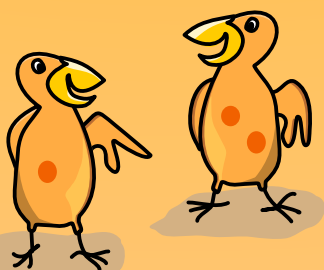




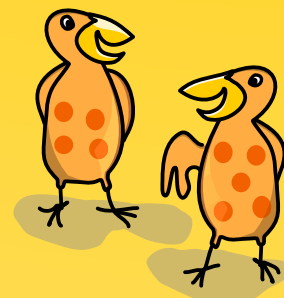
**INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA**

Exercices et solutions 2021

Années HarmoS 5/6



<https://www.castor-informatique.ch/>



Éditeurs :

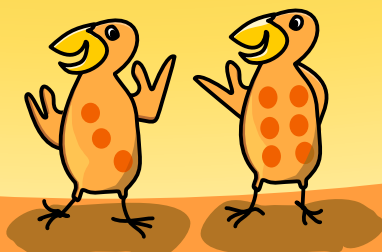
Susanne Datzko, Elsa Pellet, Jean-Philippe Pellet,
Fabian Frei, Gabriel Parriaux



010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SS!E

www.svia-ssie-ssii.ch
schweizerischerverein für informatik in d
erausbildung // société suisse pour l'infor
matique dans l'enseignement // società sviz
zera per l'informatica nell'insegnamento





Ont collaboré au Castor Informatique 2021

Masiar Babazadeh, Susanne Datzko, Fabian Frei, Martin Guggisberg, Gabriel Parriaux, Jean-Philippe Pellet

Cheffe de projet : Nora A. Escherle

Nous adressons nos remerciements pour le travail de développement des exercices du concours à :
JuraJ Hromkovič, Michael Barot, Christian Datzko, Jens Gallenbacher, Dennis Komm, Regula Lacher,
Peter Rossmann : ETH Zurich, Ausbildungs- und Beratungszentrum für Informatikunterricht
Bernadette Spieler : Pädagogische Hochschule Zürich

Le choix des exercices a été fait en collaboration avec les organisateurs de Bebras en Allemagne, Autriche, Hongrie, Slovaquie et Lituanie. Nous remercions en particulier :

Valentina Dagienė, Tomas Šiaulyš, Vaidotas Kinčius : Bebras.org

Wolfgang Pohl, Hannes Endreß, Ulrich Kiesmüller, Kirsten Schlüter, Michael Weigend : Bundesweite Informatikwettbewerbe (BWINF), Allemagne

Wilfried Baumann, Liam Baumann, Anoki Eischer, Thomas Galler, Benjamin Hirsch, Martin Kandlhofer, Katharina Resch-Schobel : Österreichische Computer Gesellschaft

Gerald Futschek, Florentina Voboril : Technische Universität Wien

Zsuzsa Pluhár : ELTE Informatikai Kar, Hongrie

Michal Winzcer : Université Comenius de Bratislava, Slovaquie

La version en ligne du concours a été réalisée sur l'infrastructure cuttle.org. Nous remercions pour la bonne collaboration :

Eljakim Schrijvers, Justina Dauksaite, Arne Heijenga, Dave Oostendorp, Andrea Schrijvers, Alieke Stijf, Kyra Willekes : cuttle.org, Pays-Bas

Chris Roffey : UK Bebras Administrator, Royaume-Uni

Pour le support pendant les semaines du concours, nous remercions en plus :

Hanspeter Erni : Direction, école secondaire de Rickenbach

Christoph Frei : Chragokyberneticks (Logo Castor Informatique Suisse)

Dr. Andrea Leu, Maggie Winter, Brigitte Manz-Brunner : Senarclens Leu + Partner AG

Ces brochures sont dédiées à la mémoire de Martin Guggisberg.

La version allemande des exercices a également été utilisée en Allemagne et en Autriche.

L'adaptation française a été réalisée par Elsa Pellet et l'adaptation italienne par Christian Giang.



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Le Castor Informatique 2021 a été réalisé par la Société Suisse pour l'Informatique dans l'Enseignement (SSIE) et soutenu par la Fondation Hasler.

HASLERSTIFTUNG

Cette brochure a été produite le 24 août 2022 avec le système de composition de documents \LaTeX . Nous remercions Christian Datzko pour le développement et maintien de la structure de génération des 36 versions de cette brochure (selon les langues et les degrés). La structure actuelle a été mise en place de manière similaire à la structure précédente, qui a été développée conjointement avec Ivo Blöchliger dès 2014. Nous remercions aussi Jean-Philippe Pellet pour le développement de la série d'outils `bebras`, qui est utilisée depuis 2020 pour la conversion des documents source depuis les formats Markdown et YAML.

Tous les liens dans les tâches ci-après ont été vérifiés le 1^{er} décembre 2021.



Les exercices sont protégés par une licence Creative Commons Paternité – Pas d'Utilisation Commerciale – Partage dans les Mêmes Conditions 4.0 International. Les auteur·e·s sont cité·e·s en p. 27.



Préambule

Très bien établi dans différents pays européens et plus largement à l'échelle mondiale depuis plusieurs années, le concours « Castor Informatique » a pour but d'éveiller l'intérêt des enfants et des jeunes pour l'informatique. En Suisse, le concours est organisé en allemand, en français et en italien par la SSIE, la Société Suisse pour l'Informatique dans l'Enseignement, et soutenu par la Fondation Hasler dans le cadre du programme d'encouragement « FIT in IT ».

Le Castor Informatique est le partenaire suisse du concours « Bebras International Contest on Informatics and Computer Fluency » (<https://www.bebas.org/>), initié en Lituanie.

Le concours a été organisé pour la première fois en Suisse en 2010. Le Petit Castor (années HarmoS 5 et 6) a été organisé pour la première fois en 2012.

Le Castor Informatique vise à motiver les élèves à apprendre l'informatique. Il souhaite lever les réticences et susciter l'intérêt quant à l'enseignement de l'informatique à l'école. Le concours ne suppose aucun prérequis quant à l'utilisation des ordinateurs, sauf de savoir naviguer sur Internet, car le concours s'effectue en ligne. Pour répondre, il faut structurer sa pensée, faire preuve de logique mais aussi de fantaisie. Les exercices sont expressément conçus pour développer un intérêt durable pour l'informatique, au-delà de la durée du concours.

Le concours Castor Informatique 2021 a été fait pour cinq tranches d'âge, basées sur les années scolaires :

- Années HarmoS 5 et 6 (Petit Castor)
- Années HarmoS 7 et 8
- Années HarmoS 9 et 10
- Années HarmoS 11 et 12
- Années HarmoS 13 à 15

Les élèves des années HarmoS 5 et 6 avaient 9 exercices à résoudre : 3 faciles, 3 moyens, 3 difficiles. Les élèves des années HarmoS 7 et 8 avaient, quant à eux, 12 exercices à résoudre (4 de chaque niveau de difficulté). Finalement, chaque autre tranche d'âge devait résoudre 15 exercices (5 de chaque niveau de difficulté).

Chaque réponse correcte donnait des points, chaque réponse fautive réduisait le total des points. Ne pas répondre à une question n'avait aucune incidence sur le nombre de points. Le nombre de points de chaque exercice était fixé en fonction du degré de difficulté :

	Facile	Moyen	Difficile
Réponse correcte	6 points	9 points	12 points
Réponse fautive	-2 points	-3 points	-4 points

Utilisé au niveau international, ce système de distribution des points est conçu pour limiter le succès en cas de réponses données au hasard.



Chaque participant·e obtenait initialement 45 points (ou 27 pour la tranche d'âge «Petit Castor», et 36 pour les années HarmoS 7 et 8).

Le nombre de points maximal était ainsi de 180 (ou 108 pour la tranche d'âge «Petit Castor», et 144 pour les années HarmoS 7 et 8). Le nombre de points minimal était zéro.

Les réponses de nombreux exercices étaient affichées dans un ordre établi au hasard. Certains exercices ont été traités par plusieurs tranches d'âge.

Pour de plus amples informations :

SVIA-SSIE-SSII Société Suisse pour l'Informatique dans l'Enseignement

Castor Informatique

Gabriel Parriaux

<https://www.castor-informatique.ch/fr/kontaktieren/>

<https://www.castor-informatique.ch/>



Table des matières

Ont collaboré au Castor Informatique 2021	i
Préambule	iii
Table des matières	v
1. Les tampons de Mika	1
2. Le bon maillot	5
3. Construction de pont	7
4. Cadeau favori	9
5. Porte-clés	11
6. Timber!	15
7. Chemins tortueux	17
8. Les moulins de castor Max	21
9. Jeu de balles	25
A. Auteur-e-s des exercices	27
B. Sponsoring : Concours 2021	28
C. Offres ultérieures	30



1. Les tampons de Mika

Mika a quatre tampons avec des images différentes. Elle prend chaque tampon dans sa main une fois et tamponne deux fois avec. Elle fait ainsi l'image suivante :




Quel tampon Mika a-t-elle utilisé en premier ?

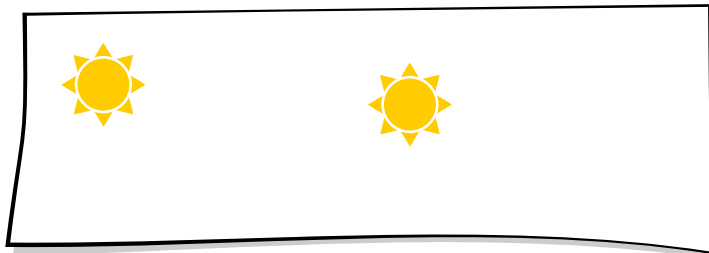




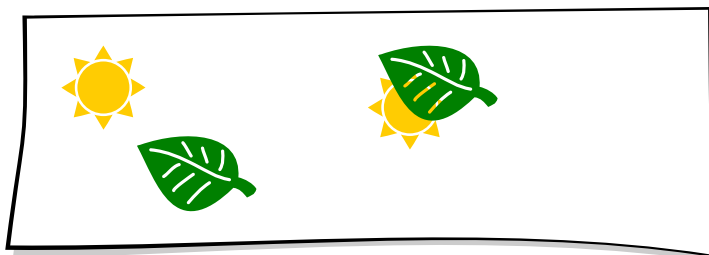
Solution

La bonne réponse est C: 

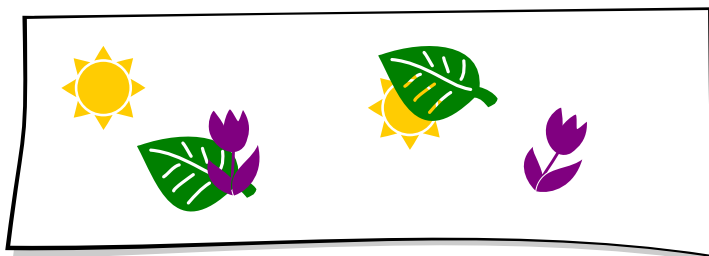
On peut reconnaître l'ordre dans lequel Mika a utilisé les tampons en regardant quelles images sont superposées à d'autres et quelles images sont en dessous d'autres images. Le soleil est en dessous de toutes les autres images : Mika a donc d'abord utilisé le tampon avec le soleil.



La feuille est superposée au soleil, mais est en dessous de la fleur et de la maison. Mika a donc utilisé le tampon avec la feuille en deuxième :



La fleur est superposée à la feuille et au soleil :



Mika ne peut pas avoir utilisé le tampon avec la fleur en dernier, car la fleur se trouve sous la maison à deux endroits. Mika a donc utilisé le tampon avec la fleur en troisième et celui avec la maison en dernier.





C'est de l'informatique !

Le dessin de Mika est une sorte d'image ou de *modèle* de la réalité utilisant des images tamponnées sur quatre plans : un plan avec des soleils, un plan avec des feuilles, un plan avec des fleurs et un plan avec des maisons. La superposition des images des différents plans permet à Mika de créer une illusion de profondeur et d'espace (tridimensionnel) sur une surface (bidimensionnelle) de papier.

Lorsque l'on fait un modèle, on ne représente en général que les aspects qui sont nécessaire à une certaine tâche ou une certaine fonction. La réalité est donc représentée de manière simplifiée. La modélisation est un principe important en informatique.

C'est ainsi que des programmes informatiques peuvent permettre d'analyser de manière rapide et précise des parties du monde réel et de mieux comprendre celles-ci. Pour créer un tel programme, il faut commencer par construire un modèle contenant les éléments du monde réel qui sont essentiels au problème étudié. Ce modèle permet de modéliser une partie du monde réel sous forme de croquis ou de programme. On peut ensuite obtenir de nouvelles connaissances sur une partie du monde réel à l'aide de ce programme.

Mots clés et sites web

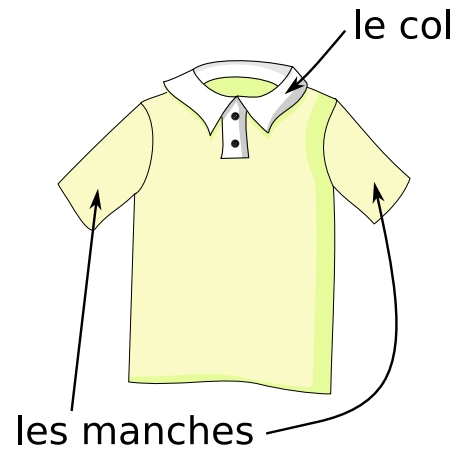
- Modélisation : <https://fr.wikipedia.org/wiki/Modélisation>
- Calques, traitement d'images : [https://fr.wikipedia.org/wiki/Calque_\(infographie\)](https://fr.wikipedia.org/wiki/Calque_(infographie))





2. Le bon maillot

Anne prépare son sac pour aller au match. Aujourd'hui, elle doit prendre le maillot avec des manches claires et un col noir, mais sans rayures.



Quel maillot Anne met-elle dans son sac ?





Solution



La bonne réponse est le maillot B.



Les maillots A et D ne vont pas aujourd'hui parce qu'ils ont des manches noires, et le noir n'est pas une couleur claire.



Le maillot C a des rayures et ne va donc pas pour le match d'aujourd'hui.

Le maillot B est parfait pour aujourd'hui : il a des manches claires, un col noir et n'a pas de rayures.

C'est de l'informatique !

Dans cet exercice du castor, tu devais trouver dans un ensemble d'objets celui qui remplit ou ne remplit pas certaines *conditions*.

Ici, plusieurs sous-conditions ont été définies, comme par exemple la couleur des manches et le motif du tissu, et ont été combinées pour former une condition globale. Pour ce genre de combinaisons, on utilise en informatique des *connecteurs logiques*.

Lorsque toutes les sous-conditions doivent être remplies en même temps, on utilise le connecteur *ET* : « la couleur des manches est claire » *ET* « le col est noir ». S'il suffit qu'au moins une des sous-conditions soit remplie, on utilise le connecteur *OU*. Si l'une des sous-conditions ne peut pas être remplie, on peut utiliser le connecteur *NON*, comme par exemple *NON* (le maillot a des rayures).

Pour effectuer des recherches dans des bases de données, des langages de requête peuvent être utilisés pour formuler des conditions très complexes. Pour cela, les conditions doivent être clairement définies. Par exemple, la condition que les manches doivent être claires n'est pas forcément très bien définie. En informatique, dans un cas comme celui-ci, on écrit un programme ou une fonction qui vérifie si une couleur est claire ou pas. Pour cela, il faut avoir une définition exacte de quand une couleur est claire, sinon, c'est impossible d'écrire un programme qui fonctionne.

Mots clés et sites web

- Algèbre de Boole : [https://fr.wikipedia.org/wiki/Algèbre_de_Boole_\(logique\)](https://fr.wikipedia.org/wiki/Algèbre_de_Boole_(logique))
- Connecteurs logiques : https://fr.wikipedia.org/wiki/Connecteur_logique



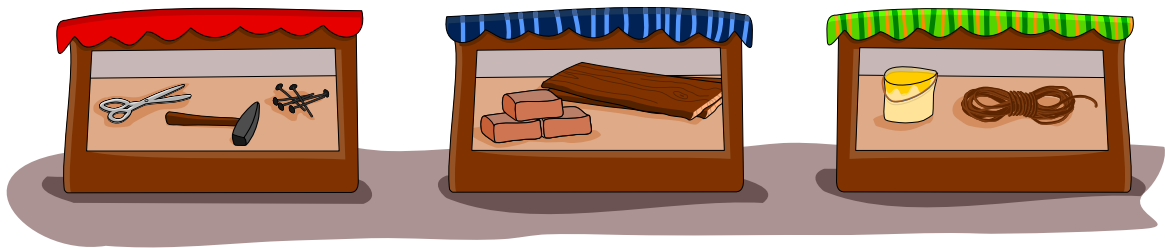
3. Construction de pont

Bella aimerait construire un pont par-dessus un ruisseau. Elle a besoin d'un marteau, de clous, de planches et d'une corde. Elle trouve un marteau et une corde à la cave.



Elle doit acheter les autres objets. En bas, tu vois trois magasins et ce qu'ils vendent.

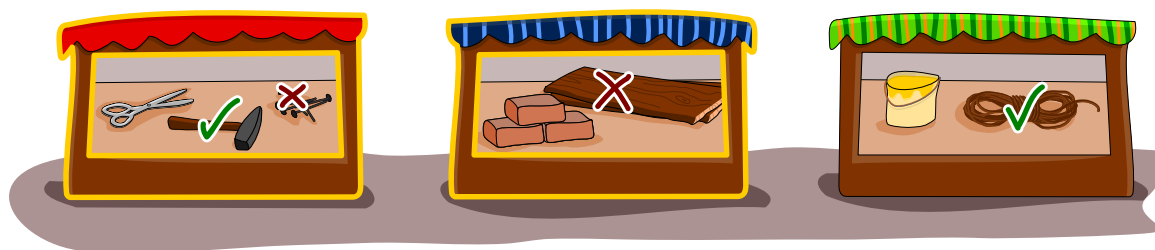
Dans quels magasins Bella peut-elle acheter les autres objets ?





Solution

La bonne réponse est :



C'est de l'informatique !

Dans cet exercice, les magasins vendent en tout sept objets différents : des ciseaux, un marteau, des clous, des briques, des planches, une corde et un seau. C'est un grand ensemble d'objets ! Les objets que Bella doit acheter sont une *partie de cet ensemble*. On peut le représenter de la façon suivante : on montre tous les objets de l'ensemble complet et note pour chaque objet s'il appartient au sous-ensemble que Bella doit acheter ou pas :



De la même manière, on peut représenter quels objets sont vendus dans les magasins, par exemple dans le magasin de gauche :



On voit ainsi au premier coup d'œil ce que Bella peut acheter dans le magasin de gauche : Les clous sont marqué d'une coche verte dans le sous-ensemble de Bella et dans le sous-ensemble du magasin.

Les programmes informatiques doivent aussi souvent comparer des ensembles. Pour chaque objet pouvant être présent, on a besoin d'un *bit*. Un ordinateur peut enregistrer une de deux valeurs dans un bit, comme par exemple « oui » ou « non ». Dans notre cas, on enregistre si un objet appartient à un ensemble (« oui ») ou pas (« non »). Un programme peut ensuite comparer deux ensembles de la façon suivante : il vérifie si le bit correspondant à un objet vaut « oui » dans un ensemble et vaut aussi « oui » dans l'autre ensemble. Une telle comparaison de deux bits est très rapide pour un ordinateur. C'est pour cela qu'en informatique, les ensembles sont souvent décrits en utilisant des bits.

Mots clés et sites web

- Ensemble : [https://fr.wikipedia.org/wiki/Ensemble_\(informatique\)](https://fr.wikipedia.org/wiki/Ensemble_(informatique))
- Bits : <https://fr.wikipedia.org/wiki/Bit>
- Tableau de bits : https://fr.wikipedia.org/wiki/Tableau_de_bits



4. Cadeau favori

La famille castor a trois cadeaux pour ses trois enfants. Chaque enfant indique d'abord son cadeau favori, puis son second choix. Les cadeaux doivent être bien distribués :

1. Le plus d'enfants possible doivent recevoir leur cadeau favori.
2. Les autres enfants doivent recevoir leur second choix.

Donne les bons cadeaux aux enfants.



1: , 2: 



1: , 2: 

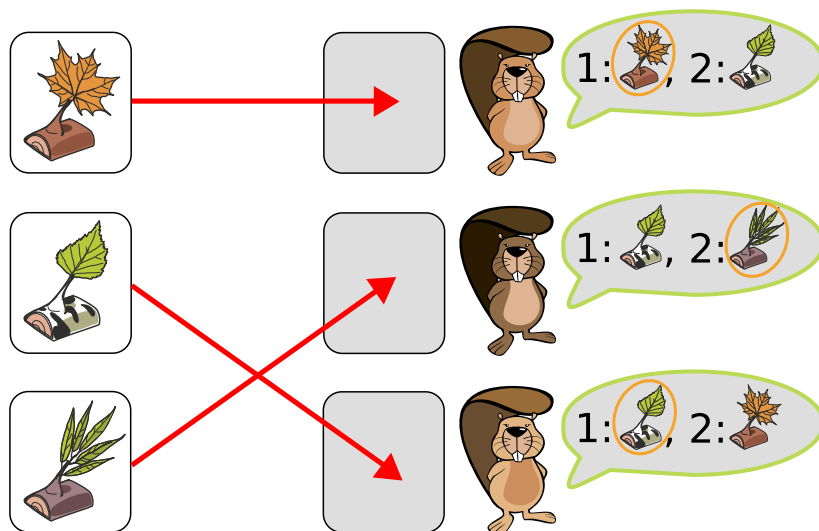


1: , 2: 



Solution

Voici la seule manière de distribuer les cadeaux en respectant les deux conditions.



Seul le deuxième castor désire le troisième cadeau, c'est donc lui qui doit le recevoir. Sinon, un autre castor recevrait un cadeau qui n'est ni son cadeau favori, ni son deuxième choix. La distribution des deux autres cadeaux est claire : chaque castor reçoit son cadeau favori.

C'est de l'informatique !

Dans cet exercice, nous avons affaire à un *problème d'affectation* univoque : nous voulons affecter les cadeaux de manière à ce que tous les enfants reçoivent un cadeau. Les enfants n'ont ici pas qu'un seul souhait, mais une liste de préférence. De tels problèmes d'affectation avec listes de préférence peuvent devenir très compliqués. L'informatique nous aide à résoudre de tels problèmes rapidement.

Une possibilité est de donner une valeur aux affectations : le cadeau favori a la valeur 1 et le deuxième choix la valeur 2. Un *couplage* (*matching* en anglais) est optimal s'il n'existe pas d'autre couplage avec plus de premiers choix distribués. Un tel couplage est appelé *couplage parfait de poids minimum*. Il existe beaucoup de problèmes d'affectation. L'un d'eux est appelé *problème des mariages stables* (*Stable Marriage Problem* en anglais). Intéressant ? L'informatique est une branche très variée !

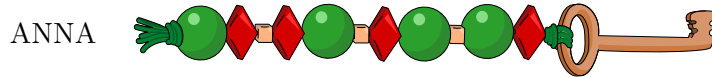
Mots clés et sites web

- Problème d'affectation : https://fr.wikipedia.org/wiki/Problème_d'affectation
- Couplage : [https://fr.wikipedia.org/wiki/Couplage_\(théorie_des_graphes\)](https://fr.wikipedia.org/wiki/Couplage_(théorie_des_graphes))

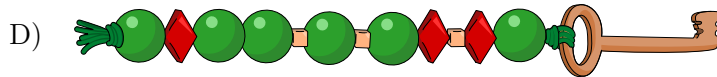
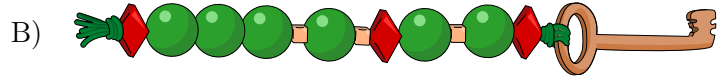
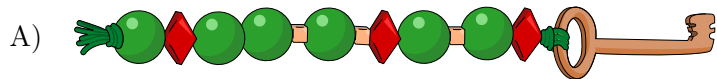


5. Porte-clés

ANNA, BELLA et LENA font des porte-clés avec leur nom. Elles utilisent deux sortes de perles pour les lettres : ● et ◆. Différentes lettres sont séparées par la perle ◻.



Quel porte-clés LENA a-t-elle fait ?





l'aide d'un télégraphe. Un point représente une courte durée de transmission et un trait une durée trois fois plus longue. Une pause sépare les lettres, et une pause plus longue sépare les mots. L'alphabet Morse est encore utilisé aujourd'hui pour le signal SOS. Un SOS en Morse $\bullet\bullet\bullet - - - \bullet\bullet\bullet$ (3x court, 3x long, 3x court) peut être transmis facilement en criant, tapant ou avec une lampe de poche.

Dans le traitement de données informatiques, les caractères sont encodés par des valeurs numériques pour être transmis ou enregistrés.

Mots clés et sites web

- Codage des caractères : https://fr.wikipedia.org/wiki/Codage_des_caractères
- Alphabet Morse : https://fr.wikipedia.org/wiki/Code_Morse_international



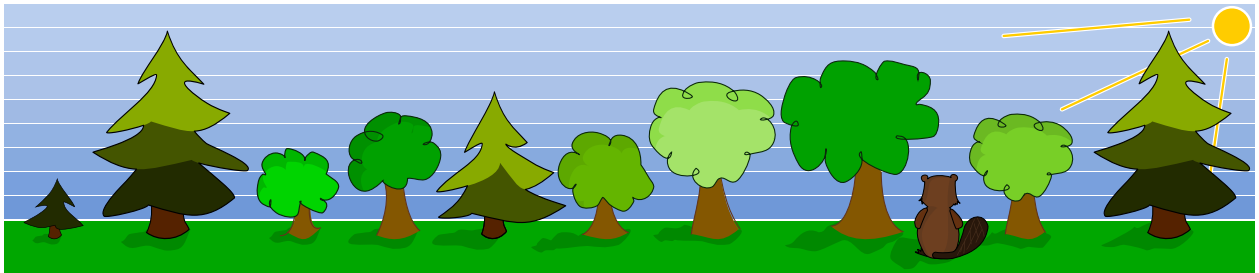


6. Timber !

Un castor aimerait construire un barrage. Afin de toujours abattre les bons arbres, il s'est fixé deux règles. Il n'abat un arbre que si :

- un arbre plus petit pousse directement à sa gauche et
- un arbre plus grand pousse directement à sa droite.

Quels arbres le castor va-t-il abattre ?





Solution

Seuls les arbres à la quatrième et septième positions remplissent les conditions données: il y a un arbre plus petit directement à gauche ET un arbre plus grand directement à droite.



C'est de l'informatique !

En informatique, il faut souvent résoudre des problèmes qui sont spécifiés par une série de *contraintes* logiques. La tâche consiste à trouver une solution qui respecte toutes les contraintes. Des problèmes plus complexes que celui-ci peuvent être résolus en utilisant des *opérateurs logiques* pour combiner les contraintes. La conjonction (\wedge ou encore opérateur ET) donne par exemple le résultat « vrai » pour l'expression $A \wedge B$ lorsque les deux contraintes A et B sont vraies. Dans cet exercice, ce serait donc: « l'arbre de gauche est plus petit » \wedge « l'arbre de droite est plus grand ». On retrouve ce principe fondamental dans presque tous les domaines de l'informatique, par exemple dans beaucoup d'algorithmes de tri comme le *tri à bulles* lors duquel la satisfaction aux contraintes de plusieurs objets d'une liste est évaluée avant de les déplacer, si nécessaire, à une autre position. Ce procédé est répété jusqu'à ce que la liste soit complètement triée.

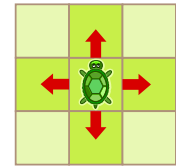
Mots clés et sites web

- Pensée algorithmique (*algorithmic thinking*)
- Opérateur logique: https://fr.wikipedia.org/wiki/Connecteur_logique
- Tri à bulles: https://fr.wikipedia.org/wiki/Tri_à_bulles
- Tri: <https://sorting.at/>
- Problème de satisfaction de contraintes:
https://fr.wikipedia.org/wiki/Problème_de_satisfaction_de_contraintes



7. Chemins tortueux

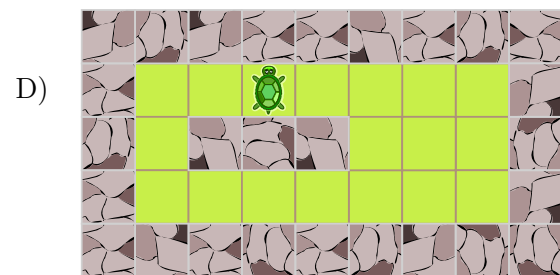
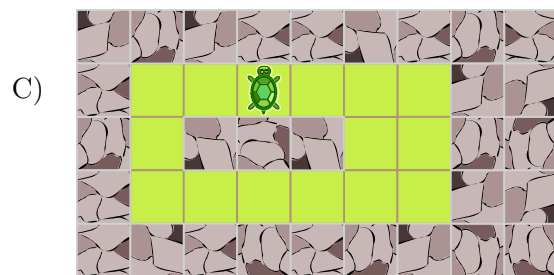
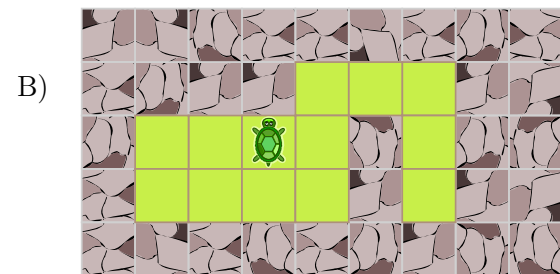
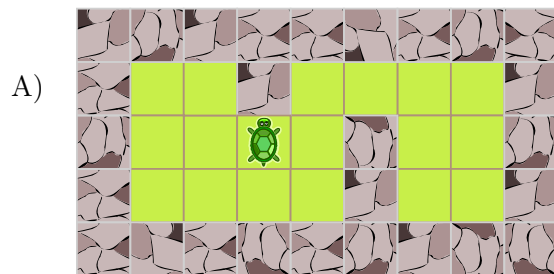
Une tortue doit brouter plusieurs jardins. Chaque jardin est divisé en carrés qui sont recouverts soit de gazon, soit de pierres. La tortue ne peut pas traverser un carré avec des pierres, mais elle peut passer d'une case d'herbe à une autre case d'herbe qui se trouve directement à côté.

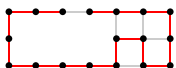


La tortue doit complètement brouter les jardins. elle commence sur la case sur laquelle elle est sur l'image. À la fin, elle doit avoir passé exactement une fois sur chaque case d'herbe.

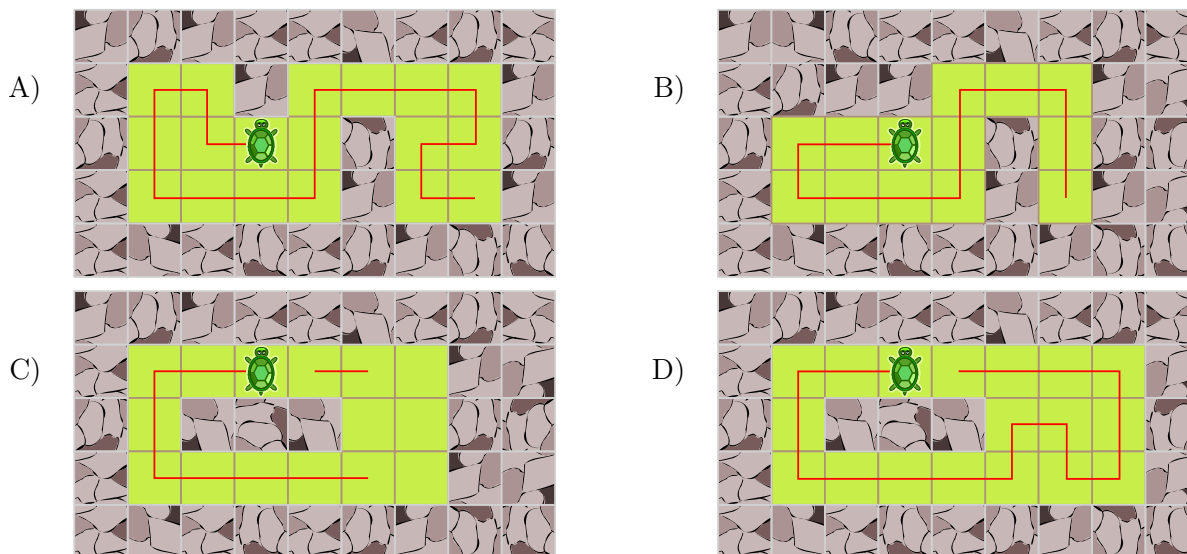
Il y a malheureusement un jardin qu'elle ne peut pas brouter complètement de cette façon.

De quel jardin s'agit-il ?





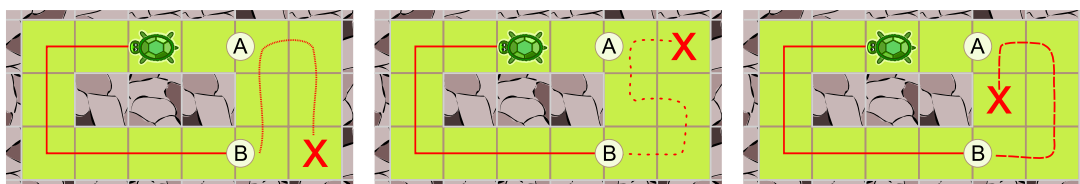
Solution



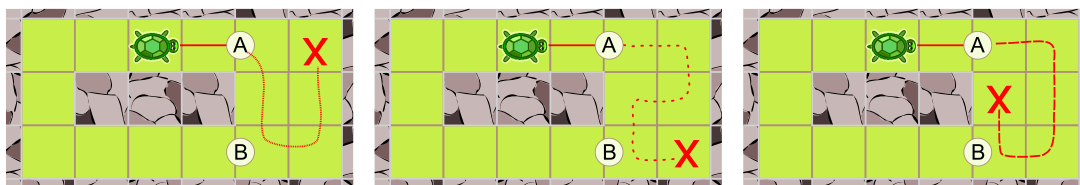
La tortue peut brouter les jardins A, B et D complètement.

La tortue ne peut pas brouter le jardin C de cette façon. La tortue a deux possibilités depuis son point de départ :

- Si elle part d'abord vers la gauche, elle arrive au point B. Depuis là, elle devrait brouter les 6 cases de droite de manière à terminer au point A, mais aucun des chemins possibles depuis le point B ne se termine au point A.



- Si la tortue part d'abord vers la droite, elle arrive au point B et devrait brouter les 6 cases de manière à atteindre le point A à la fin. On peut utiliser le même argument que si dessus en inversant le haut et le bas. Il n'y a donc pas non plus de chemin adapté.

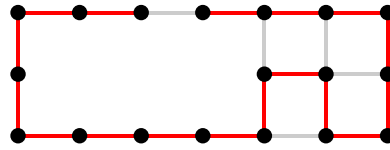


C'est de l'informatique !

La tortue doit trouver un chemin à travers un jardin en passant exactement une fois par chaque case d'herbe. Le problème de cet exercice est un problème du type *chemin hamiltonien*



Les carrés d'herbes du jardin de la tortues peuvent être considérés comme des *nœuds*. On représente alors le jardin D comme ceci :



Au XIX^e siècle, Sir William Hamilton se demanda s'il existait pour de telle structures (appelées *graphes* en informatique et en mathématiques) un chemin les long des arêtes passant exactement une fois par chaque nœud. C'est pour cela que l'on appelle un tel chemin un *chemin hamiltonien*. La question de savoir si un chemin hamiltonien existe dans un certain graphe est en règle générale très difficile à résoudre. Personne ne connaît d'*algorithme* permettant de déterminer si un chemin hamiltonien existe dans un graphe quelconque de manière efficace (assez rapidement pour que ce soit utile). Cela est vrai de tous les problèmes dits *NP-complets*, dont le problème du chemin hamiltonien est le plus connu.

Mots clés et sites web

- Graphe hamiltonien, chemin hamiltonien :
https://fr.wikipedia.org/wiki/Graphe_hamiltonien



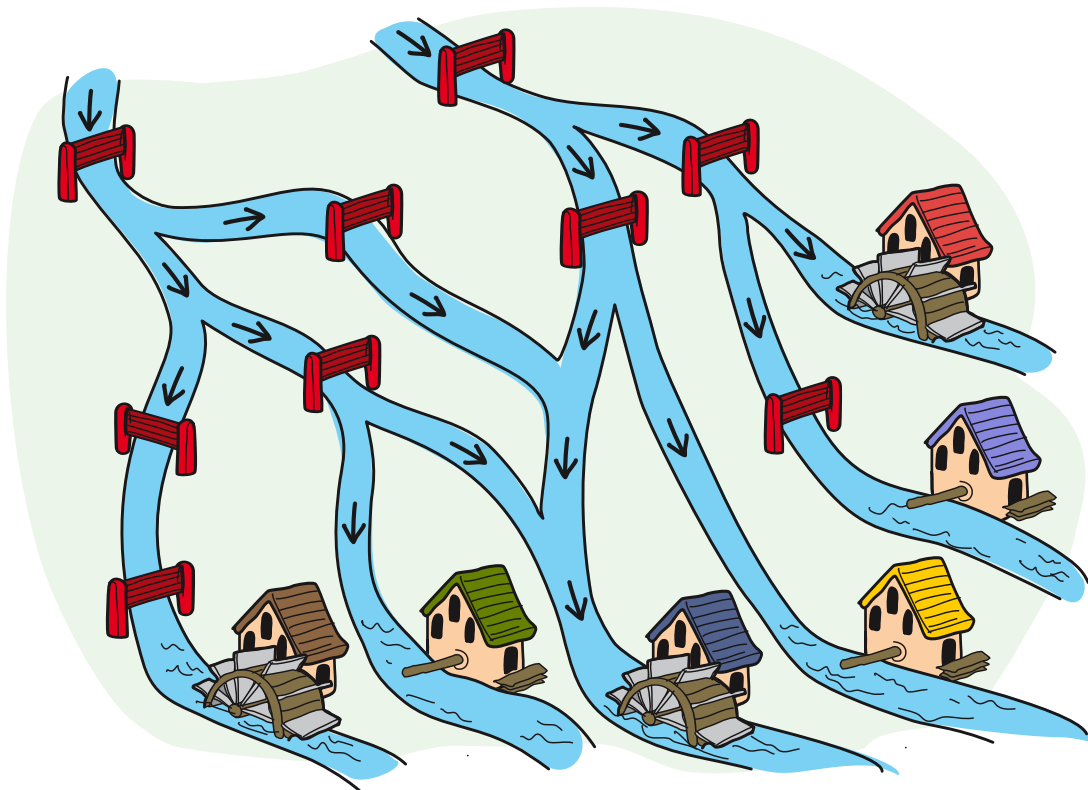


8. Les moulins de castor Max

Le meunier Max a six moulins. Il doit encore fixer la roue de trois d'entre eux. Pour cela, il doit empêcher l'eau d'arriver à ces moulins. L'eau doit par contre continuer de couler jusqu'aux autres moulins.

L'eau ne peut couler que vers le bas. Un clapet fermé empêche l'eau de couler.

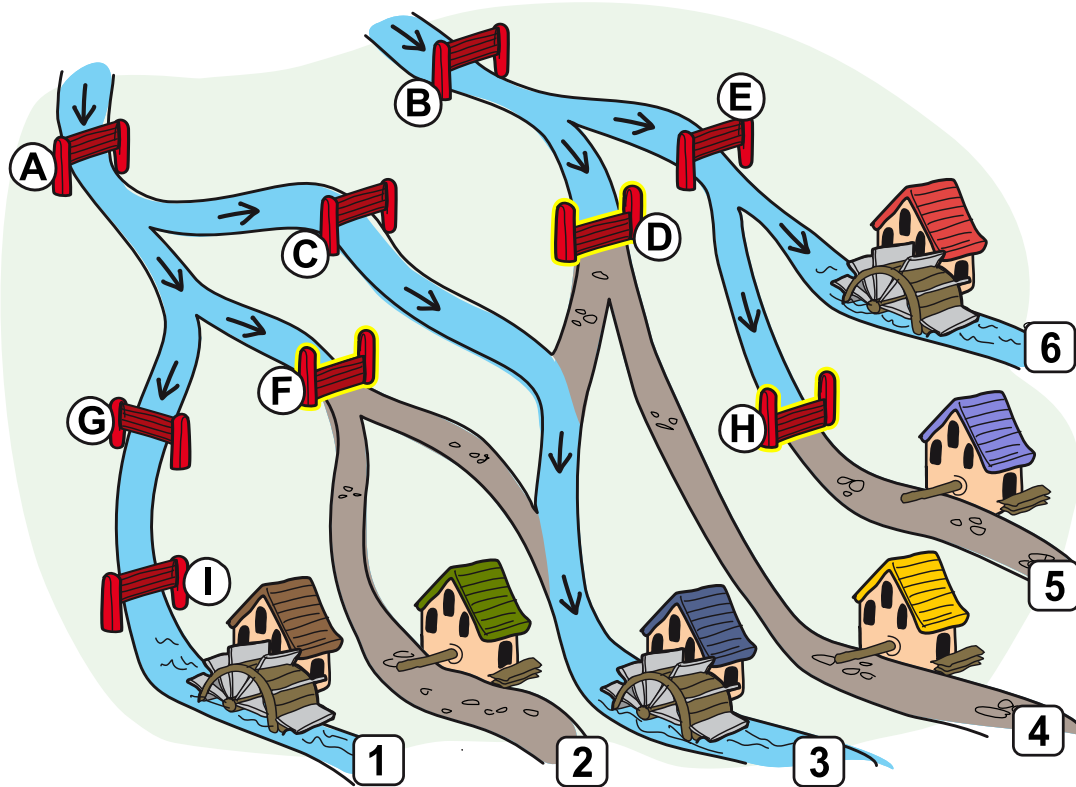
Quels clapets faut-il fermer ?





Solution

La bonne réponse est qu'il faut fermer les trois clapets qui sont nommés D, F et H dans le dessin suivant.



C'est la seule possibilité permettant de couper l'eau aux moulins 2, 4 et 5 tout en continuant d'alimenter en eau les trois moulins 1, 3 et 6 :

- Les clapets A, G et I doivent rester ouverts, car sinon le moulin 1 n'est pas alimenté en eau.
- Les clapets B et E doivent également rester ouverts pour que le moulin 6 soit alimenté en eau.
- Comme les clapets B et E sont ouverts, le clapet H doit être fermé pour éviter que l'eau n'arrive au moulin 5.
- Comme le clapet A est ouvert, le clapet F doit être fermé pour éviter que l'eau n'arrive au moulin 2.
- Comme le clapet B est ouvert, le clapet D doit être fermé pour éviter que l'eau n'arrive au moulin 4.
- Comme les clapets D et F sont fermés, le clapet C doit être ouvert pour que le moulin 3 soit alimenté en eau.

C'est de l'informatique !

Dans cet exercice, l'écoulement de l'eau est régulé par des *conditions*. Par exemple, l'eau ne coule jusqu'au moulin 6 que si les deux clapets B et E sont ouverts. Voici un autre exemple un peu plus



complexe : l'eau ne coule jusqu'au moulin 3 que si au moins l'une des deux ou les deux conditions suivantes sont remplies :

- Le clapet A est ouvert et l'un des deux clapets C ou F est ouvert.
- Les deux clapets B et D sont ouverts.

De telles combinaisons de conditions sont obtenues à l'aide des *opérateurs logiques* ET (symbolisé par \wedge) ou OU (symbolisé par \vee). De tels opérateurs connectent des valeurs de vérité comme vrai et faux. Si A et B sont deux valeurs de vérité, on peut indiquer quelles valeurs de vérité les expressions « A ET B » et « A OU B » ont en fonction de A et B :

A	B	A ET B	A OU B
faux	faux	faux	faux
vrai	faux	faux	vrai
faux	vrai	faux	vrai
vrai	vrai	vrai	vrai

En informatique (et en mathématiques), l'expression « A OU B » est donc aussi considérée comme juste si A et B sont les deux justes. L'affirmation « le moulin 6 est alimenté » est équivalente à :

« le clapet B est ouvert » ET « le clapet E est ouvert ».

Dans le deuxième exemple, l'affirmation « le moulin 3 est alimenté » est équivalente à :

(« le clapet A est ouvert » ET (« le clapet C est ouvert » OU « le clapet F est ouvert »)) OU (« le clapet B est ouvert » ET « le clapet D est ouvert »).

En programmation, il est important de formuler les conditions de manière exacte. Chaque ET et chaque OU combine deux affirmations. Les parenthèses déterminent dans quel ordre les affirmations sont combinées. Les combinaisons à l'aide d'opérateurs logiques et de parenthèses sont utiles pour formuler des conditions complexes. Des conditions sont utilisées aussi bien pour des branchements avec `if` que pour des boucles `while` afin de guider le déroulement d'un programme.

Mots clés et sites web

- Instruction conditionnelle : [https://fr.wikipedia.org/wiki/Instruction_conditionnelle_\(programmation\)](https://fr.wikipedia.org/wiki/Instruction_conditionnelle_(programmation))
- Variable booléenne : <https://fr.wikipedia.org/wiki/Booléen>
- Algèbre de Boole : [https://fr.wikipedia.org/wiki/Algèbre_de_Boole_\(logique\)](https://fr.wikipedia.org/wiki/Algèbre_de_Boole_(logique))

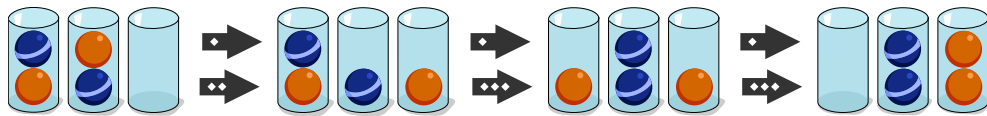




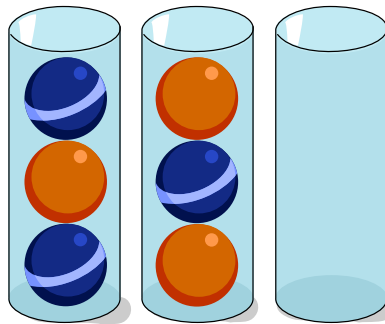
9. Jeu de balles

Les castors aimeraient trier des balles par couleur. À la fin, toutes les balles doivent se trouver dans deux verres. Toutes les balles qui se trouvent dans un verre doivent avoir la même couleur. Ils doivent suivre les règles suivantes :

- ➡ Règle 1 : Seule la balle la plus haute d'un verre peut être déplacée dans un autre verre.
- ➡ Règle 2 : Une balle peut toujours être mise dans un verre vide.
- ➡ Règle 3 : Une balle peut être mise dans un verre non vide uniquement s'il y reste de la place et que la balle en dessous a la même couleur que la balle déplacée.



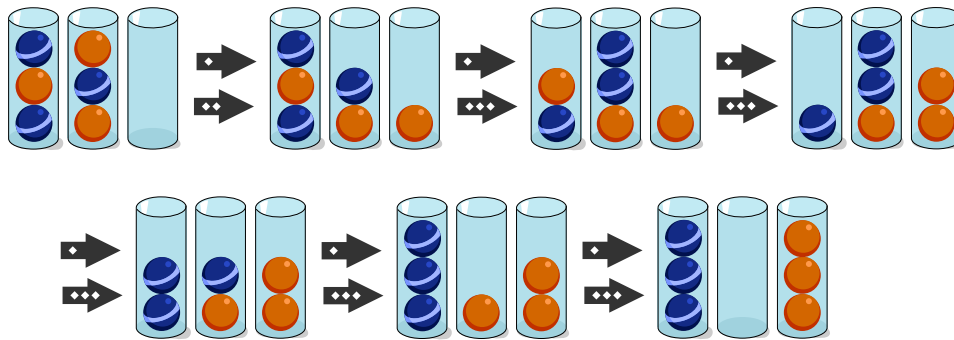
Trie les balles en les déplaçant d'après les trois règles.





Solution

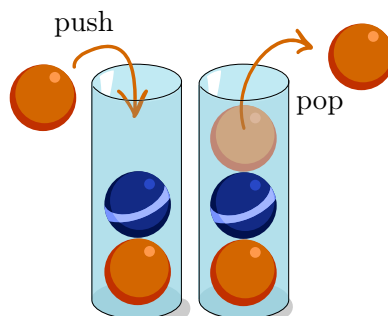
Les balles peuvent être déplacées dans l'ordre suivant :



Il faut au moins six étapes pour réarranger les balles. Il existe d'autres possibilités de réarranger les balles en seulement six étapes.

C'est de l'informatique !

Dans cet exercice, tu déplaces les balles comme un ordinateur gère les données enregistrées dans une *pile* : il ne peut qu'*empiler* (*push* en anglais) un élément en haut de la pile et *dépiler* (*pop* en anglais) l'élément du haut de la pile.




L'ordinateur ne peut accéder aux éléments du bas que si les balles du dessus ont d'abord été retirées. L'élément qui a été enregistré en dernier va être retiré en premier par l'ordinateur. Ce principe est appelé *dernier arrivé, premier sorti* en informatique.

Mots clés et sites web

- Pile: [https://fr.wikipedia.org/wiki/Pile_\(informatique\)](https://fr.wikipedia.org/wiki/Pile_(informatique))



A. Auteur·e·s des exercices


 Daumilas Ardickas

 Michael Barot

 Liam Baumann


 Wilfried Baumann


 Carmel Carroll

 Christian Datzko

 Susanne Datzko

 Nora A. Escherle


 Lidia Feklistova

 Fabian Frei


 Gerald Futschek

 Yasemin Gülbahar

 Ezgi Arzu Güneş

 Benjamin Hirsch

 Andrea Hrušecká

 Tiberiu Iorgulescu

 YongJu Jeon


 Soojin Jun


 Ungyeol Jung

 Filiz Kalelioğlu


 Martin Kandlhofer

 Dong Yoon Kim

 Jihye Kim


 Vaidotas Kinčius

 V́ictor Koleszar


 Taina Lehtimäki

 Tom Naughton


 Graciela Oyhenard

 Elsa Pellet

 Jean-Philippe Pellet

 Zsuzsa Pluhár


 Wolfgang Pohl


 Rosario Schunk

 Bernadette Spieler

 Troy Vasiga

 Florentina Voboril

 Kyra Willekes

 Hongjin Yeh



B. Sponsoring : Concours 2021

HASLERSTIFTUNG <http://www.haslerstiftung.ch/>



<http://www.baerli-biber.ch/>



<http://www.verkehrshaus.ch/>
Musée des transports, Lucerne



Kanton Zürich
Volkswirtschaftsdirektion
Amt für Wirtschaft und Arbeit

Standortförderung beim Amt für Wirtschaft und Arbeit Kanton Zürich



i-factory (Musée des transports, Lucerne)



<http://www.ubs.com/>



<http://www.oxocard.ch/>
OXOcard
OXON



<https://educatec.ch/>
educaTEC



<http://senarclens.com/>
Senarclens Leu & Partner



<http://www.abz.inf.ethz.ch/>
Ausbildungs- und Beratungszentrum für Informatikunterricht der ETH Zürich.



hep/ haute
école
pédagogique
vaud

<http://www.hepl.ch/>
Haute école pédagogique du canton de Vaud

PH LUZERN
PÄDAGOGISCHE
HOCHSCHULE

<http://www.phlu.ch/>
Pädagogische Hochschule Luzern

n|w Fachhochschule
Nordwestschweiz

<https://www.fhnw.ch/de/die-fhnw/hochschulen/ph>
Pädagogische Hochschule FHNW

Scuola universitaria professionale
della Svizzera italiana

<http://www.supsi.ch/home/supsi.html>
La Scuola universitaria professionale della Svizzera italiana
(SUPSI)

SUPSI

PÄDAGOGISCHE
HOCHSCHULE
ZÜRICH

PH
ZH

<https://www.phzh.ch/>
Pädagogische Hochschule Zürich



C. Offres ultérieures

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SS!E

www.svia-ssie-ssii.ch
schweizerischervereinfürinformatikind
erausbildung//sociétésuissepourl'infor
matique dans l'enseignement//societasviz
zeraperl'informaticanell'insegnamento

Devenez vous aussi membre de la SSIE

<http://svia-ssie-ssii.ch/la-societe/devenir-membre/>

et soutenez le Castor Informatique par votre adhésion

Peuvent devenir membre ordinaire de la SSIE toutes les personnes qui enseignent dans une école primaire, secondaire, professionnelle, un lycée, une haute école ou donnent des cours de formation ou de formation continue.

Les écoles, les associations et autres organisations peuvent être admises en tant que membre collectif.