

## Edição 2023

### Categoria

**Benjamins** (5º e 6º ano de escolaridade)

### Tempo

45 minutos

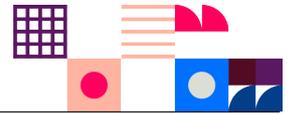
Resolve tantos problemas quanto possível em 45 minutos.

Não é esperado que consigas resolver todos!

Responde apenas na folha de respostas.

É uma folha única, à parte, que deves identificar com o teu nome.

Os enunciados e folhas de rascunho devem ser obrigatoriamente recolhidos no final da prova.



O **Bebras** é uma iniciativa internacional destinada a promover o pensamento computacional e a Informática (Ciência de Computadores). Foi desenhado para motivar alunos de todas as idades mesmo que não tenham experiência prévia.

Esta iniciativa começou em 2004 na Lituânia e todos os anos participam mais de 3 milhões de aluno de todo o mundo. O seu nome original vem dessa origem - "bebras" significa "castor" em lituano. A comunidade internacional adotou esse nome, porque os castores buscam a perfeição no seu dia-a-dia e são conhecidos por serem muito trabalhadores e inteligentes.

## O que é o Pensamento Computacional?

O pensamento computacional é um conjunto de técnicas de resolução de problemas que envolve a maneira de expressar um problema e a sua solução de modo a que um computador (seja um humano ou máquina) a possa executar. É muito mais do que simplesmente saber programar. O desafio do Bebras promove precisamente este tipo de habilidades e conceitos como a capacidade de partir um problema complexo em problemas mais simples, o desenho de algoritmos, o reconhecimento de padrões ou a capacidade de generalizar e abstrair.

## Organização Portuguesa

O Bebras começou em **Portugal** em 2019 e ano passado contou com a participação de mais de 70 mil estudantes de cerca de 500 escolas de todo o país.

É organizado por uma equipa de pessoas ligadas à Educação e à Ciência de Computadores da **TreeTree2** e do Departamento de Ciência de Computadores da Faculdade de Ciências da Universidade do Porto (**DCC/FCUP**)

## Estrutura da Prova

Existe apenas uma fase a nível nacional, a qual é constituída por uma prova individual com 12 questões de três níveis de dificuldade diferentes, cuja pontuação é da seguinte forma:

Dificuldade	Correto	Incorreto	Não respondido
fácil	+6 pontos	-2 pontos	0 pontos
média	+9 pontos	-3 pontos	0 pontos
difícil	+12 pontos	-4 pontos	0 pontos

## Sobre os Problemas



CC BY-NC-SA 4.0

Os problemas aqui colocados foram criados pela comunidade internacional da iniciativa Bebras e estão protegidos por uma licença da Creative Commons Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional.

Os problemas da edição portuguesa foram escolhidos, traduzidos e adaptados pela organização portuguesa. Para a deste ano foram usados problemas com autores originários dos seguintes países:

 - Arábia Saudita	 - Canadá	 - Chéquia	 - China	 - Eslováquia
 - Estados Unidos	 - Filipinas	 - Hungria	 - Índia	 - Irlanda
 - Itália	 - Japão	 - Lituânia	 - Nova Zelândia	 - Paquistão
 - Perú	 - Portugal	 - Suiça	 - Taiwan	 - Turquia
 - Uruguai	 - Vietname			



# 1. Máscaras de Carnaval

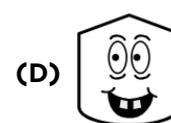
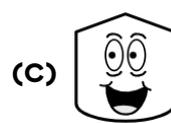
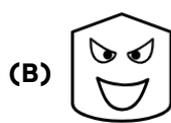
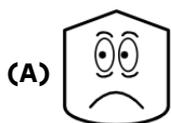
Uma máquina de fazer máscaras utiliza olhos e bocas diferentes para criar novas combinações. No quadro seguinte, apresentam-se exemplos de algumas das máscaras criadas:

				
				
				
	?			
				

## Pergunta

Que máscara será criada no quadrado azul com o ponto de interrogação?

## Respostas possíveis



# 1. Máscaras de Carnaval (Resolução)

## Solução

(D)

## Resolução

Nesta tabela de dupla entrada cada máscara apresenta a boca da sua linha e os olhos da sua coluna. Isto apresenta uma forma de organizar elementos de acordo com as suas características partilhadas e de trabalhar com eles para criar novas combinações. Desse modo, a solução só pode ser a (D) por corresponder à boca e olhos necessários.

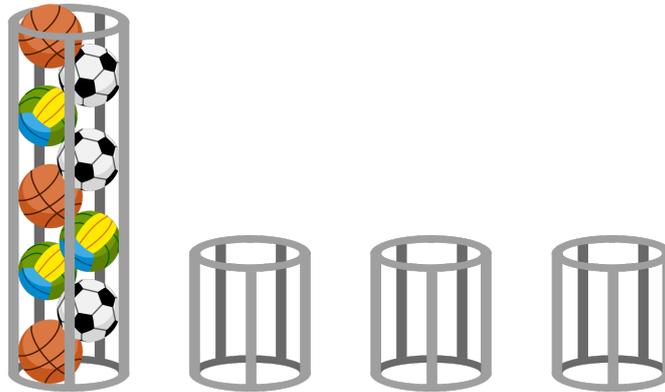
				
				
				
				
				

## Isto é Pensamento Computacional!

A conceção de *sistemas de informação* envolve a extração de propriedades que compõem todo o sistema a partir de um problema do mundo real. Neste problema, as características faciais, como os olhos e a boca, são seleccionadas como propriedades e armazenadas na base de dados. Estas propriedades, por vezes designadas por dimensões, permitem a produção de diversas faces através da sua combinação. Neste problema, a base de dados bidimensional é representada como uma tabela, organizada em linhas e colunas. Em cenários mais complexos do mundo real, particularmente no armazenamento de dados e no processamento analítico em linha, é construída e utilizada uma base de dados multidimensional para gerir e analisar dados de forma eficaz.

## 2. Organizar Bolas

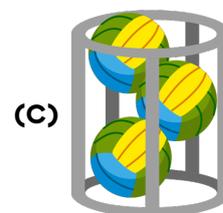
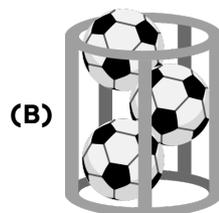
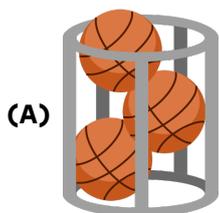
A Sara tem 9 bolas de 3 estilos diferentes num grande recipiente e quer organizar cada estilo de bolas numa recipiente separado. Para as dividir, a Sara retira as bolas uma a uma do recipiente (a partir de cima) e coloca-as no seu próprio recipiente.



### Pergunta

Com base na imagem acima, que recipiente de bolas é que a Sara vai encher primeiro?

### Respostas possíveis



## 2. Organizar Bolas (Resolução)

### Solução

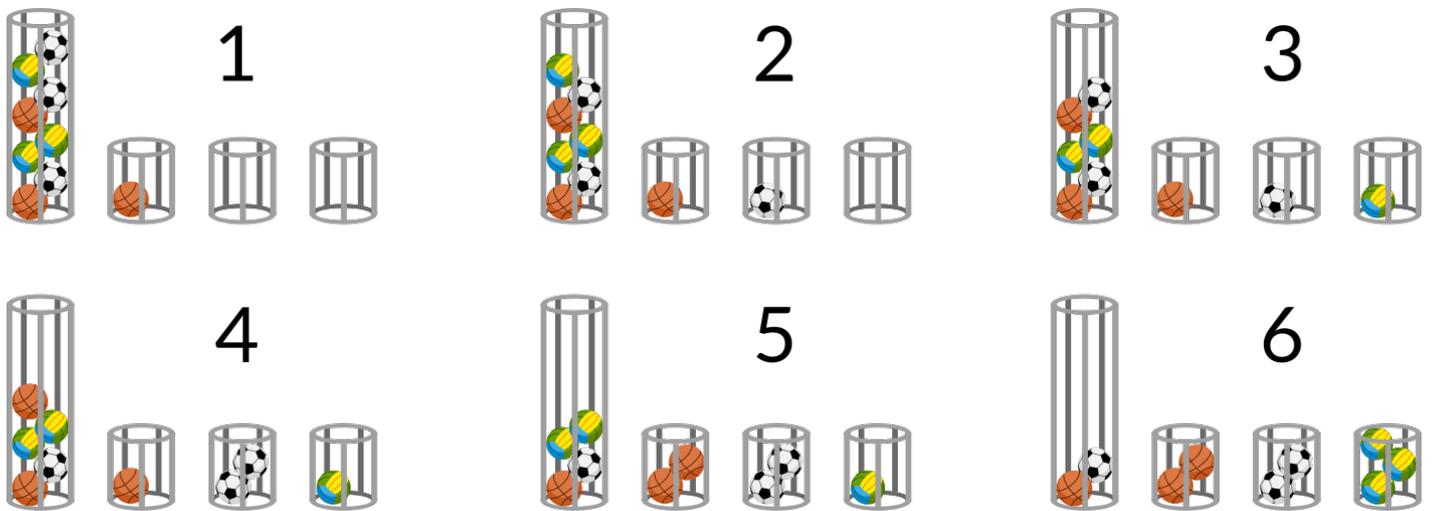
(C)

### Resolução

A resposta correcta é (C).

Como a Sara pega nas bolas uma a uma, pega nelas por ordem, de cima para baixo. Ela vai encher o primeiro recipiente de bolas quando apanhar a terceira bola desse estilo.

A terceira bola que sairá primeiro do tubo é a bola de voleibol, ou seja, a opção (C).



### Isto é Pensamento Computacional!

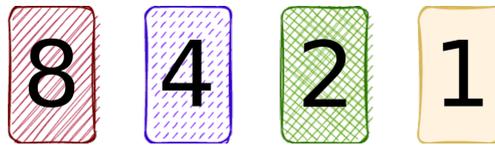
A Sara retira as bolas do topo do recipiente. Em Ciência de Computadores, esta estrutura de dados é designada por *pilha*. A pilha é uma estrutura de dados utilizada para armazenar dados, adicionando ou retirando elementos de uma extremidade chamada topo da pilha.

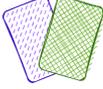
Existem muitos exemplos reais de uma pilha, como uma pilha de pratos ou de livros, em que se pode retirar primeiro o elemento do topo e o elemento do fundo será o último a ser retirado.

Para retirar as bolas e saber qual é o primeiro recipiente a ficar cheio, é necessário utilizar uma sequência de operações que mediante a bola no topo escolhe um dos recipientes. Esta sequência de operações é conhecida como um *algoritmo*, ou seja, uma forma sistemática de executar passos lógicos que podem ser repetidos para resolver o problema original e problemas semelhantes.

### 3. Idades Codificadas

A Beatriz mostra ao seu irmão Simão como usar as cartas abaixo para mostrar as suas idades.



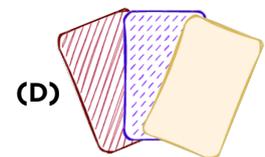
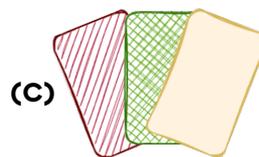
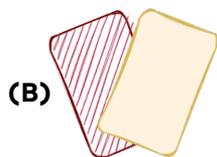
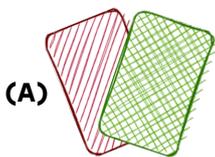
A Beatriz tem **6** anos, portanto ela usa as cartas  ( $4 + 2 = 6$ ).

O Simão tem **7** anos, portanto ele usa as cartas  ( $4 + 2 + 1 = 7$ ).

#### Pergunta

A irmã mais velha Carla tem **9** anos. Que cartas é que ela vai usar?

#### Respostas possíveis



### 3. Idades Codificadas (Resolução)

---

#### Solução

(B)

#### Resolução

A resposta correcta é a opção B ( $8 + 1 = 9$ ).

A opção A está incorrecta porque representa a idade de 10 anos ( $8 + 2 = 10$ ).

A opção C está incorrecta porque representa a idade de 11 anos ( $8 + 2 + 1 = 11$ ).

A opção D está incorrecta porque representa a idade de 13 anos ( $8 + 4 + 1 = 13$ ).

#### Isto é Pensamento Computacional!

A utilização de diferentes *representações para os números* é comum em Ciência de Computadores. Num único computador podem existir muitas representações diferentes para os números, todas a serem utilizadas ao mesmo tempo. Por exemplo, as idades (números inteiros), as fracções (valores reais) e as datas teriam todas regras de representação diferentes. Compreender a representação dos números neste problema é um passo para compreender a notação posicional dos números (decimal ou binária, por exemplo) que torna os cálculos numéricos (por exemplo, adição, multiplicação) muito eficientes com lápis e papel e também num computador. As grandes organizações internacionais, como a ISO e o IEEE, fazem um grande esforço para chegar a acordo sobre as representações que devem ser usados para diferentes tipos de números (os chamados formatos de números) nos computadores.

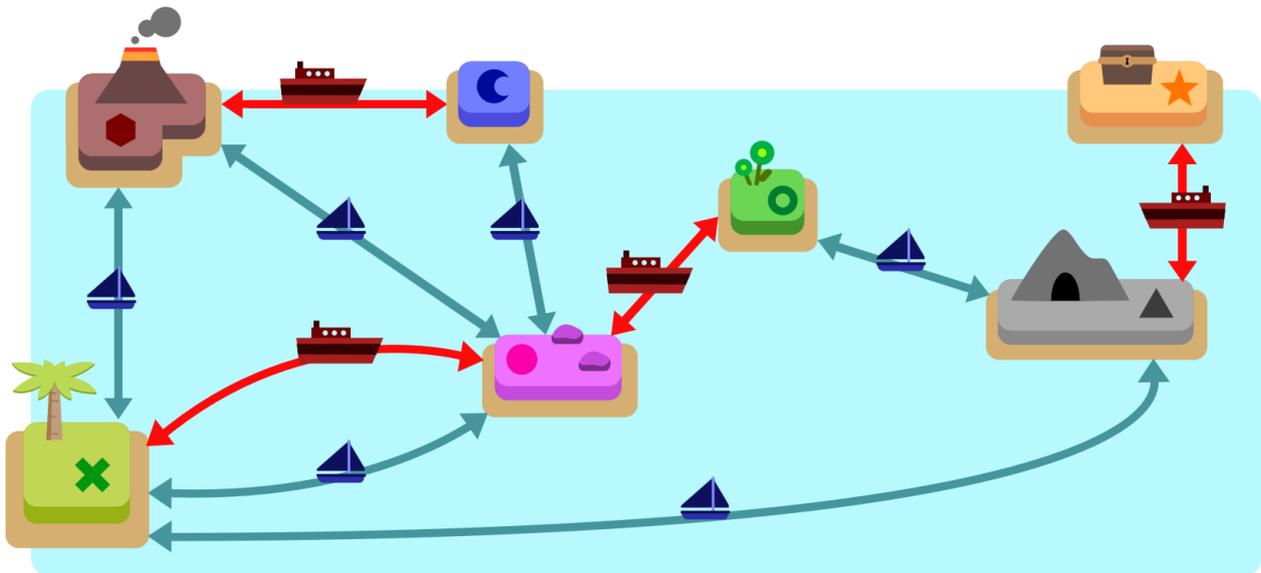
Inspirados no *sistema numérico binário*, os números nos cartões deste problema são cuidadosamente escolhidos para serem potências de 2, de modo a que apenas seja necessário um cartão de cada cor para representar números entre 0 e 15. Em comparação com uma notação posicional como a binária, a representação de números baseada na cor dos cartões, é mais flexível, na medida em que a ordem dos cartões não é importante. No entanto, tem duas desvantagens principais que a tornam pouco prática. Em primeiro lugar, os humanos e os computadores terão dificuldade em distinguir entre as diferentes cores se tentarmos representar números muito grandes. Em segundo lugar, para efetuar cálculos numéricos eficientes, teríamos de converter as cores numa representação diferente.



## 4. Ilha do Tesouro

O Bruno está a passar férias no Reino das Ilhas. Há dois tipos de navios que navegam entre as ilhas:  e .

Podes ver as suas rotas na imagem. Quando as pessoas visitam as ilhas, escolhem um navio para ir para a ilha seguinte.



### Pergunta

Qual das seguintes viagens com início na ilha  não pode fazer com que o Bruno chegue à ilha .

### Respostas possíveis

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



## 4. Ilha do Tesouro (Resolução)

### Solução

(D)

### Resolução

Assumimos que o Bruno parte da ilha .

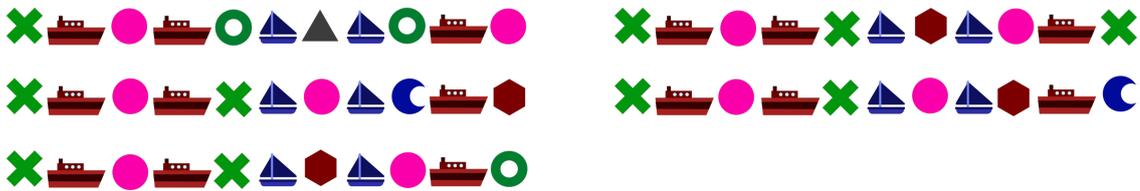
Um itinerário possível para a opção (A) é: 

Um itinerário possível para a opção (B) é: 

Um itinerário possível para a opção (C) é: 

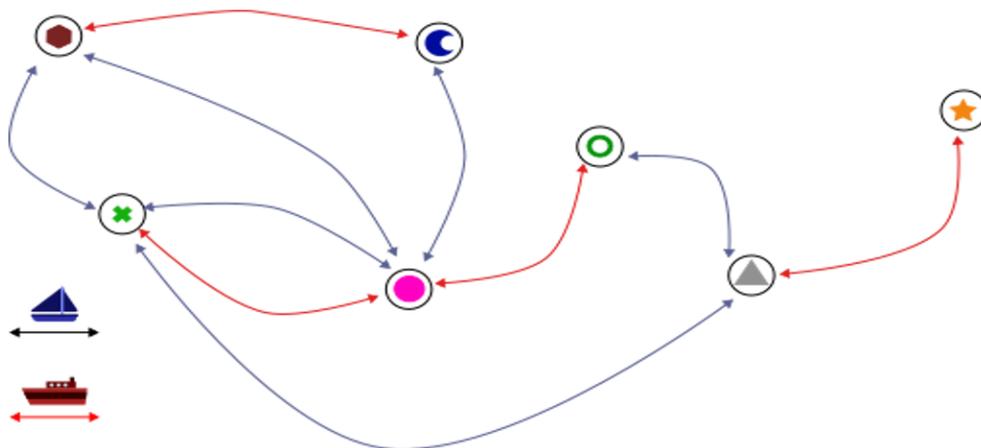
Apenas com a opção (D) é impossível chegar à ilha .

De facto, é possível chegar a todas as ilhas com exceção de  ou . Por exemplo:



### Isto é Pensamento Computacional!

Para resolver problemas utilizando computadores, precisamos primeiro de representar os problemas de uma forma que os computadores possam compreender. Neste caso específico, estamos a lidar com redes de transporte, que podem ser representadas como um *grafo*. Um grafo é um tipo de estrutura de dados que mostra como coisas diferentes estão ligadas umas às outras. Ua outra maneira mais simples de desenhar o grafo deste problema poderia ser:



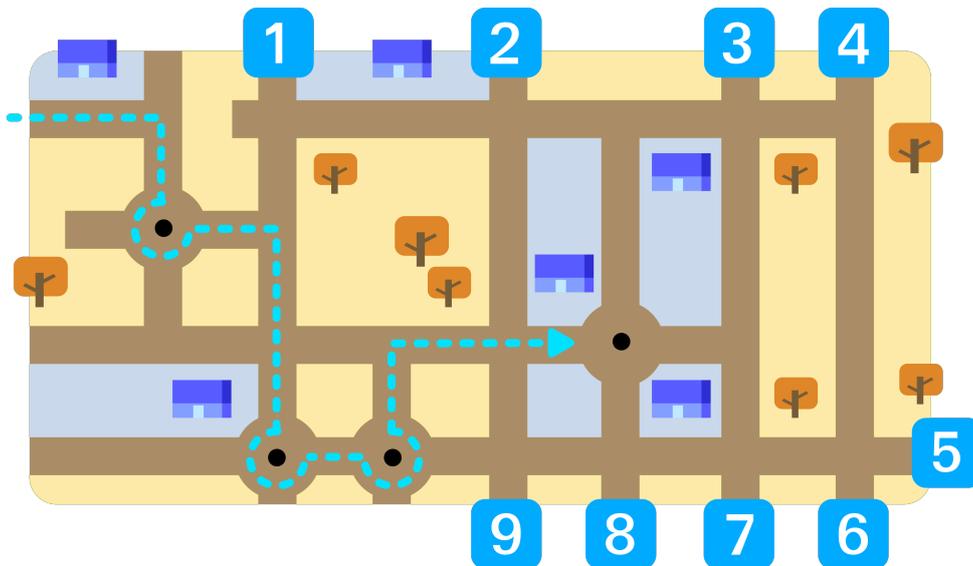
Ao utilizar um grafo, os computadores podem ajudar-nos a resolver vários problemas. Por exemplo, podem encontrar a rota mais económica entre dois locais, determinar o caminho mais curto, resolver o problema de uma rede de comunicações e muito mais. Neste problema, pede-se aos alunos que encontrem todas as rotas possíveis da ilha  para a ilha  utilizando a representação gráfica.



## 5. Carro Autónomo

O Filipe está a viajar no seu carro autónomo que anda de forma automática.

Sempre que o carro chega ao mesmo tipo de intersecção segue sempre o mesmo comportamento e usa sempre a mesma saída. Por exemplo, sempre que chega a uma rotunda ele sai na terceira saída possível.



### Pergunta

A qual número num quadrado chega o carro do Filipe no final da sua viagem?

### Respostas possíveis

- (A) 1    (B) 2    (C) 3    (D) 4    (E) 5    (F) 6    (G) 7    (H) 8    (I) 9



## 5. Carro Autónomo (Resolução)

### Solução

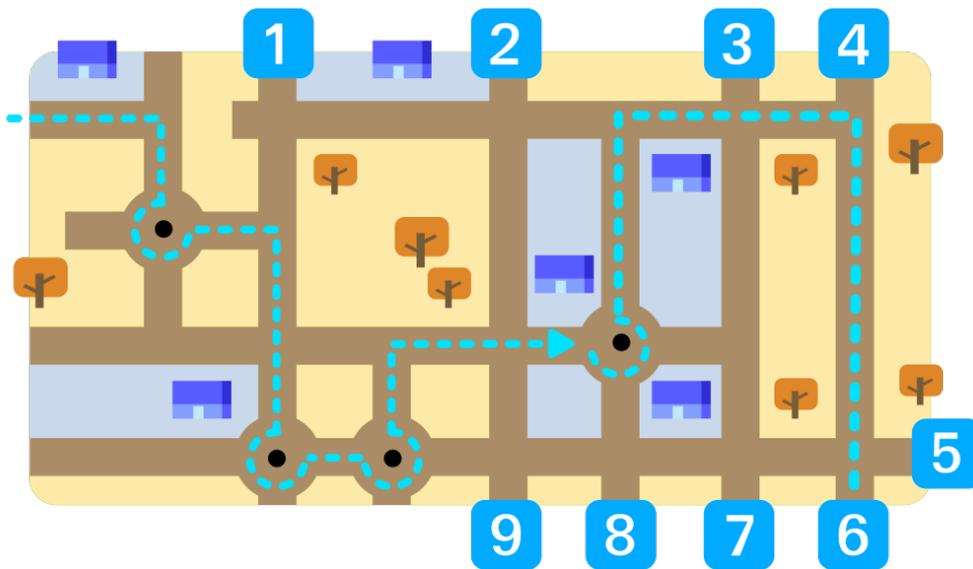
(F)

### Resolução

A partir da imagem, pode concluir-se que o sistema de navegação segue as seguintes regras para cada tipo de cruzamento:

- Num cruzamento em T, o carro vira sempre à direita.
- Numa rotunda, o carro sai sempre na terceira saída para direita
- Num cruzamento +, o carro segue sempre em frente.

O cruzamento seguinte é uma rotunda, pelo que o carro vai pela terceira saída, indo para cima. O cruzamento seguinte é um cruzamento em T, pelo que o carro vai para a direita. Segue em frente no cruzamento + e vira à direita no cruzamento cruzamento em T na parte superior do mapa. Finalmente, o carro passa em frente pelo cruzamento + e pára em 6. O mapa abaixo mostra este trajeto:



### Isto é Pensamento Computacional!

É uma situação típica em Ciência de Computadores a observação de um *sistema dinâmico* que é controlado por um *algoritmo* que segue determinadas regras. O comportamento futuro do sistema só pode ser previsto através de uma análise cuidadosa do algoritmo e da determinação das regras exactas que este segue. O sistema que é estudado pode ser um sistema da vida real ou um sistema de *software*, por exemplo, para efeitos de perceber possíveis erros de um programa.



## 6. Árvore Mágica

O castor David tem uma árvore mágica a crescer perto da sua casa.

Sempre que um pássaro pousa nela (  ), a árvore faz nascer 2 maçãs.

Sempre que um esquilo a trepa (  ), a árvore deixa cair 1 maçã (se tiver alguma).

Sempre que uma cobra visita a árvore (  ), todas as maçãs desaparecem instantaneamente!

Uma manhã, o David verifica que a árvore mágica contém **25** maçãs. O David passa então o resto do dia a fazer desenhos de todos os animais que vêm à árvore. Os desenhos, por ordem, são:



### Pergunta

Quantas maçãs tem a árvore no final do dia?

### Respostas possíveis

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



## 6. Árvore Mágica (Resolução)

---

### Solução

(B)

### Resolução

A resposta é a opção (B). Há 7 maçãs na árvore ao fim do dia.

Como todas as maçãs desaparecem instantaneamente sempre que uma cobra visita a árvore, podemos ignorar tudo o que acontece antes da chegada de uma cobra (). Depois da última cobra, quatro pássaros () pousam na árvore, o que significa que dela brotarão  $4 \times 2 = 8$  maçãs. Depois, um esquilo () sobe à árvore, o que faz cair uma maçã, deixando  $8 - 1 = 7$  maçãs.

### Isto é Pensamento Computacional!

Este problema introduz as ideias subjacentes a dois conceitos fundamentais de programação.

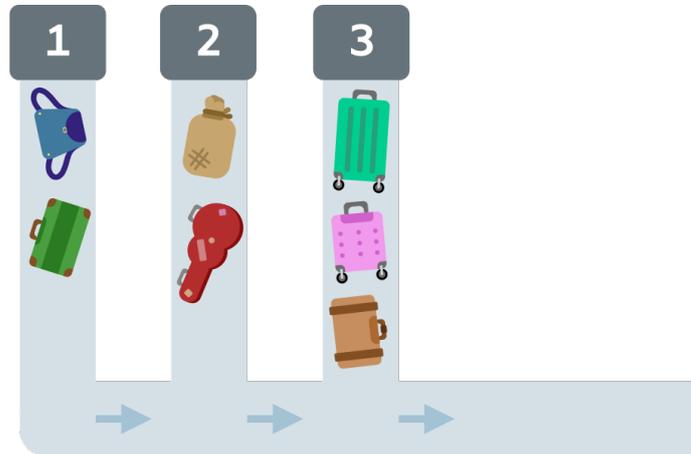
O primeiro é a ideia de uma *variável*. Uma variável é usada para armazenar informações que um programa de computador precisa. O valor de uma variável pode mudar consoante o resto das instruções do programa.

Neste problema, o número de maçãs na árvore é uma variável e o seu valor pode aumentar () , diminuir () ou reiniciar () .

Para decidir como alterar o valor de uma variável, um programa de computador precisa de ter a capacidade de tomar decisões. Este é o segundo conceito fundamental de programação e é designado por *seleção*. A tomada de decisões faz-se utilizando instruções especiais chamadas instruções condicionais que permitem seleccionar entre diferentes resultados possíveis. Normalmente, estas instruções assumem a forma "se isto, então aquilo". Neste problema, uma expressão condicional seria "se um pássaro pousar na árvore, então aumenta o número de maçãs em 2". Consegues encontrar outras expressões condicionais neste problema?

## 7. Check-in no Aeroporto

No aeroporto de Castorlândia, os passageiros podem deixar as suas malas em qualquer um dos 3 balcões de check-in mostrados abaixo.



Os operadores dos balcões colocam as malas, uma de cada vez, na passadeira rolante vertical. À medida que cada nova mala é colocada, o operador carrega num botão para fazer avançar a mala. Quando a mala atinge a passadeira horizontal, é automaticamente transferida para a mesma.

As malas que já estão nas passadeiras verticais são mostradas na imagem acima.

### Pergunta

Qual das opções mostra uma possível ordem das malas na passadeira horizontal?

### Respostas possíveis

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

## 7. Check-in no Aeroporto (Resolução)

### Solução

(A)

### Resolução

O que sabemos é que a ordem relativa das malas na passadeira horizontal tem de respeitar a ordem relativa em que estavam no mesmo balcão, ou sejam, mantêm a ordem entre si que tinham quando estava na mesma passadeira vertical.

A resposta correta é (A). A primeira mala é  do balcão 2, a segunda mala é  do balcão 1, a terceira mala é  do balcão 1, a quarta mala é  do balcão 3, a quinta mala é  do balcão 3, a sexta mala é  do balcão 2 e a sétima mala é  do balcão 3.

A seguir exemplificamos uma inconsistência em cada uma das hipóteses incorretas.

(B) está errada porque  deveria vir antes de  para respeitar a ordem em que estavam no balcão 1.

(C) está errada porque  deveria vir antes de  para respeitar a ordem em que estavam no balcão 2.

(D) está errada porque  deveria vir antes de  para respeitar a ordem em que estavam no balcão 3.

### Isto é Pensamento Computacional!

Um *fila* é uma estrutura de dados utilizada para representar informações por ordem sequencial, sendo que primeiro elemento a entrar numa fila é sempre o primeiro a sair dela.

Neste problema, temos diferentes filas iniciais (representadas pelas passadeiras verticais) e, em todas elas, as malas são os seus elementos. Assim, para todos os sacos das filas verticais, essa propriedade da primeira mala a entrar ser a primeira a sair deve ser mantida.

A passadeira horizontal pode ser entendida como um programador de um sistema operativo, que gere os pedidos de vários utilizadores (as passadeiras verticais). Consequentemente, há muitas possibilidades de retirar elementos das filas e, por isso, há muitas seqüências possíveis de malas na passadeira horizontal.

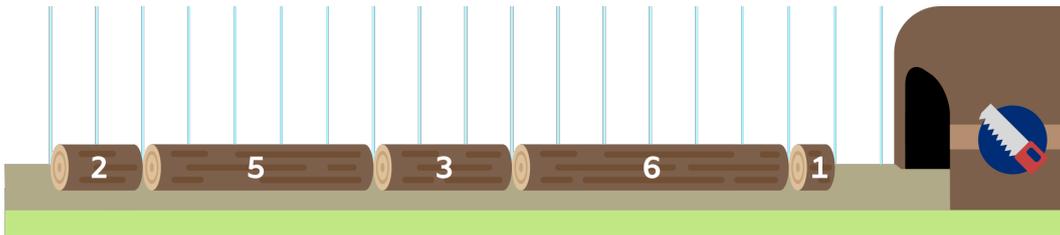


## 8. Armazenamento de Troncos

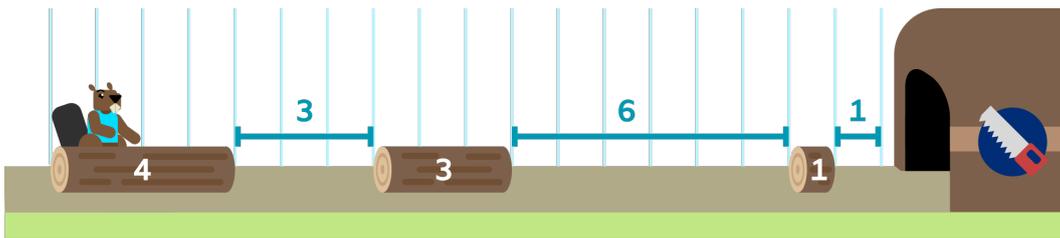
A castora Joana corta troncos de madeira de vários comprimentos e depois vende-os. Sempre que acaba de cortar um tronco, coloca-o no chão ao longo da estrada estreita de 18 metros, um a seguir ao outro, porque os troncos não cabem lado a lado.

Quando a Joana coloca um tronco no chão, coloca-o no primeiro lugar disponível a partir da esquerda onde o tronco cabe. Quando vende um determinado tronco, este é simplesmente retirado do lugar que ocupava anteriormente.

A Joana preparou, por esta ordem, troncos com comprimentos 2, 5, 3, 6 e 1 metros. Isto leva a esta disposição ao longo da estrada:



A seguir, ela vende os troncos com tamanhos de 6, 2 e 5 metros. De seguida, corta um novo tronco de 4 metros. Como o coloca o mais à esquerda possível, a estrada fica com este aspeto:



### Pergunta

A Joana tem de cortar troncos com comprimentos de 1, 2, 3 e 4 metros. Qual das seguintes ordens lhe permitirá armazená-los todos na estrada, desde que utilize a mesma regra?

### Respostas possíveis

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

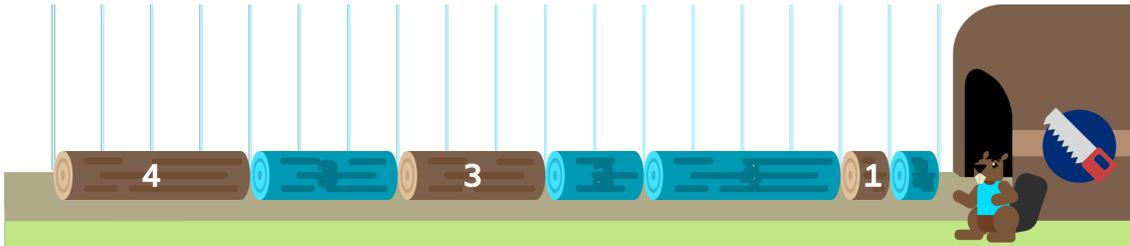


## 8. Armazenamento de Troncos (Resolução)

### Solução

(C)

### Resolução



A resposta correcta é (C). Todas as outras respostas levam a que a Joana não consiga colocar o último tronco.

O estado da estrada antes de a Joana cortar os novos troncos, como se mostra na pergunta, tem três lugares livres: o lugar mais à esquerda tem 3 metros de comprimento, o lugar do meio tem 6 metros de comprimento e o último, à direita, tem 1 metro de comprimento.

A resposta (C) preenche totalmente o lugar mais à esquerda com o primeiro tronco, depois preenche totalmente o lugar do meio com os dois troncos seguintes e o último tronco entra diretamente no lugar mais à direita: funciona.

A resposta (A) preenche totalmente o lugar mais à esquerda com os dois primeiros troncos, mas depois coloca o tronco de 3 metros de comprimento no lugar do meio. Agora, o lugar do meio ficou reduzido a apenas 3 metros, e o último tronco, que tem 4 metros de comprimento, não cabe nem aí nem no lugar mais à direita.

A resposta (B) coloca o tronco de 1 metro de comprimento no lugar mais à esquerda e, a seguir, o tronco de 4 metros no lugar do meio. O lugar mais à esquerda ainda pode ser completamente preenchido com o terceiro tronco, que tem 2 metros de comprimento, mas não há espaço contíguo suficiente para colocar o último tronco.

A resposta (D) tem um problema semelhante, mas ainda mais cedo: o terceiro tronco de 4 metros já não pode ser colocado. De qualquer modo, os lugares mais à esquerda e mais à direita são demasiado pequenos e o lugar do meio está meio ocupado pelo segundo tronco de 3 metros.

Na resposta (E) é o último tronco que não consegue ser colocado. Consegues ver porquê?

### Isto é Pensamento Computacional!

Podemos ver a estrada como uma memória dentro de um computador, e os registos a serem guardados e, mais tarde, vendidos, como processos informáticos que requerem uma certa quantidade de memória para funcionar. Neste exemplo, embora haja espaço suficiente para colocar teoricamente todos os registos, a ordem (e a estratégia) pela qual eles entram e saem torna por vezes impossível encaixá-los todos. O mesmo pode acontecer nos computadores: dependendo da ordem pela qual os pedaços de memória são atribuídos e libertados, pode ser impossível atribuir um novo pedaço de memória de um determinado tamanho, mesmo quando o total de memória livre é superior a esse tamanho.

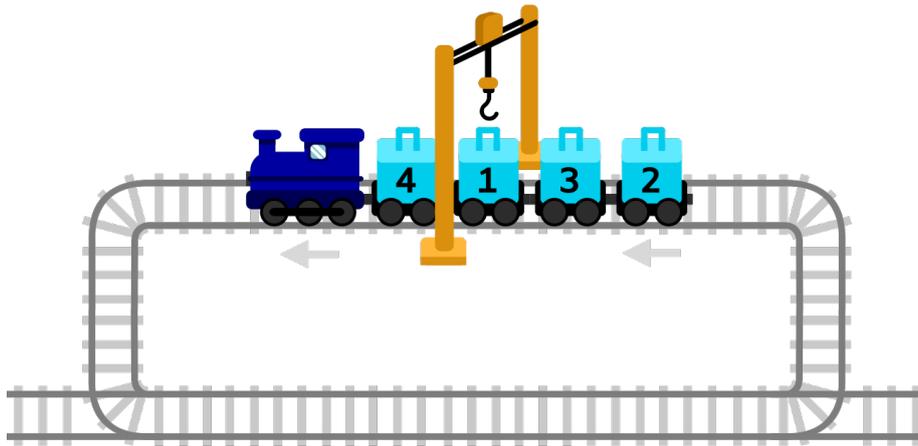
Este problema é conhecido como *fragmentação de memória*. As estratégias de alocação mais inteligentes do que a da Joana ou os modelos de memória que permitem, por exemplo, a deslocação de blocos, ajudam a atenuar o problema da fragmentação. A fragmentação não é apenas um problema com a memória: a atribuição de espaço num disco para escrever ficheiros de vários tamanhos leva a problemas muito semelhantes.



## 9. Descargas

Um comboio de mercadorias tem várias carruagens, cada um com uma caixa numerada. Uma única grua é utilizada para descarregar. A grua está numa posição fixa. Para descarregar uma caixa, esta tem de ser posicionada diretamente por baixo da grua.

As caixas têm de ser descarregadas por ordem crescente a partir da caixa 1. O comboio só pode deslocar-se para a frente. Está numa via circular, pelo que pode dar a volta à via e regressar para que mais caixas possam ser descarregadas pela grua.



No exemplo acima, as caixas têm de ser descarregadas na sequência 1, 2, 3, 4.

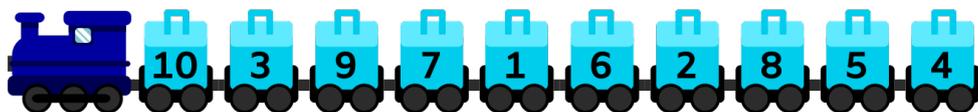
Na primeira volta de descarga, o comboio salta a caixa 4, descarrega a caixa 1, salta a caixa 3 e descarrega a caixa 2.

Na segunda volta, salta a caixa 4 e descarrega a caixa 3.

O comboio tem de voltar para uma terceira volta e descarrega a última caixa, a 4.

### Pergunta

Quantas voltas serão necessárias para descarregar todas as caixas do comboio seguinte?



### Respostas possíveis

- (A) 4      (B) 5      (C) 6      (D) 7      (E) 8      (F) 9      (G) 10



## 9. Descargas (Resolução)

---

### Solução

(D)

### Resolução

A ordem necessária para descarregar é 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Se seguirmos o procedimento descrito acima, durante a primeira volta as carruagens 1 e 2 serão descarregadas em conjunto, depois 3 e 4 em conjunto, depois 5, depois 6, depois 7 e 8 em conjunto, depois 9 e finalmente 10. Isto corresponde a 7 voltas.

Em alternativa, pode observar-se o princípio geral de que, para cada número desta sequência 1, 2, ..., se o número seguinte surgir à sua esquerda no comboio, é necessária uma volta adicional. No caso da posição, se o 3 aparecer à esquerda do 2, então o 3 será ignorado para descarregar o 2, pelo que é necessária uma volta extra para colocar o 3 debaixo da grua. No desafio dado, o número de pares que estão fora de ordem é (2,3), (4,5), (5,6), (6,7), (8,9) e (9, 10), pelo que são necessárias 6 voltas extra, num total de 7 voltas.

### Isto é Pensamento Computacional!

Para qualquer número da carruagem, se o número maior seguinte estiver à sua esquerda no comboio, chamamos a isto uma "inversão". Para cada inversão, é necessária uma volta extra. Se contarmos o número de inversões, obtemos a resposta.

A contagem das inversões em relação a uma sequência desejada tem muitas aplicações. Para alguns algoritmos de ordenação, como o *bubble sort*, o número de inversões indica-nos quantas trocas são necessárias para ordenar uma dada sequência. Se dois clientes classificarem o mesmo conjunto de itens por ordem de preferência, o número de inversões nas suas classificações diz-nos até que ponto os seus gostos estão alinhados. Isto pode ser utilizado por exemplo por lojas *online* para identificar clientes "semelhantes" e fazer recomendações de produtos.

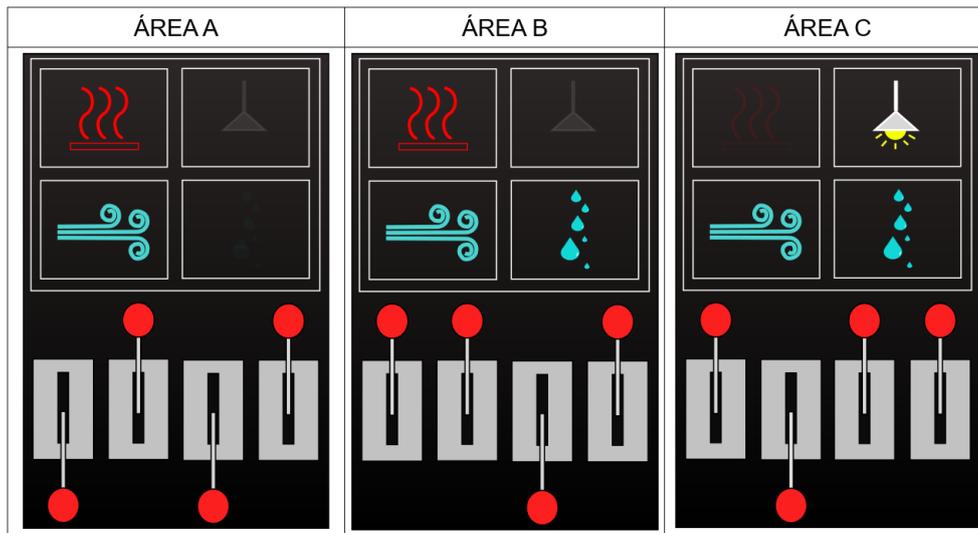


## 10. Alavancas

A Estação Espacial tem 3 diferentes áreas, todas com o mesmo painel de controlo com 4 alavancas para controlar os sistemas de calor, ventilação, luz e humidade. Todas as alavancas de cada painel funcionam da mesma forma e estão na mesma ordem.

Infelizmente alguém se esqueceu de colocar etiquetas para saber qual alavanca controla qual sistema!

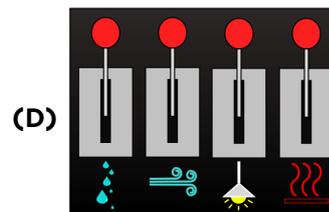
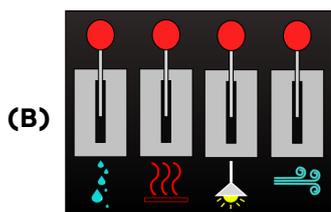
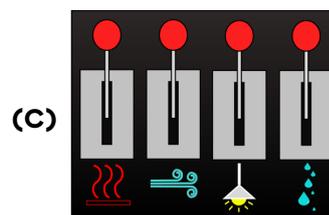
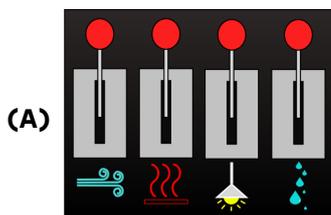
Felizmente, observando as posições atuais das alavancas e o estado de cada sistema, é possível descobrir qual alavanca controla qual sistema:



### Pergunta

Qual é o sistema (aquecimento, ventilação, luz ou humidade) que está a ser controlado por cada alavanca?

### Respostas possíveis



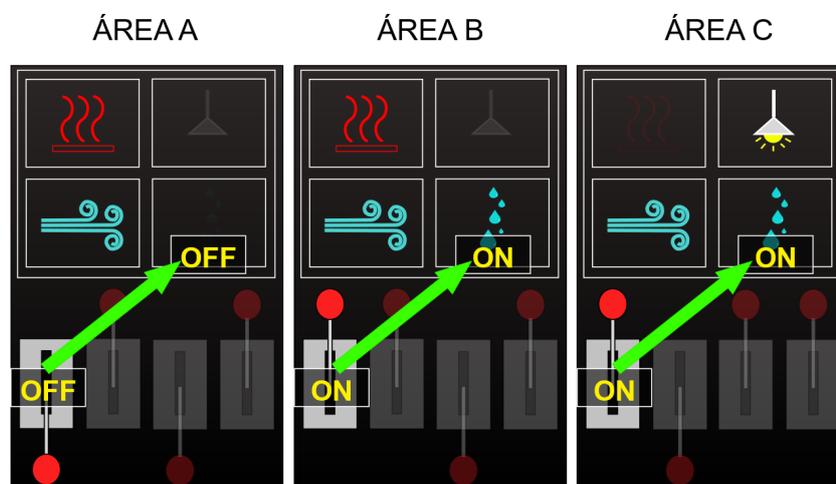
## 10. Alavancas (Resolução)

### Solução

(B)

### Resolução

A resposta correcta é a (B) (HUMIDADE / AQUECIMENTO / LUZ / VENTILAÇÃO). A forma mais fácil de detetar a função de cada alavanca é verificar o padrão de alteração de cada alavanca em cada posição e compará-lo com a alteração de cada sistema em todos os painéis. Em primeiro lugar, olhamos para as áreas A e B. As únicas diferenças entre elas são a humidade e a posição da 1ª alavanca. A partir daí, podemos deduzir que a alavanca posicionada para cima liga o sistema correspondente e que a 1ª alavanca controla a humidade.



As áreas A e B têm 2ª alavanca na posição LIGADO, mas está na posição DESLIGADO na área C. O sistema nos estados LIGADO-LIGADO-DESLIGADO é AQUECIMENTO. Portanto, a 2ª alavanca controla o AQUECIMENTO. Como LUZ está LIGADO apenas na área C, encontramos a alavanca que está na posição LIGADO apenas na Área C. Deduzimos que a 3ª alavanca controla a LUZ. Portanto, a 4ª alavanca controla a VENTILAÇÃO.

É muito importante avaliar sempre a posição das alavancas em todos os 3 painéis. Se avaliarmos apenas 2 painéis, as respostas erradas podem parecer as correctas por não termos informação suficiente.

A expressão "funcionam da mesma forma" ajuda a compreender que as posições LIGADO/DESLIGADO são as mesmas para todas as alavancas. Caso contrário, não é possível determinar o AQUECIMENTO e a LUZ.

### Isto é Pensamento Computacional!

Este problema pode ajudar-nos a ter uma ideia geral dos conceitos de *representação de dados e números binários*. Aqui, as posições das alavancas são basicamente binárias, no sentido em que cada alavanca só pode ter 2 posições. Podemos chamar a essas posições CIMA BAIXO, LIGADO e DESLIGADO, ou podemos representar esses estados com números, como, por exemplo, 1 e 0. O estado e os dados recolhidos em dispositivos digitais são representados através de números. Carrega-se num botão e pode aparecer um 0 na memória do dispositivo. Roda-se uma alavanca e um 1 ou um 0 é armazenado algures na memória do dispositivo.

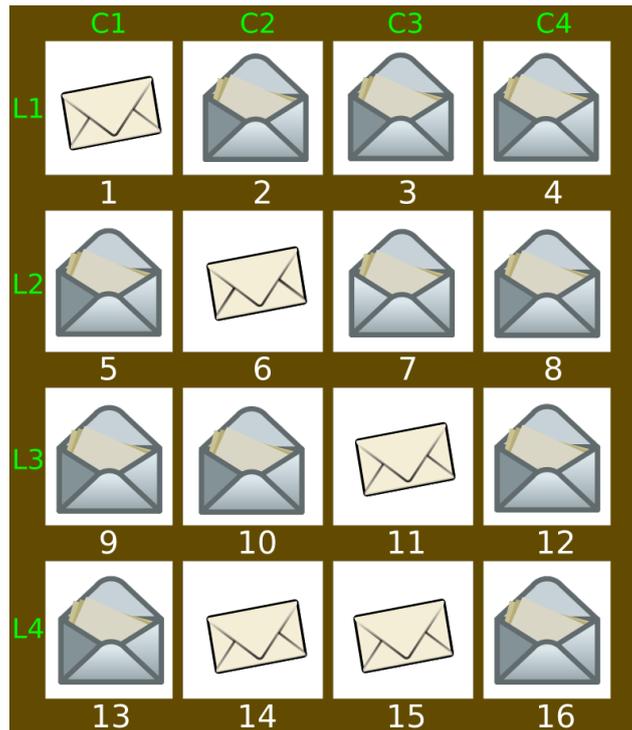
Mesmo quando os dados estão a ser transferidos entre dispositivos, todos estes 0s e 1s continuam presentes sob a forma de impulsos eléctricos. É assim que o seu o seu telemóvel envia comandos para os seus altifalantes sem fios para colocar a sua música favorita. São todos 1's e 0's sob a forma de electrões ou ondas de rádio, mas que representam todo o tipo de informação rica... como as posições das alavancas numa Estação Espacial!



## 11. Cartas Fechadas

A República dos Castores mantém um armário cheio de cartas secretas. Entre as 16 cartas desse armário, numeradas de 1 a 16, 10 tinham sido abertas, enquanto as outras 6 ainda estavam seladas dentro dos seus envelopes.

Uma noite, um espião inimigo entrou sorrateiramente e abriu **uma** das cartas seladas. No entanto, esqueceu-se de a selar novamente. Na manhã seguinte, a República dos Castores inicia uma investigação depois de constatar que existem agora 11 cartas abertas, como se pode ver abaixo:



O guarda não se lembra de todos os pormenores, mas tem a certeza de que, **antes** de o espião se ter infiltrado:

- O número de cartas abertas nas colunas C2 e C4 combinadas era par.
- O número de cartas abertas nas colunas C3 e C4 combinadas era par.
- O número de cartas abertas nas linhas L2 e L4 combinadas era par.
- O número de cartas abertas nas linhas L3 e L4 combinadas era par.

### Pergunta

Qual das cartas foi aberta pelo espião inimigo?

### Respostas possíveis

- (A) 5                      (B) 9                      (C) 10                      (D) 13
- (E) Não existem informações suficientes para identificar a carta aberta



## 11. Cartas Fechadas (Resolução)

---

### Solução

(D)

### Resolução

A resposta correcta é 13 cartas e podemos seguir esta linha de raciocínio:

1. Há um número par de cartas abertas nas colunas C2 e C4 combinadas, o que corresponde à recordação do guarda. Como só há uma carta aberta pelo espião, isso implica que a carta aberta pelo espião deve estar na coluna C1 ou C3.
2. Há um número par de cartas abertas nas colunas C3 e C4 combinadas, o que corresponde à recordação do guarda. Dada a afirmação anterior, isto implica que a carta aberta pelo espião deve estar na coluna C1.
3. Há um número ímpar de cartas abertas nas linhas L2 e L4 combinadas, o que não corresponde à recordação do guarda. Isto implica que a carta aberta pelo espião deve estar na linha L2 ou L4.
4. Há um número ímpar de cartas abertas nas linhas L3 e L4 combinadas, o que não corresponde à recordação do guarda. Dada a afirmação anterior, isto implica que a carta aberta pelo espião deve estar na linha L4.

Portanto, a carta lida pelo espião está na coluna C1 e na linha L4, o que aponta para (D) 13.

### Isto é Pensamento Computacional!

Este desafio introduz o conceito de *códigos de correção de erros*. Os dados digitais são essencialmente uma sequência de *bits*: 1s e 0s. Quando passam por uma rede, podem ser corrompidos devido a ruído ou à ação de entidades maliciosas. Os dados armazenados em DVD também podem ser corrompidos se estes dispositivos de armazenamento forem riscados. Assim, é importante dispor de um esquema capaz não só de detetar a ocorrência de corrupção (*deteção de erros*), mas também de corrigir os *bits* corrompidos (*correção de erros*).

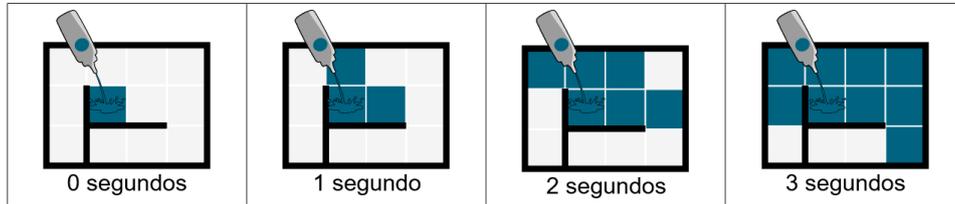
O primeiro código moderno de correção de erros foi introduzido por Richard Hamming em 1950. Este desafio inspira-se especificamente na ideia do *código de Hamming [15,11]*, em que as cartas fechadas e abertas correspondem a 0s e 1s, respetivamente. Este esquema de codificação pega num dado de 11 *bits* e insere 4 *bits* de paridade. Estes *bits* de paridade ocupam os lugares 2, 3, 5 e 9, enquanto os *bits* dos dados originais ocupam os restantes lugares (exceto o lugar 1). Os *bits* de paridade são definidos de modo a que haja um número par de 1s na 2ª e 4ª colunas, 3ª e 4ª colunas, 2ª e 4ª linhas, e 3ª e 4ª linhas. Finalmente, o *bit* no lugar 1 é definido de modo a que haja um número par de 1s no total.

Como se vê na solução apresentada, este esquema permite que o *bit* corrompido seja identificado e posteriormente corrigido, desde que haja no máximo 1 *bit* corrompido. Embora sejam necessários esquemas de correção de erros mais sofisticados para corrigir 2 ou mais *bits* corrompidos, a eficiência do código de Hamming contribui para ainda hoje em dia seja utilizado em sítios como memórias de computador.

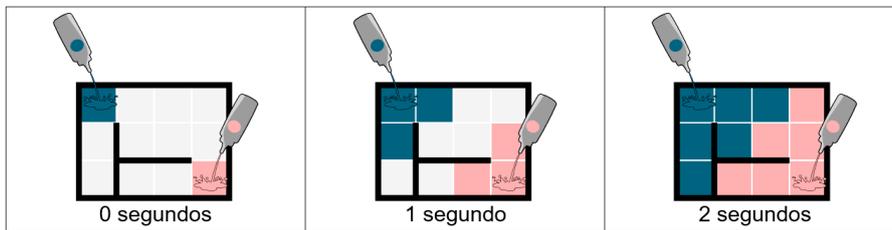


## 12. Aguarela

Quando os castores põem aguarela num labirinto, a cor espalha-se para os quadrados vizinhos a cada segundo. A cor não se espalha através das paredes, como podes ver abaixo.

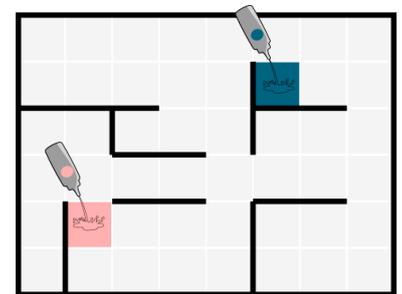


Se os castores puserem mais do que uma aguarela no labirinto, a primeira cor que chegar ao quadrado vai preenchê-lo na totalidade. Quando as cores chegam a um quadrado ao mesmo tempo, o quadrado fica da cor mais escura.

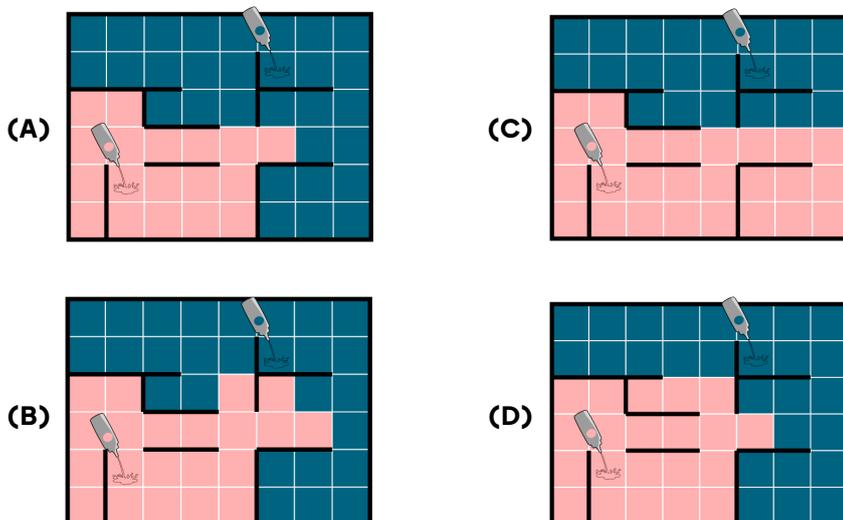


### Pergunta

Os castores puseram duas cores no labirinto mostrado na imagem à direita. Qual será o aspeto do labirinto quando todos os quadrados estiverem preenchidos com uma cor?



### Respostas possíveis



## 12. Aguarela (Resolução)

### Solução

(A)

### Resolução

A resposta correta é a (A). A imagem abaixo mostra o estado do labirinto a cada segundo:



### Isto é Pensamento Computacional!

A configuração do problema assemelha-se a uma matriz bidimensional, o que basicamente significa uma tabela com linhas e colunas. Esta forma de representar os dados é bastante funcional para simular o estado do labirinto segundo a segundo e resolver o problema.

No entanto, também é possível representá-lo como um *grafo* em que cada quadrado está ligado aos seus vizinhos. Este grafo permite-nos identificar rapidamente que cor chegará a um quadrado sem simular todo o cenário segundo a segundo. Poderia usa-se uma *pesquisa em largura*, que começa numa qualquer nó do grafo e visita todos os nós no nível de profundidade atual antes de passar para os nós no nível de profundidade seguinte. A pesquisa em largura pode ser utilizada para resolver muitos problemas em teoria de grafos.

