuesday.

### Connections to Computer Science

All *data* can be thought of as a sequence of zeros and ones. Each zero or one is called a *bit* and the sequence is called a *binary code*, *binary representation*, or *binary number*.

In this task, we can model an empty space as 0 and a space with a car as 1; each parking space corresponds to a bit. We get a sequence of bits by viewing the parking spaces in order. For example, we can read the top row from left to right and then the bottom row from left to right to get 101001001010 from the parking lot on Monday and 100100000111 from the parking lot on Tuesday. This task asks to determine which of the twelve positions contain a 1 in either of these binary numbers. This is a logical operation named OR. Notice how we can compute the correct answer by seeing that

$$\begin{array}{r} 101001001010 \\ \text{OR } 100100000111 \\ \hline 101101001111 \end{array}$$

This resulting binary number has four zeros in it.

### Country of Original Author

Canada

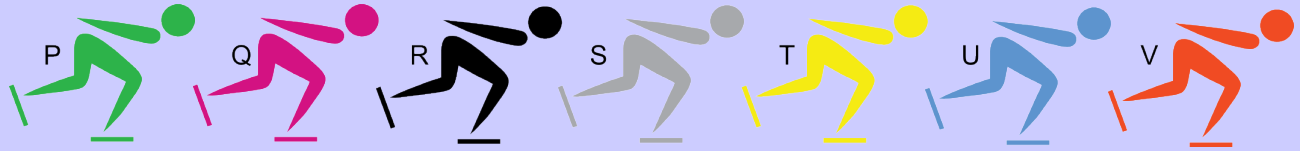




# Skaters

## Story

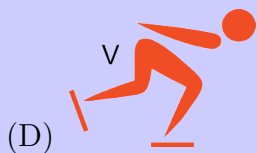
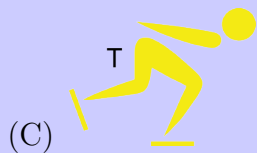
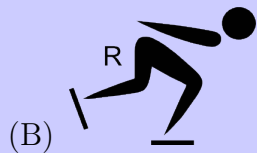
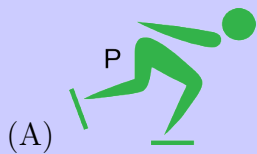
Seven people are skating in a line on a very long, frozen canal. They begin as shown below.



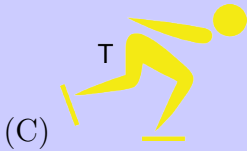
After every minute the person at the front of the line moves to the end of the line. For example, after one minute, U will be in front of the line, since V will move behind P.

## Question

Which skater will be at the front of the line after 9 minutes?



## Answer



## Explanation of Answer

We can simply trace out the positions after each minute:

- after 1 minute, U is in front
- after 2 minutes, T is in front
- after 3 minutes, S is in front
- after 4 minutes, R is in front
- after 5 minutes, Q is in front
- after 6 minutes, P is in front
- after 7 minutes, V is in front (again)
- after 8 minutes, U is in front (again)
- after 9 minutes, T is in front (again)

Thus, after 9 minutes, T is in front.

## Connections to Computer Science

This task is focussed on *tracing an algorithm* and remembering the *state* or *configuration* of all information. Specifically, we start in a given configuration, follow a sequence of (nine, in this task) steps, and we must determine the final configuration. This process of tracing is used by computer scientists to ensure that that computer programs perform the intended process, and if they do not, to find the location or step where the program is not correct.

## Country of Original Author

Canada

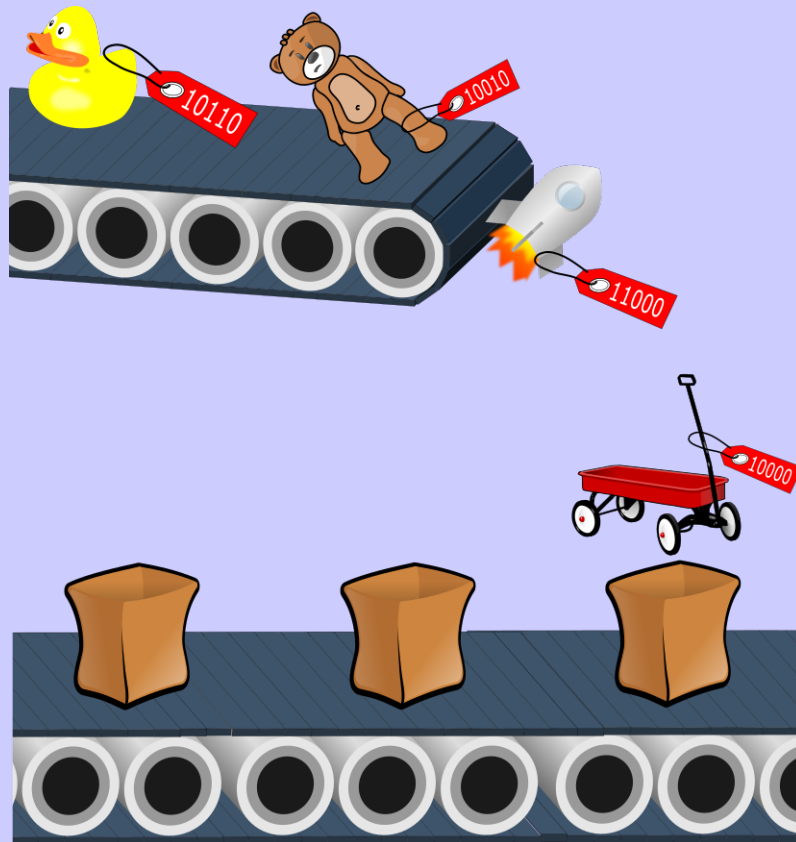


## Part B

# Toy Factory

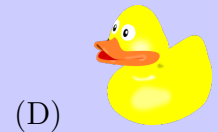
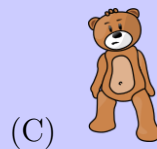
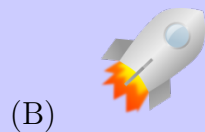
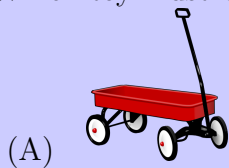
## Story

Toys fall from a high conveyor belt into bags on a low conveyor belt. The toys should fall into the bags with their numerical product codes in increasing order. One of the toys is in the wrong place and needs to be removed so that the remaining toys are in the right order.



## Question

Which toy must be removed?



### Answer

(B)



### Explanation of Answer

The toys come out of the machine with bar codes: 10000, 11000, 10010, 10110

Removing the 10000 or the 10110 does not leave the remaining three in increasing order since  $11000 > 10010$ . Removing the 10010 does not leave the remaining toys in increasing order since  $11000 > 10110$ . Thus, the only possibility is removing the toy corresponding to 11000 (the rocket) and this choice yields three toys in increasing order since  $10000 < 10010$  and  $10010 < 10110$ .

### Connections to Computer Science

*Ordering* objects is very important in computer science. In this task, the toys are required to be in the correct order as they enter the bag otherwise they will be shipped to the incorrect location. When instructions in computer science are out of order they can cause faulty behaviour and even render the program useless, invalid or incorrect.

The integers above are also all written using zeros and ones. Numbers written using only these two digits are called *binary numbers*. These numbers are the kinds of numbers that a computer uses to store information. A computer interprets these binary numbers differently than we interpret normal decimal numbers. However, binary numbers still respect the same ordering properties (such as  $101 < 110$  being true) that our decimal numbers follow.

### Country of Original Author

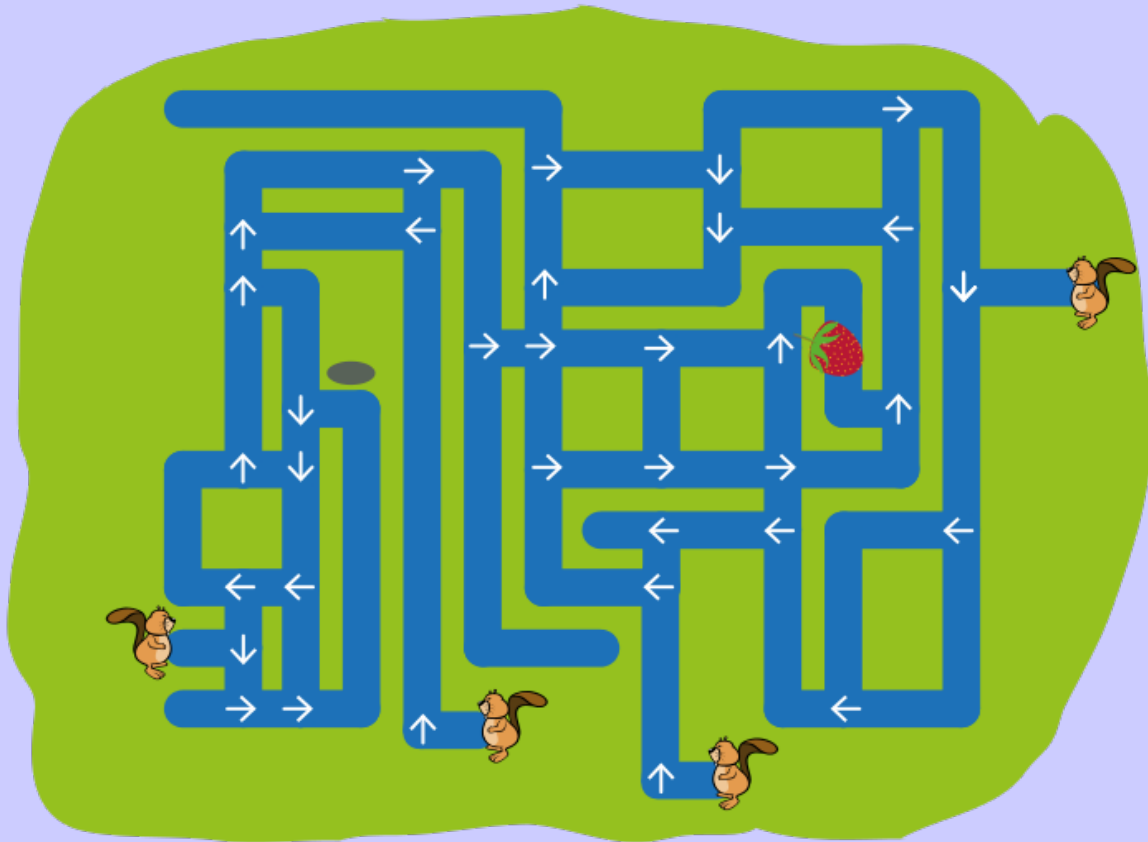
Canada



# Strawberry Hunt

## Story

Four beavers swim through canals in an attempt to find a strawberry. They start at different places and always move in the direction of the arrows shown below.



Each beaver either finds the strawberry, swims in a loop forever, or reaches and remains at a dead-end.

## Question

How many beavers find the strawberry?

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



### Connections to Computer Science

The system of canals in which the beaver are swimming has two main elements: canals (where the beavers can just swim through) and crossings (where the beavers have to decide by the arrow into which canal to swim next). In Computer Science this system is called a *graph* with *edges* (the canals) and *nodes* (the crossings). In this case the nodes have extra information attached to them: which canal should the beaver swim into next. Therefore, we could call this graph a *directed* graph.

Graphs can be used to describe many real-life situations similar to these tasks. For example:

- how traffic flows in a city, where there are one way streets and multiple intersections;
- how electricity, water or natural gas are transported around a series of homes;
- how information flows between computers on the internet.

The analysis of graphs in terms of *maximum flow* or *shortest paths* is an active area of research in computer science.

### Country of Original Author

Slovenia

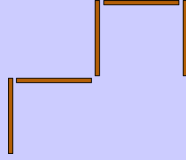




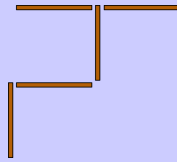
## Five Sticks

### Story

Nola puts five sticks on the table making this shape:



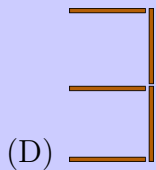
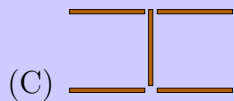
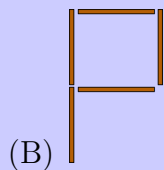
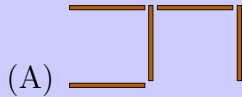
Then Adam moves one stick to a different place making this shape:



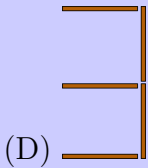
Now, Vera wants to move one stick to a different place.

### Question

Which shape is Vera **not** able to make?

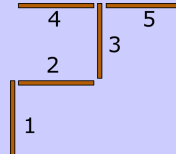


Answer



Explanation of Answer

Let us number the sticks for easier explanation.

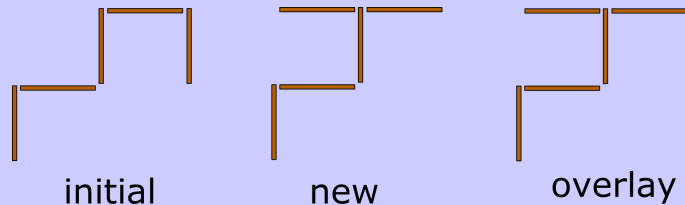


Sticks placed *horizontally* have left and right ends. Sticks placed *vertically* have top and bottom ends.

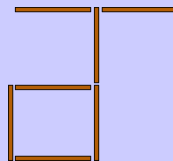
- (A) can be made by placing the stick 1 vertically down from the right end of stick 5.
- (B) can be made by placing stick 5 vertically between the left end of stick 4 and top end of stick 1.
- (C) can be made by placing stick 1 horizontally to the right of the right end of stick 2.
- (D) cannot be made by moving any one stick.

Alternatively, when you create a shape by moving one stick, you can put both images onto each other. The resulting picture will need to have exactly 6 sticks.

This is how it looks when overlaying the first shape (Adam's) and the second shape (Nola's), where we have the maximum number of overlapping sticks possible.



If you overlay the shape Nola created with the alternatives, only alternative (D) contains more than 6 sticks:



### Connections to Computer Science

To solve this problem, we need to think *abstractly* and also consider many possibilities or *choices* in our *search space*.

By *thinking abstractly*, we can imagine the consequences of moving the sticks without having the physical sticks in front of us. In computer science, since the information is stored inside the computer, as electronic signals, we cannot “touch” the information, so we need to imagine and visualize the information in other, non-concrete ways.

In order to determine the correct answer, we can consider every possible movement of every possible stick. That is, starting from the *initial configuration*, we can consider moving each of the 5 sticks to touch each “corner” of the figure, and each of these possibilities (around 30 or so configurations) is one step in an *exhaustive search* technique, which is guaranteed to solve the problem, although perhaps not as efficiently as other techniques.

### Country of Original Author

Slovakia



## Toy Storage

### Story

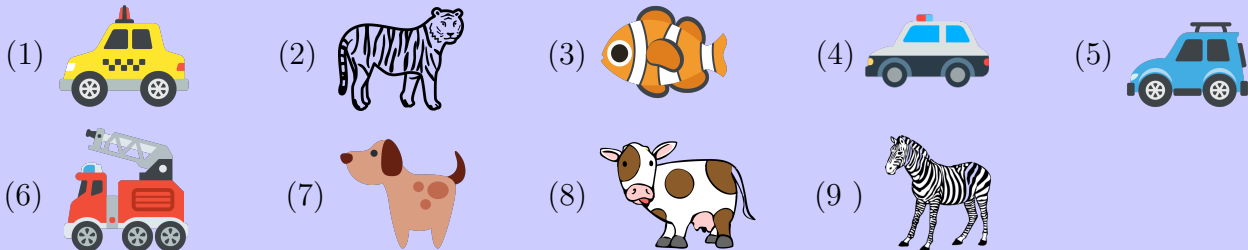
Tom has two types of toys: animal toys and vehicle toys. Tom fills three boxes by putting three toys in each box. As long as there is room, he puts

- vehicles into box A,
- animals with striped bodies into box B, and
- animals with spotted bodies into box C.

However,

- Anytime he tries to put a toy into box A and it is full, he then tries to put the toy into box B.
- Anytime he tries to put a toy into box B and it is full, he then tries to put the toy into box C.
- Anytime he tries to put a toy into box C and it is full, he then tries to put the toy into box A.

Tom puts the following nine toys into boxes in the following order:



### Question

Where does Tom put the dog and zebra?

- (A) Tom puts the dog in box C, and the zebra in box B.
- (B) Tom puts both in box A.
- (C) Tom puts both in box B.
- (D) Tom puts both in box C.

### Answer

(D) Tom puts both in box C.

### Explanation of Answer

For toys 1 to 5, we can simply follow the rules and put them into corresponding box.

When Tom tries to put the firetruck into box A according to rule (1), he realizes that box A is full. He puts the truck into box B, which still has room for one toy.

The dog and cow have spots so Tom puts them in box C, which still has room for three toys.

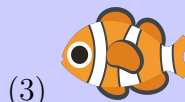
When Tom tries to put the zebra into box B according to rule (2), he realizes that box B is full. He puts the zebra into box C, which still has room for one toy.



Box A



Box B



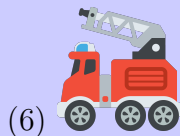
Box B



Box A



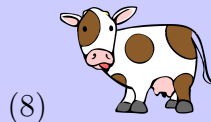
Box A



Box B



Box C



Box C



Box C

### Connections to Computer Science

When storing information, one important goal is to be able to quickly *retrieve* a particular piece of information. One method to accomplish this goal is to store information in a *hash table*.

A *hash table* consists of several locations where we can store information. The position of where to store a piece of information is determined by a *hash function*. For this task, the hash function for the toys was to put vehicles into box A, animals with stripes into box B, and animals with spots into box C.

As was also realized in solving this task, sometimes there are *collisions* in the hash function, where there is no more room in a particular location to store a piece of information. The *collision resolution strategy* used in this problem was to try the “next” box: in computer science, this strategy is formally called *linear probing*.

Hashing can provide extremely efficient lookup times even with a massively large amount of data.

### Country of Original Author

Taiwan

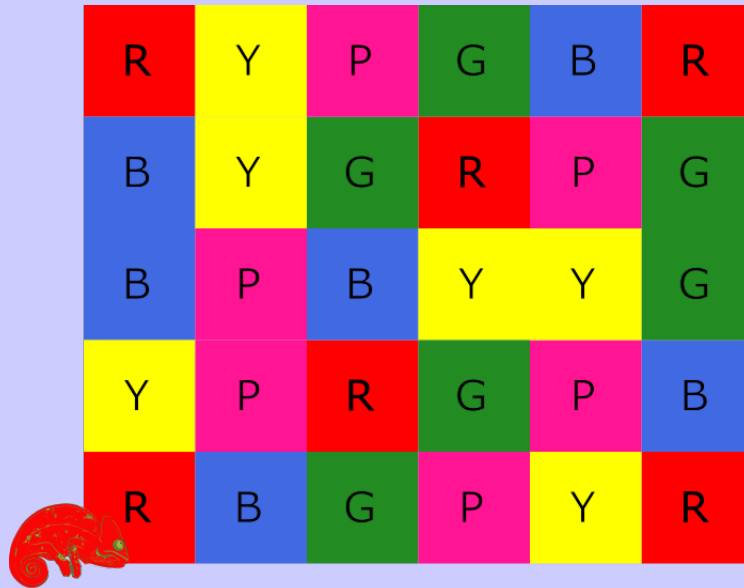


## Part C

## Chameleon

### Story

A chameleon travels on the grid below. It moves between adjacent cells either horizontally, vertically or diagonally. In a cell, a chameleon has the same colour as the colour of the cell.



### Question

What is the minimum number of different colours that the chameleon has when traveling from the lower left of the grid to the upper right?

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

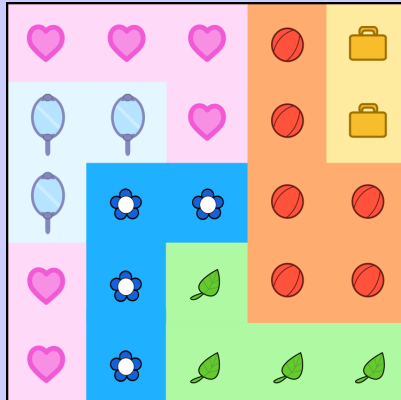




# Wallpaper

## Story

Robyn covers a wall with six overlapping rectangular sheets of wallpaper as shown. Each sheet of wallpaper is designed using a different image in a repeating pattern.



## Question

What is the order of the wallpaper pieces from the one placed first to the one placed last?

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

### Answer



### Explanation of Answer

The yellow wallpaper with the briefcases is the only wallpaper that isn't cut off by another one, so that must be the last one. You can see the suitcase cuts off the basketball wallpaper, that the basketball wallpaper cuts off the leaf wallpaper, the leaf wallpaper cuts off the flower wallpaper, the flower wallpaper cuts off the mirrors and the mirrors cut off the hearts.

### Connections to Computer Science

Using the same wallpaper rectangles in a different order would have given a totally different image: the *ordering of operations* are very important in computer science.

The *order* in which you execute statements is crucially important in programming or using a computer: an incorrect order of statements will yield incorrect results. Thinking of a mathematical expression like  $3 + 2 \times 7$ , we “know” from our BEDMAS rules that we must evaluate the  $2 \times 7$  first before adding it to 3. Similarly in *computer algorithms*, the order of evaluating steps influences what the end result will be.

This task is also an example of using *layers*: digital image editing uses layering to alter parts of an image. For example, images can be *composed* so that foreground images, such as a person, can be placed in a background image, such as a beach or forest. The software to enable this requires algorithms such as *boundary detection* in order to know where images overlap or intersect.

### Country of Original Author

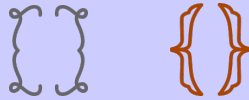
United States



# Bracket Bracelet

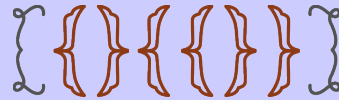
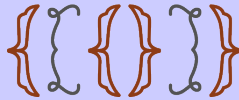
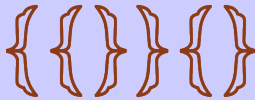
## Story

A jewelry shop produces chains used to make bracelets. The chains are built by continually adding matching pairs of bracket-shaped ornaments. There are two types of pairs:



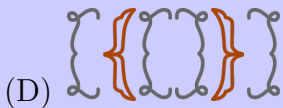
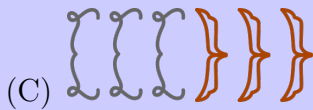
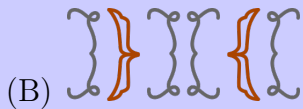
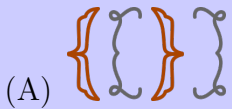
After choosing a starting pair, a second pair is either added to the end of the chain or inserted between the previously added pair. This process can be repeated any number of times.

Examples of three different chains that can be produced are shown below.

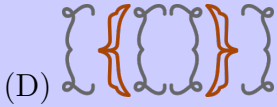


## Question

Which of the following chains can also be produced?



### Answer



### Explanation of Answer

They started off with two bracelets, placed a pair in between and then placed a third pair between the second pair.

All other bracelets cannot be constructed according to the methods stated in the task:

- (A) Position 3 is wrong: right side of ornament 2 is before the right side of ornament 1;
- (B) Position 1 is wrong: it starts off with a right side of an ornament instead of a left side, which is not correct;
- (C) Position 2 is wrong: there are 3 left sides of one ornament and then 3 right sides of another ornament, so there is no pairing between bracelets.

### Connections to Computer Science

The rules for the bracelets are exactly like the rules for *bracket expressions*. Bracket expressions occur in mathematics when certain expressions are to be grouped together: for example,  $(2 * (6 / (4 - 2)))$  is one such bracket expression

Computer scientists call correct expressions *well-formed*. Expressions containing errors are called *malformed*. In this task, we were asked to find the well-formed expression amongst the malformed expressions.

An expression that is well-formed can also be called *syntactically correct*. The *syntax* of a statement is the correct form or structure: for example, the sentence “My dog likes walks” is syntactically correct, but the sentence “Likes walks dog my” is not syntactically correct, since the form or order of the words is not correct. Most programming languages in computer science require that programs are syntactically correct: in fact, computers are much more strict about syntax since computers do not have the same ability as humans to *infer meaning* from malformed sentences. For example, most English speakers could understand what the meaning of “The stairs up I walked”, but an equivalently scrambled statement in a *computer programming language* would be rejected by the computer as a *syntax error*.

### Country of Original Author

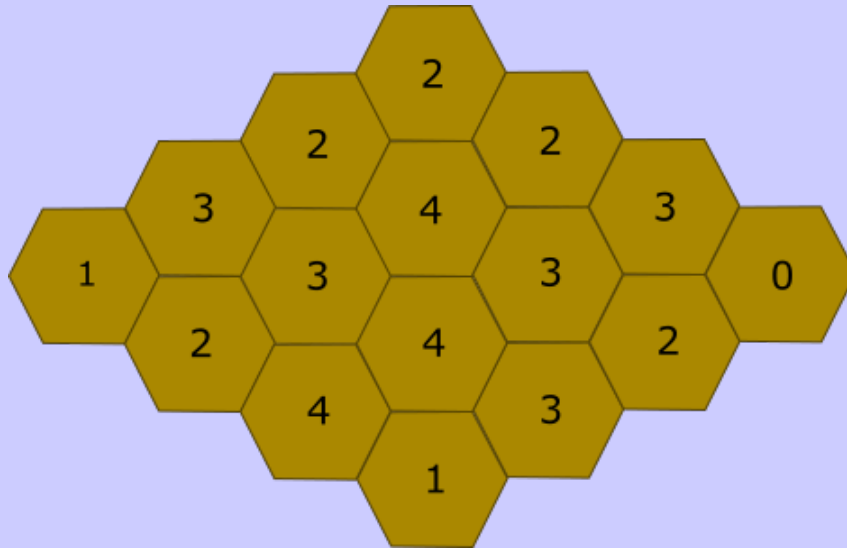
Austria



## Beehive

### Story

A bear studies how many hexagons in a honeycomb contain honey. For each hexagon, the bear records how many *other* hexagons touching this hexagon contain honey. So this number could be 0, 1, 2, 3, 4, 5 or 6. The results of the bear's study are below.



### Question

How many hexagons contain honey?

- (A) 7
- (B) 8
- (C) 9
- (D) 10

Answer

(C) 9

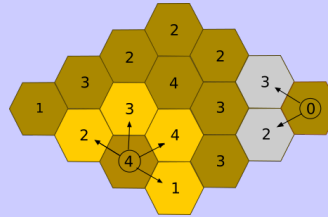
Explanation of Answer

One way to solve this task is to start from the cells for which the content of their adjacent cells is known. It is the case for two cells, that have a circle in the picture below:

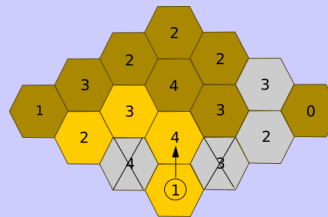
All the adjacent cells of the cells with a 0 are empty

All the adjacent cells of the cells with a 4 are full

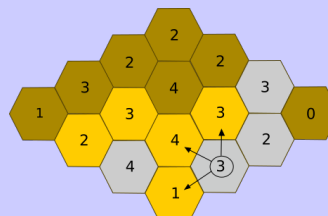
We will use yellow to represent honey in a hex, grey to represent no honey, and brown for undetermined.



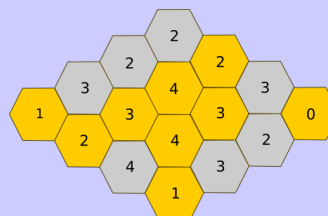
After that, we can easily find some others cells, near the zone that we have changed, for which the content of their adjacent cells is now known. For instance the cell with a circle in the picture below, has only 1 full adjacent cell. It is the one already colored in yellow, just above it. So its other adjacent cell is empty.



Then, the cell with a 3 on it, which we just coloured grey, must have exactly 3 full adjacent cells. We have already two of them on its left, the one on the right is empty, so the one above it must be full.



We can go on from one adjacent cell to the others, in order to cover all the beehive.



### Connections to Computer Science

This task concerns *logic* and *inference*.

The logic in this problem is to understand that, for example, a cell marked as 0 means that none of its neighbours contain honey. From this fact, we can *infer* further facts, and build up a solution to the larger problem.

The method of building a solution using inference is what is called a *bottom-up* algorithm: we start with a solution to a very small part of problem (the “bottom” of the problem) and then build up larger and larger solutions until we have solved the entire problem (the “top” of the problem).

### Country of Original Author

Iran

