

Edição 2025

Categoria

Juniores (9º e 10º ano de escolaridade)

Tempo

45 minutos

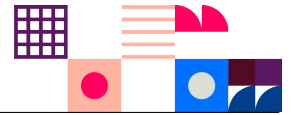
Resolve tantos problemas quanto possível em 45 minutos.

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 alunos 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 no ano passado contou com a participação de 136 232 alunos, de 849 escolas de Portugal, Angola, Moçambique, Cabo Verde e Timor-Leste.

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

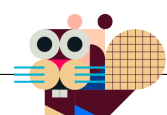


CC BY-NC-SA 4.0

Os problemas aqui colocados foram criados pela comunidade internacional da iniciativa Bebras e estão protegidos por uma licença da Creative Commons Atribuição-NãoComercial-Compartilha Igual 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:

 Alemanha	 Argentina	 Brasil	 Bulgária	 Canadá
 Chéquia	 Coreia do Sul	 Costa Rica	 Eslovénia	 Eslováquia
 Indonésia	 Japão	 Lituânia	 Malásia	 México
 Montenegro	 Paquistão	 Portugal	 Reino Unido	 Suíça

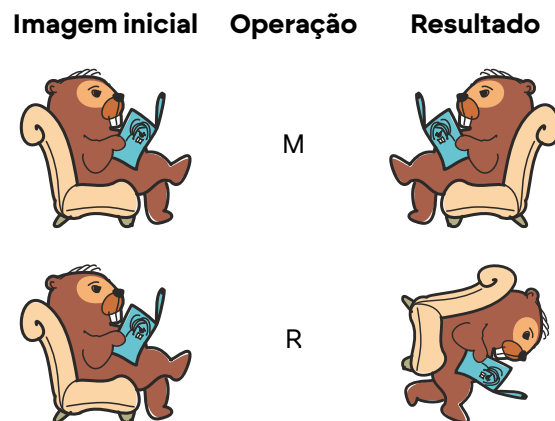


1. Transformações Espaciais

O castor Xavier quer programar um jogo e precisa de aprender a transformar imagens. O Xavier começa por aprender duas operações para transformar uma imagem:

- M: espelha a figura usando o eixo vertical;
- R: roda a figura 90 graus no sentido dos ponteiros do relógio.

Exemplo



Pergunta

O Xavier começa com esta figura...



... e quer obter esta figura:



Qual das seguintes sequências de operações **não** transforma a figura original na que o Xavier quer utilizar?

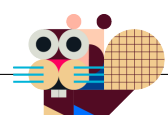
Respostas possíveis

(A) M R

(B) R R R M

(C) R M

(D) M R M R M R



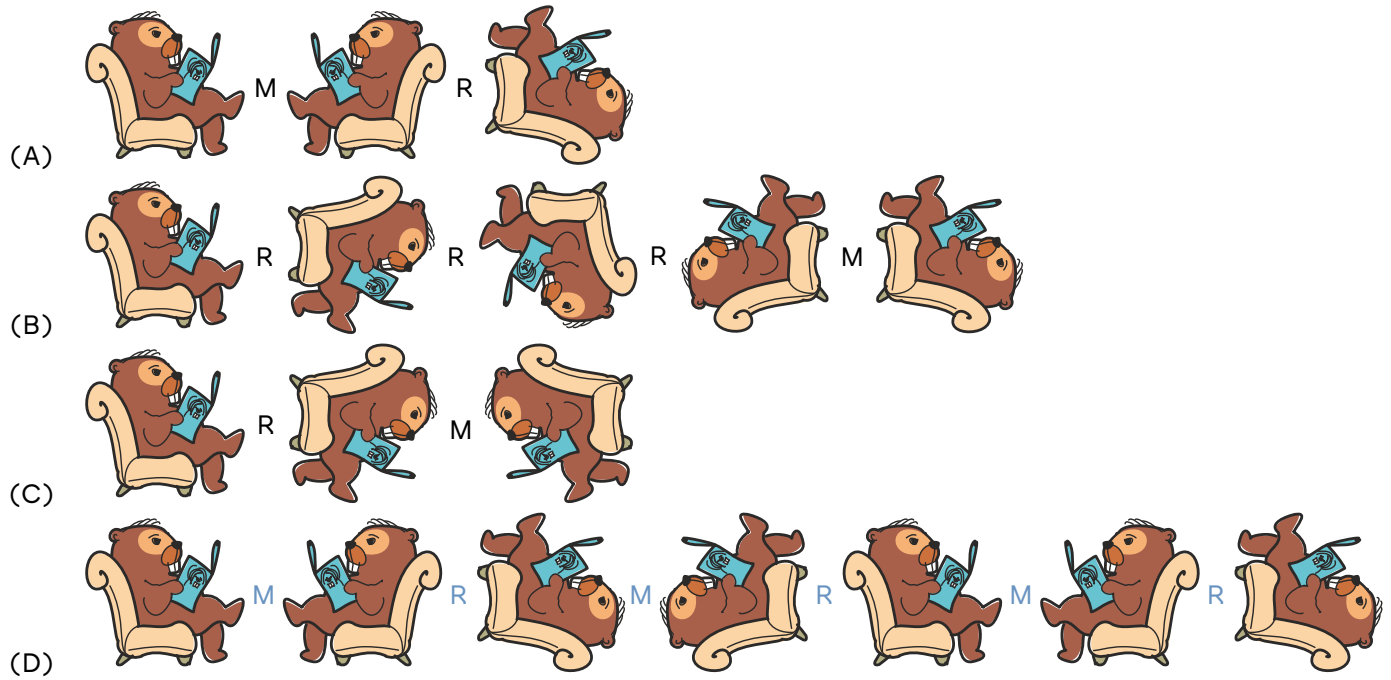
1. Transformações Espaciais (Resolução)

Solução

(C)

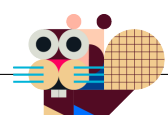
Resolução

Os resultados de cada uma das sequências de operações são os seguintes:



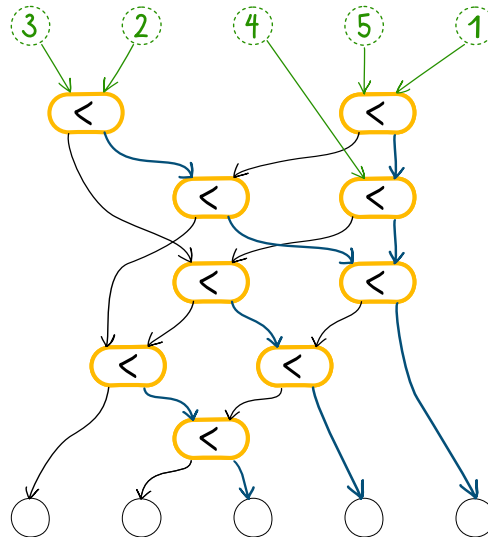
Isto é Pensamento Computacional!

A tarefa exige uma verificação cuidadosa das sequências apresentadas na resposta. Rodar e espelhar uma imagem são operações complexas, mas representá-las com letras (símbolos) simples, constitui, em si mesmo, uma forma de abstração.

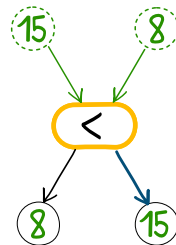


2. Uma Máquina Especial

Os castores têm uma máquina especial que recebe cinco números nas entradas do topo (por exemplo 3, 2, 4, 5 e 1). Através de setas e interruptores, os números percorrem a máquina até chegarem às saídas na parte inferior.



Cada um dos nove interruptores < compara os dois números que recebe e encaminha o número mais pequeno para a esquerda, e o número maior para a direita, como representado no exemplo abaixo.

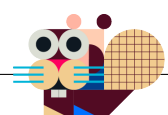


Pergunta

Que tarefa é executada por esta máquina?

Respostas possíveis

- (A) Ordena os números pela ordem inversa da entrada (resultado: 1, 5, 4, 2, 3)
- (B) Ordena os números por ordem crescente (resultado: 1, 2, 3, 4, 5)
- (C) Ordena os números por ordem decrescente (resultado: 5, 4, 3, 2, 1)
- (D) Mantém a ordem de entrada (resultado: 3, 2, 4, 5, 1)

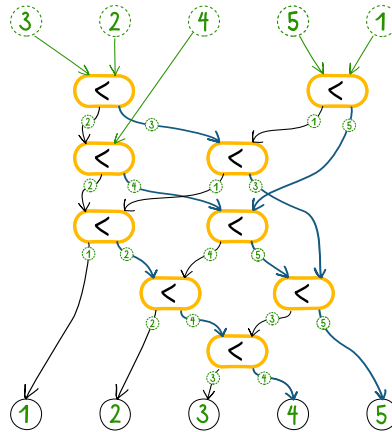


2. Uma Máquina Especial (Resolução)

Solução

(B)

Resolução



Isto é Pensamento Computacional!

Em informática, a máquina especial desta tarefa Bebras é conhecida como “rede de ordenação” (*sorting network*). É construída a partir de vários componentes idênticos e muito simples: os comparadores. Cada comparador recebe dois valores numéricos em duas linhas e compara-os, encaminhando o valor mais pequeno para a linha da esquerda e o valor maior para a linha da direita.

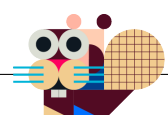
Combinando um número suficiente de comparadores, qualquer sequência de números pode ser ordenada por valor! Para ordenar cinco números, são necessários pelo menos nove comparadores. Para seis números, são necessários doze comparadores.

As redes de ordenação têm relevância prática pelas seguintes razões:

1. Em geral, uma ordenação rápida e eficiente facilita a pesquisa, por exemplo, em listas de resultados de motores de busca.
2. O algoritmo de ordenação é adequado para execução paralela, podendo ser executado de forma muito rápida.
3. Estas redes são muito adequadas para implementação em hardware, por exemplo em GPUs (*Graphics Processing Units*), porque os comparadores podem ser construídos como dispositivos eletrónicos muito simples e, portanto, baratos.



Algumas desvantagens incluem:







1. As redes de ordenação são concebidas apenas para entradas de comprimento fixo.
2. O número de comparadores necessários aumenta rapidamente com o comprimento da entrada.













3. Da Folha à Madeira

A Eduarda e os seus amigos adoram fazer caminhadas. Durante as caminhadas, os amigos da Eduarda recolhem informações sobre as árvores que veem e anotam-nas em tabelas da seguinte forma:








1. O Santiago recolhe informações sobre as formas das folhas () e as espécies de árvores correspondentes ()

	
  ⋮	  ⋮

2. A Catarina recolhe informações sobre os frutos () , as espécies de árvores correspondentes () e se são coníferas ()

		
  ⋮	X X ⋮	  ⋮

3. A Laura recolhe informações sobre a cor da madeira () , as espécies de árvores correspondentes () e se tem uma boa madeira para os castores construírem abrigos ()

		
  ⋮	  ⋮	✓ X ⋮

Para além disso, cada formato de folhas é característico de um tipo único de árvore, ou seja, duas árvores de espécies diferentes têm necessariamente folhas com formatos diferentes.

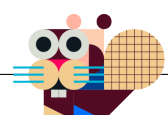
Pergunta

A Eduarda encontrou uma folha na floresta e conhece a sua forma. Ela quer saber se essa espécie de árvore tem uma boa madeira para os castores construírem abrigos.

A quais dos seus amigos deve a Eduarda perguntar, e em que ordem, para encontrar a resposta certa?

Respostas possíveis

- (A) Apenas à Laura.
- (B) Primeiro ao Santiago, depois à Catarina.
- (C) Primeiro ao Santiago, depois à Laura.
- (D) Primeiro à Catarina, depois ao Santiago, depois à Laura.







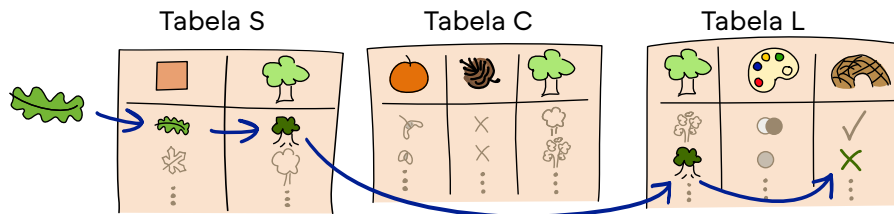
3. Da Folha à Madeira (Resolução)


Solução

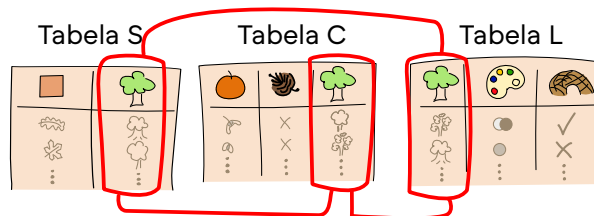
(C)

Resolução

A informação sobre a madeira de castor () só pode ser encontrada na tabela da Laura (tabela L). Mas, se a Eduarda apenas tiver informação sobre a folha, não conseguirá selecionar uma linha da tabela L. Ela precisa de obter informação sobre a espécie da árvore () ou sobre a cor da madeira (). A tabela da Catarina (tabela C) não tem referência a folhas, mas a tabela do Santiago (tabela S) tem. Sabendo a forma da folha, a Eduarda pode selecionar a linha correta da tabela S e obter a informação em falta sobre a espécie da árvore (). Depois, pode selecionar a linha correta da tabela L para obter a informação em falta sobre a madeira. Por exemplo, se a folha for de carvalho, então a Eduarda pode selecionar a linha correta na tabela L e descobrir que o carvalho não tem uma boa madeira para um castor para construir um abrigo.

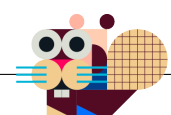


As tabelas estão ligadas entre si através de uma informação comum: a espécie da árvore (). Se conseguires selecionar uma informação específica em qualquer uma das tabelas, tens acesso a todas as informações em todas as tabelas.



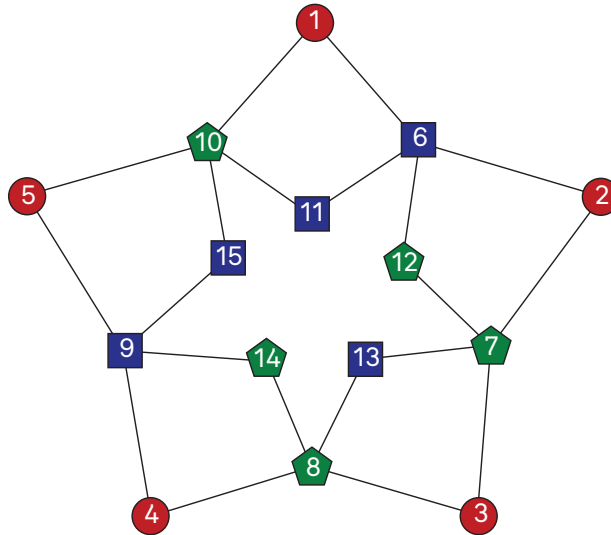
Isto é Pensamento Computacional!

O participante precisa de **abstração** para resolver esta tarefa. Embora o exemplo dado tenha uma solução nas tabelas apresentadas, o participante tem de analisar os dados fornecidos e pensar numa forma de os **interligar**.



4. Luzes Programáveis

A Sofia tem 15 luzes programáveis de três tipos: vermelhas (círculos), azuis (quadrados) e verdes (pentágonos). Cada luz tem um número associado.



A Sofia programa as luzes da seguinte forma:

- Cada luz vermelha é controlada por um interruptor;
- Uma luz azul acende se as duas luzes com menores números que lhe estão ligadas estiverem ambas acesas;
- Cada luz verde acende se exatamente uma das duas luzes com menores números que lhe estão ligadas estiver acesa (por outras palavras, quando uma destas luzes está acesa mas a outra não).

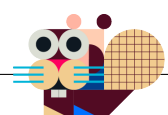
Num dado momento, todas as luzes estão apagadas. Depois, a Sofia acende as luzes 1, 2 e 4. Isto faz com que outras luzes se acendam, seguindo a programação da Sofia.

Pergunta

Quando as luzes param de mudar, quais das luzes 11 a 15 estão acesas?

Respostas possíveis

- (A) Luzes 11, 12, 13, 14 e 15
- (B) Apenas as luzes 12, 13 e 15
- (C) Apenas as luzes 11, 12, 13 e 14
- (D) Apenas as luzes 11, 13 e 14



4. Luzes Programáveis (Resolução)

Solução

(D)

Resolução

Das luzes vermelhas, apenas as luzes 1, 2 e 4 são ligadas inicialmente. Para determinar quais das restantes luzes se acendem, temos de caminhar em direcção ao centro da estrela:

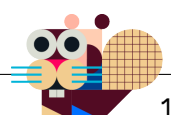
- Como a luz 6 é uma luz vermelha e as luzes 1 e 2 estão ambas acesas, a luz 6 acende;
- Como a luz 7 é uma luz verde e a luz 2 está acesa mas a 3 está apagada, a luz 7 acende;
- Como a luz 8 é uma luz verde, e a luz 4 está acesa mas a 3 está apagada, a luz 8 acende;
- Como a luz 9 é uma luz azul, e as luzes 4 e 5 não estão ambas acesas, a luz 9 permanece apagada;
- Como a luz 10 é uma luz verde, e a luz 1 está acesa mas a 5 está apagada, a luz 10 acende;
- Como a luz 11 é uma luz azul, e as luzes 6 e 10 estão ambas acesas, a luz 11 acende;
- Como a luz 12 é uma luz verde, mas as luzes 6 e 7 estão ambas acesas, a luz 12 permanece apagada;
- Como a luz 13 é uma luz azul, e as luzes 7 e 8 estão ambas acesas, a luz 13 acende;
- Como a luz 14 é uma luz verde, e a luz 8 está acesa mas a 9 está apagada, a luz 14 acende;
- Como a luz 15 é uma luz azul, mas as luzes 9 e 10 não estão ambas acesas, a luz 15 permanece apagada.

Portanto, das cinco luzes mais próximas do centro, só as luzes 11, 13 e 14 acendem.

Isto é Pensamento Computacional!

Esta tarefa exige **raciocínio algorítmico**, uma vez que os alunos têm de começar pelo que sabem sobre as luzes vermelhas exteriores e avançar em direcção ao centro da estrela, raciocinando sobre a forma como as luzes azuis (que funcionam como a operação lógica **E**) e verdes (que funcionam como a operação lógica **OU exclusivo / XOR**) estão programadas, para determinar quais das restantes luzes irão acender.

Esta tarefa também exige **decomposição**, pois os alunos precisam apenas de analisar as duas luzes de menor número ligadas a cada luz, para decidir se esta se acende ou não.



5. Mensagem Oculta

Um grupo de castores usa um código secreto que substitui cada letra pelo número que corresponde à sua posição no alfabeto. A tabela abaixo mostra o alfabeto completo com o número correspondente.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Cada letra é depois codificada numa linha de 9 quadrados, dividida em duas partes (i e ii). Os passos para codificar cada letra são os seguintes:

1. Encontrar o número da letra no alfabeto (por exemplo, o número do S é 19).
2. Dividir este número por 5, arredondando para baixo, e pintar de preto a caixa desta posição na secção (i). Por exemplo, $19/5$ arredondado para baixo dá 3.

i					ii			
1	2	3	4	5	1	2	3	4
		■						

3. Calcular o resto do número da letra dividido por 5 e pintar de preto a caixa desta posição na secção (ii). Por exemplo, $19/5$ tem resto 4.

i					ii			
1	2	3	4	5	1	2	3	4
		■						■

Assim, quando um castor quer enviar a palavra "BEBRAS", ela será codificada como:

	i					ii			
	1	2	3	4	5	1	2	3	4
B							■		
E	■								
B							■		
R			■					■	
A							■		
S			■						■

Os castores vão transformar a tabela de letras codificadas numa imagem, mas não vão usar a parte de cima onde estão os títulos.

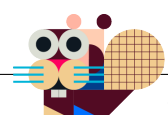
Pergunta

Qual a mensagem codificada na seguinte grelha?

		■					■	
						■		
								■
■								■
		■						

Resposta

Escreve a mensagem codificada na folha de respostas.



5. Mensagem Oculta (Resolução)

Solução

RADIO

Resolução

Para encontrar as letras escondidas, temos de encontrar primeiro o número codificado pela grelha para cada linha através da seguinte fórmula:

número da letra = posição da caixa preta em (i) \times 5 + posição da caixa preta em (ii)

i					ii			
1	2	3	4	5	1	2	3	4

$$\text{número} = (3 \times 5) + 3 = 18 \text{ (R)}$$

$$\text{número} = (0 \times 5) + 1 = 1 \text{ (A)}$$

$$\text{número} = (0 \times 5) + 4 = 4 \text{ (D)}$$

$$\text{número} = (1 \times 5) + 4 = 9 \text{ (I)}$$

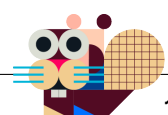
$$\text{número} = (3 \times 5) + 0 = 15 \text{ (O)}$$

Isto é Pensamento Computacional!

Esta tarefa introduz o conceito de **codificação**, ou seja, a conversão de dados de um formato para outro. Neste caso, as letras do alfabeto são primeiro transformadas em números e, depois, esses números são representados como um padrão visual utilizando quadrados. O desafio consiste em descodificar essa representação visual e reconstruir a mensagem original — uma sequência de letras.

Na ciência de computadores, os dados nem sempre são escritos como texto simples ou números. Por vezes, são transformados num formato visual que pode ser lido e descodificado por um computador. Dois exemplos do dia a dia são os **códigos de barras** e os **códigos QR**.

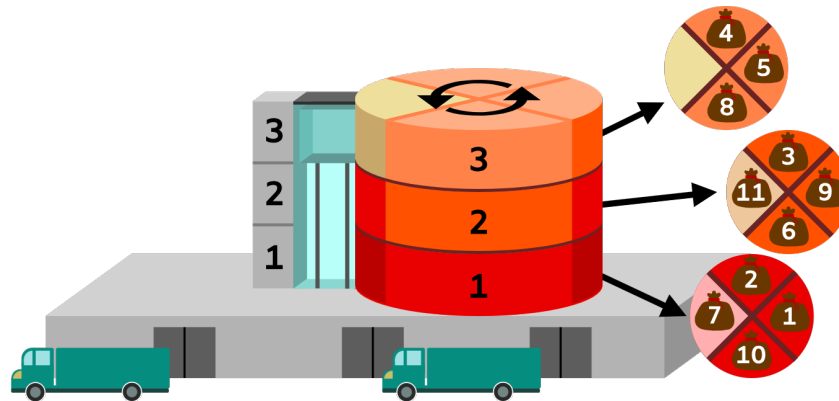
Nesta tarefa, a divisão com arredondamento e resto é usada como método para codificar e descodificar dados. Esta técnica desempenha um papel importante em muitas áreas da ciência de computadores e dos sistemas digitais. Por exemplo, sistemas de encriptação como o **RSA** utilizam operações de módulo (restos) para transformar de forma segura dados em mensagens codificadas e depois recuperá-las.



6. Fluxo de Presentes

És um(a) programador(a) a trabalhar na “GiftFlow”, um armazém de entrega de presentes. O teu trabalho é escrever instruções para um robô que move presentes do armazém para a zona de descarga.

O armazém tem 4 níveis: zona de descarga, Nível 1, Nível 2 e Nível 3. Cada nível de armazenamento é uma plataforma rotativa dividida em 4 setores. Os presentes são guardados em sacos nestes setores, havendo no máximo um saco por setor. A zona de descarga encontra-se no rés-do-chão.



Os comandos que podes dar ao robô são os seguintes:

- MU: subir um andar no elevador
- MD: descer um andar no elevador
- R: rodar a plataforma do andar em que o robô se encontra um setor no sentido contrário ao dos ponteiros do relógio
- L: carregar o presente do setor a que o robô tem acesso para o elevador
- U: descarregar o presente que está há mais tempo no elevador

Exemplo

Imagina que o robô começa no andar 2 e quer descarregar os presentes 8 e 9 no rés-do-chão. Uma sequência de comandos que permite executar esta tarefa é a seguinte: **MU, R, R, R, L, MD, R, R, L, MD, MD, U, U**.

Esta lista de comandos move o robô um andar para cima (MU), roda esse nível três vezes (R, R, R), carrega o saco 8 (L), desce um nível (MD), roda duas vezes (R, R), carrega o saco 9 (L) e depois leva os dois sacos para baixo para serem descarregados (MD, MD, U, U).

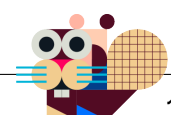
Pergunta

O elevador está atualmente no nível 1 e está vazio. Os sacos estão dispostos conforme mostrado na imagem. O robô executa a seguinte sequência: **R, R, L, MU, MU, R, R, MD, R, L, MD, MD, U, U**

Que sacos são descarregados na área de descarga?

Respostas possíveis

- | | |
|----------|-------------|
| (A) 3, 5 | (C) 1, 3, 5 |
| (B) 1, 5 | (D) 1, 3 |



6. Fluxo de Presentes (Resolução)

Solução

(D)

Resolução

Temos apenas dois comandos de carga (L) e dois comandos de descarga (U), por isso carregamos/descarregamos apenas dois sacos. Assim, a opção c) não é correta. As opções a) 3, 5 e b) 1, 5 não são correctas porque o robô não carregou o saco 5. Se analisarmos os comandos em detalhe, temos:

- R, R, L — carrega o saco 1
- MU, MU, R, R, MD — move para o nível 2
- R, L — carrega o saco 3
- MD, MD, U, U — desce e descarrega os dois sacos, 1 e 3

Isto é Pensamento Computacional!

Decomposição: A tarefa global (mover presentes entre os níveis do armazém) é decomposta em sub-tarefas mais pequenas e geríveis:

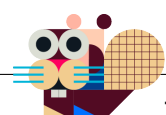
- Mover o elevador verticalmente (MU, MD).
- Rodar o nível do armazém (R).
- Carregar um presente (L).
- Descarregar um presente (U).

Pensamento algorítmico:

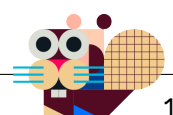
- *Instruções passo a passo:* As ações do robô são definidas como uma sequência precisa de passos (a sequência de comandos). Isto é um aspeto central do pensamento algorítmico.
- *Desenvolver uma estratégia:* Para mover os presentes, é necessário conceber uma estratégia (um algoritmo) que envolva:
 - Determinar as localizações iniciais e finais dos presentes.
 - Calcular os movimentos verticais necessários do elevador.
 - Calcular as rotações necessárias.
 - Inserir os comandos de carregar e descarregar nos pontos corretos.
- *Exemplo como algoritmo:* O exemplo dado (MU, R, R, R, L, MD, R, R, L, MD, MD, U, U) é um algoritmo específico para mover os presentes 8 e 9.
- *Otimização:* O pensamento algorítmico também envolve considerar a eficiência. Poderia o movimento dos presentes ser realizado com menos passos?

Raciocínio espacial:

- *Compreensão da disposição do armazém:* O robô precisa de compreender a disposição espacial dos níveis e setores. Isto envolve visualizar as posições relativas das diferentes localizações.

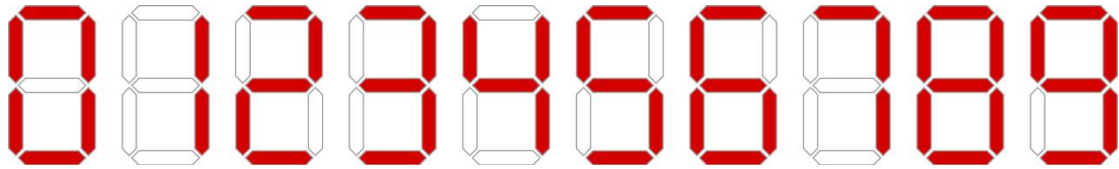


- *Calcular rotações:* Determinar o número de rotações necessárias para alcançar um setor específico requer raciocínio espacial. O robô tem de acompanhar mentalmente o seu setor atual e o setor alvo.
- *Visualizar movimentos verticais:* O robô precisa de perceber a relação vertical entre os níveis para executar corretamente os comandos do elevador.
- *Gestão do elevador:* O robô precisa de acompanhar a ordem dos presentes dentro do elevador e visualizar qual presente será descarregado a seguir.
- *Posições relativas:* O comando «rodar o andar em que o robô se encontra um setor no sentido contrário ao dos ponteiros do relógio» exige raciocínio espacial, pois o robô deve visualizar o movimento dos presentes dentro do elevador.



7. Segmento Desgastado

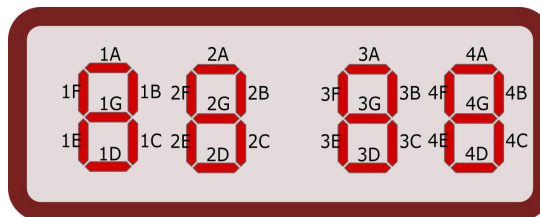
Um relógio digital padrão, que muda a cada minuto, mostra as horas usando quatro dígitos. Cada dígito contém 7 segmentos luminosos. Utilizando esses segmentos, cada dígito consegue representar os números de 0 a 9:



Os segmentos desgastam-se cada vez que são ativados (ou seja, quando passam do estado desligado para ligado). O segmento que for ativado mais vezes será o primeiro a precisar de ser substituído.

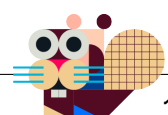
Pergunta

Qual dos 28 segmentos do relógio digital é o primeiro a precisar de ser substituído?



Resposta

Na folha de respostas, indica o código que representa o segmento, seguindo o esquema representado acima.



7. Segmento Desgastado (Resolução)

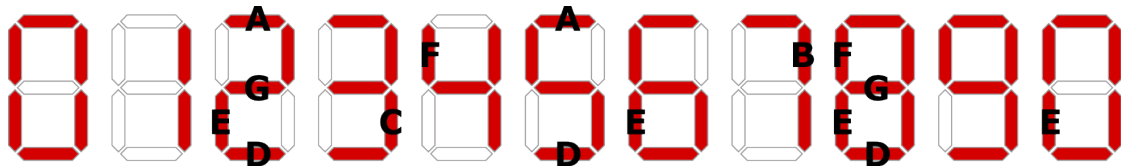
Solução

4E

Resolução

Cada dígito no mostrador representa um período de tempo diferente: dezenas de horas, horas, dezenas de minutos e minutos. O dígito mais à direita — ou seja, o dígito 4 — é o que muda com mais frequência, uma vez que representa os minutos individuais. Antes que qualquer outro dígito mude, todos os segmentos do dígito mais à direita já terão mudado várias vezes. O segmento mais desgastado encontra-se, portanto, no dígito 4, e o seu código de segmento é E.

Devemos então analisar todos os 7 segmentos do dígito mais à direita para determinar quantas vezes cada um é ativado (ou seja, ligado) quando o dígito representado passa de 0 a 0 novamente:



A imagem mostra essa verificação. Um segmento ativado (acabado de acender) é aquele que está iluminado no dígito atual, mas não estava iluminado no dígito imediatamente anterior. Cada ativação de um segmento é assinalada com a letra correspondente a esse segmento. O número total de ativações de um segmento é o número de vezes que a sua letra aparece na imagem.

Segmento	Dígito que causa a ativação do fragmento	Número de vezes que o segmento é ativado
A	2, 5	2
B	7	1
C	3	1
D	2, 5, 8	3
E	2, 6, 8, 0	4
F	4, 8	2
G	2, 8	2

A tabela resume esta verificação:

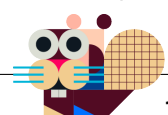
- A coluna central mostra os dígitos em que o segmento foi ativado.
- A coluna da direita contabiliza essas ativações.

O segmento que muda mais vezes é o E no dígito 4, e o seu código é 4E.

Isto é Pensamento Computacional!

Este problema exige que os alunos usem o raciocínio lógico para perceber que o dígito mais à direita de um relógio digital padrão é o que muda com mais frequência. Assim, o problema reduz-se a analisar o padrão de mudanças dos segmentos no dígito mais à direita.

De seguida, os alunos precisam de analisar o número de mudanças em cada segmento do visor de sete segmentos mais à direita durante um ciclo completo de mudança. Esta mudança é representada pela transição dos segmentos do visor de desligado para ligado. Este processo requer a capacidade de reconhecer padrões.

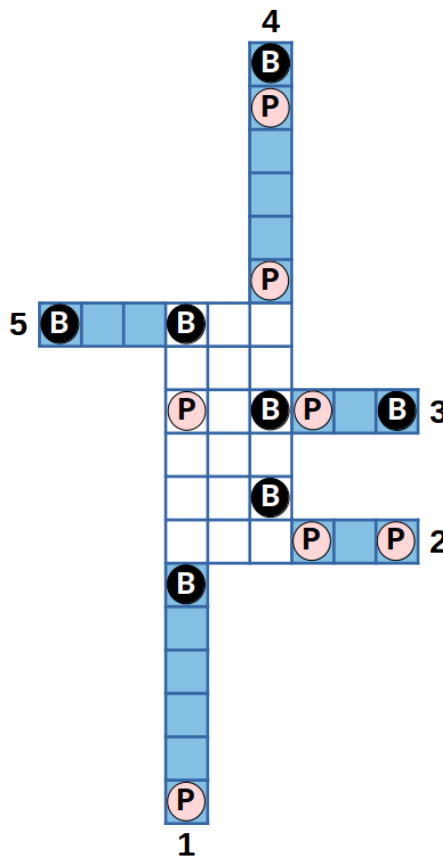


8. Dobragem Perfeita

A figura abaixo representa uma grelha 3 por 6 com algumas linhas e colunas prolongadas (numeradas e a azul). Na grelha e nos prolongamentos, alguns quadrados contêm pinos (P) vermelhos, outros contêm buracos (B) pretos e outros estão limpos. O Lucas quer dobrar as 5 extensões azuis da figura de forma a obter uma dobragem perfeita, ou seja, uma grelha 3 por 6 sem extensões. Durante a dobragem, o Lucas deve considerar as seguintes regras:

- Ao sobrepor um pino sobre um buraco, ou um buraco sobre um pino, o quadrado resultante fica limpo;
- Ao sobrepor um buraco sobre outro buraco, o buraco mantém-se;
- Quando um quadrado limpo é sobreposto a um buraco, ou um buraco a um quadrado limpo, o buraco prevalece;
- É impossível sobrepor um pino a outro pino;
- É impossível sobrepor um quadrado limpo a um pino, ou um pino a um quadrado limpo;

Para além disso, uma dobragem só é considerada válida se todos os quadrados da grelha ficarem limpos no final do processo.

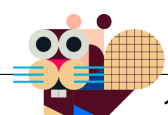


Pergunta

Por quantas ordens diferentes podemos dobrar os segmentos para obter uma dobragem perfeita?

Resposta

Escreve o número de possibilidades na folha de resposta.



8. Dobragem Perfeita (Resolução)

Solução

6

Resolução

Para chegar à lista de possibilidades, é possível listar todas as ordens de dobragem e excluir aquelas que não funcionam. No entanto, isto corresponde a considerar $5! = 120$ possibilidades, pelo que idealmente o problema deve ser resolvido com uma análise mais global:

- A aba 2 tem de ser depois da 1 e da 4;
- A aba 3 tem de ser antes da 1;
- A aba 5 tem de ser antes da 4.

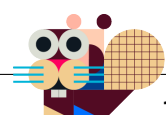
Isto significa que a aba 2 tem de ser a última. Assim, o problema fica reduzido a escolher a posição de 2 dos números (por exemplo, 3 e 1, uma vez que as posições do 5 e do 4 ficam autodeterminadas ao colocar estes dois números) das quatro posições disponíveis, o que resulta em 6 ordens possíveis:

- 3 1 5 4 2
- 3 5 1 4 2
- 3 5 4 1 2
- 5 3 1 4 2
- 5 3 4 1 2
- 5 4 3 1 2

Isto é Pensamento Computacional!

Resolver esta tarefa exige a aplicação de várias ferramentas mentais do pensamento computacional, tais como:

- **Decomposição:** dividir um problema complexo em partes mais simples e geríveis — analisar a estrutura da grelha e a localização dos pinos, buracos e espaços limpos, identificar possíveis estratégias de dobragem e avaliar se a dobragem conduz a uma solução válida;
- **Desenvolvimento de algoritmos:** o problema é um caso de ordenação com restrições;
- **Abstração:** podemos representar a grelha como um array de valores (por exemplo, 1 para pinos, -1 para buracos, 0 para limpo) e, em vez de nos focarmos nos detalhes gráficos, pensar em como modificar os valores da matriz ao realizar as dobras.



9. Raios Cósmicos

A Marta está a contar a uma amiga uma história estranha sobre a sua conta bancária. Um dia, a Marta consultou o saldo no telemóvel e viu que tinha 8095 euros. De repente, o número mudou para 7071 euros. Como não tinha feito nenhuma transação nem pagamento, a Marta ficou confusa e enviou uma mensagem ao banco a perguntar o que tinha acontecido.

Os funcionários do banco analisaram a situação e concluíram que não havia erro no software nem tinha havido nenhum ciberataque. Acabaram por perceber que tinha havido uma alteração aleatória no valor de um bit na memória, provocada por radiação cósmica.

Nos sistemas informáticos, os números são armazenados em binário. O sistema de numeração binária usa apenas dois dígitos: 0 e 1. Os valores das posições aumentam como múltiplos de dois à medida que se avança no número da direita para a esquerda (do dígito menos significativo para o mais significativo).

A tabela abaixo mostra como ler o número de 8 bits "00100011":

128	64	32	16	8	4	2	1
0	0	1	0	0	0	1	1

Em formato decimal, este número é $32+2+1=35$.

Na imagem seguinte um único bit mudou, nomeadamente o **8º bit** (porque contamos as posições da direita para a esquerda) e ficamos com o valor $128+32+2+1=163$.

128	64	32	16	8	4	2	1
1	0	1	0	0	0	1	1

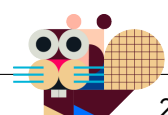
Como podes ver, a alteração de um único bit pode mudar substancialmente o valor do número. Um saldo bancário teria de ser armazenado usando um grande número de bits, já que o saldo pode ser muito elevado.

Pergunta

Qual foi o bit do saldo da Marta que foi alterado pela radiação cósmica?

Respostas possíveis

- (A) 11º bit de 1 para 0
- (B) 12º bit de 1 para 0
- (C) 11º bit de 0 para 1
- (D) 12º bit de 0 para 1



9. Raios Cósmicos (Resolução)

Solução

(A)

Resolução

Primeiro, descobrimos quanto mudou o saldo da conta, subtraindo os valores:

$$8095 - 7071 = 1024$$

Pelas tabelas, podemos ver que o valor de posição de cada bit duplica à medida que avançamos da direita para a esquerda. Para chegar ao valor de 1024 precisamos de encontrar o 11.º bit.

Podemos então verificar os números e ver que, na forma binária, eles diferem apenas num dígito, no 11.º bit:

$$8095 \text{ (base 10)} = 1\ 1111\ 1001\ 1111 \text{ (base 2)} \text{ vs. } 7071 \text{ (base 10)} = 1\ 1011\ 1001\ 1111 \text{ (base 2)}$$

Isto é Pensamento Computacional!

Os sistemas informáticos armazenam números em forma binária. Estes números são guardados na memória do computador, que pode estar sujeita a erros e avarias. Normalmente, a origem é elétrica, e os valores 0 ou 1 são armazenados carregando ou descarregando um componente.

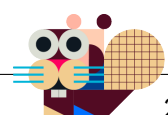
É um facto que a radiação cósmica, composta por partículas carregadas rápidas, também pode atingir a superfície da Terra e, portanto, atravessar a memória de um computador. Quando atinge uma posição da memória que guarda o valor de um bit de informação, pode carregá-lo ou descarregá-lo. A probabilidade é muito pequena, mas existe.

As naves espaciais estão em ainda maior perigo, porque a radiação cósmica é muito mais forte no espaço. É por isso que usam três blocos de memória para armazenar os mesmos valores. Durante o funcionamento, verificam constantemente se os valores são diferentes entre si e, caso sejam, descobrem qual dos três valores é diferente dos outros e corrigem-no.

Existem vários métodos específicos de deteção e correção de erros (como os códigos de Hamming, os códigos de Reed-Solomon, etc.), com diferentes níveis de complexidade e capacidades de correção de erros, para lidar com erros de dados.

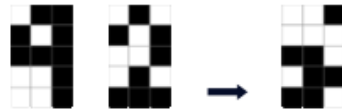
As componentes de Pensamento Computacional avaliadas neste problema incluem:

- **Abstração:** reconhecer o que é importante numa solução e concentrar-se apenas nessa componente
- **Reconhecimento de padrões:** padrões no valor de posição dos números binários

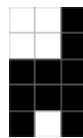


10. Quadriculas em Código

A Gema está aborrecida durante uma aula e começa a preencher quadrados na sua folha quadriculada para desenhar números. Ela inventa uma forma especial de representar o seu número favorito de dois dígitos, o 42. A partir das duas representações dos algarismos "4" e "2" numa grelha 3 × 5, mostradas abaixo, ela cria uma nova grelha em que cada quadrado é pintado de preto se, e só se, exatamente um dos dois quadrados correspondentes nas imagens do "4" e do "2" estiver a preto.



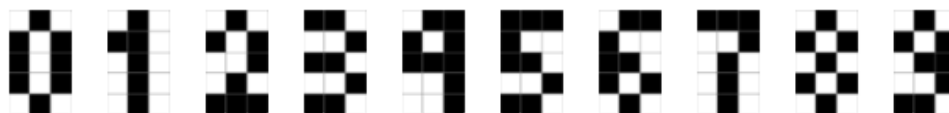
A Gema utilizou o mesmo método para representar um outro número de 2 dígitos, e obteve a seguinte figura:



Pergunta

Qual foi o número de 2 dígitos escolhido pela Gema para obter esta representação?

Se houver mais do que uma resposta correta, indica apenas uma. As imagens iniciais correspondentes a cada dígito são dadas a seguir.



Resposta

Escreve dois dígitos na folha de respostas.

10. Quadriculas em Código (Resolução)

Solução

Há duas respostas corretas possíveis, 26 e 62.

Resolução

Combinar estes dois dígitos seguindo a regra definida pela Gema resulta no padrão indicado:



Note-se que o método de combinação é simétrico em relação aos dois dígitos do número selecionado, o que significa que dois números com o mesmo par de dígitos em ordem inversa (por exemplo, 26 e 62) dão origem à mesma representação.

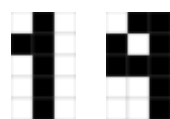
Para resolver esta tarefa, temos de encontrar formas de determinar os dois algarismos corretos evitando procurar por todas as combinações possíveis, uma vez que há 100 no total, de "00" a "99". Como normalmente omitimos o zero à esquerda em números de um só algarismo (escrevemos "7" e não "07"), isto reduz o número de combinações para 90, ou até para 45 se considerarmos a simetria mencionada acima. Ainda assim, este número é demasiado elevado para verificar exaustivamente cada possibilidade.

A primeira coisa a compreender é que a operação de combinação produz um pixel preto numa determinada posição sempre que esse pixel é diferente nos dois algarismos de origem. Isto resulta do facto de que obtemos um pixel preto no resultado sempre que há apenas um pixel preto ao considerar as imagens de origem. Se olharmos para um único pixel e listarmos todas as possibilidades, obtemos esta tabela de correspondência: [inserir tabela]

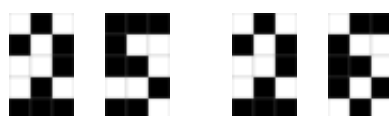
Ao perceber isto, podemos de imediato excluir todos os números formados pelo mesmo algarismo repetido, como "22" ou "99", já que todos os pixels seriam iguais entre os dois algarismos e isso resultaria numa imagem combinada completamente branca.

Uma forma de abordar o resto da procura pela solução é notar que a coluna mais à direita da imagem combinada é toda preta, o que indica que os dois algarismos de origem têm de ter todos os pixels diferentes nessa coluna.

Um par de algarismos óbvio que satisfaz esta condição é "1"-"4" — a última coluna está toda vazia no "1" e totalmente preenchida no "4":



Infelizmente, ao observarmos, por exemplo, a segunda linha ou a linha inferior, podemos excluir esse par, pois o padrão resultante não corresponde à imagem combinada. Os dois outros pares, um pouco menos óbvios, em que todos os pixels são diferentes na última coluna, são "2"-"5" e "2"-"6":



Ao inspecionarmos os pixels restantes da imagem combinada (por exemplo, o primeiro pixel na linha superior), podemos excluir o par "2"-"5". Ficamos assim apenas com o par "2"-"6", que resulta na representação desejada. Considerando a simetria mencionada acima, concluímos que tanto 26 como 62 são respostas possíveis.

Isto é Pensamento Computacional!

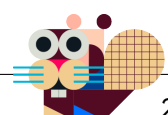
A operação central utilizada na criação da imagem combinada é, na verdade, a operação booleana OU-exclusivo (**XOR**). Uma operação booleana atua sobre valores verdadeiro/falso (chamados valores booleanos) e produz outro valor booleano. Esta é a tabela de verdade da operação booleana XOR, onde Z é o resultado:
[inserir tabela]

Nota-se a semelhança com a tabela de correspondência de pixels mostrada acima, se substituirmos os 0 por quadrados vazios e os 1 por quadrados preenchidos a preto.

Outras operações booleanas incluem **E** e **OU**. Estas operações são amplamente usadas no desenho de sistemas de hardware sob a forma de portas lógicas. São blocos de construção que recebem dois sinais como entrada e “calculam” a saída de acordo com a respetiva tabela de verdade.

Em criptografia, o XOR desempenha um papel central em esquemas de encriptação simétrica devido à sua natureza reversível: aplicar a mesma operação XOR duas vezes com a mesma chave resulta no valor original (neste exemplo, combinar a imagem combinada com uma das imagens dos algarismos dará como resultado... a outra imagem do algarismo!). Esta propriedade torna o XOR útil tanto para codificar como para decodificar dados. O XOR é também amplamente utilizado em técnicas de deteção e correção de erros, como checksums e códigos de redundância cíclica (CRC), onde esta operação ajuda a identificar discrepâncias entre os dados transmitidos e recebidos.

Esta tarefa treina sobretudo o **reconhecimento de padrões**, ao verificar pixels nas imagens de origem e na imagem combinada.



11. Tecla Mágica

O Tiago tem um rato de computador especial que tem uma tecla mágica! O rato mantém um contador, que é incrementado em 1 cada vez que esta tecla mágica é utilizada.

Para além disso, cada vez que esta tecla é pressionada, todos os números inteiros entre o valor atual do contador e 1 são escritos num ficheiro especial. Como o contador começa com o valor 1, depois de o Tiago carregar na tecla 5 vezes, este ficheiro especial contém a seguinte sequência: **1, 2, 1, 3, 2, 1, 4, 3, 2, 1, 5, 4, 3, 2, 1**

Nº de cliques	Conteúdo do ficheiro
1	1
2	1, 2, 1
3	1, 2, 1, 3, 2, 1
4	1, 2, 1, 3, 2, 1, 4, 3, 2, 1
5	1, 2, 1, 3, 2, 1, 4, 3, 2, 1, 5, 4, 3, 2, 1

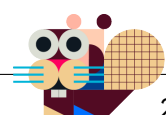
Pergunta

O Tiago está fascinado com este fenómeno e carregou na tecla mágica várias vezes, gerando um ficheiro com uma sequência muito comprida.

Qual o número que se encontra na 127ª posição da sequência?

Resposta

Escreve um número inteiro (entre 1 e 16) na folha de respostas.



11. Tecla Mágica (Resolução)

Solução

10

Resolução

O tamanho do ficheiro (em termos da contagem de números nele escritos) é, após carregar na tecla mágica algumas vezes: 1, 3, 6, 10, 15, 21, 28... De forma geral, se n é o número vezes que se carregou na tecla, o tamanho do ficheiro será $T(n) = n(n+1)/2$.

Para obter a resposta, podemos recorrer a uma simulação. Podemos tentar responder à seguinte questão: qual é o menor n tal que $T(n) \geq 127$? Claramente, qualquer n que produza um valor mais pequeno não pode ser a resposta, pois isso significaria que menos de 127 números foram escritos no ficheiro. Assim, aumentamos progressivamente n até encontrarmos o primeiro número que satisfaça a condição. Verificamos que $T(15) = 120$, que é demasiado pequeno, mas $T(16) = 136$.

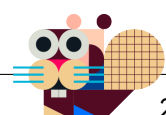
Quando carregamos no botão pela 16.^a vez, os números escritos serão **[16, 15, 14, 13, 12, 11, 10, ..., 1]**. Sabemos que o 16 é o 121.^o número escrito. Então, basta contar para baixo a partir do 16 até chegarmos ao 127.^o número, que será **10**.

Isto é Pensamento Computacional!

Para resolver esta tarefa, é essencial aplicar conhecimentos de reconhecimento de padrões. Reconhecer a sequência pela qual o tamanho do ficheiro aumenta é semelhante a identificar padrões que podem ajudar a produzir melhores algoritmos.

A capacidade de decomposição também é útil nesta tarefa. O estudante pode decompor o espaço do problema em duas partes: todos os números que podem ser ignorados e o conjunto de números relevantes onde a resposta deve ser encontrada.

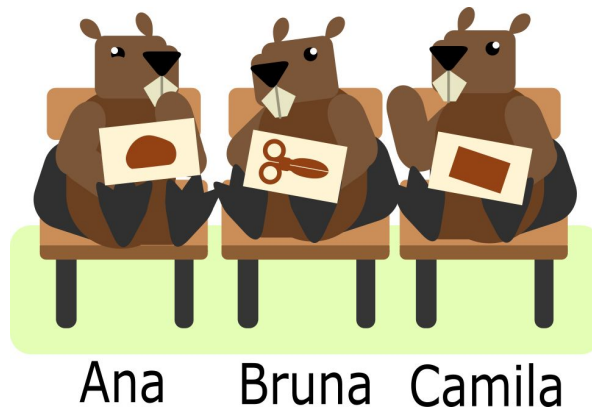
Estas competências promovem uma compreensão profunda do pensamento computacional através de cenários simples.



12. Pedra, Papel, Tesoura, Troca

A Ana, a Bruna e a Camila inventaram uma nova versão do jogo pedra, papel, tesoura. Na versão original deste jogo, a pedra ganha à tesoura, a tesoura ganha ao papel e o papel ganha à pedra.

Na versão modificada, a Ana, a Bruna e a Camila sentam-se em cadeiras e seguram um cartão com o objeto que vão utilizar, de forma a que todas possam ver a sua jogada. Antes de determinar quem ganha a quem, as jogadoras têm de decidir entre si quantas trocas vão efetuar. Uma troca é uma operação realizada entre duas jogadoras apenas, que trocam os seus cartões. Depois de decidir quantas trocas vão realizar, as três amigas têm de escolher que jogadoras estarão envolvidas em cada uma delas.



Pergunta

O único objetivo da Bruna nesta ronda é ganhar à Camila. Que estratégia garante que a Bruna tem sucesso na sua tarefa?

Respostas possíveis

- (A) A Bruna só tem de garantir um número ímpar de trocas com a Camila
- (B) Independentemente do número de trocas realizadas, a Bruna nunca deve trocar de cartão com a Camila
- (C) Independentemente do número de trocas realizadas, a Bruna deve trocar sempre de cartão com a Camila
- (D) A Bruna só tem de garantir que há um número par de trocas, independentemente das jogadoras nelas envolvidas

12. Pedra, Papel, Tesoura, Troca (Resolução)

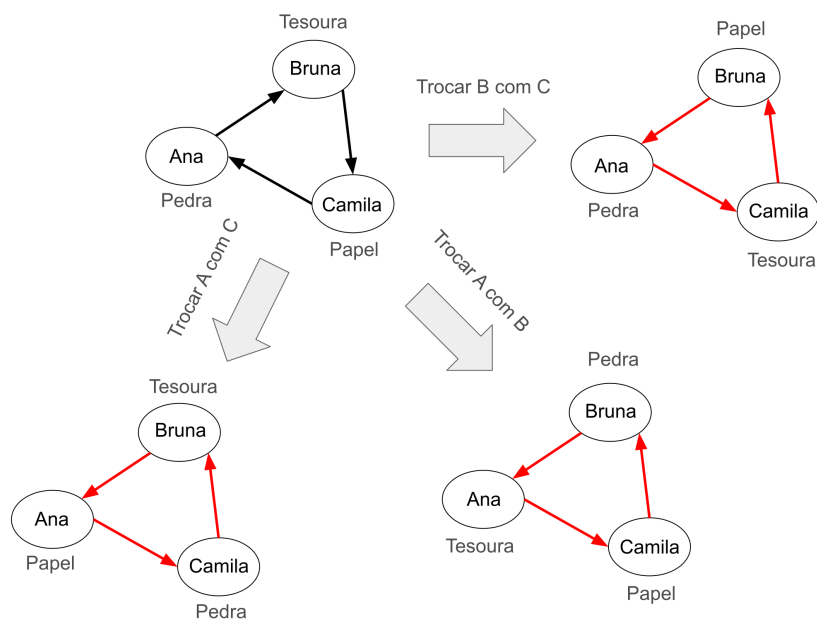
Solução

(D)

Resolução

Para resolver este problema, é preciso reparar que, quando ocorre uma troca, independentemente de que carta está com cada jogador, todas as vitórias ficam invertidas.

[inserir imagem]



Nesta figura, as setas indicam como se resolve uma partida. Assim, após um número **par** de trocas, as vitórias mantêm-se iguais à situação inicial. Pelo contrário, após um número **ímpar** de trocas, as vitórias ficam invertidas.

Como a Bruna quer ganhar à Camila e isso acontece na situação inicial, ela só precisa de garantir que ocorre um número **par** de trocas.

Isto é Pensamento Computacional!

O reconhecimento de padrões é uma competência fundamental para resolver esta tarefa. À primeira vista, a tarefa pode parecer muito difícil porque parece haver demasiados resultados possíveis a considerar. No entanto, se forem analisados sistematicamente alguns casos, não é difícil perceber que afinal não há assim tantas variações e que os casos existentes podem ser agrupados de acordo com a **paridade** do número de trocas.

