



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

Exercices et solutions 2017 Années HarmoS 5/6

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

Éditeurs :

Julien Ragot, Gabriel Parriaux, Jean-Philippe Pellet, Nicole Müller, Christian Datzko, Hanspeter Erni

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
01001001010010010010001

SS!E

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



Ont collaboré au Castor Informatique 2017

Andrea Adamoli, Christian Datzko, Susanne Datzko, Olivier Ens, Hanspeter Erni, Martin Guggisberg, Per Matzinger, Carla Monaco, Nicole Müller, Gabriel Parriaux, Jean-Philippe Pellet, Julien Ragot, Silvan Stöckli, Beat Trachsler.

Nous adressons nos remerciements à :

Juraj Hromkovič, Giovanni Serafini, Urs Hauser, Regula Lacher, Ivana Kosírová : ETHZ

Valentina Dagienė : Bebras.org

Hans-Werner Hein, Wolfgang Pohl : Bundesweite Informatikwettbewerbe (BWINF), Allemagne

Anna Morpurgo, Violetta Lonati, Mattia Monga : Italie

Gerald Futschek, Wilfried Baumann : Austrian Computer Society, Austria

Zsuzsa Pluhár : ELTE Informatikai Kar, Hongrie

Eljakim Schrijvers, Daphne Blokhuis : Eljakim Information Technology bv, Pays-Bas

Roman Hartmann : hartmannGestaltung (Flyer Castor Informatique Suisse)

Christoph Frei : Chragokyberneticks (Logo Castor Informatique Suisse)

Pamela Aeschlimann, Andreas Hieber, Aram Loosmann, Daniel Vuille, Peter Zurflüh : Lernetz.ch (page web)

Andrea Leu, Maggie Winter, Brigitte Maurer : Senarclens Leu + Partner

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

L'adaptation française a été réalisée par Nicole Müller et la version italienne par Andrea Adamoli.



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Le Castor Informatique 2017 a été réalisé par la Société Suisse de l'Informatique dans l'Enseignement SSIE. Le Castor Informatique est un projet de la SSIE, aimablement soutenu par la Fondation Hasler.

HASLERSTIFTUNG

Tous les liens ont été vérifiés le 1^{er} novembre 2017. Ce cahier d'exercice a été produit le 9 octobre 2019 avec le logiciel de mise en page L^AT_EX.



Les exercices sont protégés par une licence Creative Commons Paternité – Pas d'Utilisation Commerciale – Partage dans les Mêmes Conditions 4.0 International. Les auteurs sont cités p. 24.



Préambule

Très bien établi dans différents pays européens 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» (<http://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 2017 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). Chaque autre tranche d'âge devait résoudre 15 exercices (5 faciles, 5 moyens et 5 difficiles).

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.

Les participants disposaient de 45 points (Petit Castor 27) sur leur compte au début du concours.

Le maximum de points possibles était de 180 points (Petit Castor 108), le minimum étant de 0 point.

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 de l'Informatique dans l'Enseignement)

Castor Informatique

Julien Ragot

castor@castor-informatique.ch

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


 <https://www.facebook.com/informatikbiberch>



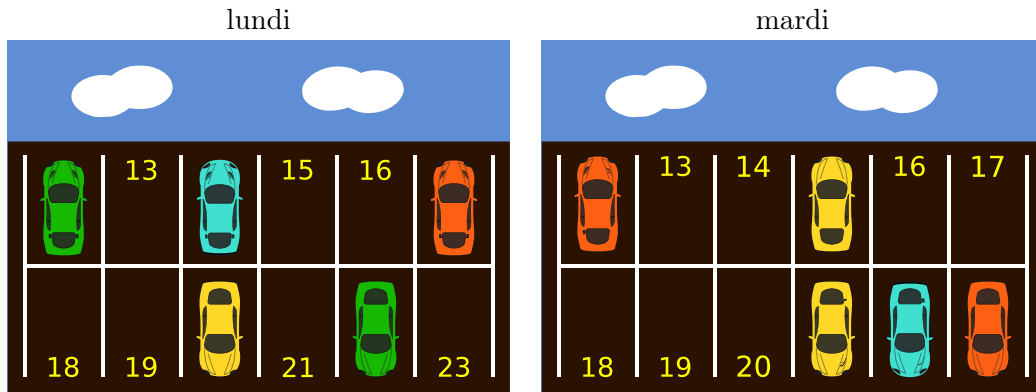
Table des matières

Ont collaboré au Castor Informatique 2017	i
Préambule	ii
1. Des places de parking libres	1
2. Nichoirs pour oiseaux	3
3. Trouve le passage !	5
4. Le portail binaire	7
5. Service de transmission de messages	9
6. La chasse à la fraise	11
7. Le castor à un bras	13
8. Enlever des murs	17
9. Cinq petits bouts de bois	21
A. Auteurs des exercices	24
B. Sponsoring : Concours 2017	25
C. Offres ultérieures	27



1. Des places de parking libres

Le parking des castors permet de stationner 12 voitures. Chaque place de parking est numérotée. Les images ci-dessous montrent les places de parking qui étaient occupées lundi dernier et celles qui étaient occupées mardi dernier.



Combien de places de parking étaient libres ces deux journées, lundi et mardi ?

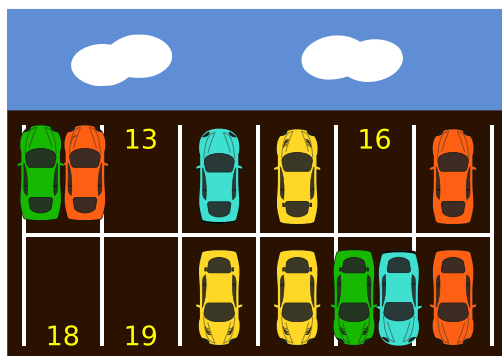
- A) 3
- B) 4
- C) 5
- D) 6



Solution

La réponse B) est correcte puisque les places 13, 16, 18 et 19 sont restées libres pendant les deux journées en question.

Quand on représente les voitures garées sur un seul parking, nous voyons tout de suite quelles places sont restées libres. En les additionnant, on arrive à quatre places qui n'ont pas été occupées, ni lundi ni mardi.



C'est de l'informatique !

Il est possible d'encoder tous les types de données sous forme d'une série de «zéros» et de «uns». Une telle série de données s'appelle *code binaire*. Chaque zéro et chaque un est dénommé *bit* («binary digit»).

Dans la tâche présente, nous pouvons encoder la présence d'une voiture sur le parking sous forme de un (1) et l'absence d'une voiture sous forme de zéro. L'occupation du parking peut ainsi être représentée sous forme d'une série de *bits* : celle de lundi correspondra donc à la série 101001001010 et celle de mardi à la série 100100000111. À l'aide d'un opérateur logique *OU (OR)* qui, lui, indique que le résultat sera VRAI si la valeur A OU la valeur B sont vraies, toutes les places qui sont occupées au moins pendant une des deux journées seront encodées sous forme de un (1). Par la suite, nous calculons la réponse correcte en superposant les deux séries de données et nous obtenons le résultat suivant :

$$\begin{array}{r}
 101001001010 \\
 \text{ou} \\
 100100000111 \\
 = \\
 101101001111
 \end{array}$$

Ce code binaire comporte quatre zéros ce qui correspond à quatre places de parking libres.

Mots clés et sites web

bits, binaire, code binaire, OU (OR), opérateur logique

— https://fr.wikipedia.org/wiki/Disjonction_logique



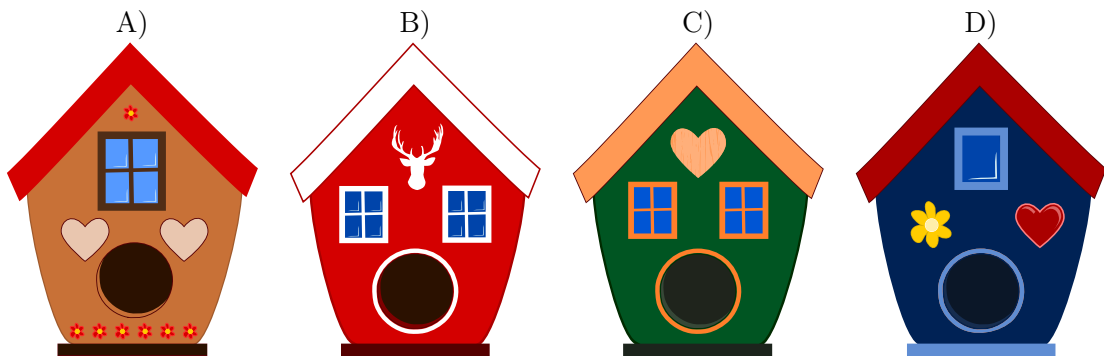
2. Nichoirs pour oiseaux

Maman castor aimerait bien acheter un nichoir pour sa fille qui fêtera son anniversaire demain. Pour ne pas se tromper de nichoir, elle demande à sa fille lequel des nichoirs lui ferait plaisir. Sa fille lui répond :

«J'aimerais un nichoir avec deux fenêtres et un coeur.»

La mère va donc au magasin pour animaux acheter un nichoir.

Lequel de ces quatre nichoirs maman castor achètera-t-elle pour sa fille ?





Solution

La réponse C) est correcte car le nichoir comporte 2 fenêtres et 1 coeur.

La réponse A) est fausse parce que le nichoir n'a qu'une seule fenêtre. En outre, il a 2 coeurs au lieu d'un seul.

La réponse B) est fausse parce que le nichoir comporte 2 fenêtres, mais aucun coeur.

La réponse D) est fausse parce que le nichoir ne comporte qu'une seule fenêtre.

C'est de l'informatique !

Cette tâche est difficile car il faut se concentrer sur un nichoir qui corresponde aux désirs de la fille de maman castor. Pour choisir le bon nichoir, il faut donc ignorer toutes les autres caractéristiques des nichoirs comme par exemple la couleur de la maison ou d'autres détails qui ne sont pas importants. Cette vision «limitée» est appelée abstraction, ce qui veut dire que l'on ne prendra en considération que des propriétés préalablement déterminées. En informatique, ceci est souvent nécessaire.

Mots clés et sites web

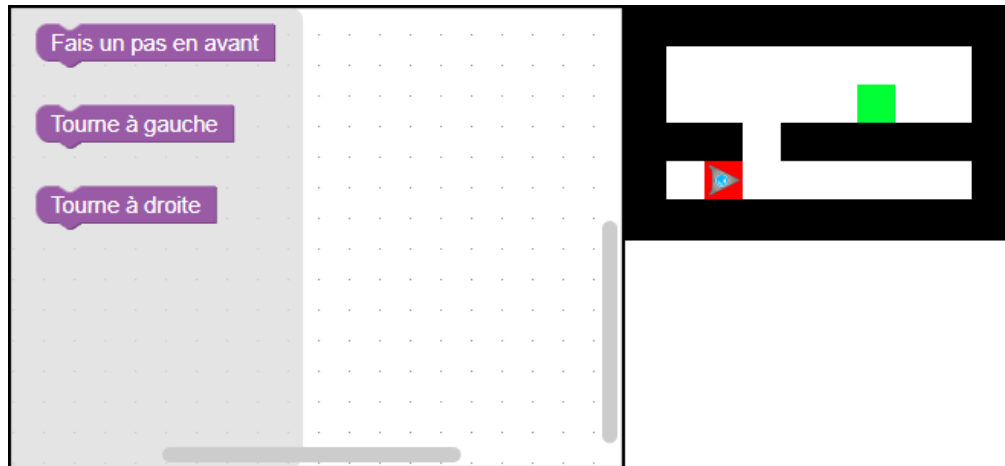
modèle (pattern), reconnaissance des formes (pattern recognition), propriété, abstraction (informatique)

- <https://en.wikipedia.org/wiki/Abstraction>
- https://en.wikipedia.org/wiki/Pattern_recognition



3. Trouve le passage !

Un robot ayant une forme triangulaire se trouve sur la position de départ marquée en rouge. Il doit atteindre la zone cible marquée en vert.

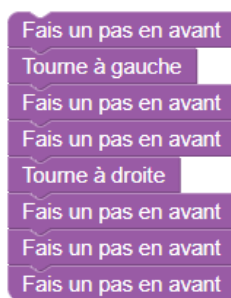


Pour programmer le robot afin qu'il arrive jusqu'à la zone cible, tu dois insérer des instructions dans la fenêtre de programme tout en respectant l'ordre correct de la séquence d'instructions.



Solution

Voici une solution possible :



Pour que le robot atteigne la zone cible, il doit d'abord avancer d'un pas afin de se trouver à côté de la brèche dans le mur. A partir de là, s'il tourne à gauche et avance de deux pas, il traverse la brèche. Finalement, s'il tourne à droite et avance de trois pas, il arrive directement à la zone cible.

C'est de l'informatique !

En robotique «mobile», la navigation pose toujours certains problèmes. La résolution de problèmes tels que celui de cet exercice consistant à devoir sortir d'un labyrinthe n'est pas si courante, elle exige des facultés de pensée computationnelle. Pour résoudre ce problème, on se sert d'un robot automatisé. Il existe différents types de labyrinthes : avec ou sans boucle, avec ou sans quadrillage. Le robot dans notre tâche ne possède pas de capteurs qui lui permettraient de détecter les murs. Il est quasiment aveugle. C'est la raison pour laquelle il dépend d'instructions très précises pour trouver le passage et, par conséquent, le bon chemin pour arriver à la zone cible.

Mots clés et sites web

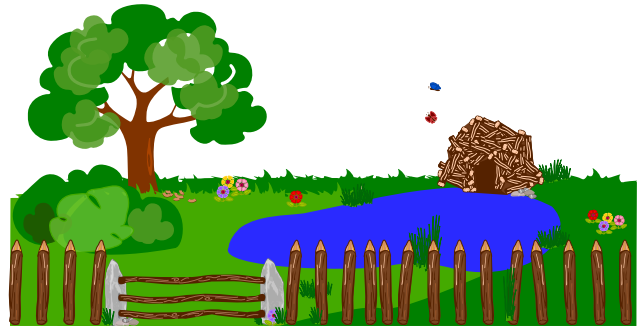
séquence, programmer

- <https://fr.wikipedia.org/wiki/Blockly>
- <http://primalogo.ch/fr>
- <http://www.abz.inf.ethz.ch/primarschulen-stufe-sek-1/programmieren-fur-kinder/>
- http://www.swisseduc.ch/informatik/programmiersprachen/scratch_werkstatt/
- <http://ilearnit.ch/de/2b/explain.html>
- <https://scratch.mit.edu/>
- <http://kinderlabor.ch/informatik-fuer-kinder/programmieren-mit-scratch/>



4. Le portail binaire

Les castors ont l'habitude de se rendre visite assez régulièrement. Cependant, il arrive que l'un ou l'autre ne soit pas chez lui et que les castors arrivent en vain. Afin que le castor qui a quitté sa maison puisse laisser un message, par exemple pour informer ses amis quand il reviendra, les castors ont inventé une méthode très pratique. Le portail du jardin comporte deux piliers en pierre et trois bâtons en bois que l'on peut placer dans des paires de trous prédéterminés qui se trouvent dans les faces opposées des deux piliers en pierre. Cette construction permet de créer des messages courts.



Les castors se sont mis d'accord sur quatre messages :



Nous sommes à la maison, venez nous voir.



Nous serons de retour à midi déjà.



Nous serons de retour dans la soirée.



Nous passons une soirée avec nos amis et nous serons de retour vers minuit.

Il serait parfaitement possible que les castors se mettent d'accord sur d'autres messages encore, sans qu'ils aient besoin ni d'autres bâtons en bois ni de trous supplémentaires.

Devine, avec deux piliers en pierre, trois bâtons en bois et six trous, sur combien de messages différents les castors pourraient-ils se mettre d'accord au total (bien entendu, y compris les quatre messages mentionnés ci-dessus).



Solution

Les castors peuvent se mettre d'accord sur huit messages différents au total :



C'est de l'informatique !

Dans la présente tâche, les castors se servent en fait d'un système de numération appelé binaire à trois positions. Les porteurs d'information sont représentés par les paires de trous qui se trouvent dans les faces opposées des piliers. Une telle paire de trous représente deux valeurs, à savoir «bâton en bois placé» et «bâton en bois non placé». La position des paires de trous nous renseigne donc sur sa valeur. Ainsi, le nombre des différents messages correspondra au nombre des valeurs des paires de trous (2) multiplié par le nombre des paires de trous (3), donc : $2^3 = 2 \cdot 2 \cdot 2 = 8$.

En général, les castors comprennent leurs messages amicaux et si jamais il arrive qu'ils se trompent, il n'y aura pas de graves conséquences. En informatique, par contre, étant donné qu'il existe différents systèmes globaux interconnectés, toutes les personnes impliquées dépendent d'une communication simple et correcte qui exclut les malentendus.

De grandes organisations s'occupent de la standardisation des systèmes de numération et de la certification des produits. Des comités d'experts multinationaux discutent et décident du caractère et du sens des signes, puis, les parlements de nombreux états approuvent ces décisions et intègrent les systèmes de numération validés dans leur législation. Voilà comment beaucoup d'hommes prennent soin que tous les ordinateurs du monde se comprennent.

Mots clés et sites web

codage, code binaire, standard

— https://fr.wikipedia.org/wiki/Système_binaire



5. Service de transmission de messages

Violette aimerait bien envoyer un message à Léo, mais elle a besoin d'aide de la part des castors. Elle découpe le message original en quelques petits billets dont chacun comportera trois lettres. Chaque castor messenger recevra un de ces petits billets.

Sachant que les castors peuvent être facilement détournés de leur tâche lors du trajet entre Violette et Léo et que leur arrivée peut donc être désordonnée, Violette numérote chaque petit billet avant de le remettre à un des castors. Plus tard, après avoir reçu les billets, Léo n'aura qu'à les remettre dans le bon ordre pour lire le message complet.

Voilà un exemple : afin d'envoyer le message FETONSCESOIR, elle découpe quatre petits billets qui comportent les lettres suivantes :



L'autre jour, Léo a reçu la séquence de petits billets suivante :



Quel était le message original ?

- A) APPELELONBALORT
- B) LONBALELEORTAPP
- C) APPORTELEBALLON
- D) ELEAPPORTBALLON



Solution

La réponse correcte est C) APPORTELEBALLON.

Quand on classe les petits billets dans l'ordre correct, on reçoit le message original suivant :



C'est de l'informatique !

Afin de transmettre des données sur Internet comme par exemple des e-mails, des images ou des vidéos, celles-ci sont découpées en plusieurs entités appelées paquets de données (TCP/IP en anglais, Transmission Control Protocol/Internet Protocol). Chaque paquet ne dépasse pas la taille maximale de 65536 octets (soit $2^{16} = 65536$ octets = 64 Ko).

Puis, à l'aide de différents routeurs, ces paquets numériques, souvent enrichis d'informations supplémentaires qui dévoilent les spécificités du paquet (expéditeur, destinataire, numéro de séquence, ...), seront dirigés à travers le réseau informatique. Toutes ces informations supplémentaires sont nécessaires pour que le destinataire puisse reconstruire le message original même si ce dernier est transmis sous forme de paquets partiels.

Remarque : Le nouveau *protocole Internet* «IPv6» permet de gérer des paquets de plus grande taille encore.

Mots clés et sites web

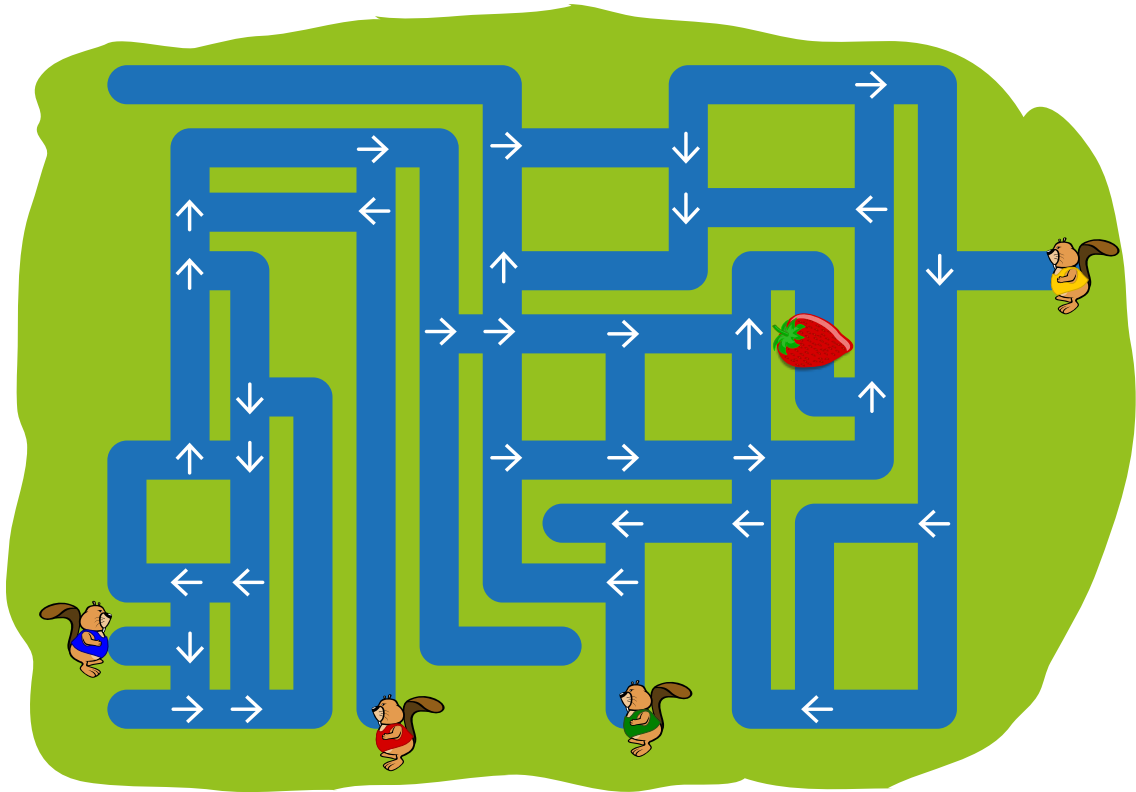
lettre, petit billet, message, numéro, ordre des petits billets, paquet de données, transmission de messages, routeur, protocole Internet

- [https://fr.wikipedia.org/wiki/Paquet_\(réseau\)](https://fr.wikipedia.org/wiki/Paquet_(réseau))
- https://en.wikipedia.org/wiki/IP_fragmentation



6. La chasse à la fraise

Quatre castors commencent à nager à partir de quatre points de départ différents. Ils nagent droit devant eux et suivent les flèches à chaque fois qu'ils arrivent à une intersection.



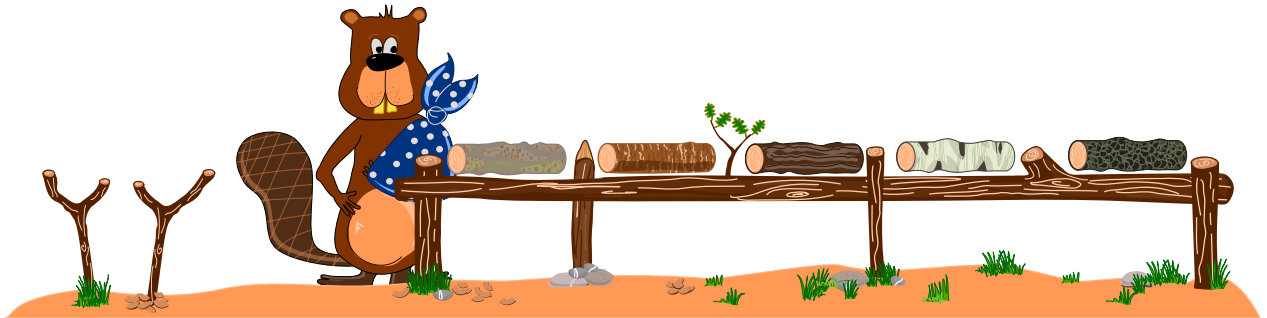
Combien de castors arriveront jusqu'à la fraise ?

- A) 0
- B) 1
- C) 2
- D) 3
- E) 4



7. Le castor à un bras

Le pauvre David s'est cassé le bras gauche et ne peut travailler qu'avec le bras droit. Il aimerait bien classer sa collection de bûches de bois, mais à cause de sa blessure, il ne peut soulever qu'une bûche à la fois. Ce qu'il peut faire, pourtant, c'est déposer une bûche sur le support qui se trouve à sa gauche et qui peut lui servir d'entrepôt.

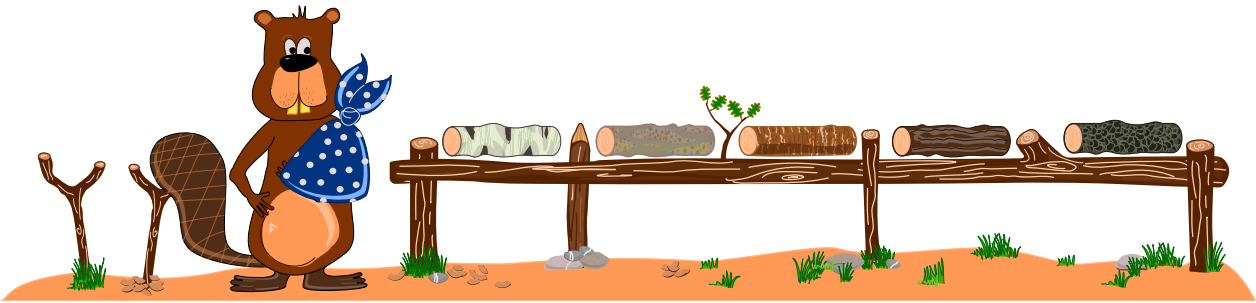


David te demande de l'aider à classer les différentes bûches selon leur couleur, de la plus claire à la plus foncée. Il aimerait bien que la bûche la plus claire se trouve à l'extrême gauche et la bûche la plus foncée à l'extrême droite.



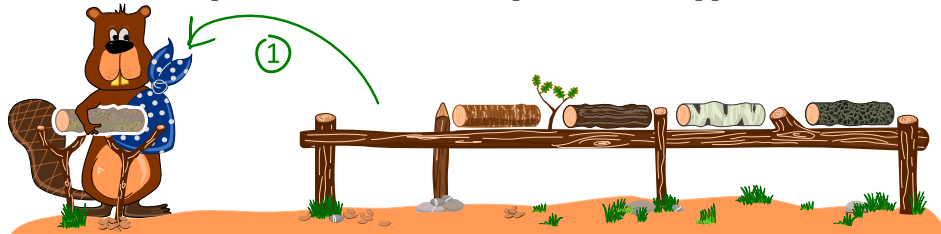
Solution

La réponse correcte est :

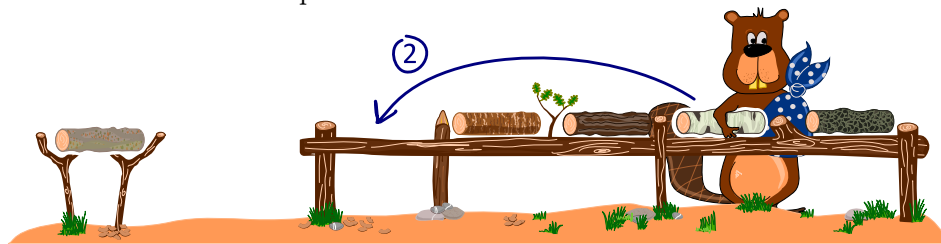


Pour que David puisse résoudre ce problème, il est nécessaire qu'il utilise le support à sa droite. Il existe plusieurs possibilités pour classer les bûches correctement, mais la plus simple est qu'il échange deux bûches en entreposant une des deux bûches sur le support :

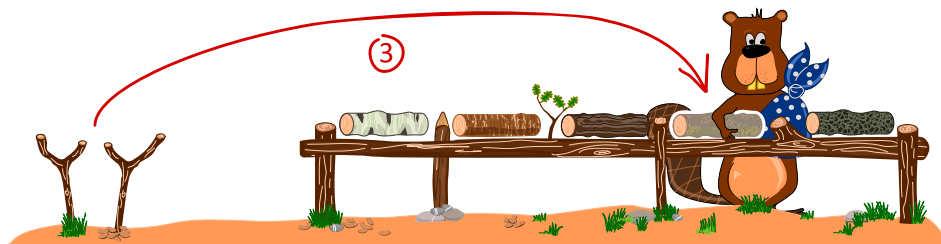
- Enlever une bûche de sa position initiale et la déposer sur le support.



- Déposer une autre bûche sur la place devenue libre.



- Reprendre la bûche qui repose actuellement sur le support et la déposer sur la place devenue libre.



Ainsi, en échangeant progressivement les bûches deux par deux, on arrivera à l'ordre désiré.

C'est de l'informatique !

En informatique, cette méthode d'échanger deux bûches de bois en utilisant une place supplémentaire comme entrepôt, est courante. Les «places» sont en général des *variables* (par exemple, appelées *a* et *b*) qui comprennent des *valeurs*. Si on veut intervertir la valeur des variables *a* et *b*, il faut faire appel à une troisième variable appelée *t*. D'abord, on enregistre temporairement la valeur de la variable *a* dans la variable *t*. Ensuite, on enregistre la valeur de la variable *b* dans la variable *a* pour enregistrer finalement la valeur de la variable temporaire *t* (qui comprend la valeur de la variable *a*) dans la variable *b*. Nous pouvons décrire cette procédure comme suit :


$$t \leftarrow a$$
$$a \leftarrow b$$
$$b \leftarrow t$$

Afin de classer un ensemble de plusieurs variables à l'aide d'une seule variable temporaire, on peut utiliser l'algorithme *tri par sélection* :

- De la première variable jusqu'à la dernière :
 - Sélectionne la plus petite valeur de l'ensemble des variables qui n'ont pas encore été triées
 - Invertis cette valeur avec la valeur de la variable actuellement en question à l'aide de la variable temporaire

Outre l'algorithme simple *tri par sélection*, il existe d'autres algorithmes de tri simples comme par exemple le *tri par insertion* ou le *tri à bulles*. Un algorithme de tri plus efficace encore mais par-delà aussi plus complexe est le tri rapide (en anglais, *quicksort*).

Mots clés et sites web

intervertir des variables, (algorithme de) tri

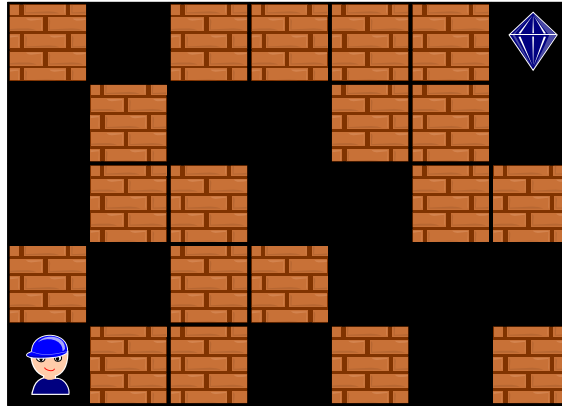
- [https://fr.wikipedia.org/wiki/Permutation_\(informatique\)](https://fr.wikipedia.org/wiki/Permutation_(informatique))
- https://fr.wikipedia.org/wiki/Algorithme_de_tri





8. Enlever des murs

Pour arriver jusqu'au trésor qui se trouve en haut à droite, Pierre doit enlever des murs. Son but est d'en enlever le moins possible.

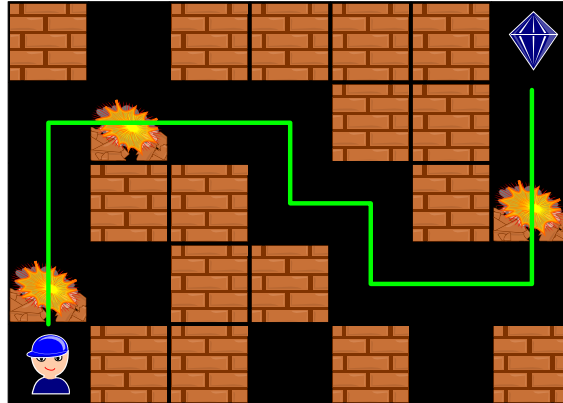


Quels murs au minimum doivent être enlevés pour libérer le chemin jusqu'au trésor ?



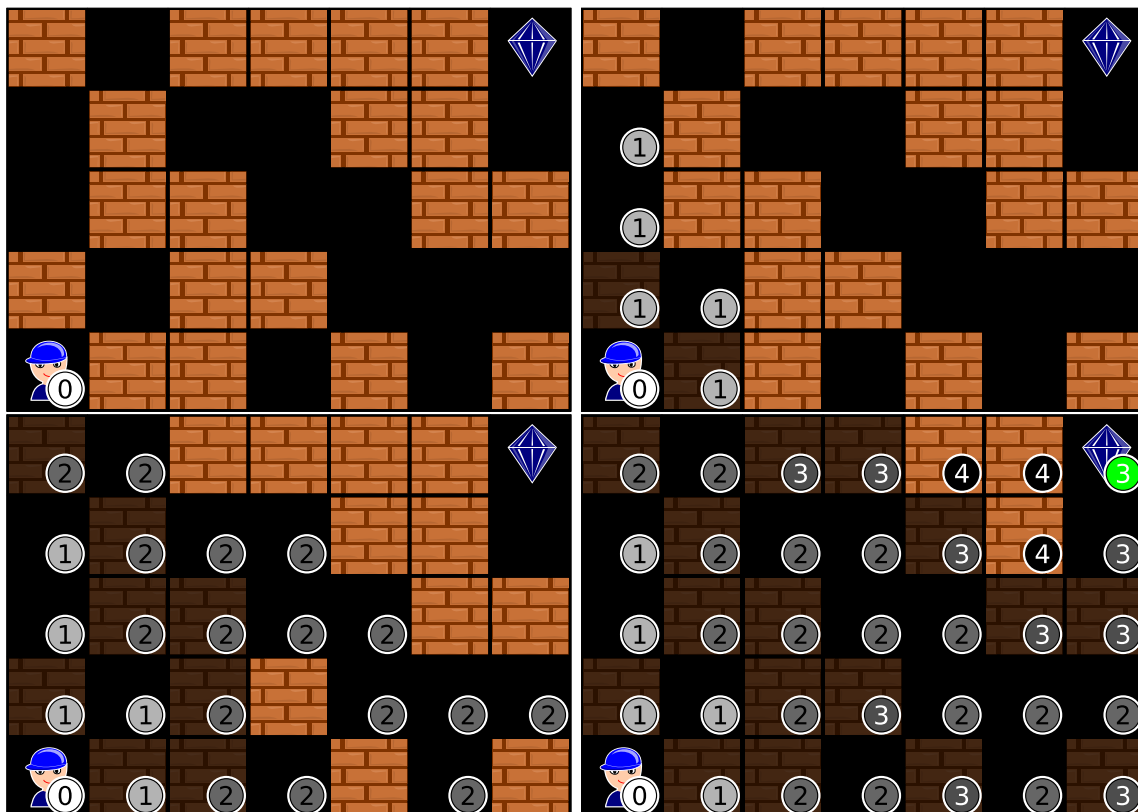
Solution

Il faut enlever au moins trois murs. La figure ci-dessous montre le nombre de murs qu'il faut enlever (les murs à enlever sont représentés par le symbole d'une petite explosion). Le chemin qui mène ainsi directement au trésor est marqué en vert.



On peut résoudre ce problème en procédant de manière systématique : on note dans chaque cellule le nombre de murs qu'il a fallu enlever pour y arriver. Nous commençons au point de départ en bas à gauche. Nous notons dans la cellule le chiffre 0. Nous marquons toutes les cellules noires que l'on peut atteindre directement à partir de ce point par le chiffre 0. Ensuite, nous marquons tous les murs avoisinant les 0 par le chiffre 1 et nous notons dans toutes les cellules noires que l'on peut atteindre en enlevant ces murs (qui avoisinent les 0) le chiffre 1.

Puis, nous marquons tous les murs et toutes les cellules que l'on peut atteindre en enlevant deux murs par le chiffre 2, et toutes les cellules que l'on peut atteindre en enlevant 3 murs par le chiffre 3, et ainsi de suite. En procédant de cette manière, nous noterons finalement dans la zone cible le chiffre 3 qui, lui, correspondra à notre solution mentionnée ci-dessus.





C'est de l'informatique !

Le problème du plus court chemin dans un labyrinthe est un problème informatique très répandu. Les systèmes de navigation, par exemple, se servent de différentes techniques informatiques pour trouver le chemin le plus court, tout en respectant des conditions particulières comme par exemple d'éviter des routes à péage. Dans notre cas, il faut simplement trouver le chemin le plus court à travers les murs, la longueur réelle des chemins n'a pas d'importance.

Pour trouver la solution à notre problème, il faut un raisonnement logique et une démarche (un algorithme) qui tient compte des résultats intermédiaires. En procédant ainsi, on fait appel à un algorithme particulier appelé *algorithme de parcours en largeur*.

Mots clés et sites web

labyrinthe, plus court chemin, algorithme de parcours en largeur.

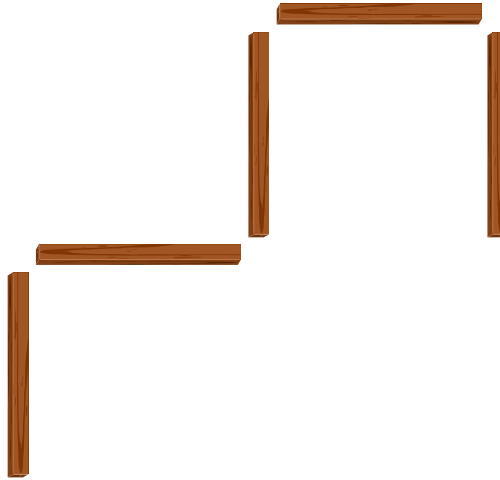
- https://fr.wikipedia.org/wiki/Problème_de_plus_court_chemin
- https://fr.wikipedia.org/wiki/Algorithme_de_parcours_en_largeur



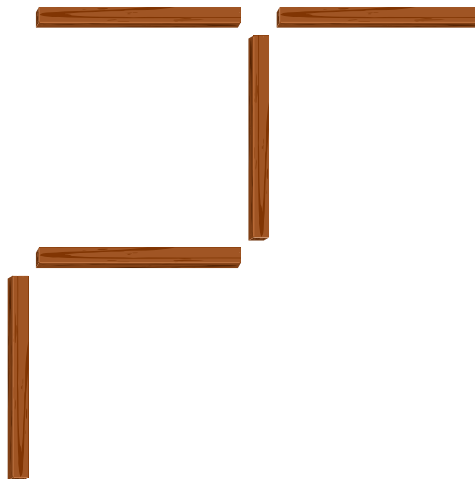


9. Cinq petits bouts de bois

Sur une table, il y a 5 petits bouts de bois arrangés de la manière suivante :

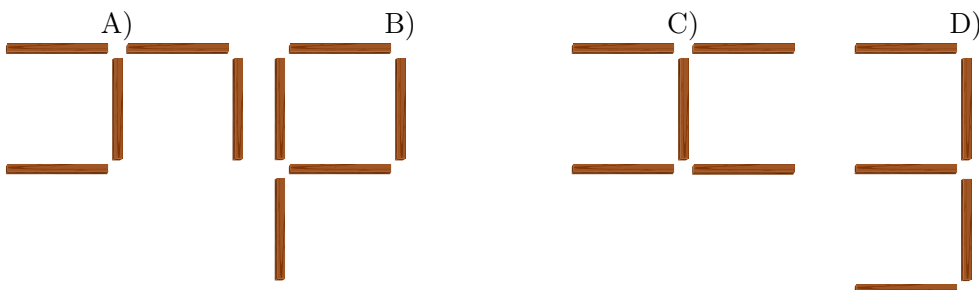


Nola prend un petit bout de bois et le déplace comme suit :



Ensuite, Bernard prend un autre petit bout de bois et le déplace aussi.

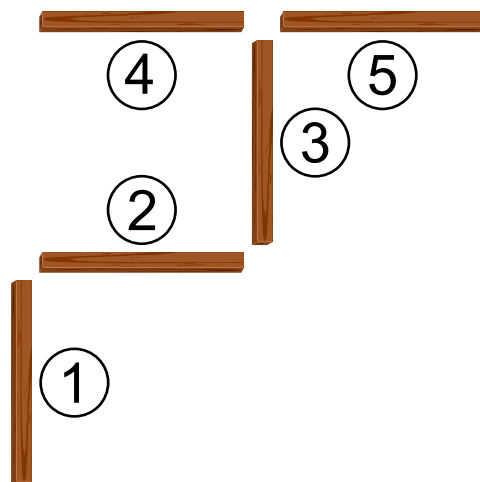
Lequel des arrangements de petits bouts de bois suivants ne correspond pas à un arrangement possible après la dernière opération ?





Solution

Pour expliquer la solution, nous numérotons les petits bouts de bois. Après l'opération effectuée par Nola, les petits bouts de bois sont arrangés comme suit :



Pour que les petits bouts de bois soient arrangés comme illustré dans la figure A), Bernard doit déplacer le petit bout de bois n^o 1 à un autre endroit.

Pour que les petits bouts de bois soient arrangés comme illustré dans la figure B), Bernard doit déplacer le petit bout de bois n^o 5 à un autre endroit.

Pour que les petits bouts de bois soient arrangés comme illustré dans la figure C), Bernard doit déplacer le petit bout de bois n^o 1 à un autre endroit.

Pour que les petits bouts de bois soient arrangés comme illustré dans la figure D), Bernard devrait déplacer deux petits bouts de bois, les petits bouts de bois n^o 1 et 5. Le problème, c'est qu'il ne peut déplacer qu'un seul petit bout de bois.

C'est de l'informatique !

Pour changer la position des petits bouts de bois, Nola et Bernard ont effectué une opération simple : ils ont choisi un seul petit bout de bois et l'on déplacé à un endroit de leur choix. Ils suivent ainsi une instructions simple qui est : «Prends un petit bout de bois quelconque et déplace-le.» Cependant, cette instruction n'est pas tout à fait évidente car ce n'est pas clairement indiqué lequel des petits bouts de bois il faudra déplacer et, par conséquent, nous ne saurons pas à coup sûr lequel des arrangements sera l'arrangement final.

Quand on programme des ordinateurs, on doit se servir d'instructions précises pour que l'ordinateur les comprenne sans équivoque. L'ordinateur est déjà bien familiarisé avec certaines instructions. Quand on veut en créer de nouvelles, on peut procéder comme suit : on peut, par exemple, créer des instructions plus complexes en concaténant différentes instructions. On peut aussi créer un programme qui exécute une seule et même instruction à plusieurs reprises ou on peut définir qu'une instruction ne soit exécutée que dans des conditions particulières. Voilà les trois possibilités les plus importantes afin de transformer des instructions simples en des instructions plus complexes.

Mots clés et sites web

instructions, changement d'état (informatique), programmer

— [https://en.wikipedia.org/wiki/Statement_\(computer_science\)](https://en.wikipedia.org/wiki/Statement_(computer_science))




- https://fr.wikipedia.org/wiki/Programmation_informatique
- https://fr.wikipedia.org/wiki/Diagramme_états-transitions




A. Auteurs des exercices

 Nursultan Akhmetov


 Adil Aliyev

 Haim Averbuch

 Raluca Constantinescu

 Darija Dasović Rakijašić

 Christian Datzko

 Susanne Datzko


 Hanspeter Erni


 Gerald Futschek

 Arnheiður Guðmundsdóttir


 Martin Guggisberg

 Hans-Werner Hein

 Fredrik Heintz


 Jia-Ling Koh

 Wolfgang Pohl


 J.P. Pretti

 Chris Roffey


 Frances Rosamond

 Eljakim Schrijvers

 Monika Tomcsányiová

 Willem van der Vegt

 Troy Vasiga

 Corina Elena Vint

 Khairul A. Mohamad Zaki



B. Sponsoring : Concours 2017

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

ROBOROBO <http://www.roborobo.ch/>

d digitec.ch

<http://www.digitec.ch/> & <http://www.galaxus.ch/>



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



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



Standortförderung beim Amt für Wirtschaft und Arbeit Kanton Zürich



i-factory (Musée des transports, Lucerne)



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



<http://www.bbv.ch/>



<http://www.presentex.ch/>



PH LUZERN
PÄDAGOGISCHE
HOCHSCHULE

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

ABZ

AUSBILDUNGS- UND BERATUNGSZENTRUM
FÜR INFORMATIKUNTERRICHT

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

n|w

Fachhochschule
Nordwestschweiz

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

Z

hdk

Zürcher Hochschule der Künste
Game Design

<https://www.zhdk.ch/>
Zürcher Hochschule der Künste



ZUBLER & PARTNER AG
Informatik

<http://www.zubler.ch/>
Zubler & Partner AG Informatik

senarclens
leu+partner
strategische kommunikation

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



C. Offres ultérieures

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SS!E

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