Answer:

Emil ⇒ Beat ⇒ Alicia ⇒ David ⇒ Caroline

Explanation:

This task has to do with dependencies.

- Emil is the only friend with no dependencies so he must go first.
- Beat only depends on Emil so he can go second.
- Alicia depends on Beat and Emil, so she can go third.
- David depends on Alicia so he can go next,
- Finally Caroline depends on Alicia and David so she can go last.

No other sequence can fulfil the dependencies.

It's Computational Thinking:

Breaking down problems into parts (Decomposition)
Algorithms – following and describing
Data interpretation – modelling

Satisfying dependencies is a computational problem that happens frequently in real life.

This problem of dependency and ordering can be modelled with a graph. A graph is composed of vertices (here the friends) and arcs from some vertices to other ones (here there is an arc from X to Y if X needs to appear after Y. Our graph here is special: since there exists an ordering, it has no cycles. A cycle would be a path that starts at one vertex, follows some arcs and returns to the starting vertex. The graph in this problem is a directed acyclic* graph (DAG). The problem is then called a topological sort: we sort the vertices according to the dependency relation.

One way to do so is to select all the vertices where no arcs exit (hence those tasks can be done first), and put them at the beginning of our answer. Then we can remove these vertices from the graph, and continue with the new graph.

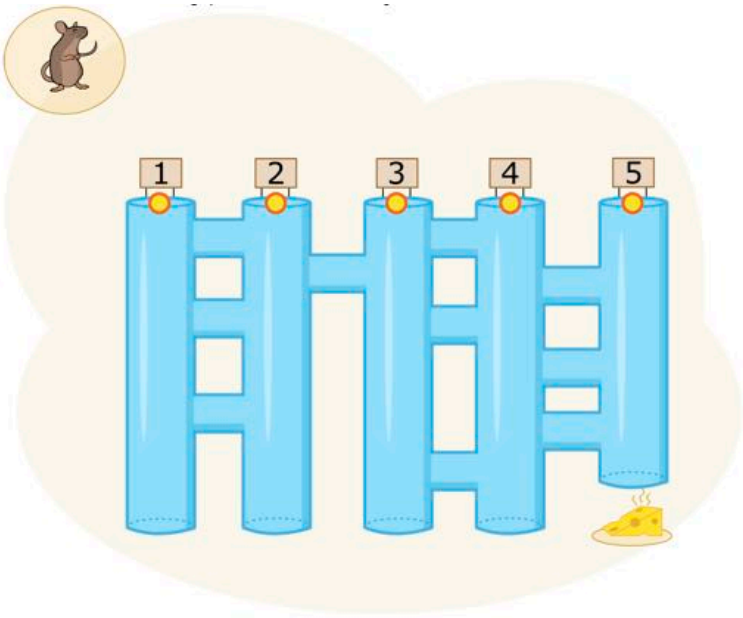
The existence of a vertex where no arcs exit is guaranteed by the fact that the graph is acyclic* (*having no graph cycles).

Tube System

Years 3+4 B Years 7+8 A
Years 5+ 6 A Years 9+10
Years 11+12

A mouse is at the entrance of a tube system. It wants to reach the cheese at the end of tube 5. The mouse always follows these commands:

1. Go downwards until a crossing
2. At the crossing, move through to the next vertical tube
3. Go to command 1



Question:

In which tube should the mouse start so that it reaches the cheese?

1 2 3 4 or 5

Answer:

3

Explanation:

From tube 1 the mouse always reaches tube 3.
From tube 2 it reaches tube 1.
From tube 4 it reaches tube 2.
From tube 5 the mouse gets to tube 4.

It's Computational Thinking:

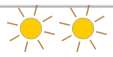
Algorithms – following and describing

Many robots are programmed so that they have to follow exact commands. This mouse does the same thing: it follows the commands 'go downwards' and 'change directions at the next crossing' over and over again. These kinds of commands depend on the choice of the tube entrance as to which way the mouse runs in the tube system. Most computer programs are deterministic: if you always input the same data, the program performs the same calculations and delivers the same output.

Barbara has been given two stamps.

With one she can produce a little flower, with the other a little sun.

Being a clever girl, she thinks of a way to write her own name by using the code below:

Letter	B	A	R	E	Y
Code					

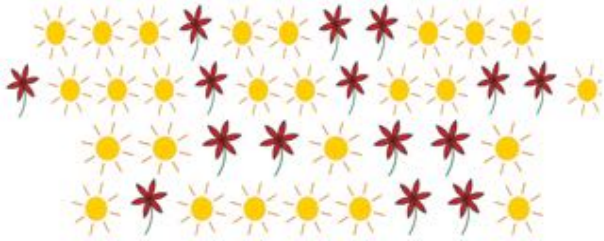
So Barbara" becomes:

She then writes the names of her friends. Unfortunately, they all got mixed up.

Question:

Drag the sun-flower-codes to the names of her four friends.

Abby	
Arya	
Barry	
Ray	



Answer:

3

Explanation:

This problem is most easily solved by noting that Abby starts with an A and a B and so we look for a code with two suns and a flower at the start. There is only one of these so this is assigned. Next it is noted that Arya's code begins with three suns and a flower. Again, there is only one of these so this is assigned. By continuing in this way, all the codes are quickly assigned to the correct names.

It's Computational Thinking:

Breaking down problems into parts (Decomposition)
Data representation - symbolism
Data interpretation - patterns
Algorithms – following and describing
Digital systems

Often in Computational Science, instead of storing data in a simple and straightforward way, we can devise a scheme to store it more efficiently, using less space.

For instance, computers store information about characters that can be typed on the keyboard in what is called an ASCII encoding. Each letter corresponds to a different sequence of 8 bits (0's and 1's). In ASCII, every character takes the same amount of space.

However, letters have different frequencies in texts (for example, the letter "E" is the most common letter in English words), and we can use these frequencies to improve our encoding.

Specifically, we encode frequent letters with smaller codes: in this question, B should be frequent and takes one symbol, A two, and the other letters more. There is a famous and widely used algorithm to do this for texts, named the Huffman coding. You cannot, however, use any encoding you wish: you have to make sure the code is unambiguous. For example, suppose the code was the following: Letter B was one flower, letter A was two flowers.

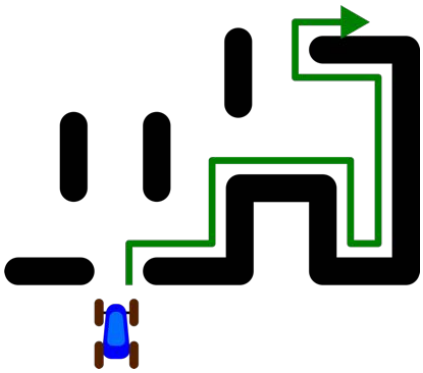
What do two flowers mean? It could be BB or A, but we have no way to know which one for sure. One way to achieve unambiguity is to make sure the code is prefix-free; that is if we take the code of a letter, it is not the beginning of any other code.

Also, since the Huffman code used depends on the text itself (it depends on the frequencies of letters), it is necessary to store the correspondence between the code and the actual letters. This takes a bit of space, but is negligible for long texts.

A robotic car uses a simple rule to drive through a maze:

Turn right whenever possible.

The picture on the right gives an example of how the robot would drive through a maze.



Question:

In how many of the following mazes will the car reach the red dot if it uses this system?

Maze A

Maze B

Maze C

Maze D

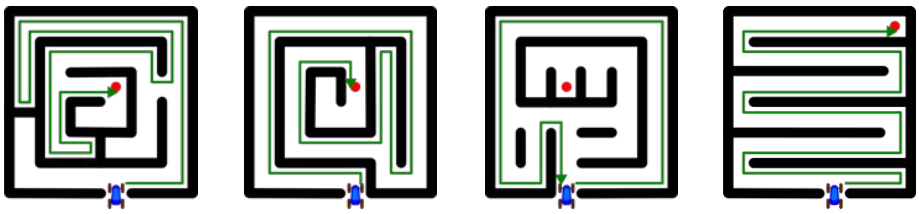
Choose from: 0 1 2 3 or 4

Answer:

3

Explanation:

In the pictures below, the green arrow indicates the path taken by the car. In Maze C the whole centre part of the maze is not visited and the red dot is not reached. In the 3 other cases the red dot is reached.



It's Computational Thinking:

Algorithms - following

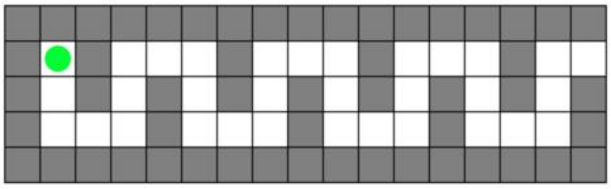
The method which is used by our car, is called the wall follower. It is one of the simpler techniques (algorithms) to solve a maze for which you do not know the layout in advance. By following this rule, you will never get lost: you will always return to the starting point eventually. However, as can be seen by our example, it does not guarantee that you will visit the entire maze.

Robot Exit

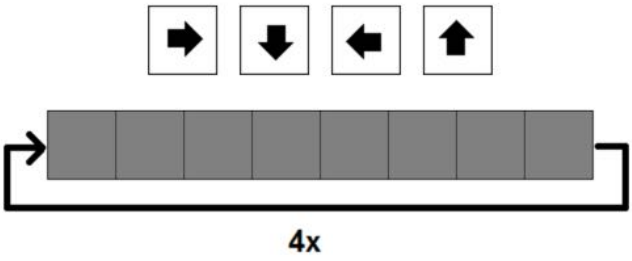
Years 3+4 A Years 7+8
Years 5+ 6 A Years 9+10
Years 11+12

Help the green robot to exit the maze.

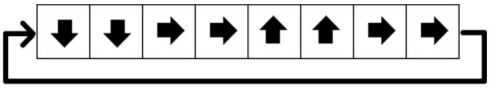
The robot will repeat your instructions 4 times.



Question:
Drag the arrows to form a set of instructions.



Answer:



Explanation:
In mobile robotics, maze problem solving is one of the most common problems. To solve this problem, an autonomous robot is used. Mazes can be of different kinds; having loops, without any loops, grid systems or without a grid system. In this short loop maze algorithm, the robot is instructed to follow a preference of directions.

It's Computational Thinking:
Algorithms – following
Algorithms – describing

In computer programming, a loop is a sequence of instruction's that is continually repeated until a certain condition is reached.

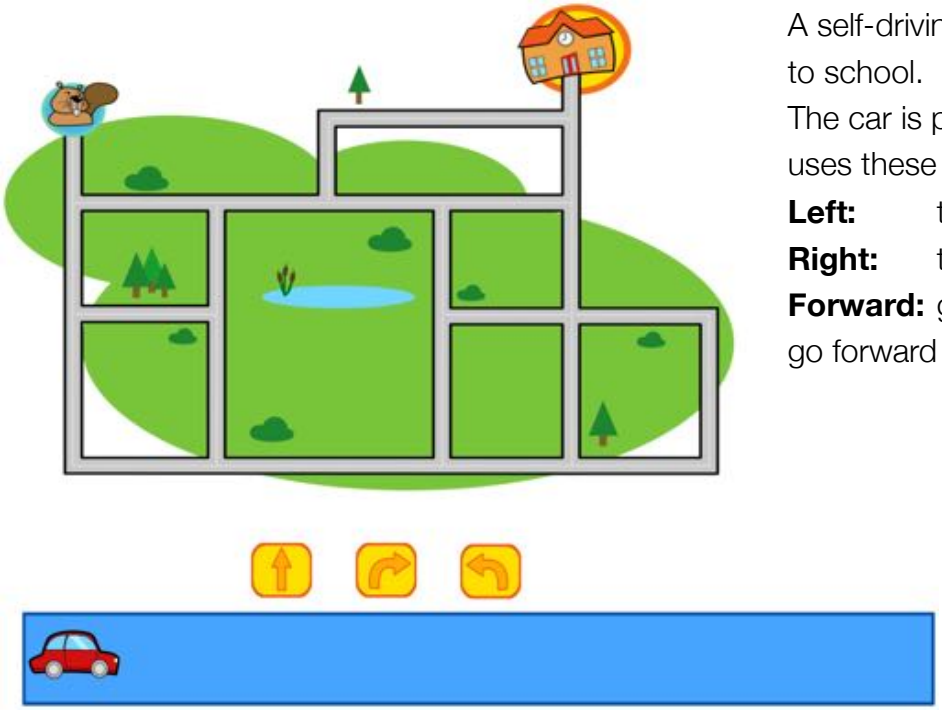
Typically, a certain process is done, such as getting an item of data and changing it, and then some condition is checked such as whether a counter has reached a prescribed number.

If it hasn't, the next instruction in the sequence is an instruction to return to the first instruction in the sequence and repeat the sequence. If the condition has been reached, the next instruction "falls through" to the next sequential instruction or branches outside the loop.

A loop is a fundamental programming idea that is commonly used in writing programs. It is sometimes referred to as 'repetition' or 'iteration'.

Car Trip

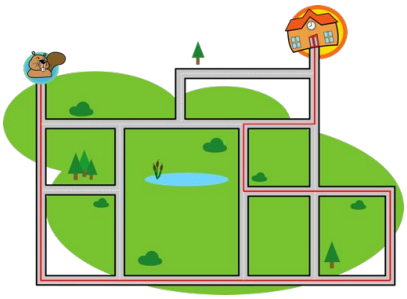
Years 3+4 B Years 7+8 A
Years 5+ 6 A Years 9+10
Years 11+12



A self-driving car needs to take a student to school.
The car is programmed so that it only uses these 3 instructions:

Left: turn 90° left
Right: turn 90° right
Forward: go forward until you cannot go forward anymore

Question:
Write a set of instructions (a program) that will get the beaver to his school. You can do this by dragging the three instruction blocks next to the car.



Answer:

Explanation:

The important thing for participants to remember is that there is no forward movement when turning 90 degrees, so the 'straight' command has to be entered between every turn.

It's Computational Thinking:

Breaking down problems into parts (Decomposition)
Algorithms – following
Algorithms – describing

The task is similar to giving instructions to a robot: writing a computer program requires step by step execution. Programs are essential to our use of computers: they tell computers what sequences of operations they must do. Computers and robots are good at computing fast, doing repetitive things, but they cannot think just by themselves, and require instructions to perform tasks. As shown in this task, the order of the operations is very important: the right set of instructions in the wrong order will not give the expected result.

Party Banner

Years 3+4 Years 5+ 6 | Years 7+8 Years 9+10 Years 11+12 | B

Beaver Bert has a long strip of coloured paper for a party.
The strip has three different colours (yellow, red, blue) in a regularly repeating pattern.
Bert's friend, James, has cut out a section of the paper, as shown in the diagram below.

James says that he will give back the missing piece of paper if Bert can correctly guess the size of the piece cut out.

Question:

How many coloured squares does the missing piece of paper have?

31 32 33 or 34

Answer:

31

Explanation:

We know the pattern ended with YRR, meaning that the James has cut out at least one B. After that, he cuts out some number of sequences of 4 (i.e., YRRB). After that, the right side of his piece of paper must have YR, since the second piece begins with RB. So, the length of his piece of paper is 1 (for B) + 4*X (where X is the number of repeated patterns YRRB) + 2 (for the YR). So, the length of her paper is 4X+3.

Looking at the possible answers, we see that 31/4 has remainder 3: that is, 31 = 4*7 + 3. So, our equation is solved when X=7. None of the other answers can be written as 4X+3.

It's Computational Thinking:

Abstraction
Breaking problems into parts (decomposition)
Data representation - symbolism
Data interpretation – patterns and contexts
Algorithms – following and describing

Finding a pattern in information is important for a variety of problems. For instance, sequences of DNA are composed of patterns, and finding repetitions or substrings that satisfy a certain property is an important research area in genetics and medicine. To solve these sorts of problems, we use text processing algorithms and pattern-matching programs to help determine whether certain strings appear in a sequence of text.

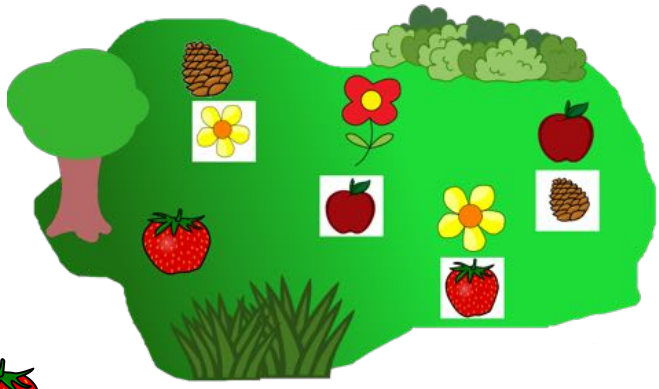
This problem also involves some abstraction, data interpretation and algorithm decomposition: We take a sequence of information and represent it as a formula or equation which we can solve. In order for computer scientists to solve problems, they need to take an explanation and convert it into something more concrete, formalised and mathematical in order to write a program to solve it.

Secret Recipe

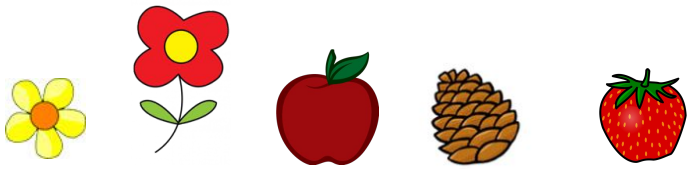
Years 3+4 C Years 7+8
Years 5+ 6 B Years 9+10
Years 11+12

Esther has asked Ivan to cook a special cake made of five ingredients.
She has put labels next to the ingredients in the garden. One ingredient has no label.
The labels tell Ivan in which ingredient must be added next in the sequence.

The garden looks like this:



Question:
Which ingredient should be added first?



Answer:

Explanation:

If Esther starts with the flower, she can add all five ingredients in the right order. The first added ingredient must be the one with no referring image.

Choosing the strawberry, she could not have continued to the next as there is no paper with it. The apple is not correct because if she had started with the apple, she would have skipped the red flower. The pine cone is not correct because if she had started with the cone, she would have skipped the red flower and the apple.

It's Computational Thinking:

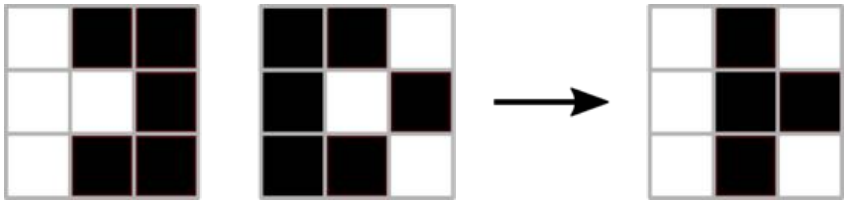
Data representation - symbolism
Breaking down problems into parts (Decomposition)
Data interpretation – patterns
Algorithms – following

The data structure used here is called a linked list in which there can be an arbitrary number of items. A linked list is a linear collection of data elements that consist of an item and a reference point (pointer) showing the next item. The first item of the linked list is very important as the list starts from there and it is the only point that refers to the whole list.

The recipe here is a linked list. The ingredients are the items and each label is the pointer to the next item in line. In other words, the plants are the data and the labels are the pointers. The first component is that ingredient which is not referred by any label, but accompanied by a label.

The benefit of the linked list is that items of different types and sizes can be stored together, just like fruits and flowers in this question.

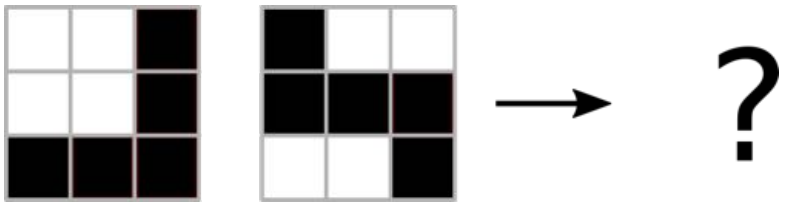
Combining Card A and Card B, you get Card C:



Card A Card B Card C

Question:

How many black cells will Card F have after combining Card D and Card E?

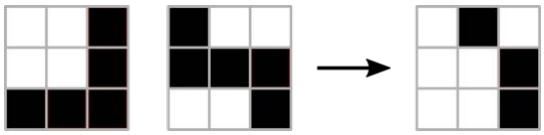


Answer:

3

Explanation:

Combining the cards obeys the following rule. When the colour of the corresponding cells is the same the resulting colour is black. Otherwise the resulting colour is white.



It's Computational Thinking:

Abstraction
Data interpretation – patterns
Algorithms – following
Digital Systems

A Boolean circuit is an example of a mathematical computation model. An equivalence is one of the basic Boolean operations. If the white cell is interpreted as 0 or FALSE and the black cell as 1 or TRUE, this operation could be described this way:

1 ⇔ 1 → 1
0 ⇔ 1 → 0
1 ⇔ 0 → 0
0 ⇔ 0 → 1