



UNIVERSITY OF  
**WATERLOO**



The CENTRE for EDUCATION in  
MATHEMATICS and COMPUTING



2018  
*Beaver  
Computing  
Challenge*  
(Grade 9 & 10)

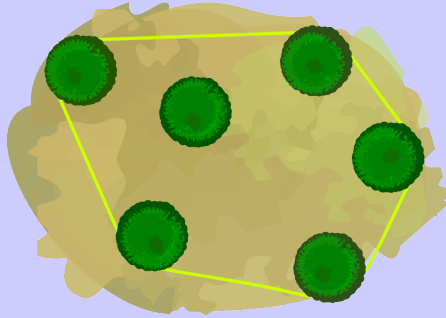
*Questions,  
Answers,  
Explanations,  
and  
Connections*

# Part A

## Roped Trees

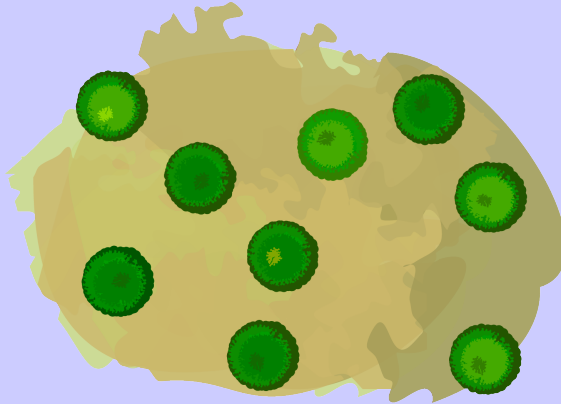
### Story

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



### Question

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



- (A) 4
- (B) 5
- (C) 6
- (D) 7

## Answer

(C) 6

## Explanation of Answer

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



## Connections to Computer Science

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

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

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

## Country of Original Author

Canada



## Rotation Game

### Story

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

Red	Green
Blue	Yellow

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

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

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

Yellow	Blue
Green	Red

### Question

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

- (A) Red Green Blue Green Yellow
- (B) Red Blue Green Blue Red
- (C) Red Blue Yellow Red Green
- (D) Red Red Yellow Red Blue

### Answer

(B) Red Blue Green Blue Red

### Explanation of Answer

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

Blue	Red	Green	Yellow	Blue	Red	Red	Green
Yellow	Green	Red	Blue	Yellow	Green	Blue	Yellow

### Connections to Computer Science

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

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

### Country of Original Author

Canada



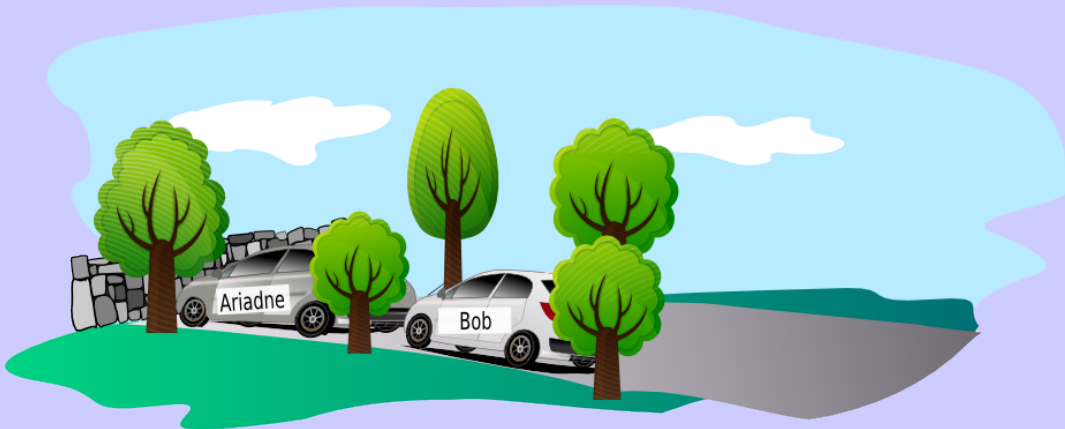
## Sharing a Driveway

### Story

Neighbours share a very long and narrow driveway. Cars parked in the driveway can only leave by backing out. A schedule has been created so that nobody is ever blocked in when they need to leave the driveway. On each day, any cars which need to leave do so before any other cars enter. Before Monday, there are no cars in the driveway. The table below gives details of how the driveway is shared over one week.

Day	Number of cars leaving	Number of cars entering	Owners of cars and order they enter
Monday	0	2	Ariadne, Bob
Tuesday	1	3	Kate, Ben, Roy
Wednesday	2	1	Daisy
Thursday	0	2	Finn, Rose
Friday	3	1	Vincent

The driveway at the end of Monday is shown below:



### Question

Whose cars will be parked on the driveway at the end of Friday?

- (A) Bob, Vincent, Daisy
- (B) Vincent, Ariadne, Rose
- (C) Ariadne, Kate, Vincent
- (D) Ariadne, Daisy, Vincent

### Answer

(C) Ariadne, Kate, Vincent

### Explanation of Answer

We record which cars are parked in the driveway at the end of each day listing them in the order they entered:

End of Monday: Ariadne, Bob

End of Tuesday: Ariadne, Kate, Ben, Roy

End of Wednesday: Ariadne, Kate, Daisy

End of Thursday: Ariadne, Kate, Daisy, Finn, Rose

End of Friday: Ariadne, Kate, Vincent

### Connections to Computer Science

The task relies on the concept of *stack*. A stack is an *abstract data type* where the element most recently stored is the first element removed from the structure. The stack can be modified using two operations: *push*, which puts an item into the stack, and *pop*, which removes an element from the stack. Stacks are sometimes described as *LIFO* (*last in first out*) structures.

In this problem, each day involves a sequence of pop operations, followed by a sequence of push operations.

Underlying this problem is the skill of remembering the *state* of the the driveway. In many programming languages, there are *variables* which can take on various values, and the analysis of these programs involves determining the state of each variable.

### Country of Original Author

Cyprus



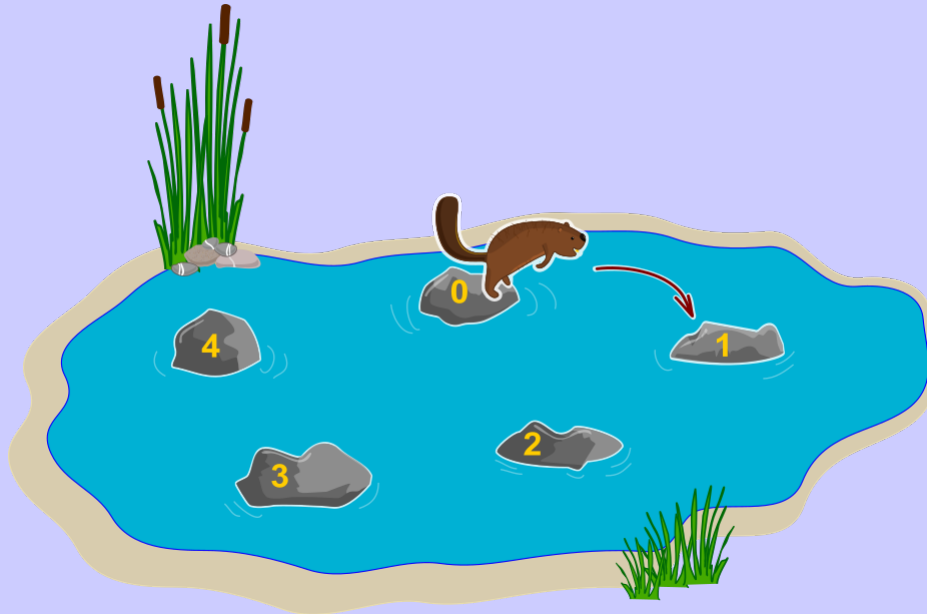


## Beaver Jump Challenge

### Story

Beavers take part in an annual challenge. Starting from rock number 0, they jump clockwise from rock to rock. For example, if a beaver jumps 8 times, it ends up on rock number 3:

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 3$



### Question

One of the beavers showed off and jumped an astonishing 129 times. On which rock did it end up?

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

### Answer

(A) 4

### Explanation of Answer

If a beaver jumps 5 times, it ends up where it starts. Let's call it a "lap". Since  $129 = 25 \times 5 + 4$ , the beaver will do 25 laps and have 4 jumps left. In these last four jumps, the beaver goes from rock number 0 to rock number 4.

### Connections to Computer Science

You may have seen this operation in math class before. It's part of what you might know as *Long Division* or *Euclidean division* which calculates an integer quotient and a remainder.

In this case you're asked to calculate the remainder of 129 when it is divided by 5. Since this operation is used very commonly in computers, it has a name: *modulo operation*. Usually "%" or "mod" is used as an operator. So, for our equation we could write:  $129 \% 5 = 4$ .

Typical applications of this operator are in programs containing loops (just like our beaver jumping in loops), and when variables *overflow* because they are too big. *Modular arithmetic* is also used heavily in *cryptosystems*, such as *RSA cryptography*, used to keep our passwords safe online.

### Country of Original Author

Switzerland



## Lemonade Party

### Story

James made 37 litres of lemonade at home and now he wants to bring it to a celebration at school. He has several empty bottles of various sizes but he wants to use the smallest number of them to bottle exactly 37 litres of lemonade.

He has one bottle of each of the following sizes:

- 1 litre
- 2 litre
- 4 litre
- 8 litre
- 16 litre
- 32 litre

### Question

What is the least number of bottles James needs to use?

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

### Answer

(C) 3

### Explanation of Answer

James should use a 32-litre bottle, a 4-litre bottle and a 1-litre bottle. Notice that this is three bottles and  $32 + 4 + 1 = 37$ .

There isn't a 37-litre bottle so James must use more than one bottle. You can't find two numbers (possibly the same) among the sizes 1, 2, 4, 8, 16 and 32 that add up to 37, so more than two sizes are needed. This means we know that three bottles is the least James can use.

### Connections to Computer Science

The *binary number system* is central to computation and to computers. In the binary number system, there are only two digits: 0 and 1. Why? It is related to how the *hardware* in modern computers works. The key idea is that part of hardware can be “on” or “off” meaning it only has two possible states. All of this will change if and when *quantum computers* become commonplace in society.

The volume of the bottles keeps doubling in this task, just like the values of individual *bits* in a binary number. In computer science it is important to be able to convert numbers in one *base* (i.e., base ten) to another base (i.e., binary). When converting to binary, a simple trick is to select the largest bit that still fits in the number you are trying to convert.

### Country of Original Author

United States

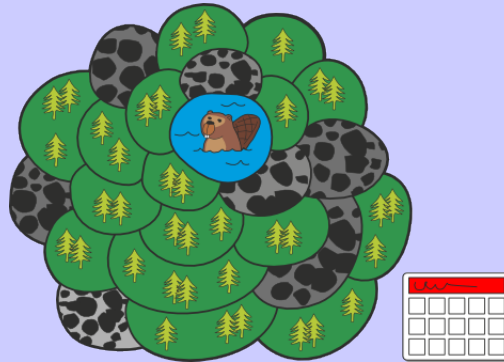


## Part B

# Beaver Lake

## Story

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



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



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

## Question

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

- (A) 4 days
- (B) 5 days
- (C) 6 days
- (D) 7 days

### Answer

(C) 6 days

### Explanation of Answer

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



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

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

### Connections to Computer Science

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

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

### Country of Original Author

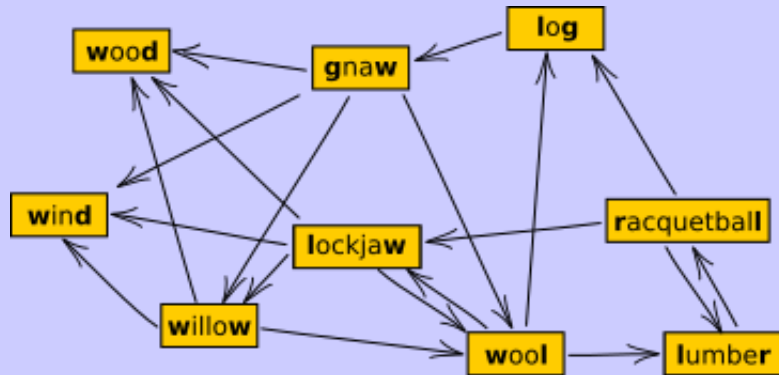
Lithuania



## Longest Word Chain

### Story

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



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

### Question

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

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



### Answer

(C) 8

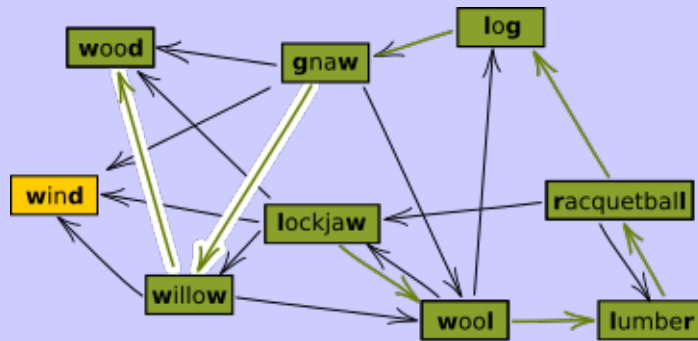
### Explanation of Answer

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

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

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

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



### Connections to Computer Science

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

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

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

### Country of Original Author










Czech Republic



## Twists and Turns


### Story

Tom lives in a city with a lot of twists and turns. His mom forgot her phone at home and asked Tom to bring it to her at work. She sent Tom the following street map to help him find his way.

	1	2	3	4	5	6
A	 ↓		↙	↙	↓	
B	↘	↘	↘	↙	↘	↓
C	↓		↓	↘	↙	
D	↙	↙	↙	↘	↘	↓
E	↓		↘			↙
F	↙	↙	↘	↘	↘	

The map is a table with rows *A*, *B*, *C*, *D*, *E*, *F*, and columns 1, 2, 3, 4, 5, 6. Tom starts in the first row and column (location A1) and his mom is located in the last row and column (location F6).

There is one symbol at each location on the map. The symbols have the following meanings:

- ↓ means Tom can only go down from this location
- ↘ means Tom can go either down or right from this location
- ↙ means Tom can go either left or down from this location
- ↗ means Tom can go either up or left from this location
-  means Tom hits an obstacle and cannot move beyond this location

### Question

According to the map, which of the following is a valid route from Tom to his mom?

- (A) A1 B1 B2 B3 C3 D3 E3 E4 F4 F5 F6
- (B) A1 B1 B2 B3 B4 C4 D4 D5 D6 E6 F6
- (C) A1 B1 B2 B3 C3 D3 E3 F3 F4 F5 F6
- (D) A1 B1 B2 B3 C3 D3 D4 D5 D6 E6 F6

### Answer

(C) A1 B1 B2 B3 C3 D3 E3 F3 F4 F5 F6

### Explanation of Answer

Answer A is not correct because there is an obstacle in E4.

Answer B is not correct because Tom cannot move from B4 to C4. From B4 he can only move either to B3 or to A4.

Answer D is nor correct because Tom cannot move from D3 to D4. From D4 he can only move either to D2 or E3.

We can follow the route given as Answer C to confirm that it agrees with all the map symbols along the route.

### Connections to Computer Science

The information described in this task uses both a *two-dimensional array*, to represent the map, and a *one-dimensional array*, to represent the order in which the two-dimensional array is moved through. Arrays are very common data structures in many programming languages.

To solve the task, we must be read the one-dimensional array and treat that data as *indices* for what the current *state* (or position) is in the two-dimensional array. Although the problem seems simple, simultaneous reading of these two arrays becomes difficult due to the *representation* of the data. If the directions were given as “up”, “down”, “left” or “right”, it would be much easier to follow the path from the top to the bottom. As one can see, deciding how to represent data is an important skill needs for processing information.

One way to find a solution(s) to this problem is to apply a *depth first search (DFS)* algorithm, which is also known as a *backtracking* algorithm. This technique tries to reach the destination by “moving forward” until either the solution is found, or there are no more possible “forward” movements, at which point we “backtrack” to an earlier position and try another path forward.

Another way to solve this problem is to start from the end of the path and try to reach the starting point.

Both methods can be combined to result in a method which is efficient if there are only a few solutions (paths, in this case).




### Country of Original Author

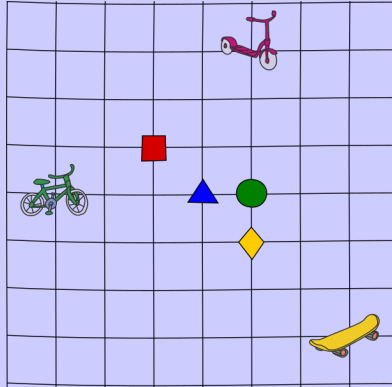
Romania



## Three Friends

### Story

Bob , Alice , and Jenny  are each playing at a different intersection in the 8-by-8 grid below. They want to choose an intersection where they can all meet to show off their different vehicles.



They will each travel to the chosen location moving only horizontally and vertically along grid lines.

### Question

Which intersection should be chosen, so the three friends travel the shortest total distance?

- (A) the intersection marked by a green circle (●)
- (B) the intersection marked by a red square (■)
- (C) the intersection marked by a blue triangle (▲)
- (D) the intersection marked by a yellow rhombus (◆)

### Answer

(A) the intersection marked by a green circle (●)

### Explanation of Answer

The total distance to the intersection marked by a green circle (●) is  $3 + 4 + 5 = 12$ .

The total distance to the intersection marked by a red square (■) is  $4 + 3 + 8 = 15$ .

The total distance to the intersection marked by a blue triangle (▲) is  $4 + 3 + 6 = 13$ .

The total distance to the intersection marked by a yellow rhombus (◆) is  $4 + 5 + 4 = 13$ .

Of these, 12 is the shortest distance.

### Connections to Computer Science

In computer science, *local search* can be used on problems that can be formulated as finding a solution maximizing a criterion among a number of candidate solutions. In this problem, there are only four possible solutions (the friends must meet at one of the four intersections), thus these can be tried in sequence, and only the best solution is kept.

The idea of local search is a *heuristic* for finding solutions to very large or difficult problems. For example, even if there are an exponential number of possible solutions (such as in the *Traveling Salesman Problem*), local search will begin with some solution, and given some limit of computing time, it will return some improved, but not necessarily optimal, solution. That is, the found solution will be *locally optimal*, but may not be *globally optimal*.

### Country of Original Author

Vietnam



## Timetabling

### Story

Bebras Tech offers the following evening classes:

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

Three beavers would like to sign up for these courses:

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

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

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

### Question

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

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

### Answer

(C) 4

### Explanation of Answer

The following schedule shows that four evenings can be used.

Evening 1: Computing (taken by Xavier and Yvette)

Evening 2: Language (taken by Xavier and Zoey) and Geography (taken by Yvette)

Evening 3: Science (taken by Yvette and Zoey)

Evening 4: Math (taken by Xavier and Zoey)

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

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

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

### Connections to Computer Science

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

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

### Country of Original Author

United Kingdom



## Part C

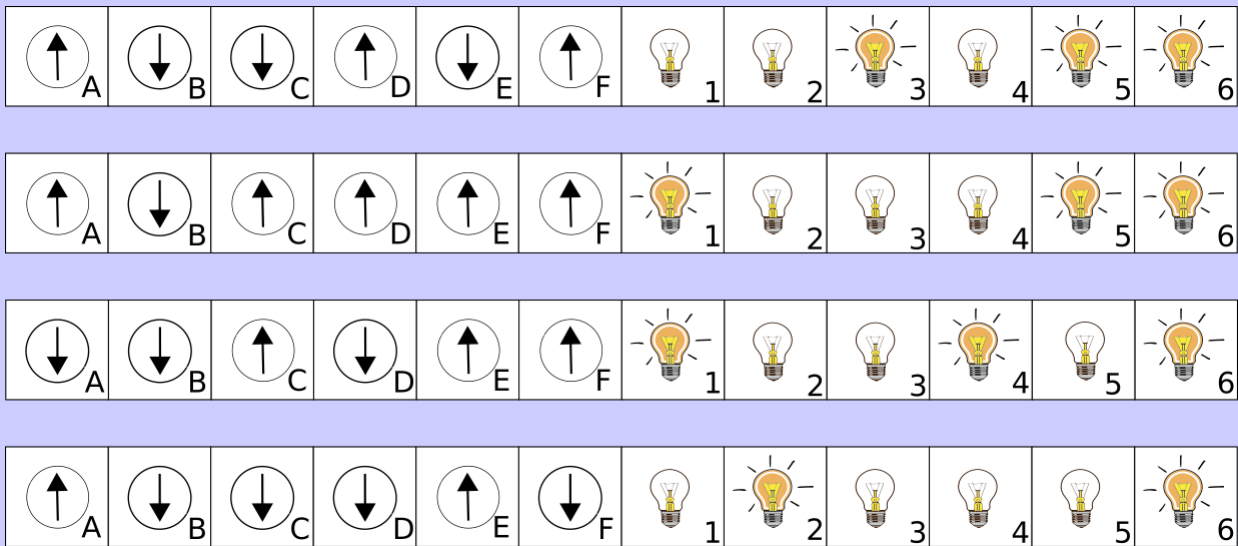


## Bulbs

### Story

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

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



### Question

Which switch is connected to which bulb?

- (A)  $C \rightarrow 1, E \rightarrow 2, D \rightarrow 3, A \rightarrow 4, F \rightarrow 5, B \rightarrow 6$
- (B)  $C \rightarrow 1, F \rightarrow 2, E \rightarrow 3, A \rightarrow 4, D \rightarrow 5, B \rightarrow 6$
- (C)  $C \rightarrow 1, F \rightarrow 2, D \rightarrow 3, E \rightarrow 4, A \rightarrow 5, B \rightarrow 6$
- (D)  $C \rightarrow 1, F \rightarrow 2, B \rightarrow 3, A \rightarrow 4, D \rightarrow 5, B \rightarrow 6$

### Answer

(B)  $C \rightarrow 1$ ,  $F \rightarrow 2$ ,  $E \rightarrow 3$ ,  $A \rightarrow 4$ ,  $D \rightarrow 5$ ,  $B \rightarrow 6$

### Explanation of Answer

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

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

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

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

### Connections to Computer Science

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

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

### Country of Original Author

Hungary

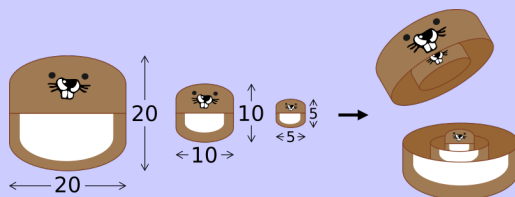


## Nesting Dolls

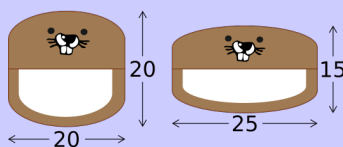
### Story

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

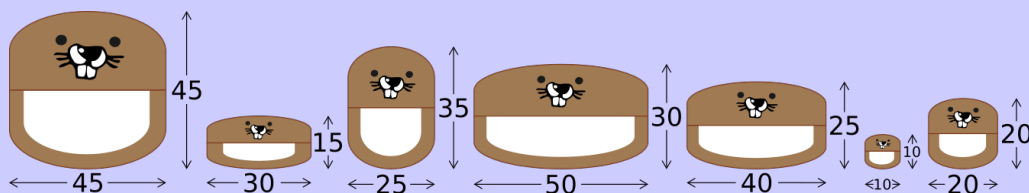
For example, a doll with width 5 and height 5 fits inside a doll with width 10 and height 10, which in turn fits inside a doll with width 20 and height 20. After this, only one doll is visible.



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



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



### Question

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

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

## Answer

(B) 2

## Explanation of Answer

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



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

## Connections to Computer Science

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

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

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

## Country of Original Author

Taiwan



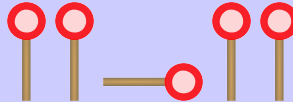
# Plane Signals

## Story

Jana and Robin play outside with their toy plane. Jana stands on a hill and Robin collects the plane after each landing. The plane always lands in long grass, which means, after landing, it is only visible from the hill and not up close. So, Jana uses a beacon and the following code to send Robin signals guiding her to the plane.

Left	Right	Towards the hill	Away from the hill

Unfortunately, there is a problem. Some signals received by Robin have multiple meanings. For example, suppose Jana sent the following signal to Robin:



Jana could mean either **Left, Towards the hill, Left** or she could mean **Left, Right, Left, Left**. Jana and Robin have to revise the code to fix this problem.

## Question

For which of the following codes do all signals have only one meaning?

	Left	Right	Towards the hill	Away from the hill
(A)				
(B)				
(C)				
(D)				

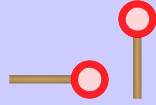
## Answer

(C)

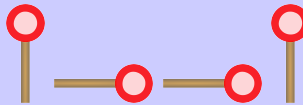
## Explanation of Answer

We shall first eliminate the other answers.

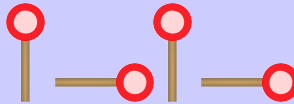
Answer B is not a good code because **Left** followed by **Right** uses the same signals as **Towards**:



Answer D is not a good code because **Left** followed by **Away** uses the same signals as **Towards**:



Answer A is not a good code because **Left** followed by **Away** uses the same signals as **Right** followed by **Right** again:



Why can we be sure that Answer C is a good code and using it ensures that all signals have only one meaning? Notice that the four commands are all coded as a raised signal followed by zero or more lowered signals. So, as soon as Robin sees a raised signal, she knows that Jana has begun signaling a new command.



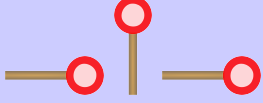
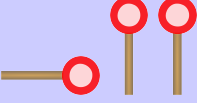
## Connections to Computer Science

When information is transmitted between computers, either through wires or through the air wirelessly, the information is sent as a rapid sequences of *signals*. Each signal can be viewed as describing one of two possibilities: either *on* or *off*. In this problem, Jana and Robin use two different signals for their messages, where we could view the beacon being vertical as being “on” and being horizontal as being “off.”

The specific way by which information is translated into signals is called a *binary code*. In this problem, a *variable length code* was used, because the number of signals used for one message or command could be different for different messages. Most computers use *fixed length codes* to encode information: for example, the most common encoding of textual information is *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 not having a corresponding original message or by having two or more possible original messages. 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 code have the property that no code is the prefix (or starting sequence) of any other code. For example, the following is a prefix-free code:

Left	Right	Towards the hill	Away from the hill
			

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

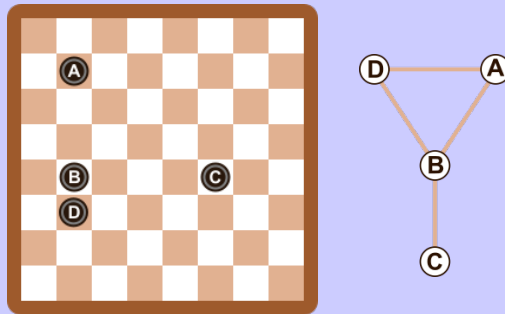
Belgium



# Rows and Columns

## Story

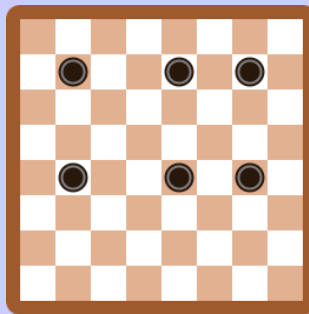
A game board with four pieces on it is shown below together with a diagram representing it.



The diagram is drawn in the following way

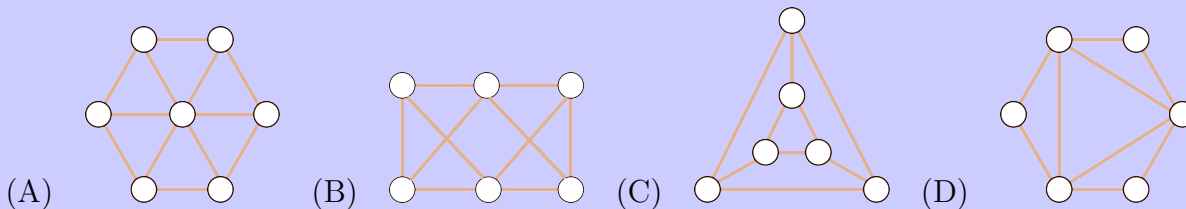
- One circle is drawn for each piece on the game board. (Letters have been shown in this example so you can see the correspondence between game pieces and circles on the diagram.)
- Whenever two pieces are in the same row or in the same column, a line is drawn between the corresponding circles. (No other lines are drawn.)

Rich draws a diagram for the following board in the same way.



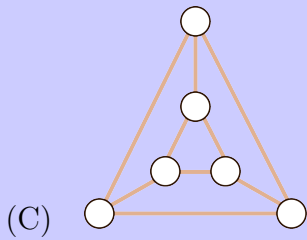
## Question

Which of the four diagrams might Rich have drawn?



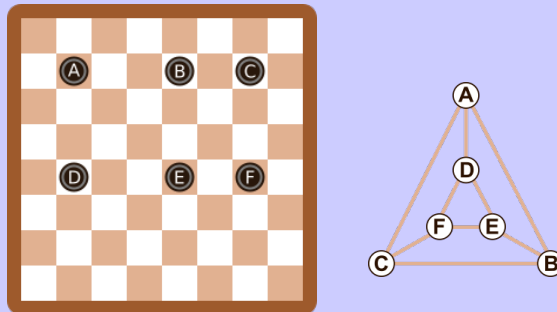


### Answer



### Explanation of Answer

By putting letters on the game pieces and corresponding circles, we can see that Answer C is correct.



Answer A cannot be correct because it has 7 circles and there are only 6 game pieces.

To see that Answers B and D are incorrect, notice that each game piece is in the same row as exactly two other pieces and in the same column as exactly one other piece. Therefore, a line must join each circle to exactly  $2 + 1 = 3$  other circles. In the diagrams in Answer B, you can find a circle that is joined to four other circles. In the diagrams in Answer C, you can find a circle that is joined to only two other circles.

## Connections to Computer Science

In computer science, diagrams like this are often used to represent the essential information of a problem. Such a diagram is called a *graph*. The circles in the diagram are called *vertices* or *nodes* of the graph, and the connections between vertices are known as *edges*.

In a graph, the key concern is whether or not pairs of vertices are connected. Where the vertices are drawn, or whether the edges are straight or curved, does not matter. The same graph can often be drawn in different ways, but different graphs will differ either in the number of vertices or how the vertices are connected.

Whether a graph can be a useful *representation* of a computer science problem depends on what information is needed to solve the problem. For instance, if we need to know whether a white piece on the board is in the same row or column as a black piece, then our graphs are not a good representation: the color of the pieces is not represented in the nodes. On the other hand, our graph is a good representation for answering questions such as “what is the minimal number of pieces you have to remove so that no piece is in the same column or row as any other piece”.

The problem also covers the *graph isomorphism* problem: given two graphs, are they *isomorphic*, meaning they are the same in terms of the vertices and the connections between vertices, even if they “look” different. There is no known *efficient* solution to the general graph isomorphism problem, but for specific small graphs, such as the graph in this problem, the problem can be solved in a reasonable amount of time.

## Country of Original Author

Belgium



## Find the Prize

### Story

Your friend is thinking of an integer between 1 and 63 (inclusive).

They offer to give you money if you guess the integer they are thinking of.

If you guess the number on your first guess, you win \$1000. Every time you guess incorrectly, your friend will take \$10 away from the prize money, but also tell you whether your guess was above or below the integer they were thinking of.

You find a strategy that *guarantees* you win at least \$ $N$ , regardless of the number your friend is thinking of.

### Question

What is the largest possible value of  $N$ ?

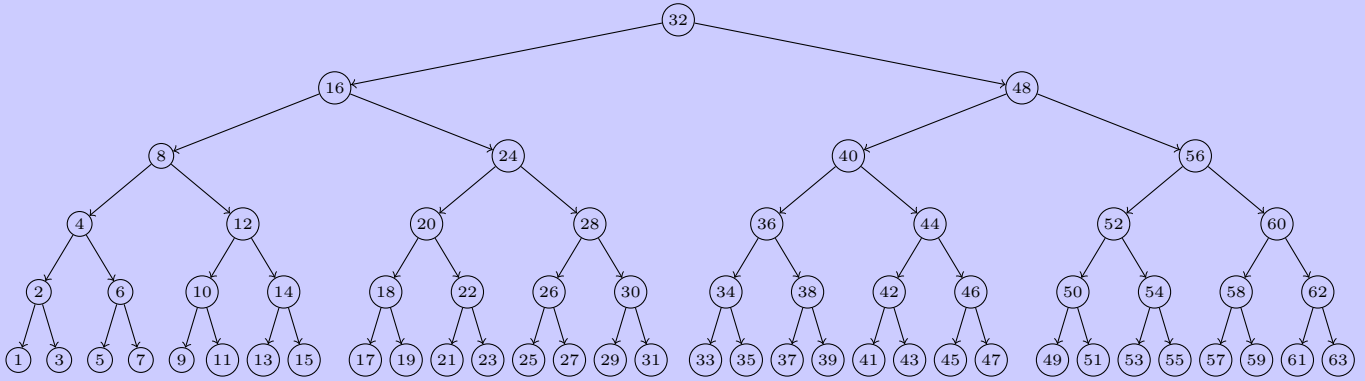
- (A) \$990
- (B) \$950
- (C) \$500
- (D) \$370

## Answer

(B) \$950

## Explanation of Answer

Consider the following *decision tree*:



To use this decision tree, start at the root, which is labelled 32. That should be the first guess. If you are correct, you win \$1000. However, if you are incorrect, and your guess was above the correct value, you should go left down the tree, and otherwise, your guess was below the correct value, and so you should go right down the tree. You then repeat this process by using the value listed as the next guess, and either winning or taking one step down the tree. Notice that every possible value between 1 and 63 is in the tree, and that you can make at most 5 incorrect guesses before you guess correctly. Thus, you are guaranteed to win at least \$950 with this strategy.

You cannot guarantee winning \$990, since there must be at least two incorrect guesses made, using the following analysis:

- if you pick a number from 1 to 32, I will say your guess was “below”; if you then guess something from 33 to 63, I say I was thinking of 64, and if you guess 64, I say I was thinking of 63.
- if you pick a number from 33 to 63, I will say your guess was “above”; if you then guess something from 1 to 31, I say I was thinking of 32, and if you guess 32, I say I was thinking of 31.

## Connections to Computer Science

As illustrated in the Explanation of Answer, we can model all of the choices as a *binary decision tree*. Since computers make decisions in a *binary* way (i.e., on/off, 0/1, yes/no), the modelling using *binary trees* is very natural.

The technique used to find the minimal number of guess is known as *binary search*. The key idea is to start in the middle, and reduce the size of the remaining *search spaces* (where to look next) by a factor of two. Thus, the *running time* for this technique for finding a particular numbers is roughly proportional to  $\log_2 N$ , where  $N$  is the number of elements. In this problem, that is why we can find every number in 6 guesses (at most 5 incorrect guesses followed by the correct guess).

Country of Original Author

South Africa

