CONCURSO BEBRAS · ESPAÑA

Bebras 2024

5.º y 6.º de Primaria · Grupo III (10–12 años)

Cuadernillo con las 15 tareas del concurso, sus soluciones explicadas y las secciones «Esto es informática» y «Esto es pensamiento computacional».

bebras.ehu.es

Presentación

Bebras es un desafío internacional de informática y pensamiento computacional que se celebra cada curso en más de 80 países. Sus tareas son problemas breves, resolubles sin conocimientos previos de programación, que acercan al alumnado a ideas fundamentales de la informática: algoritmos, estructuras de datos, lógica, grafos o representación de la información.

Este cuadernillo recoge las 15 tareas que resolvió el alumnado de 5.º y 6.º de Primaria en la edición de 2024, realizada en la plataforma ViLLE (curso «2024 Spanish III»). Cada tarea valía 10 puntos, con un máximo de 150. Para cada tarea se incluye:

- **Enunciado**, en una página que se puede fotocopiar para trabajar en clase.
- **Solución** con la respuesta correcta y su explicación.
- **Esto es informática**: el concepto informático que hay detrás de la tarea.
- **Esto es pensamiento computacional**: las destrezas que se ponen en juego al resolverla.

La dificultad indicada es la que asignó el taller internacional Bebras para la franja de 10 a 12 años; cuando una tarea no tenía asignada esa franja, se indica entre paréntesis la dificultad en la franja más próxima. Las opciones de respuesta aparecen en el orden de los materiales originales; en ViLLE el orden pudo ser distinto.

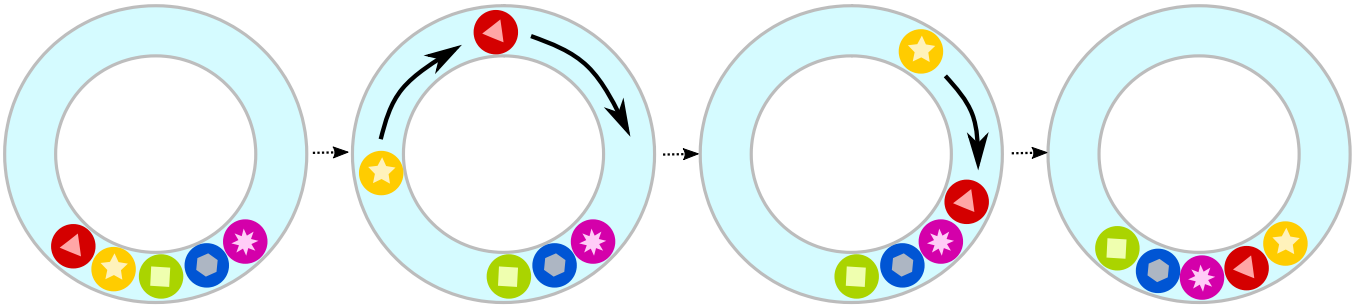
Resumen de tareas y respuestas

N.º	Tarea	Código	Dificultad	Respuesta
1	El sonajero de Óliver	2024-SK-01b	Fácil	Entre la morada y la verde: roja y amarilla; después de la verde: azul.
2	La flor milagrosa	2024-DE-03	Media	A) 5 días.
3	Cartas dibujadas	2024-IN-04	Fácil	C) Si el hombre se come una pizza, entonces no lleva corona.
4	Regalo de cumpleaños	2024-NL-02	— (fácil en 8–10)	A (la pelota con una estrella, sin rayas ni luna).
5	Rutas de árboles	2024-DE-08	— (media en 12–14)	 (roble, abedul, abeto, frutal, árbol claro).
6	Alfred en el baile	2024-HU-02	Fácil	B) Alfred puede que llevara corbata o puede que llevara bastón.
7	Superbebras	2024-DE-07	Difícil	C.
8	Un camino para el robot	2024-PL-04	Media	↑ en la 2.ª columna (3.ª fila), → en la 2.ª columna (2.ª fila) y → en la 4.ª columna (1.ª fila).
9	El tubo de bolas	2024-KR-03a	Fácil	C (blanca, negra, negra).
10	Días soleados	2024-DE-04a	Media	B) ningún castor nadó en el estanque de la cascada helada.
11	Clase en línea	2024-MY-03	Fácil	E) Hannah.
12	El cuadro de Bea	2024-FI-01	Fácil	D.
13	El mapa de los faros	2024-CZ-05	Fácil	La casilla del agua que está dentro de los círculos de B y de C, pero fuera del de A (entre los dos faros, a la derecha de la zona morada).
14	Las cartas Ricca	2024-CH-03a	Media	nombre: ABC… · ojos: 123… · patas: 123… · alas: no/sí · dientes: no/sí.
15	Plazas libres en el puerto	2024-MT-02	Fácil	C) 6.

1 El sonajero de Óliver

2024-SK-01b · Eslovaquia
Dificultad: Fácil

Óliver tiene un sonajero transparente con bolas de colores. Cuando lo agita, algunas bolas recorren el sonajero y pasan al otro lado, como se ve en la imagen.



Óliver agita el sonajero una vez más.



Coloca en los huecos blancos las bolas que faltan, tal como deberían estar en el sonajero de Óliver.

Dibuja en los huecos blancos del sonajero las bolas que faltan. Bolas disponibles:

Respuesta correcta: Entre la morada y la verde: roja y amarilla; después de la verde: azul.

Explicación

La respuesta correcta es:



Cuando las bolas recorren el sonajero no pueden cambiar de orden. Por eso, en el sonajero siempre van en este orden: después de la roja va la amarilla; después de la amarilla, la verde; después de la verde, la azul; después de la azul, la morada; y después de la morada, otra vez la roja.

En la pregunta falta una bola después de la verde (entre la verde y la morada): según ese orden, es la azul.

Además, entre la bola morada y la verde faltan dos bolas: la roja y la amarilla. Según el orden, la roja va junto a la morada y la amarilla junto a la verde.

Esto es informática

El sonajero de Óliver ilustra cómo funciona una **lista enlazada circular**. En ella, cada elemento tiene un **puntero** que indica cuál es el siguiente, y el puntero del último elemento apunta al primero, de modo que se forma un círculo. Una lista circular puede guardar cualquier cosa; aquí la lista es el sonajero y los elementos son las bolas de colores.

Es decir, de cada bola sabemos cuál va después. Para comprobar el orden se puede empezar por cualquier bola y recorrer todas en orden.

En la vida real hay muchas listas circulares. Por ejemplo, las listas de reproducción de música en bucle: cuando termina la última canción, vuelve a empezar la primera. Las listas circulares no tienen final, así que nunca se acaban los elementos, y es fácil recorrerlas sin reglas especiales para el último elemento: basta con seguir siempre en la misma dirección.

Esto es pensamiento computacional

Para resolver la tarea hay que fijarse en el orden de las bolas y completar las que faltan siguiendo el mismo orden que en la imagen original. Esto es **reconocimiento de patrones**. Reconocer patrones ayuda a ver si dos problemas se parecen o son el mismo: así no hay que resolver cada uno por separado, sino que la solución de uno ayuda a resolver el otro.

Palabras clave y enlaces: Lista enlazada circular, puntero · es.wikipedia.org/wiki/Lista_enlazada

2 La flor milagrosa

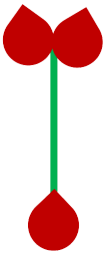
2024-DE-03 · Alemania
Dificultad: Media

Al amanecer, de cada capullo nuevo de una flor milagrosa brota un tallo. El tallo crece durante todo el día. Al atardecer, el tallo se divide en dos capullos nuevos de flor milagrosa y deja de crecer. Esto se repite día tras día, y la flor milagrosa es cada vez más espléndida.

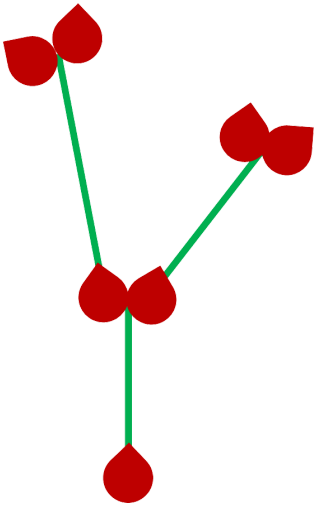
Capullo nuevo antes del amanecer



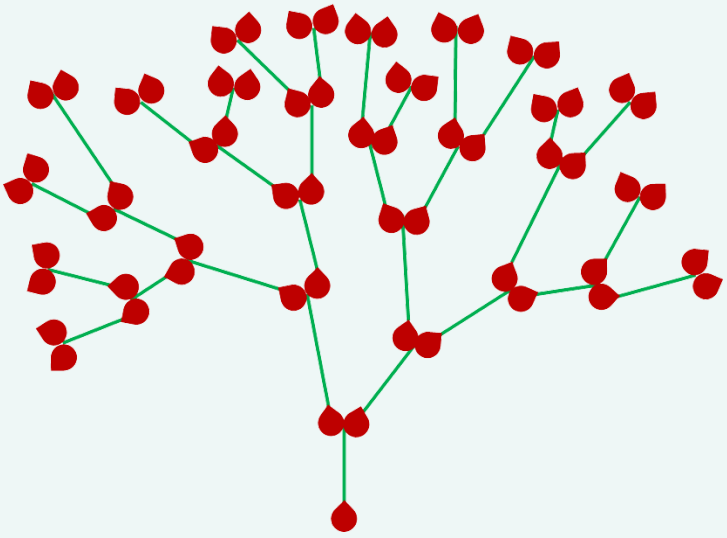
La flor después del atardecer (primer día)



La flor después del atardecer (segundo día)



¿Cuántos días lleva creciendo esta flor milagrosa?



A) 5 días
B) 11 días
C) 16 días
D) 32 días

Respuesta correcta: A) 5 días.

Explicación

La respuesta correcta es **5 días**.

Al final de cada día, de cada capullo de la flor ha salido un tallo con dos capullos nuevos. Por tanto, basta con seguir una sola rama (por ejemplo, la de más a la derecha) desde la base hasta un capullo exterior y contar los tallos: hay 5, uno por día. Es la **altura** del árbol binario que forma la flor.

- **11 días** sería el número total de capullos y tallos de una sola rama.
- **16 días** sería el número de tallos exteriores.
- **32 días** sería el número de capullos exteriores ($2^5 = 32$).

Esto es informática

Las flores milagrosas crecen según una única regla: un capullo forma un tallo y dos capullos nuevos. La regla se aplica a sí misma, y así la flor podría, en teoría, crecer indefinidamente. Este tipo de reglas se llaman **recursivas**.

En programación, una **función recursiva** es una función que se llama a sí misma; la nueva copia vuelve a llamarse, y así sucesivamente. Es imprescindible una **condición de parada**: la función solo se llama a sí misma mientras se cumpla una condición. Un ejemplo matemático es el factorial: $n! = n \cdot (n-1)!$ si $n > 1$, y $1! = 1$. Por ejemplo, $4! = 4 \cdot 3! = 4 \cdot 3 \cdot 2! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$.

Los programas recursivos suelen ser claros y fáciles de leer, aunque, por la cantidad de llamadas, pueden ser más lentos y gastar más memoria que los iterativos. Con recursión se generan también gráficos de aspecto muy natural llamados **fractales**, como los árboles de Pitágoras. Y la flor tiene forma de **árbol binario**, una estructura de datos muy usada en informática: con n niveles tiene 2^n hojas.

Esto es pensamiento computacional

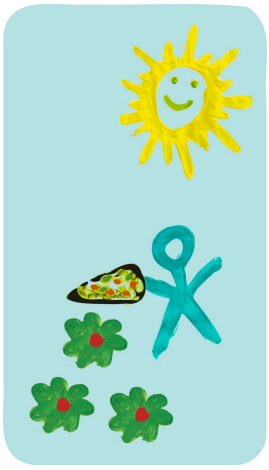
La **abstracción** exige centrarse solo en la información importante del problema e ignorar los detalles irrelevantes. Aquí basta con darse cuenta de que es suficiente mirar la longitud de una sola rama (cualquiera), sin necesidad de contar todos los capullos de la flor.

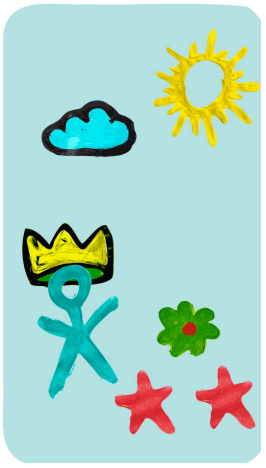
Palabras clave y enlaces: Recursión, árbol binario, fractales · es.wikipedia.org/wiki/Recursión_(ciencias_de_computación) · es.wikipedia.org/wiki/Árbol_binario

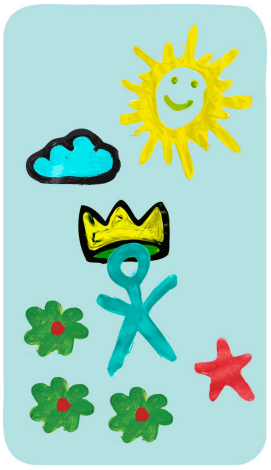
3 Cartas dibujadas

2024-IN-04 · India
Dificultad: Fácil

Manas hace dibujos en sus cartas siguiendo una única regla secreta. Ha creado cuatro cartas que cumplen la regla:









Anil observa las cartas y crea una carta nueva, pero Manas le dice que no cumple la regla.



¿Cuál de las siguientes podría ser la regla secreta que sigue Manas para dibujar las cartas?

A) Si está nublado, entonces no hay flores.
B) Tiene que haber una estrella roja o el hombre se come una pizza.
C) Si el hombre se come una pizza, entonces no lleva corona.
D) Tiene que haber un sol sonriente o una nube.

Respuesta correcta: C) Si el hombre se come una pizza, entonces no lleva corona.

Explicación

La respuesta correcta es la **C**.

La regla secreta tiene que cumplirse en las cuatro cartas de Manas y no cumplirse en la carta de Anil.

- **A** no puede ser: en la 2.ª y la 3.ª carta hay una nube y también flores.
- **D** no puede ser: la 4.ª carta no tiene ni sol sonriente ni nube.
- **B** no puede ser: las cuatro cartas la cumplen, pero la carta de Anil también (en ella el hombre se come una pizza), y Manas dice que esa carta no cumple la regla.
- **C** sí puede ser: ninguna de las cuatro cartas tiene a la vez pizza y corona, y la carta de Anil tiene las dos cosas, así que no cumple la regla.

Esto es informática

La tarea muestra la **lógica** básica que usa la informática para representar y procesar decisiones y observaciones. Los ordenadores siguen reglas claras para decidir qué hacer, de forma parecida a como observamos patrones y tomamos decisiones en la vida diaria.

Las reglas de las opciones son **condiciones** como las de los programas: «si… entonces…» (implicación) y «… o …» (disyunción). Un detalle importante: una regla «si A, entonces B» solo se incumple cuando A es verdadero y B es falso. Por eso, las cartas en las que el hombre no come pizza cumplen siempre la regla C, lleve o no corona. En programación, este tipo de reglas sirve, por ejemplo, para validar datos de entrada: un formulario puede rechazar un dato que no cumpla una condición.

Esto es pensamiento computacional

Descomposición: cada opción aísla una condición concreta y su consecuencia, y se puede comprobar carta a carta. **Abstracción**: hay que fijarse solo en los elementos que menciona cada regla (nube, flores, pizza, corona…) e ignorar el resto del dibujo. **Evaluación**: para descartar una regla basta con encontrar un contraejemplo, es decir, una carta de Manas que la incumpla o la carta de Anil si la cumple.

Palabras clave y enlaces: Lógica condicional, lógica booleana, implicación · es.wikipedia.org/wiki/Condicional_material

4 Regalo de cumpleaños

2024-NL-02 · Países Bajos
Dificultad: — (fácil en 8–10)

Fiona quiere una pelota por su cumpleaños.

Quiere que la pelota:

- no tenga rayas,
- tenga una estrella,
- no tenga una luna.

¿Qué pelota tenemos que comprarle a Fiona?

A)



B)



C)



D)



E)



F)



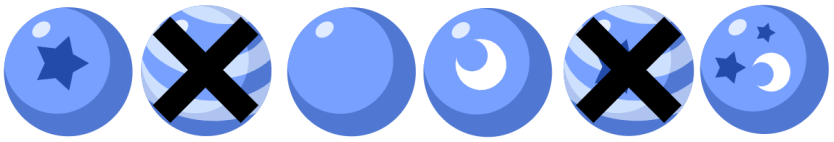
Respuesta correcta: A (la pelota con una estrella, sin rayas ni luna).

Explicación

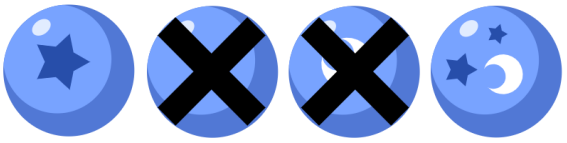
La respuesta correcta es la **A**.



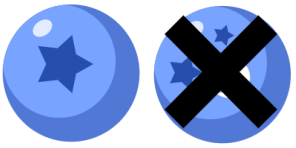
Podemos usar las reglas de Fiona para saber qué pelota quiere. Quiere una pelota sin rayas, así que quitamos las que tienen rayas:



Quiere una pelota con una estrella, así que quitamos también las que no tienen estrella:



Su última regla es: sin luna. Así que quitamos también la última pelota. La pelota con la estrella es la única que cumple las reglas de Fiona.



También se puede resolver mirando las pelotas una a una y comprobando si cada una cumple las tres condiciones de Fiona.

Esto es informática

La tarea usa un **algoritmo** muy sencillo para decidir cuál es la pelota correcta. Un algoritmo es un conjunto de reglas que hay que seguir para resolver un problema.

Se puede representar con un **diagrama de flujo**: con cada condición, el grupo de pelotas se divide en dos, las que la cumplen y las que no, y se repite hasta encontrar la solución. Esto es un **filtrado**, algo que hacen constantemente las tiendas en línea y los buscadores cuando eliges, por ejemplo, «talla M» y «color azul».

La otra forma de resolverlo, revisar las pelotas una a una, es un **bucle**: en programación, los bucles permiten repetir tareas de forma eficiente. Aquí hay que comprobar como máximo seis veces si se cumplen todas las condiciones.

Esto es pensamiento computacional

El **pensamiento algorítmico** es una forma de resolver problemas que usamos todos los días. Esta tarea es una introducción a cómo seguir un algoritmo y cómo resolver un problema paso a paso aplicando las reglas: primero una condición, luego otra, hasta que solo queda una opción.

Palabras clave y enlaces: Algoritmo, diagrama de flujo, filtrado, bucle · es.wikipedia.org/wiki/Diagrama_de_flujo

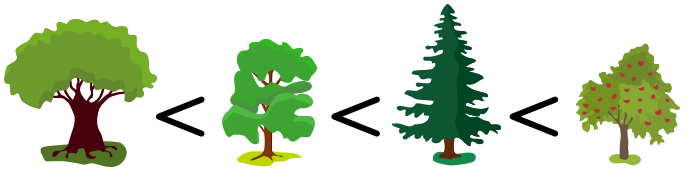
5 Rutas de árboles

2024-DE-08 · Alemania
Dificultad: — (media en 12–14)

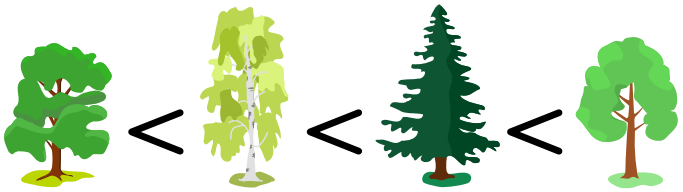
La bióloga Bea organiza rutas por el Bosque de los Castores. En cada ruta enseña algunos árboles especiales.

De tres rutas anteriores recuerda qué árboles gustaron más que otros a los visitantes (árbol 1 < árbol 2 significa que, en esa ruta, el árbol 2 gustó más que el árbol 1):

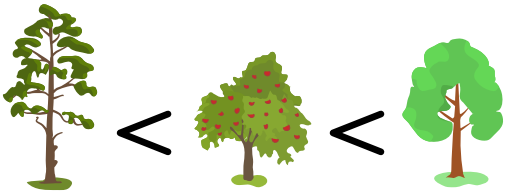
Ruta 1



Ruta 2



Ruta 3



En su próxima ruta, Bea quiere enseñar más tarde los árboles que más gustan. Quiere encontrar un **buen orden**, que no contradiga el orden de popularidad de ninguna de las rutas anteriores.

Por ejemplo, si Bea quiere enseñar los árboles  y  en un buen orden, debe enseñar  antes que . Enseñar  antes que  contradiría la ruta 1, en la que  gustó más que .

En la nueva ruta, Bea quiere enseñar estos árboles: 

Ayuda a Bea: coloca estos árboles en un buen orden.



Escribe el orden (1.º a 5.º):

Respuesta correcta: (roble, abedul, abeto, frutal, árbol claro).

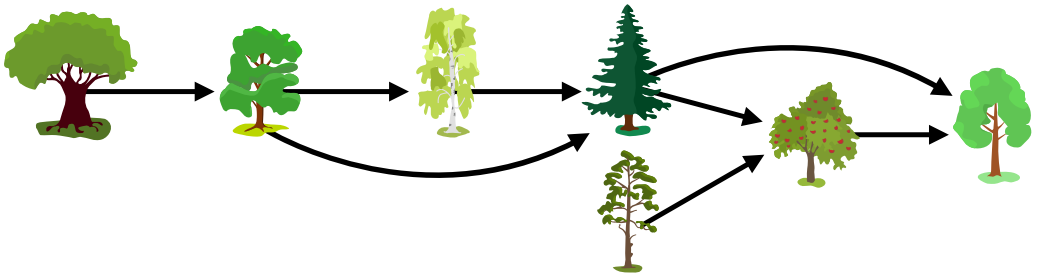
Explicación

La respuesta es:



Un buen orden no debe contradecir ninguno de los órdenes de popularidad de las tres rutas anteriores. Para evitarlo, Bea tiene que deducir el orden de la nueva ruta a partir de esos órdenes.

Entre los cinco árboles, es el único que no aparece como más popular que ningún otro, así que debe ser el primero. En la ruta 1, < … < < , así que debe ir después de . En la ruta 2, < < , así que va antes que y después. Y en la ruta 3, < . Todo ello se resume en este diagrama:



Por tanto, el orden es el único que respeta el diagrama.

Esto es informática

En esta tarea hay que ordenar un conjunto de árboles por popularidad creciente. Se sabe cómo se comparan algunas parejas de árboles, pero no todas: por ejemplo, el roble y el árbol claro no se comparan directamente en ninguna ruta. ¿Seguro que aun así se pueden ordenar?

Hay dos propiedades importantes:

1. La comparación de popularidad es **transitiva**: si árbol 1 < árbol 2 y árbol 2 < árbol 3, entonces árbol 1 < árbol 3.
2. No hay **conflictos** entre las rutas: si árbol 1 < árbol 2 en una ruta, no hay ninguna otra en la que árbol 2 < árbol 1.

Gracias a ellas, la comparación define un **orden**. A primera vista parece un orden **parcial**, porque no todas las parejas se comparan directamente. Convertir un conjunto parcialmente ordenado en una secuencia totalmente ordenada es la **ordenación topológica**, y en informática se conoce un algoritmo sencillo para hacerlo (que solo funciona si no hay conflictos). De hecho, si se mira con atención, el orden de los siete árboles es **total**: por transitividad se sabe cómo se comparan dos árboles cualesquiera, así que la ordenación de cualquier subconjunto tiene una única solución.

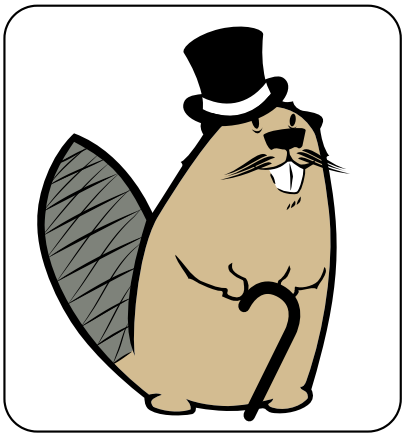
Esto es pensamiento computacional

Hay que **abstraer** la información de las tres rutas en una sola estructura (un diagrama de relaciones «va antes que»), usar el **razonamiento lógico** (la transitividad) para deducir comparaciones que no aparecen directamente y aplicar un **algoritmo**: elegir en cada paso el árbol que no tiene ningún otro pendiente por delante.

Palabras clave y enlaces: Ordenación topológica, orden parcial, transitividad · es.wikipedia.org/wiki/Ordenamiento_topológico

6 Alfred en el baile

2024-HU-02 · Hungría
Dificultad: Fácil



Cuando Alfred, un castor famoso, fue invitado al ayuntamiento, le hicieron esta foto.

Cuatro *influencers* escribieron en las redes sociales distintas afirmaciones sobre él. No todo lo que leemos en las redes sociales es verdad.

¿Cuál de las siguientes afirmaciones de los *influencers* es verdadera?

A) Alfred llevaba chistera pero no llevaba bastón

B) Alfred puede que llevara corbata o puede que llevara bastón

C) Alfred llevaba chistera y pajarita

D) Alfred llevaba pajarita o no llevaba bastón

Respuesta correcta: B) Alfred puede que llevara corbata o puede que llevara bastón.

Explicación

La afirmación verdadera es la **B**.

Hay que decidir, a partir de la foto, si cada afirmación es verdadera o falsa. Cada frase contiene dos afirmaciones unidas por los operadores lógicos Y (en la A, «pero» funciona como Y) y O.

p	q	p Y q
F	F	F
F	V	F
V	F	F
V	V	V

p	q	p O q
F	F	F
F	V	V
V	F	V
V	V	V

Una afirmación con Y solo es verdadera si las dos partes lo son; una con O es verdadera si al menos una parte lo es. En la foto, Alfred lleva chistera y bastón, pero no lleva corbata ni pajarita.

- **A)** llevaba chistera (V) pero no llevaba bastón (F): V Y F = **falso**.
- **B)** llevaba corbata (F) o llevaba bastón (V): F O V = **verdadero**.
- **C)** llevaba chistera (V) y pajarita (F): V Y F = **falso**.
- **D)** llevaba pajarita (F) o no llevaba bastón (F): F O F = **falso**.

Esto es informática

Y y O son **operadores lógicos** que permiten tomar decisiones combinando varias condiciones. Por ejemplo: si una persona tiene 18 años o más Y no lleva gafas, se le puede conceder el carné de conducir sin restricciones.

Los ordenadores son, en esencia, sistemas binarios. Los operadores lógicos (con 1 para verdadero y 0 para falso) les permiten realizar operaciones lógicas y matemáticas complejas, imprescindibles en muchos campos de la informática: la programación, el diseño de circuitos de control o las consultas a bases de datos. Las tablas que muestran el resultado de un operador para todas las combinaciones posibles se llaman **tablas de verdad**.

Esto es pensamiento computacional

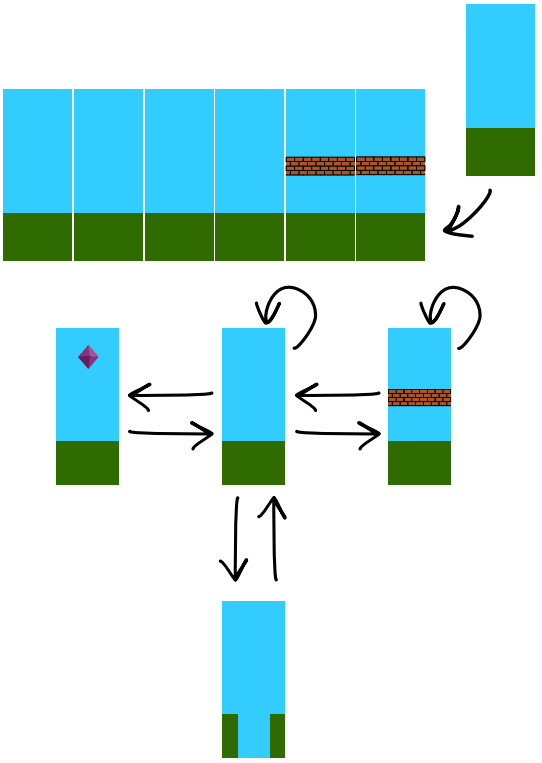
Se usa el **razonamiento lógico**: entender el problema, aplicar de forma sistemática un procedimiento (un pequeño **algoritmo**) para evaluar cada parte de la afirmación a partir de la foto y combinar los resultados según el operador. Es una forma de **evaluación**: comprobar con datos si una afirmación es cierta, algo muy útil también para no creer todo lo que se lee en las redes.

Palabras clave y enlaces: Lógica, operadores lógicos, tabla de verdad · es.wikipedia.org/wiki/Tabla_de_verdad

7 Superbebras

2024-DE-07 · Alemania
Dificultad: Difícil

En el videojuego Superbebras, el fondo es una secuencia de baldosas. El ordenador añade continuamente una baldosa nueva por la derecha y, a la vez, quita una por la izquierda. Así crea la ilusión de movimiento.

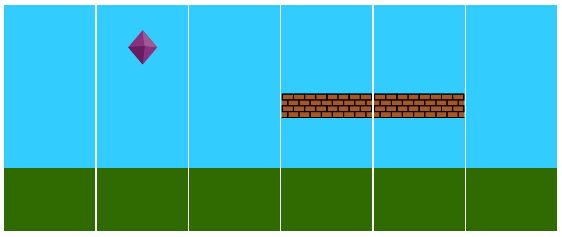


Para elegir la baldosa nueva, el ordenador usa el diagrama de la derecha: mira la baldosa anterior, comprueba las flechas que salen de ella y elige al azar una de las baldosas a las que apuntan.

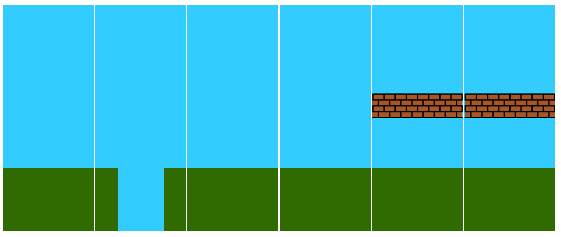
Por ejemplo, después de la baldosa  el ordenador puede elegir la baldosa  o la baldosa .

Una de las siguientes imágenes NO es un fondo válido de Superbebras. ¿Cuál?

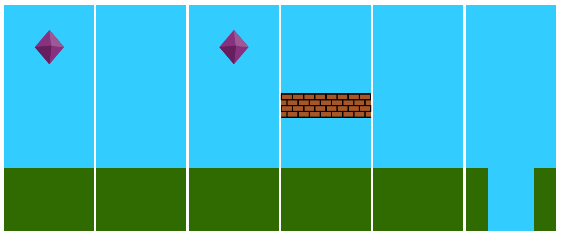
A)



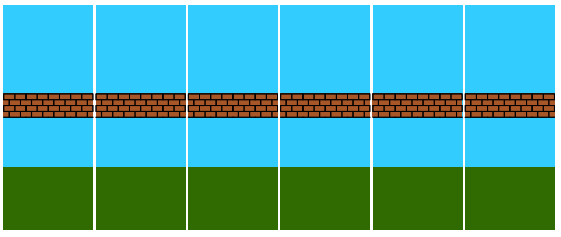
B)



C)



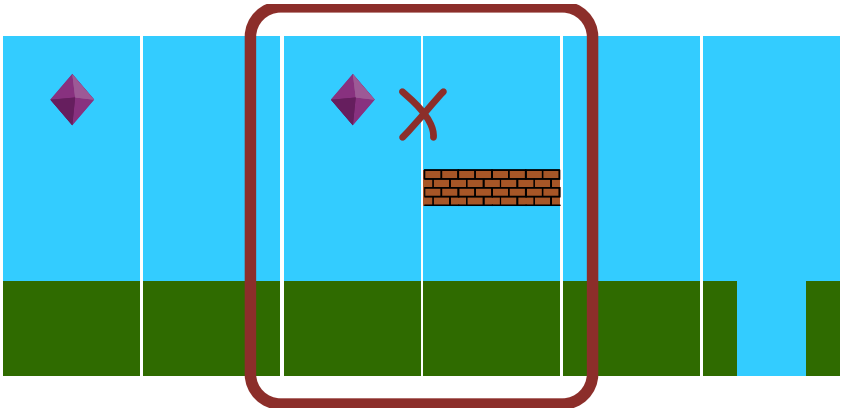
D)



Respuesta correcta: C.

Explicación

El fondo de la respuesta **C** no está construido del todo según el diagrama:



Una baldosa con un diamante solo puede ir seguida de , y aquí no es así.

Hay varias formas de encontrar la solución. Una estrategia directa es revisar cada fondo y comprobar cada baldosa con el diagrama. La más rápida es esta: empezar por las baldosas del diagrama de las que sale una sola flecha. Esas flechas son **restricciones**: A → B significa «a la baldosa A le tiene que seguir la baldosa B». Después se comprueba si cada fondo cumple esas restricciones.

Esto es informática

Algunos videojuegos, como los *endless runners* (juegos de carrera infinita), tienen un fondo que se desplaza horizontalmente y crea la ilusión de que el jugador avanza por un mundo de fantasía. A veces el fondo no es una imagen fija, sino que lo crea automáticamente el ordenador: es la **generación procedimental**. Normalmente se combinan al azar elementos pequeños para crear una enorme variedad de fondos.

Pero la combinación no puede ser totalmente aleatoria: debe cumplir ciertas reglas para no crear situaciones imposibles para el jugador. En la tarea, esas reglas se representan con un diagrama de baldosas y flechas, que es un **grafo dirigido**. Los grafos se usan para modelar todo tipo de situaciones; este en concreto funciona como un autómata que describe qué secuencias son válidas.

Esto es pensamiento computacional

Hay que leer y usar el grafo dirigido, que determina cómo se compone el fondo y es una **abstracción** del mecanismo que genera las imágenes. También hace falta **reconocimiento de patrones** para dividir visualmente cada fondo en baldosas y detectar las transiciones incorrectas. Por tanto, la tarea pone en juego el **pensamiento algorítmico**, la abstracción y el reconocimiento de patrones.

Palabras clave y enlaces: Grafo dirigido, generación procedimental, juegos de plataformas · es.wikipedia.org/wiki/Generación_por_procedimientos

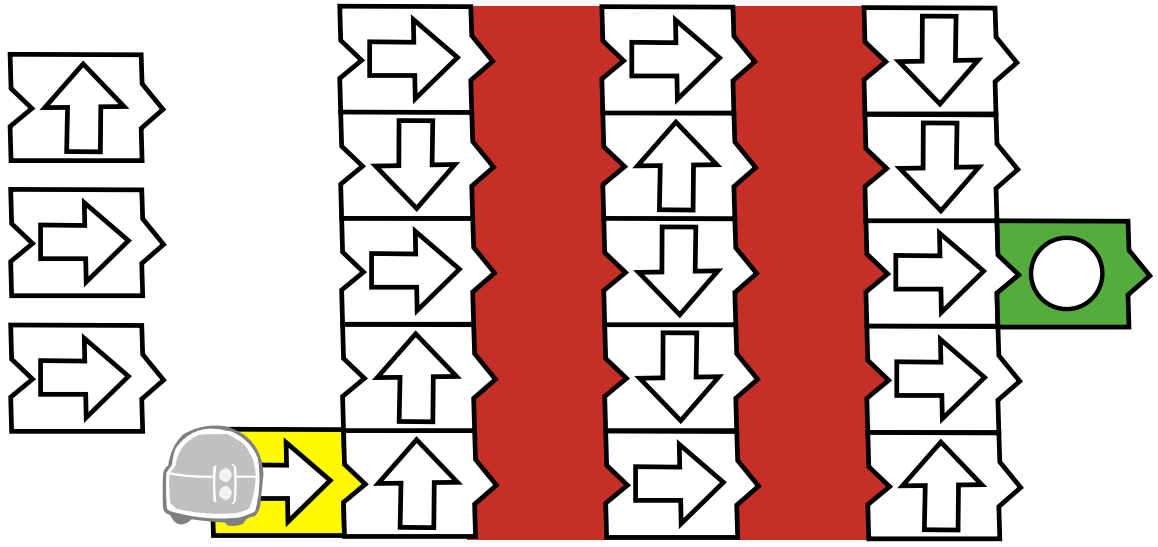
8 Un camino para el robot

2024-PL-04 · Polonia
Dificultad: Media

El robot de Tom se mueve sobre unas fichas siguiendo estas reglas:

- El robot empieza en la ficha amarilla.
- El robot avanza en la dirección de la flecha de la ficha en la que está.
- Si el robot se sale de las fichas, se detiene.

Las fichas ya están colocadas como se ve en la imagen. Tom tiene además 3 fichas extra (a la izquierda).



Ayuda a Tom: coloca las 3 fichas extra en las zonas vacías (rojas) para que su robot llegue a la ficha verde con el círculo.

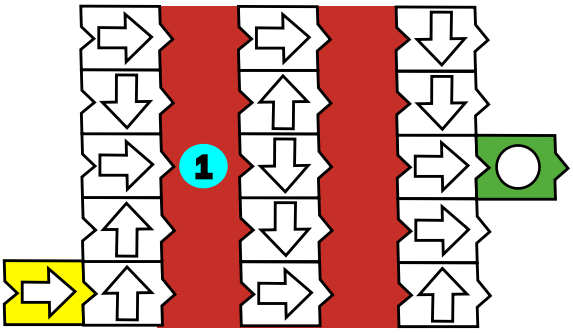
Dibuja en las zonas rojas de la imagen dónde colocarías cada una de las 3 fichas extra.

Respuesta correcta: ↑ en la 2.ª columna (3.ª fila), → en la 2.ª columna (2.ª fila) y → en la 4.ª columna (1.ª fila).

Explicación

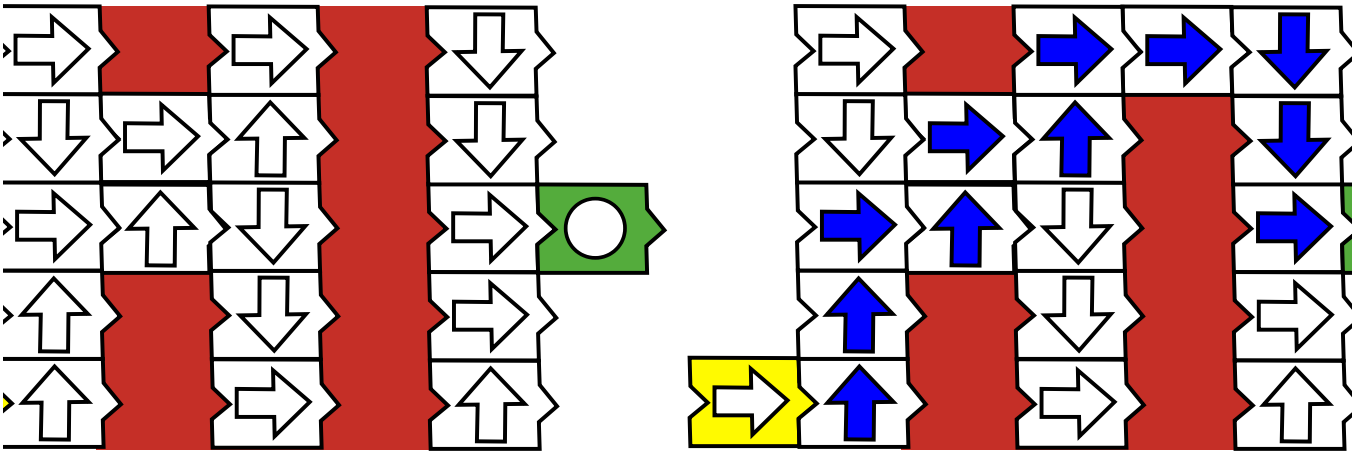
El objetivo es pasar de una columna a otra con las tres fichas hasta llegar al círculo. Solo hay una forma de hacerlo.

Para pasar de la primera columna a la segunda, el robot solo puede salir por la tercera ficha de la primera columna, así que la primera ficha se coloca en la posición marcada con 1.

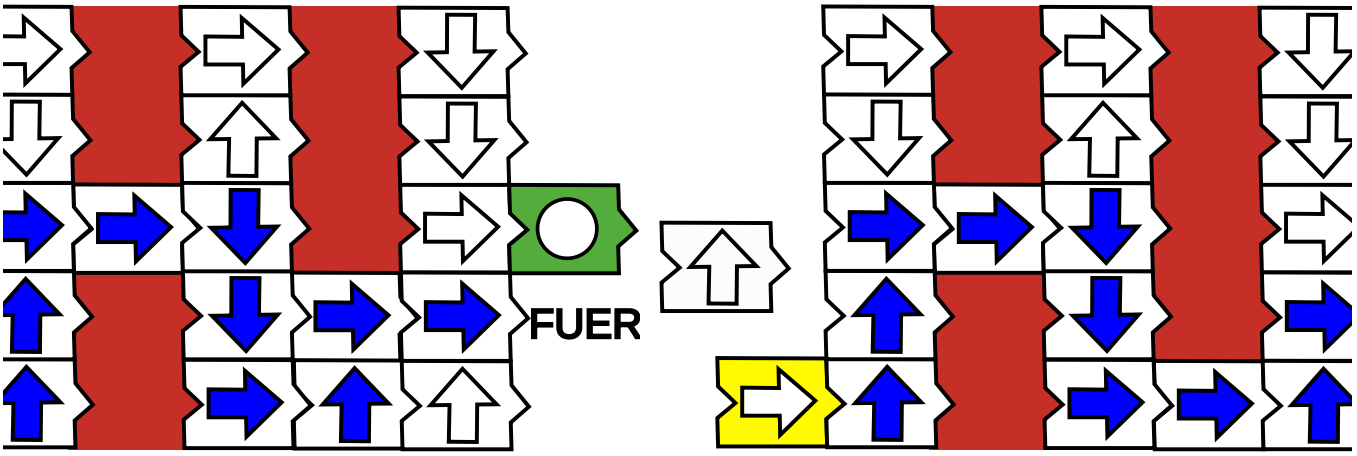


Desde ahí hay dos opciones: ir hacia arriba o hacia la derecha.

Opción 1: hacia arriba. Hace falta otra ficha para pasar a la segunda columna. Siguiendo las flechas de la segunda columna, el robot termina en su primera ficha. Con la última ficha cruza a la tercera columna y llega al círculo.



Opción 2: hacia la derecha. El robot llega a la ficha de abajo de la segunda columna y de nuevo hay dos posibilidades: a) usar la ficha hacia arriba y después la tercera ficha: el robot no llega al círculo y se sale; b) usar la ficha hacia la derecha: tampoco llega y, además, sobra una ficha, lo que va contra las reglas.



Esto es informática

En esta tarea las fichas dirigen el movimiento de un robot y hay que encontrar la ruta que lleva al final. Hay cosas que se saben de antemano: solo se pueden usar tres fichas (dos con flecha a la derecha y una hacia arriba) y solo en la zona roja, así que en cada

columna hay que encontrar una flecha a la derecha para avanzar. En informática esto se llama **restricciones**. Sin ellas el problema podría tener muchas más soluciones y sería más difícil, porque habría que pensar en muchísimos resultados posibles.

Por ejemplo, para ir de un lugar a otro puede haber muchas rutas, pero añadir algunas restricciones (pasar por una tienda, hacer parte del camino en autobús) reduce las opciones y ayuda a elegir la que mejor nos conviene. Los robots y los programas de navegación trabajan así.

Esto es pensamiento computacional

La tarea requiere sobre todo dos destrezas: **descomposición** y **evaluación**. Primero se descompone el problema en problemas más pequeños: en cada columna, cómo pasar a la siguiente. Así solo hay que pensar en una parte de la ruta y hay menos opciones. Después se evalúa el resultado de cada posibilidad y se descartan las que no llevan a una buena solución.

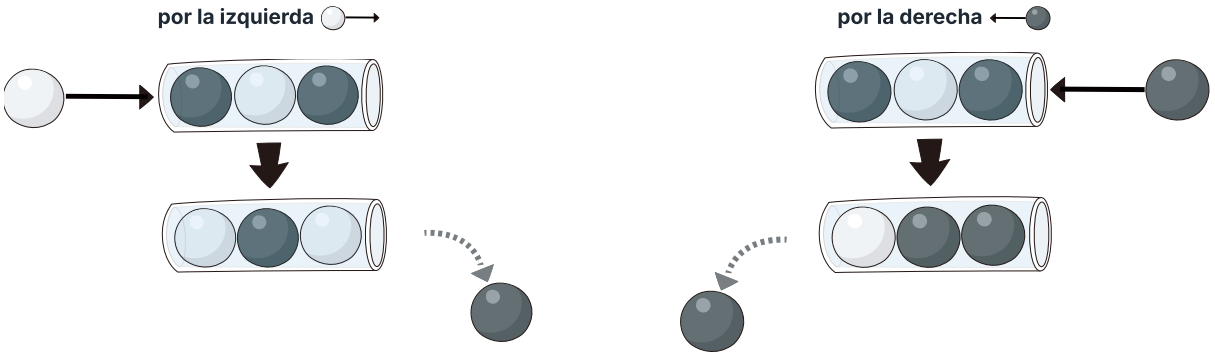
Palabras clave y enlaces: Restricciones, ruta, camino, robótica · es.wikipedia.org/wiki/Problema_de_satisfacción_de_restricciones

9 El tubo de bolas

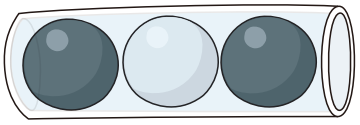
2024-KR-03a · Corea del Sur
Dificultad: Fácil

Hay un tubo  abierto por los dos lados en el que caben como máximo 3 bolas.

Estas imágenes muestran qué pasa cuando hay tres bolas en el tubo y se intenta meter una cuarta:



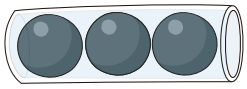
Ahora el tubo está así:



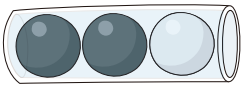
¿Cómo quedará el tubo después de meter 4 bolas en este orden?



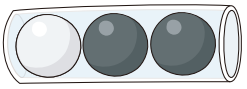
A)



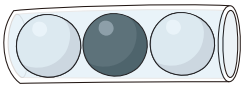
B)



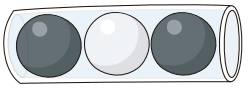
C)



D)



E)

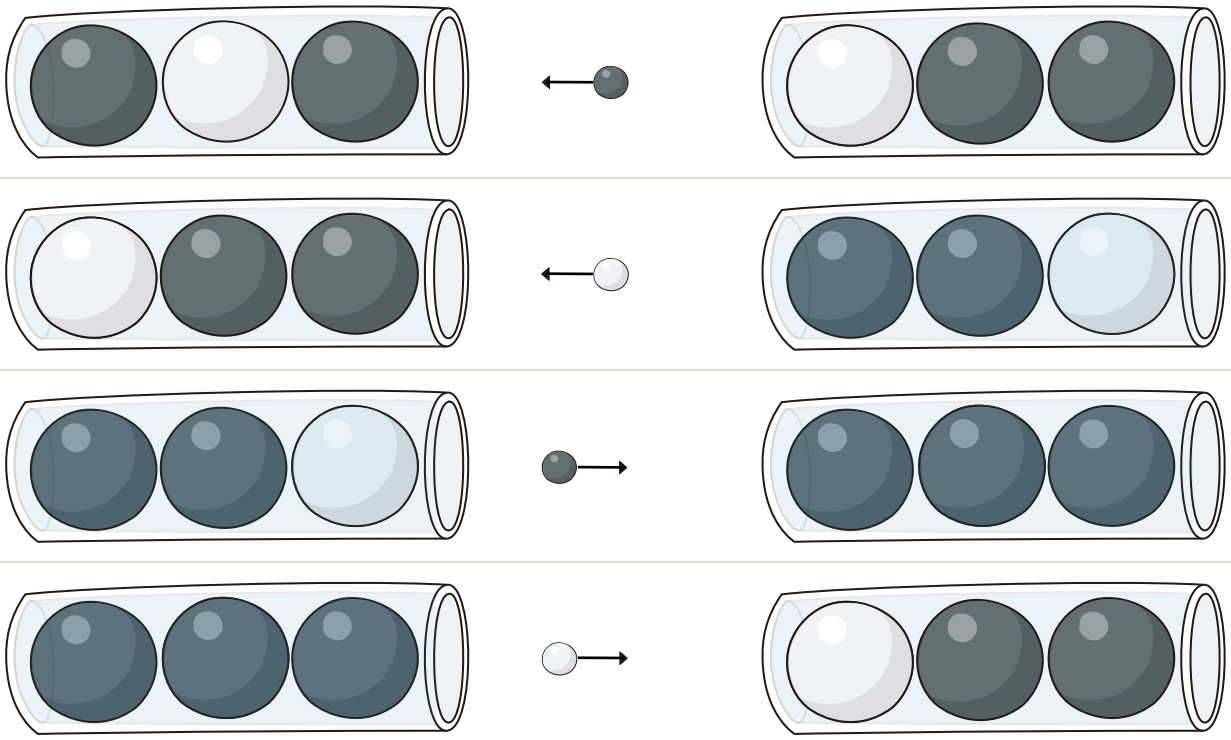


Respuesta correcta: C (blanca, negra, negra).

Explicación

La respuesta correcta es la **C**.

Hay que fijarse en por qué lado entra cada bola y seguir los pasos. Cuando el tubo está lleno, la bola nueva empuja a las demás y la del otro extremo se cae.



Esto es informática

En esta tarea, el tubo funciona de forma parecida a una **cola doble** (*deque*, del inglés *double-ended queue*), una estructura de datos que permite meter y sacar datos por los dos extremos.

Aquí tenemos una variante en la que solo se pueden meter datos (las bolas) y que tiene una **capacidad limitada**: en el tubo caben como máximo 3 bolas. Por eso, si se meten más datos de los que caben, se pierde información, igual que el tubo «pierde» una bola cuando se mete la cuarta. Es un aspecto importante del almacenamiento de datos en informática: los límites de capacidad. Por ejemplo, el historial de «deshacer» de muchos programas solo guarda las últimas acciones y va olvidando las más antiguas.

Esto es pensamiento computacional

Hay que pensar de forma **algorítmica**, paso a paso, al meter cada bola en el tubo hasta obtener la secuencia final: fijarse en el lado por el que entra cada bola, en cuántas bolas hay ya en el tubo y en la nueva secuencia, teniendo en cuenta que en cada paso puede caerse una bola. También exige **abstracción**, porque no se ve cómo entran y caen las bolas: hay que imaginarlo.

Palabras clave y enlaces: Cola doble (deque), capacidad de almacenamiento · es.wikipedia.org/wiki/Cola_doblemente_terminada

10 Días soleados

2024-DE-04a · Alemania
Dificultad: Media



El castor Tom dice: «Los días soleados, en todos los estanques hay al menos un castor nadando».

Su hermana Kim responde: «¡Eso no es verdad! El domingo pasado es un ejemplo que demuestra que te equivocas».

Supón que Kim tiene razón. Completa la frase:
El domingo pasado hizo sol y _____.

A) todos los estanques estaban llenos de castores nadando
B) ningún castor nadó en el estanque de la cascada helada
C) el castor Miguel nadó por todos los estanques
D) el castor Miguel no nadó nada

Respuesta correcta: B) ningún castor nadó en el estanque de la cascada helada.

Explicación

La solución es la **B**: ningún castor nadó en el estanque de la cascada helada.

En ese caso no todos los estanques tienen un castor nadando, porque el de la cascada helada no lo tiene. Además, sabemos que el domingo pasado hizo sol. Así que el domingo pasado es un ejemplo de día soleado con al menos un estanque sin castores nadando. Por tanto, Tom se equivoca al decir que todos los días soleados hay al menos un castor nadando en cada estanque.

Las demás opciones no demuestran que Tom esté equivocado:

- **A**: «Hizo sol y todos los estanques estaban llenos de castores nadando»: en todos los estanques había castores, lo que coincide con lo que dice Tom.
- **C**: «Hizo sol y el castor Miguel nadó por todos los estanques»: en todos había al menos un castor (Miguel), de nuevo de acuerdo con Tom.
- **D**: «Hizo sol y el castor Miguel no nadó nada»: puede que otros castores nadaran en todos los estanques, y entonces lo que dice Tom se seguiría cumpliendo.

Esto es informática

Tom y Kim discuten sobre el valor de verdad de la afirmación de Tom. Entran así en el terreno de la **lógica booleana**, en la que cada afirmación es verdadera o falsa. La lógica permite construir afirmaciones complejas con operadores como Y, O, NO y con **cuantificadores** como «para todo» y «existe al menos uno».

Para refutar una afirmación del tipo «para todo…» basta con un **contraejemplo**: un caso concreto que no la cumpla. La lógica ordena las discusiones y ayuda a evitar malentendidos. Además es la base de los ordenadores: los circuitos están hechos de puertas lógicas como NO e Y, y los lenguajes de programación incluyen condiciones lógicas como «SI es verano Y hace sol, ENTONCES enciende el aire acondicionado».

Esto es pensamiento computacional

Como los ordenadores se basan en la lógica, trabajar con ellos exige manejarla: traducir una situación del mundo real a una condición lógica, o encontrar la negación exacta de una condición para que un programa contemple todos los casos. Esta tarea entrena justamente eso, dentro del **análisis e interpretación de datos**. También hay algo de **descomposición**, porque conviene analizar cada opción por separado.

Palabras clave y enlaces: Lógica, cuantificador, contraejemplo · es.wikipedia.org/wiki/Cuantificador

11 Clase en línea

2024-MY-03 · Malasia
Dificultad: Fácil

La profesora Ava da una clase en línea desde su casa.

En la pantalla de su ordenador ve que se han conectado 9 alumnos: Emma, Maya, Bella, Lee, Raúl, Hannah, Diana, Alice y James. Cada uno usa un ordenador distinto de la biblioteca del colegio.



Como los alumnos están sentados uno al lado de otro en la biblioteca, en cada imagen Ava ve también a los compañeros que tiene al lado cada alumno. Así sabe que todos están sentados en fila.

¿Cuál de los 9 alumnos está sentado en el centro (en la 5.ª posición)?

A) Diana
B) Raúl
C) Lee
D) Bella
E) Hannah

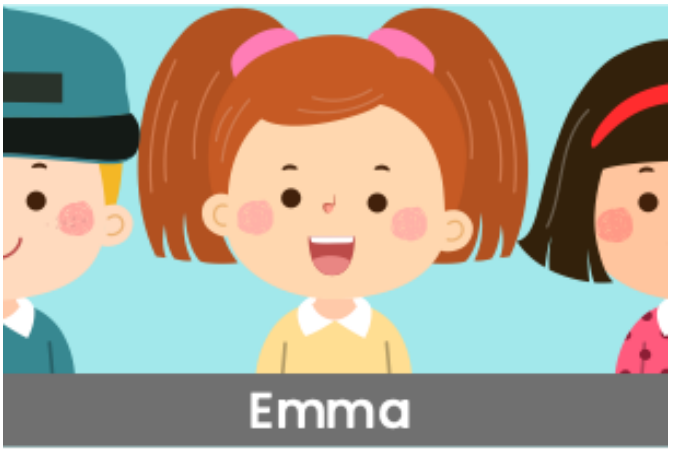
Respuesta correcta: E) Hannah.

Explicación

La respuesta es **Hannah**.

Por lo que se ve en la pantalla, sabemos quién está al lado de cada alumno. En la imagen de James vemos que se sienta junto a Emma, y en la de Emma vemos que está entre James y Diana:





Haciendo lo mismo con el resto de imágenes se obtiene el orden de los 9 alumnos:



`James · Emma · Diana · Lee · **Hannah** · Bella · Alice · Raúl · Maya`

Quien está en el centro ocupa la 5.ª posición contando desde la izquierda o desde la derecha: Hannah.

Esto es informática

En la tarea, el programa de videoconferencia muestra a los alumnos en recuadros colocados en un orden cualquiera, y hay que reconstruir el orden real para saber quién está en el centro.

La disposición de los alumnos es una estructura de datos llamada **lista doblemente enlazada**. En ella, cada nodo contiene un elemento, una referencia al nodo anterior y otra al siguiente. Aquí cada recuadro (nodo) muestra a un alumno (el elemento) y, a sus lados, a sus vecinos (las referencias). Resolver la tarea es **recorrer la lista** empezando por uno de los extremos, que se reconoce porque solo tiene un vecino.

Esto es pensamiento computacional

Hace falta **pensamiento lógico**: comprender el problema, idear un procedimiento (**algoritmo**) para ir encadenando la información de cada imagen y aplicarlo con orden. Es como montar un puzle: buscar primero las piezas de los bordes (aquí, los alumnos de los extremos) y avanzar pieza a pieza a partir de sus conexiones.

Palabras clave y enlaces: Lista doblemente enlazada, recorrido de listas · es.wikipedia.org/wiki/Lista_doblemente_enlazada

12 El cuadro de Bea

2024-FI-01 · Finlandia
Dificultad: Fácil

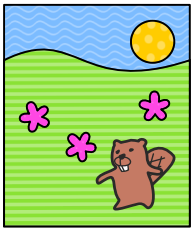
Bea empezó a pintar con cinco tubos de pintura llenos. En su cuadro, cuanto mayor es la superficie total de un color, más pintura de ese color ha gastado.

Cuando Bea terminó, esto es lo que quedaba de cada color en cada tubo:

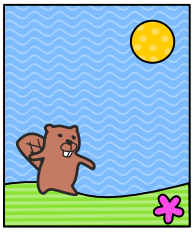


¿Cuál de estos cuadros podría ser el de Bea?

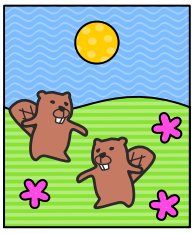
A)



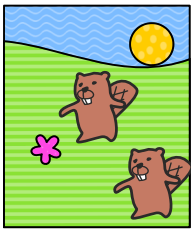
B)



C)



D)



Respuesta correcta: D.

Explicación

La respuesta correcta es la **D**. Se ve comparando la superficie total de cada color. El verde es el que más se usó; después el azul y el marrón, y luego, más o menos lo mismo, el amarillo y el rosa. Para cada color, la superficie total coincide con la cantidad de pintura que se ha gastado del tubo.

La opción **B** usa más azul que verde, y la **C** usa más o menos lo mismo de azul que de verde. Las dos son incorrectas, porque la cantidad de azul y de verde de los cuadros no coincide con lo que queda en los tubos.

Queda elegir entre la **A** y la **D**. En el tubo marrón queda mucha menos pintura que en el rosa, así que se ha usado más marrón que rosa. En la opción A hay menos marrón y más rosa, lo que no coincide con los tubos. Por tanto, la respuesta correcta es la D.

Esto es informática

Podemos ordenar los colores del cuadro de menor a mayor superficie total, y también los colores de los tubos de menor a mayor cantidad gastada. Las dos ordenaciones tienen que coincidir.

Colocar cantidades u objetos en orden se llama **ordenación**. Es uno de los problemas más fundamentales de la informática y existen muchos algoritmos para resolverlo. Cada uno tiene ventajas e inconvenientes, que empiezan a notarse cuando hay que ordenar muchísimos datos.

Esto es pensamiento computacional

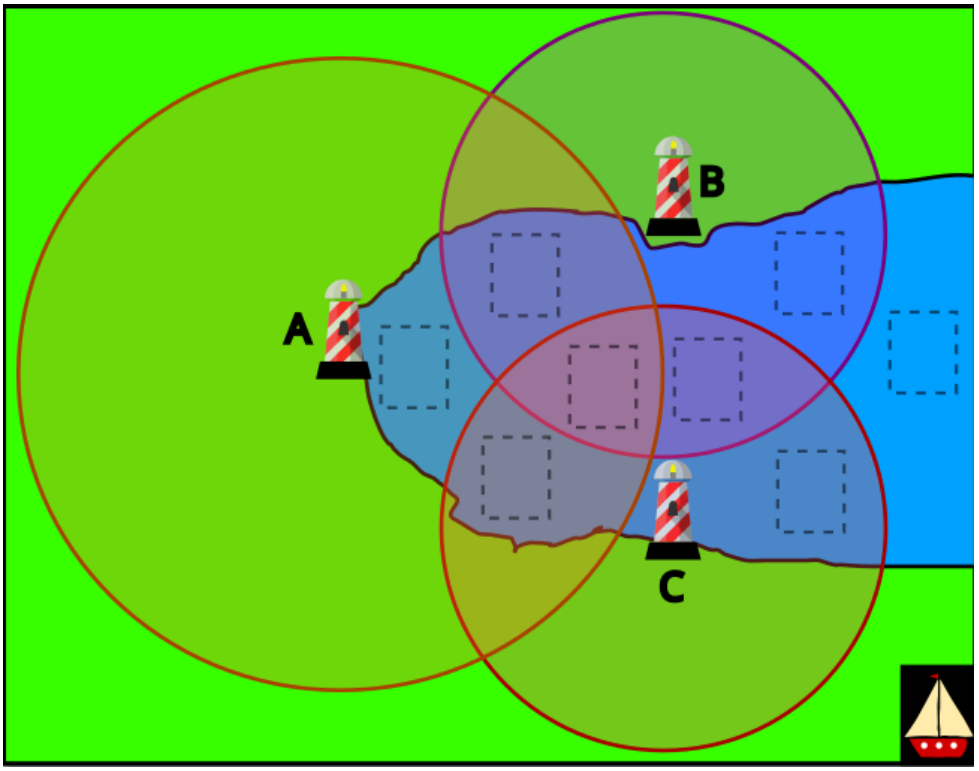
Hay que entender que la proporción de cada color en el cuadro está representada por la pintura que falta en cada tubo. Al estudiar el cuadro se usa la **descomposición**: dividir el conjunto en partes más pequeñas, aquí las cantidades de cada color. También ayuda la **evaluación**: descartar primero las opciones claramente incorrectas (las del azul y el verde) y fijarse después en los detalles.

Palabras clave y enlaces: Ordenación · es.wikipedia.org/wiki/Algoritmo_de_ordenamiento

13 El mapa de los faros

2024-CZ-05 · Chequia
Dificultad: Fácil

El capitán Ben está en su barco. Tiene un mapa del puerto, pero no sabe en qué lugar del mapa está. Los círculos del mapa indican desde dónde se ve la luz de cada faro. Con las luces, el capitán puede averiguar dónde está.



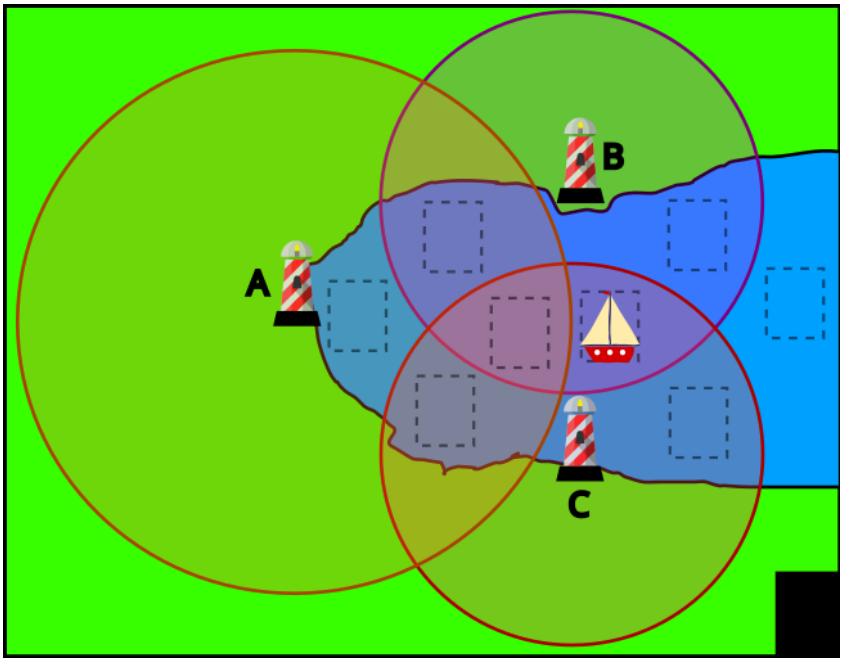
El capitán Ben ve la luz de los faros B y C. No ve la luz del faro A. Coloca el barco en el lugar del mapa donde está el capitán Ben.

Marca con una X en el mapa la casilla donde está el barco.

Respuesta correcta: La casilla del agua que está dentro de los círculos de B y de C, pero fuera del de A (entre los dos faros, a la derecha de la zona morada).

Explicación

El capitán solo ve las luces de los faros B y C. Por tanto, el barco tiene que estar dentro del círculo B y del círculo C, pero fuera del círculo A. Solo hay un lugar del mapa que cumple estas condiciones:



Esto es informática

En informática a menudo hay que describir relaciones entre objetos o conjuntos de objetos. Para ello no se usan descripciones con palabras, sino un sistema de símbolos y reglas llamado **álgebra de Boole**. Nuestro problema se escribiría así: B Y C Y NO A.

El dibujo de la tarea es un **diagrama de Venn**: usa círculos (u otras figuras) que se solapan para mostrar las relaciones lógicas entre dos o más conjuntos de elementos. Sirven para organizar las cosas de forma gráfica y ver en qué se parecen y en qué se diferencian. Los buscadores usan estas mismas operaciones cuando buscamos páginas que contengan unas palabras y no otras.

Esto es pensamiento computacional

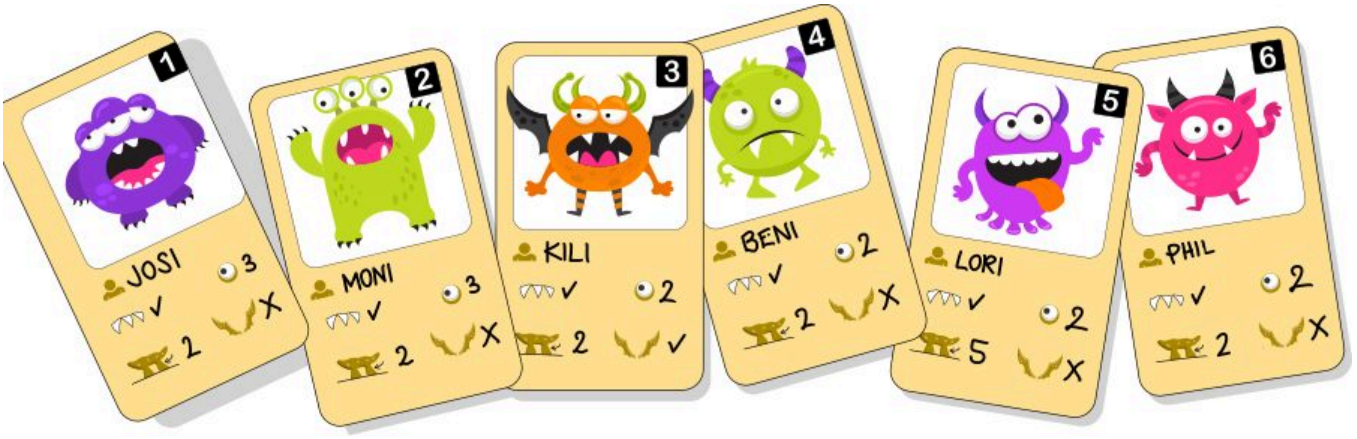
Hace falta **abstracción**: el diagrama recoge las relaciones importantes entre conjuntos y hay que saber orientarse en él. Además, se puede usar una estrategia de **evaluación**: revisar cada lugar posible y comprobar si cumple las tres condiciones (dentro de B, dentro de C, fuera de A).

Palabras clave y enlaces: Álgebra de Boole, conjunción, negación, diagrama de Venn · es.wikipedia.org/wiki/Diagrama_de_Venn

14 Las cartas Ricca

2024-CH-03a · Suiza
Dificultad: Media

Bárbara colecciona cartas de monstruos llamados «Riccas». Estas son algunas cartas de su enorme colección:



Los Riccas tienen **atributos**, como el nombre o los dientes . Cada Ricca es distinto, así que los **valores** de los atributos varían. Por ejemplo, en la carta 2 el valor del atributo es «MONI», el nombre del Ricca, y el valor del atributo es «✓», que indica que el Ricca tiene dientes.

Para tener una visión rápida de su colección, prepara una tabla en el ordenador. Pero ¿cómo anotar los valores de las cartas? A cada atributo le asigna uno de estos **tipos**:

no/sí valor «X» o «✓»
ABC... valor en forma de texto
123... valor en forma de número

Ayuda a Bárbara a terminar de preparar la tabla: indica el tipo de cada columna.

Escribe en cada casilla gris el tipo: no/sí, ABC... o 123...

Respuesta correcta: nombre: ABC… · ojos: 123… · patas: 123… · alas: no/sí · dientes: no/sí.

Explicación

Esta es la solución:

				
ABC…	123…	123…	no/sí	no/sí

Cada atributo tiene valores distintos en las cartas Ricca:

					
1	Josi	3	2	no	sí
2	Moni	3	2	no	sí
3	Kili	2	2	sí	sí
4	Beni	2	2	no	sí
5	Lori	2	5	no	sí
6	Phil	2	2	no	sí

Los valores «X» y «✓» de las alas  y los dientes  se representan con el tipo no/sí. Los nombres Josi, Moni, Kili, Beni, Lori y Phil se representan con el tipo ABC…. Y los valores 1, 2, 3… de los ojos  y las patas , con el tipo 123….

Si todos los valores se anotan correctamente y con el mismo tipo, la colección se puede ordenar por cualquiera de los atributos.

Esto es informática

Para guardar datos de forma organizada hay que pensar no solo en cómo estructurarlos, sino también en qué **tipo de dato** usar para cada valor.

Los ordenadores lo guardan todo con unos y ceros, que podrían representar cualquier cosa; los tipos de datos le dicen al ordenador cómo interpretarlos y manipularlos. En programación también sirven para comprobar que se usa la clase de información correcta. En la tabla de Bárbara hay tres tipos:

- ABC… **cadena de texto** (*string*): datos formados por caracteres de un alfabeto.
- 123… **número entero** (*integer*): se puede limitar qué números se aceptan; aquí solo tienen sentido los enteros, porque ningún Ricca tiene 3,5 patas.
- no/sí **booleano**: solo representa los valores verdadero o falso.

El tipo adecuado depende del uso que se quiera dar a los datos. Si alguien quisiera saber cuántas alas tiene cada Ricca, el booleano no bastaría y habría que usar también un número. Y hay datos que no se pueden ordenar, como el color, que puede representarse con palabras («morado») o con números, como los códigos hexadecimales de la web (#8E36D9) o las cartas RAL.

Esto es pensamiento computacional

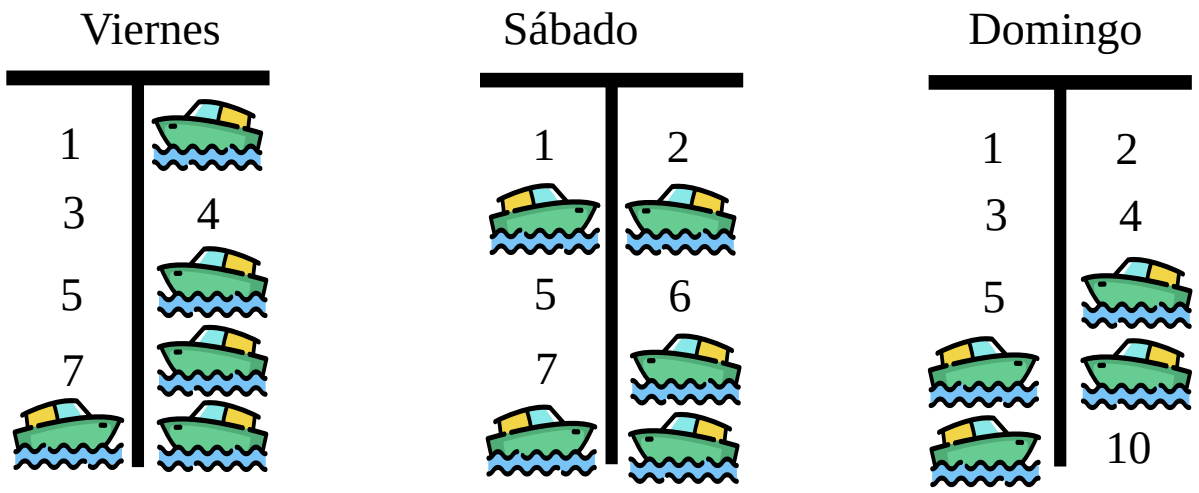
Abstracción: hay que filtrar la información relevante (los atributos) y asignar a los valores de cada uno un tipo de dato. Al crear una tabla o una base de datos es importante elegir bien los tipos para poder filtrar y buscar después. Saber montar una base de datos bien estructurada es muy útil, por ejemplo para gestionar el catálogo de una tienda o la producción de una fábrica.

Palabras clave y enlaces: Tipos de datos, cadena, entero, booleano · es.wikipedia.org/wiki/Tipo_de_dato · es.wikipedia.org/wiki/Tipo_de_dato_lógico

15 Plazas libres en el puerto

2024-MT-02 · Malta
Dificultad: Fácil

En un puerto deportivo hay diez amarres, que pueden estar reservados (con un barco) o libres, como se ve en la imagen. Por ejemplo, el amarre 1 está libre el viernes, pero el amarre 2 está reservado el viernes.



Tom tiene que elegir si llega el viernes o el sábado. Además, tiene que elegir un amarre que pueda reservar durante dos días seguidos. Por ejemplo, una opción es llegar el sábado y reservar el amarre 1.

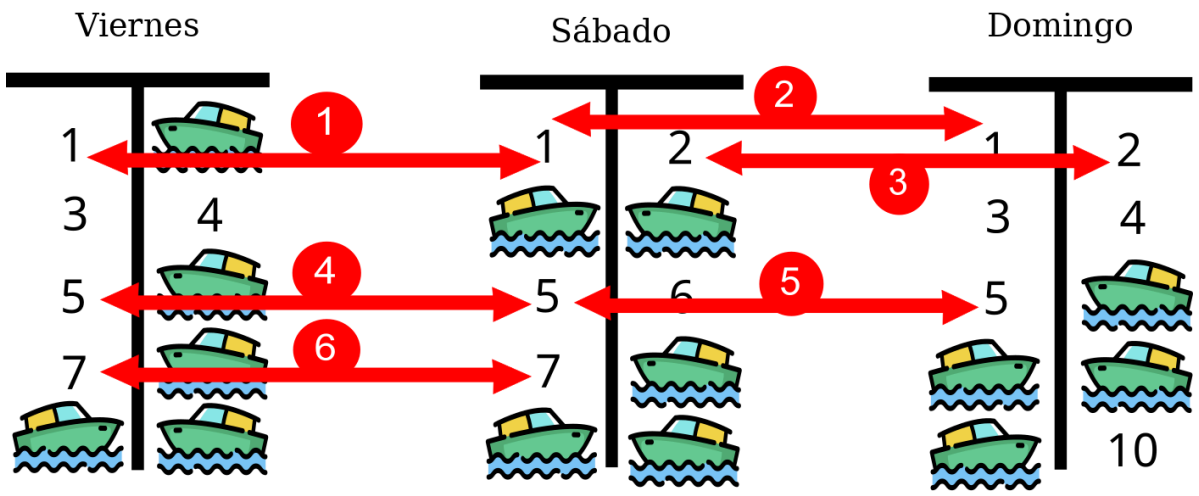
¿Cuántas opciones tiene Tom?

A) 4
B) 5
C) 6
D) 8
E) 10

Respuesta correcta: C) 6.

Explicación

La respuesta correcta es la **C) 6**. Estas son las seis opciones que tiene Tom:



- Llegando el viernes (viernes y sábado), están libres los dos días los amarres 1, 5 y 7.
- Llegando el sábado (sábado y domingo), están libres los dos días los amarres 1, 2 y 5.

En total, 3 + 3 = 6 opciones.

Nota: en el documento original la explicación indica la opción B (5), pero tanto su imagen de la solución como su propio cálculo dan 6, que es la respuesta correcta.

Esto es informática

Todos los datos se pueden ver como secuencias de ceros y unos. Cada cero o uno es un **bit**, y la secuencia es un código o número **binario**.

Aquí podemos representar un amarre libre con 0 y uno reservado con 1; cada amarre es un bit. Así, cada día se convierte en una secuencia de 10 bits (del amarre 1 al 10): viernes 0100010111, sábado 0011000111 y domingo 0000011110.

La tarea pide buscar, en dos días seguidos, los bits (amarres) que valen 0 en los dos. Eso corresponde al operador lógico **NOR** («ni... ni...»), que da 1 solo cuando las dos entradas son 0:

```
0100010111 NOR 0011000111 = 1000101000
0011000111 NOR 0000011110 = 1100100000
```

El número total de unos en los dos resultados (3 + 3) es la respuesta de la tarea.

Esto es pensamiento computacional

Hace falta **reconocimiento de patrones** para distinguir qué amarres están libres cada día y detectar las secuencias de libres y reservados a lo largo de los tres días. La **descomposición** ayuda a examinar cada amarre por separado, o cada pareja de días por separado, y sumar después los resultados.

Palabras clave y enlaces: Binario, bit, operador NOR · es.wikipedia.org/wiki/Puerta_NOR

Créditos y licencia

Las tareas proceden del taller internacional Bebras 2024 (Serbia). Cada tarea indica su código y el país que la propuso. Las imágenes son de los autores y autoras de las tareas originales.

Traducción y adaptación al castellano: Bebras España (bebras.ehu.es).

Copyright © Bebras – International Challenge on Informatics and Computational Thinking. Esta obra está bajo una licencia Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional (CC BY-SA 4.0): creativecommons.org/licenses/by-sa/4.0.