



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Exercices et solutions 2017

Années scolaires 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
010010010100100100100001

SS!E

www.svia-ssie-ssii.ch
schweizerischervereinfürinformatikind
erausbildung//sociétésuisse del'inform
atique dans l'enseignement//societàsviz
zera per l'informaticanell'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 Dagiene : 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

Tout lien a été vérifié le 1 novembre 2017. Ce cahier d'exercice a été produit le 18 novembre 2017 avec 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. 39.



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.bebbras.org/>), initié en Lituanie.

Le concours a été organisé pour la première fois en Suisse en 2010. Le Petit Castor (années scolaires 3 et 4) 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 scolaires 3 et 4 (Petit Castor)
- Années scolaires 5 et 6
- Années scolaires 7 et 8
- Années scolaires 9 et 10
- Années scolaires 11 à 13

Les élèves des années scolaires 3 et 4 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 de son 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>



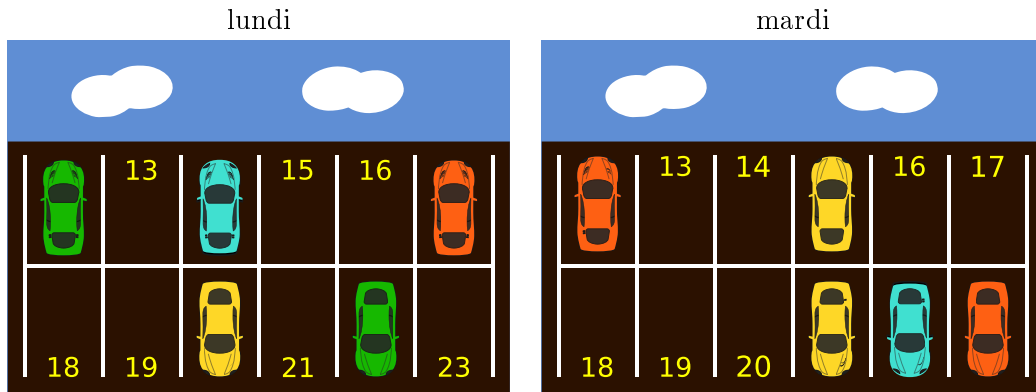
Table des matières

Ont collaboré au Castor Informatique 2017	i
Préambule	ii
1. Des places de parking libres	1
2. Le portail binaire	3
3. Service de transmission de messages	5
4. La chasse à la fraise	7
5. Déplacer des chiens	9
6. Le castor à un bras	11
7. Cinq petits bouts de bois	15
8. Concours de danse : à qui le tour ?	19
9. Le nom japonais	23
10. Un programme court	25
11. Orner des pendentifs médiévaux	27
12. Honomakato	29
13. Un art martial japonais	33
14. La confiture de grand-père	35
15. La ville riche en ronds-points	37
A. Auteurs des exercices	39
B. Sponsoring : Concours 2017	40
C. Offres ultérieures	42



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 ?

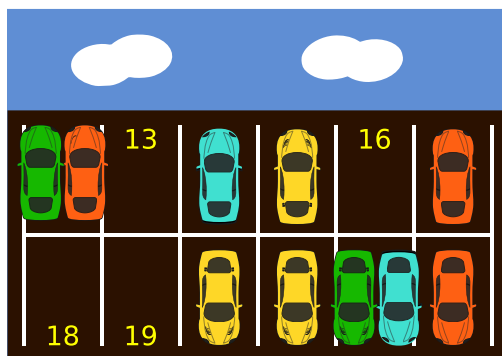
- A) 3
- B) 4
- C) 5
- D) 6



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.

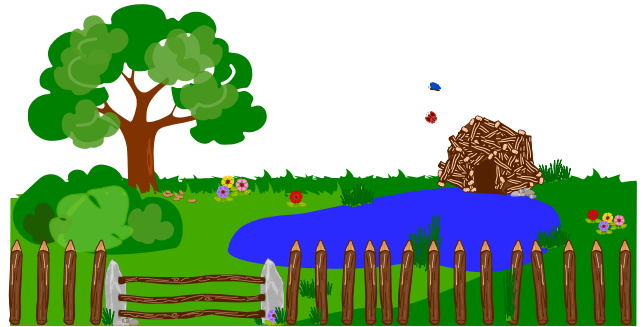
Sites web et mots clés

bits, binaire, code binaire, OU (OR), opérateur logique
— https://fr.wikipedia.org/wiki/Disjonction_logique



2. 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 :



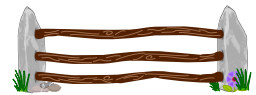
Nous sommes à la maison, venez nous voir.



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.

Sites web et mots clés

codage, code binaire, standard

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

3. 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 bits (soit $2^{16} = 65536$ bits = 64 KB).

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.

Sites web et mots clés

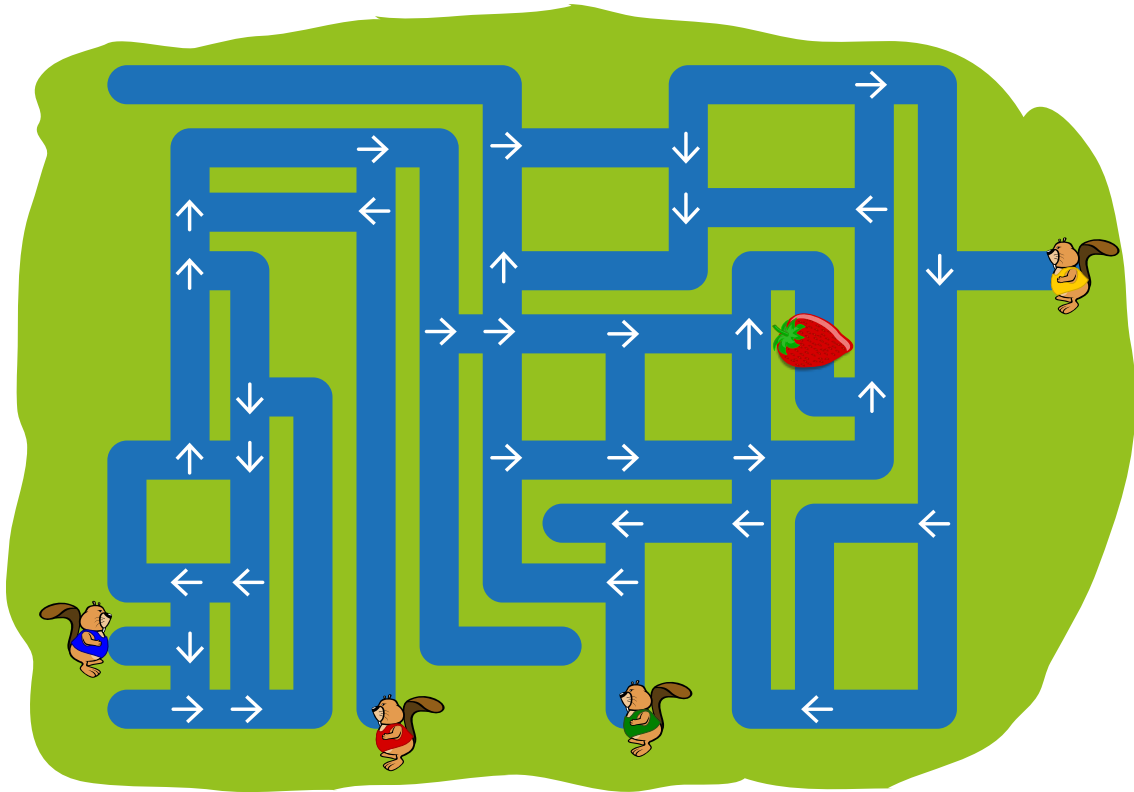
lettre, petit billet, message, numéro, ordre des petits billets, paquet de données, transmission de messages, routeur, protocole Internet

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



4. 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.



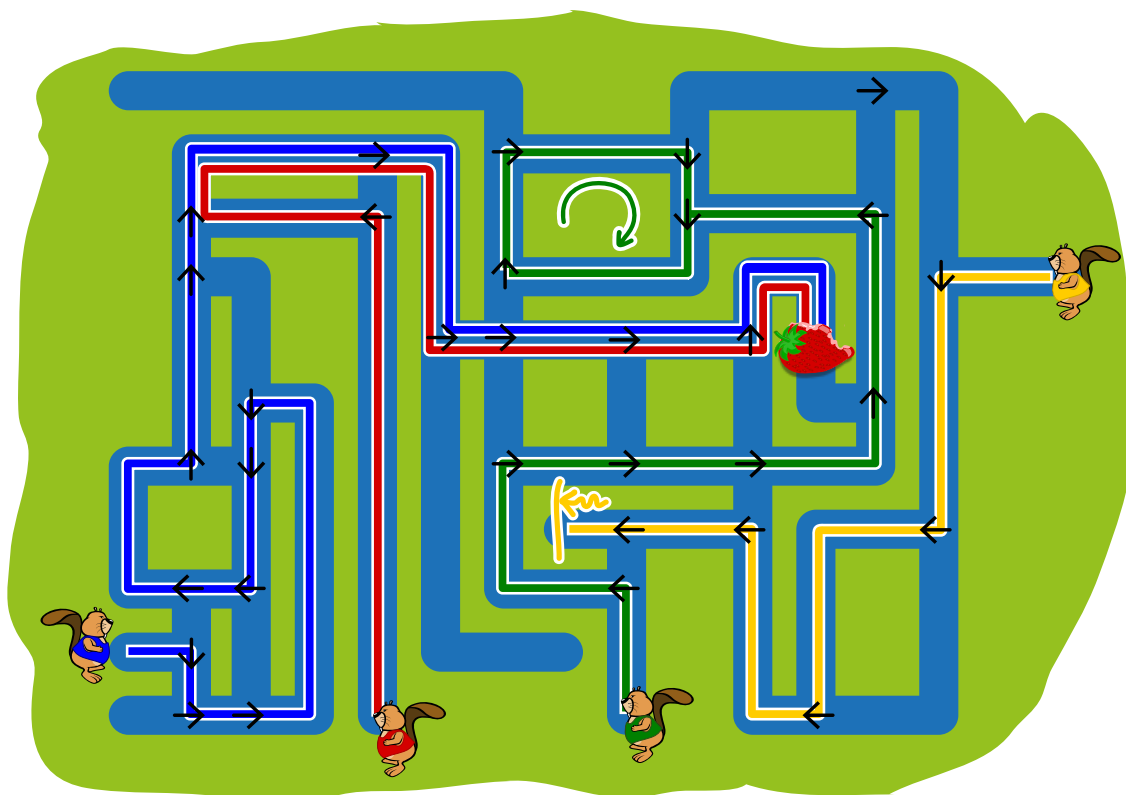
Combien de castors arriveront jusqu'à la fraise ?

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



Solution

Deux des quatre castors arriveront jusqu'à la fraise, les deux castors qui se trouvent en bas à gauche :



Le troisième castor en partant de la gauche finira par nager en rond et le castor en haut à droite, terminera son parcours dans une impasse.

C'est de l'informatique !

Le système des canaux illustré dans la figure ci-dessus distingue deux sortes d'éléments : des canaux (à travers lesquels les castors passent) et des intersections (où les flèches indiquent aux castors la direction à suivre). En informatique, on appelle un tel système un *graphe*, les canaux sont nommés *arêtes* et les intersections s'appellent *noeuds*. Les arêtes et les noeuds peuvent contenir des informations. Dans notre exercice, les noeuds indiquent la direction à suivre, c'est-à-dire les arêtes qu'il faut suivre.

Les informaticiens peuvent se servir des graphes dans maintes situations, comme par exemple pour la programmation des ordinateurs. Les noeuds d'un graphe contiennent des instructions qui indiquent à l'ordinateur quelle opération doit être exécutée ensuite. Parfois, il arrive ainsi à résoudre un problème (comme par exemple à arriver à la fraise). Parfois, par contre, son parcours se termine dans une impasse ou, pire encore, dans une boucle sans fin dont il ne sortira plus sans une intervention extérieure.

Sites web et mots clés

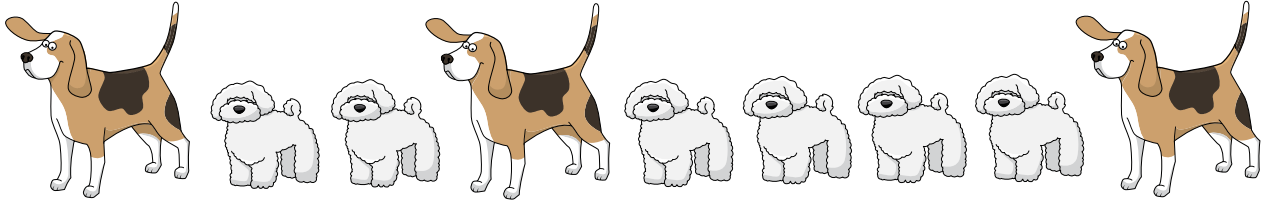
lire des graphes

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

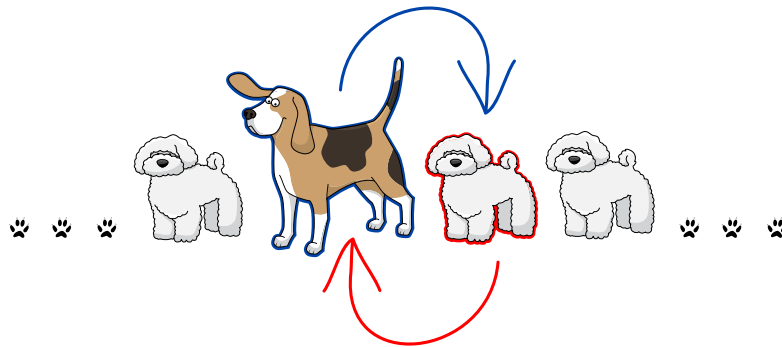


5. Déplacer des chiens

Des chiens de deux races différentes se placent en rang, l'un à côté de l'autre :



Quand deux chiens qui sont placés l'un à côté de l'autre changent leurs places, nous parlons d'un déplacement :



Suite à quelques déplacements, les trois grands chiens se retrouvent côte à côte.

Comment faut-il procéder pour déplacer les chiens le moins possible afin que les trois grands chiens se retrouvent côte à côte ?

- A) 5
- B) 6
- C) 7
- D) 8



Solution

La réponse B) (6) est correcte. D'abord, il faut déplacer le premier grand chien deux fois vers la droite. Ensuite, on déplace le dernier grand chien quatre fois vers la gauche.

Ainsi, on déplace chaque petit chien une fois puisque chaque petit chien est placé entre deux grands chiens. Le déplacement de deux petits chiens est sans effet, c'est pourquoi on préfère échanger les positions d'un petit chien avec celle d'un grand chien. Et comme il y a six petits chiens en tout, il y aura forcément au moins six déplacements.

Placer tous les grands chiens vers la droite ou vers la gauche nécessitera six déplacements supplémentaires.

C'est de l'informatique !

Afin de stocker des données, l'ordinateur utilise un ensemble de ressources appelé *mémoire* (Memory). La mémoire peut être interne (désigné par le terme RAM) ou externe (comme par exemple un disque dur ou une clé USB). Un ordinateur est équipé de plusieurs types de mémoires qu'il utilise alternativement : la mémoire interne, par exemple, permet à l'ordinateur d'accéder aux données plus rapidement que la mémoire externe. Cette dernière, par contre, est beaucoup moins chère, c'est pourquoi les ordinateurs modernes utilisent plutôt les mémoires externes qu'internes. Comme l'ordinateur peut lire et écrire plus rapidement sur la mémoire interne, les scientifiques ont intérêt à rechercher des solutions pour que l'ordinateur utilise la mémoire interne plutôt que la mémoire externe.

Dans la présente tâche, nous avons dû déplacer deux chiens c'est-à-dire deux éléments. Si nous considérons ces éléments comme des unités de données à l'intérieur de la mémoire, ce déplacement représentera l'échange de position de deux unités de données. Dans le cas où ces unités sont stockées dans une mémoire externe, il convient d'effectuer aussi peu de déplacements que possible.

Nous connaissons de tels déplacements dans le domaine de différents algorithmes de tri. Là aussi, pour que le processus soit efficace, il convient de minimiser le nombre des déplacements. Un algorithme de tri est, par exemple, le *tri à bulles*. Cet algorithme compare répétitivement des éléments consécutifs d'un *tableau* (d'une structure de données) pour les inverser s'ils sont mal triés : par exemple, il déplace les plus grandes unités de données en fin de tableau.

Sites web et mots clés

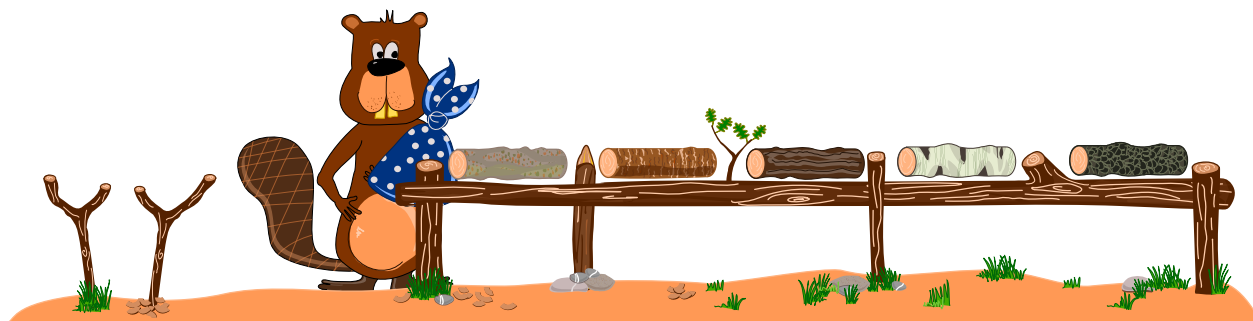
déplacement, mémoire interne, mémoire externe, tri à bulles, tableau

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



6. 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.

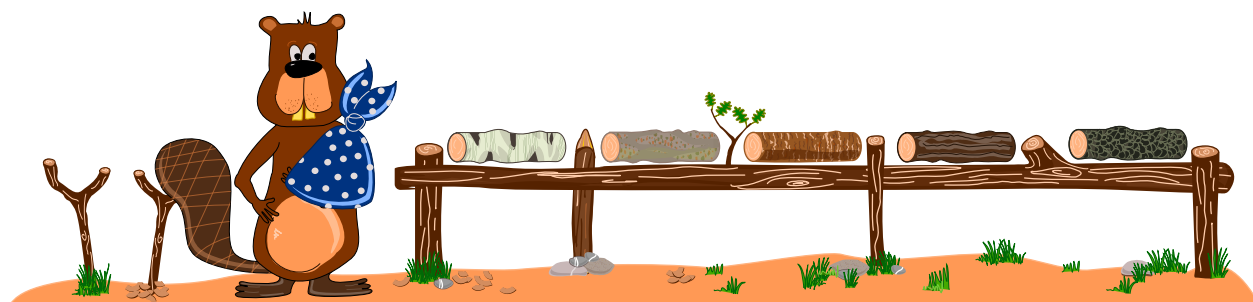


David te demande de l'aider à classer les différentes bûches selon leur couleur, de la plus claire à la plus foncée. Il aimerait bien que la bûche la plus claire se trouve à l'extrême gauche et la bûche la plus foncée à l'extrême droite.



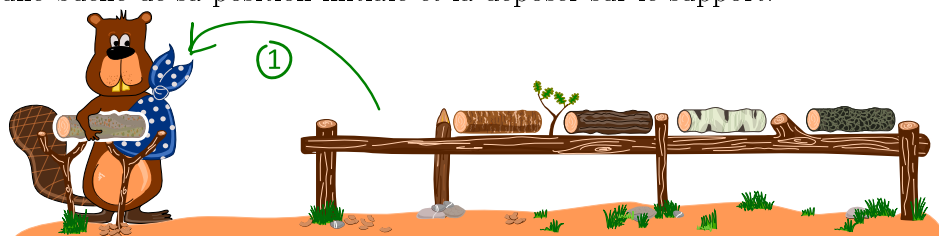
Solution

La réponse correcte est :

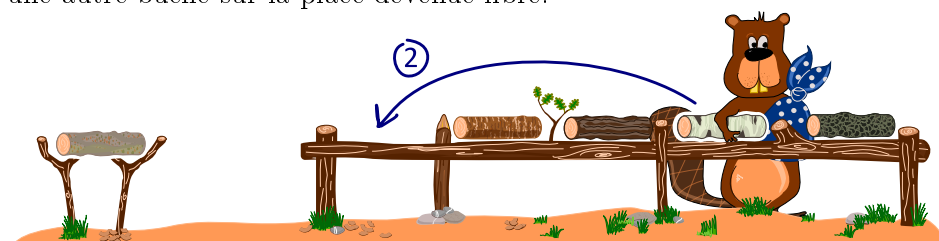


Pour que David puisse résoudre ce problème, il est nécessaire qu'il utilise le support à sa droite. Il existe plusieurs possibilités pour classer les bûches correctement, mais la plus simple est qu'il échange deux bûches en entreposant une des deux bûches sur le support :

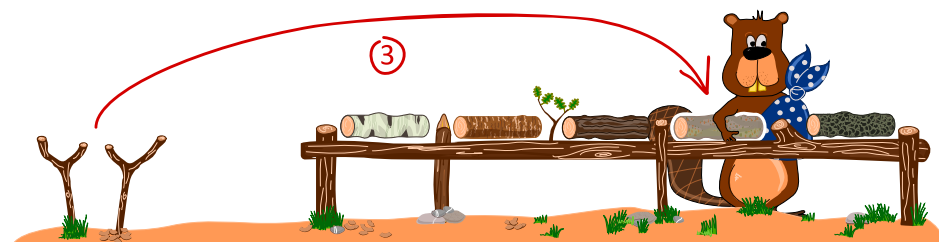
- Enlever une bûche de sa position initiale et la déposer sur le support.



- 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*).

Sites web et mots clés

intervertir des variables, (algorithme de) tri

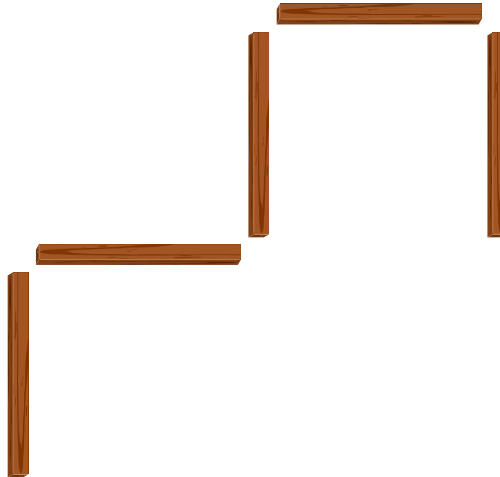
- [https://fr.wikipedia.org/wiki/Permutation_\(informatique\)](https://fr.wikipedia.org/wiki/Permutation_(informatique))
- https://fr.wikipedia.org/wiki/Algorithme_de_tri



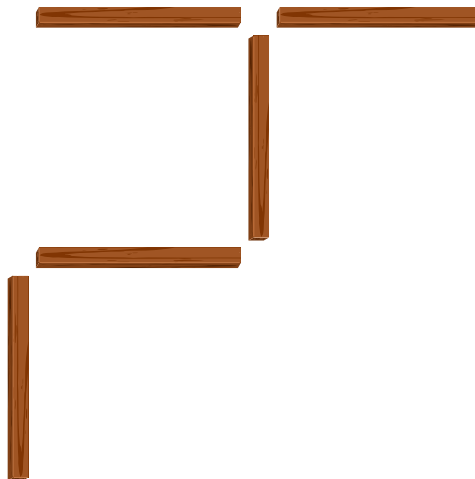


7. Cinq petits bouts de bois

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

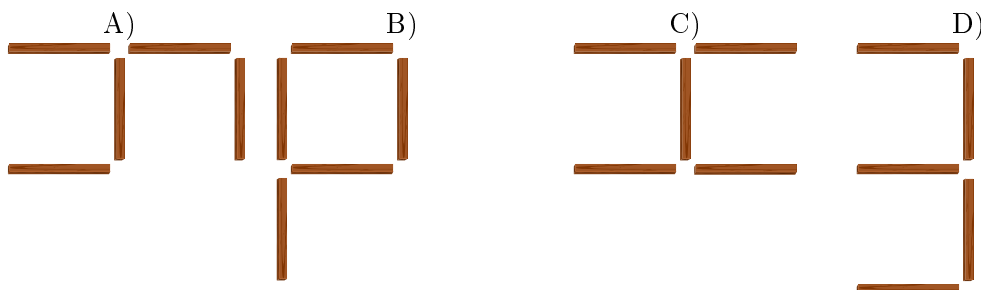


Nola prend un petit bout de bois et le déplace comme suit :



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

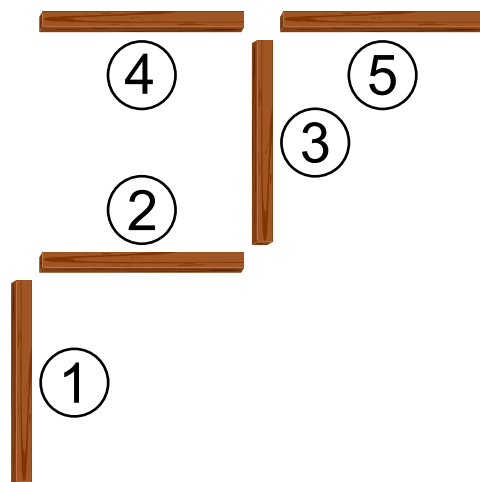
Lequel des arrangements de petits bouts de bois suivants ne correspond pas à un arrangement possible après la dernière opération ?





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° 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° 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° 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° 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.

Sites web et mots clés

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





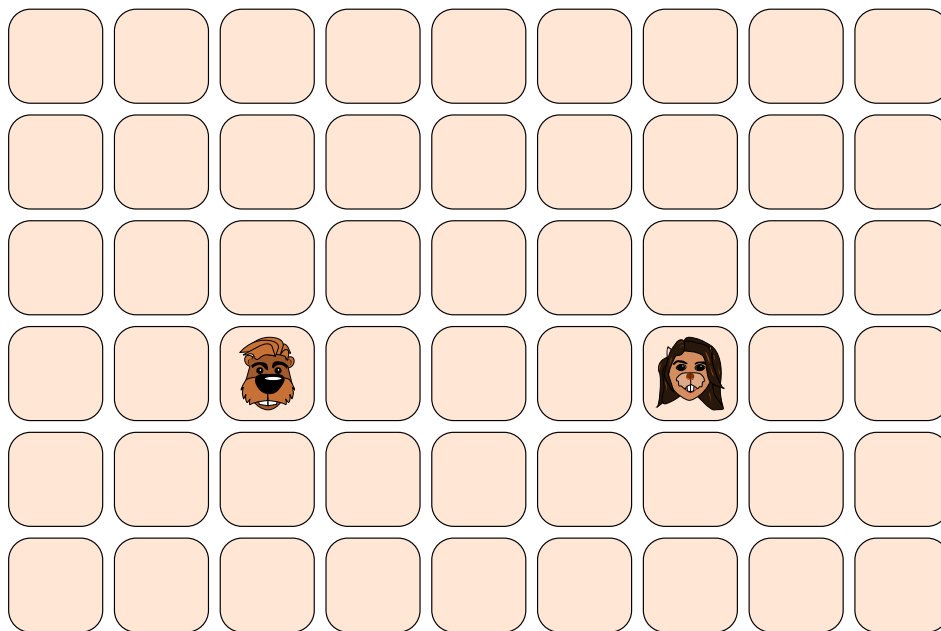
8. Concours de danse : à qui le tour ?

Un écureuil (🐿️) et un castor (🦫) participent à un spectacle de danse. Or, leurs mouvements dépendent des spectateurs : selon leurs réactions, les participants effectuent différents pas de danse. Le tableau suivant montre quelle réaction déclenche quels mouvements :

	Waouh !	Aïe !	Applaudissements !	Huées !
🐿️	← ↑	↑ ←	← ← ↑	↓ ↓
🦫	↑ →	→ ↓	↑ ↑ ↑	← ←

Quand, par exemple, les spectateurs crient «Aïe!», l'écureuil se déplacera d'abord d'une case vers le haut, puis d'une case vers la gauche ; en même temps, le castor se déplacera d'abord d'une case vers la droite, puis d'une case vers le bas.

Les deux participants commencent leur spectacle à partir des points de départ suivants :



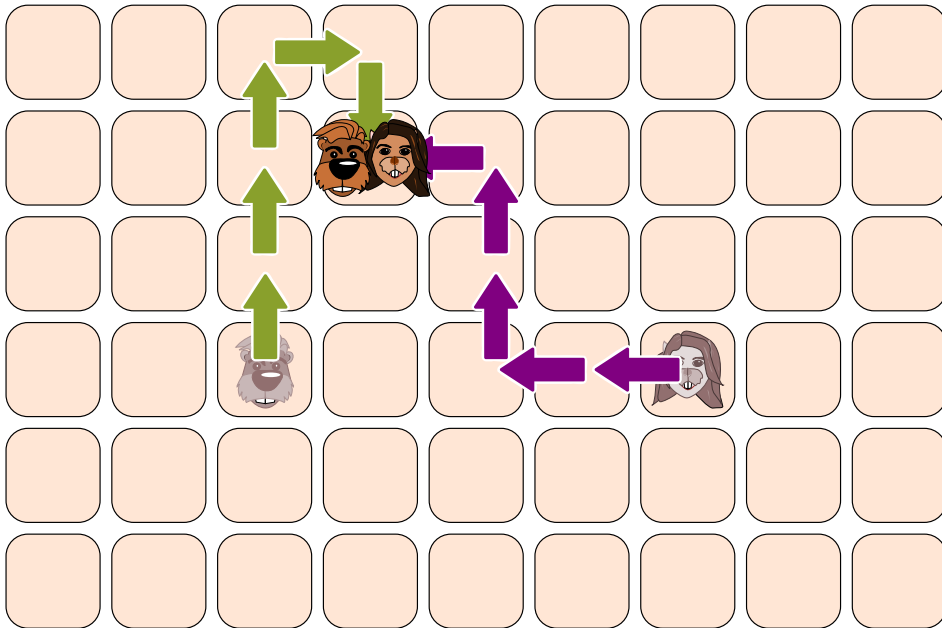
Laquelle des séquences de réactions suivantes provoquera une rencontre inévitable des deux participants sur la même case ?

- A) Huées! Aïe!
- B) Waouh! Aïe!
- C) Aïe! Aïe!
- D) Applaudissements! Aïe!



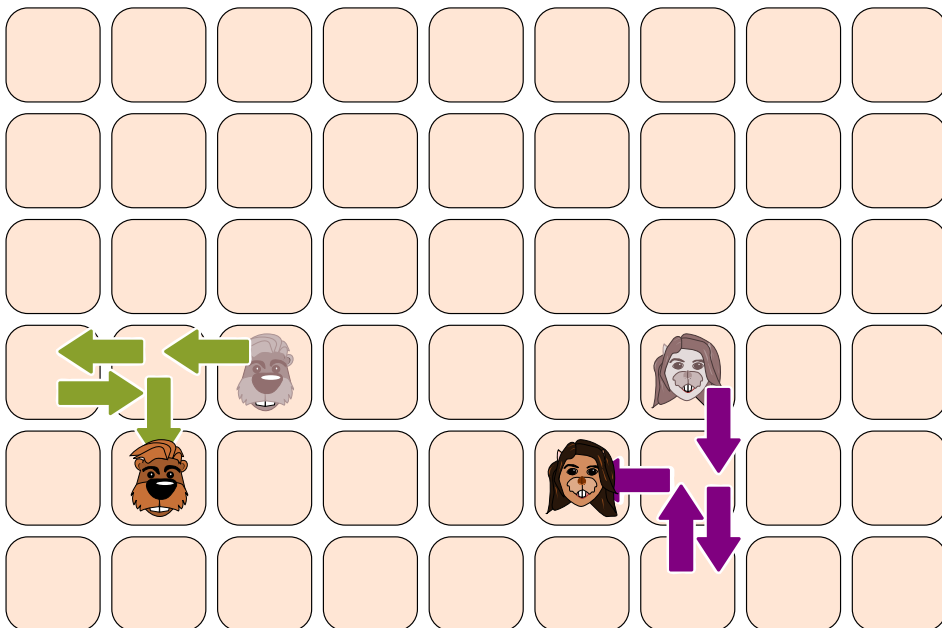
Solution

La réponse D) «Applaudissements! Aïe!» est correcte :



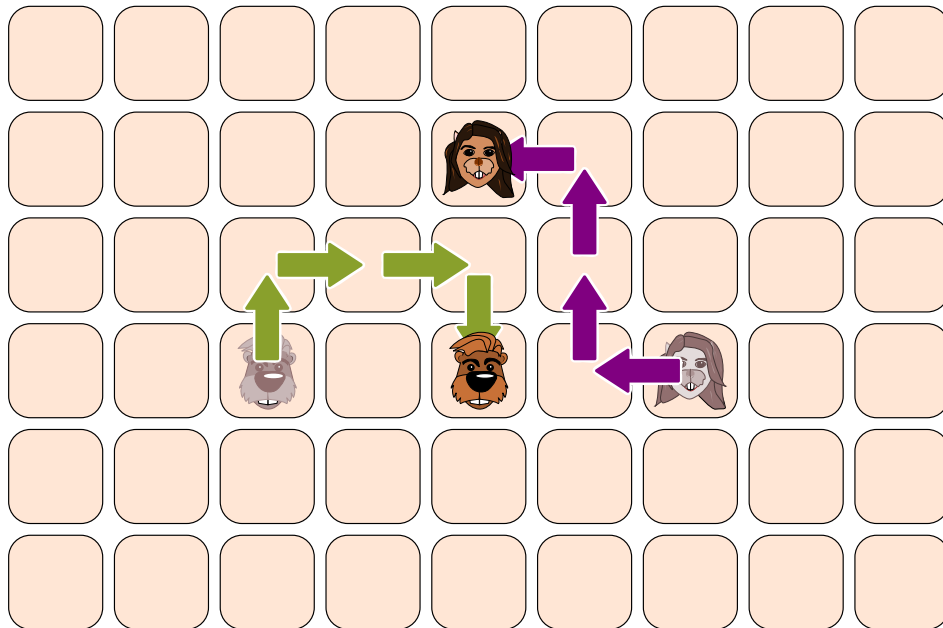
Dans tous les autres cas, l'écureuil et le castor ne se rencontreront pas sur la même case :

A) «Huées! Aïe!» :

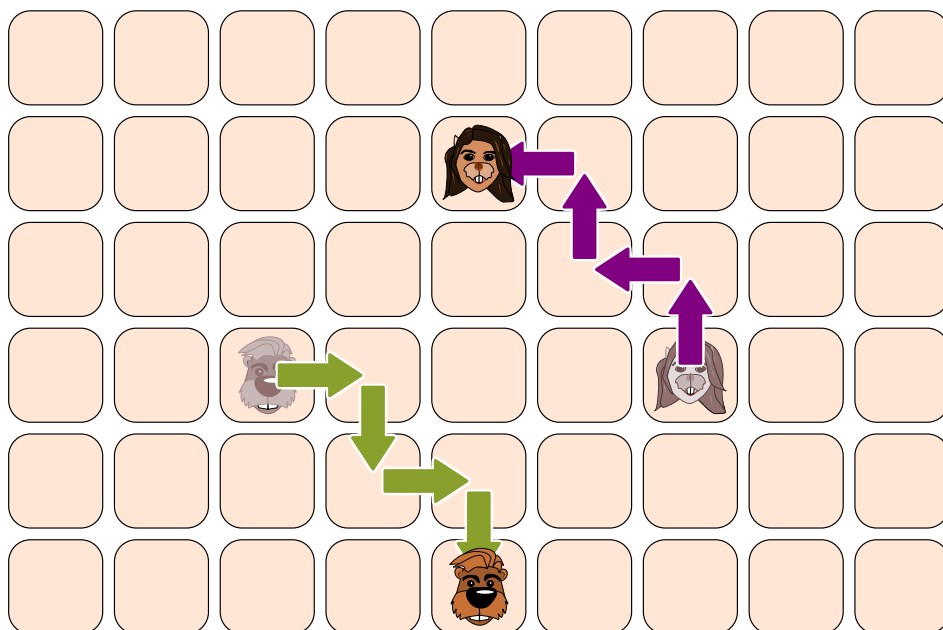




B) «Waouh! Aïe!» :



C) «Aïe! Aïe!» :



C'est de l'informatique !

La présente tâche illustre le *parallélisme* (informatique) c'est-à-dire un processus informatique qui permet de traiter des informations de manière simultanée afin de réaliser le plus grand nombre d'opérations dans un temps minimal. Dans notre cas, les deux «acteurs» bougent de manière indépendante tout en respectant des règles prédéfinies. Cependant, en informatique, au cas où deux acteurs entrent en collision, par exemple, quand ils font appel à la même ressource et ceci en même temps (une ressource peut être la mémoire vive ou un appareil raccordé à l'ordinateur comme par exemple une imprimante), il est nécessaire de préciser lequel des deux acteurs pourra accéder à la ressource en premier. Ceci est possible à l'aide des *sémaphores*, par exemple. Ainsi, un acteur peut



réserver et utiliser des ressources pour les libérer ensuite quand il n'en aura plus besoin. Seule règle : pendant qu'un acteur a réservé une ressource, aucun des autres acteurs n'a le droit d'accéder à la même ressource. Pour éviter que deux acteurs réservent une seule et même ressource, il faut donc un système central (en général, il s'agit du système d'exploitation), à l'intérieur duquel l'*ordonnanceur* gère les réservations des ressources et choisit l'ordre d'exécution des processus sur les processeurs d'un ordinateur.

Pendant que tu réfléchissais à la résolution du problème, tu as probablement effectué une *simulation* des mouvements des deux acteurs. Les *simulations informatiques* peuvent se révéler très pratiques au quotidien aussi. Ainsi, les prévisions météorologiques, par exemple, sont élaborées à l'aide des modèles numériques et de l'analyse des résultats effectuée par les prévisionnistes.

Sites web et mots clés

parallélisme (informatique), sémaphore, ordonnanceur, simulation informatique

- [https://fr.wikipedia.org/wiki/Parallélisme_\(informatique\)](https://fr.wikipedia.org/wiki/Parallélisme_(informatique))
- [https://fr.wikipedia.org/wiki/Sémaphore_\(informatique\)](https://fr.wikipedia.org/wiki/Sémaphore_(informatique))
- https://fr.wikipedia.org/wiki/Simulation_informatique



9. Le nom japonais

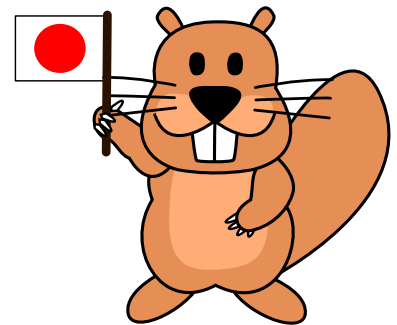
Une amie japonaise nous raconte que, selon une ancienne tradition, on peut transformer les lettres d'un prénom en un nom spirituel japonais. Il faut simplement remplacer chaque lettre par une syllabe qui lui est attribuée :

A → ka	F → lu	K → me	P → mor	U → do	Z → zi
B → pi	G → ji	L → ta	Q → ke	V → ru	
C → mi	H → ri	M → rin	R → shi	W → mei	
D → te	I → ki	N → to	S → ari	X → na	
E → ku	J → zu	O → mo	T → chi	Y → fu	

Un de ses amis provenant de la Croatie porte, par exemple, le nom spirituel «Zukame Moru».

Quel est le vrai nom de son ami croate ?

- A) Josip
- B) Jani
- C) Jakov
- D) Jurica





Solution

Son ami s'appelle Jakov. Pour trouver la bonne solution, on peut débiter par la syllabe «Zu» qui, selon le tableau, a été attribuée à la lettre «J». Après avoir trouvé, de la même manière, la lettre qui correspond à la syllabe «ka» (la lettre «A»), on réalise que les deux solutions «Josip» et «Jurica» ne sont pas correctes. De même, après avoir trouvé la lettre originale à laquelle est attribuée la syllabe «me» (la lettre «K»), on comprend que la solution C) est correcte. Pour être sûr, on peut vérifier le mot «Moru» en traduisant directement les lettres «O» («mo») et «V» («ru») en syllabes. Quand on traduit les lettres en syllabes, on peut procéder plus vite qu'à l'inverse car les lettres du tableau suivent l'ordre alphabétique, les syllabes, elles, par contre, ne sont pas triées.

Une autre approche, plus vite encore, serait de chercher dès le début la lettre originale qui correspond à la syllabe «ru». Celle-ci remplace la lettre «V» et comme Jakov est le seul nom qui se termine par un «V», on peut être sûr qu'il s'agit du prénom que l'on cherche.

C'est de l'informatique !

Il se peut que tu aies déjà entendu parler des jeux similaires. Souvent, ils sont utilisés pour créer des langages secrets. L'idée de base repose sur le fait qu'une lettre est substituée à une autre lettre ou à une syllabe distinctive. En informatique, ce langage est appelée *langage congruentiel* ou *système de réécriture* ou encore *système de semi-Thue*.

Malheureusement, ces méthodes de chiffrement ne sont pas sûres car elles ont pour base des *algorithmes de chiffrement monoalphabétiques*. Elles sont ainsi vulnérables aux attaques et peuvent être facilement déchiffrées, même sans que l'on ait recours à un ordinateur.

Notre tâche révèle un autre défi au niveau de l'informatique : dans la partie «Moru», on peut également identifier la syllabe «mor» qui remplace la lettre «P». La syllabe «mo» qui, elle, remplace la lettre «O», est en fait un préfixe de «mor». Au cas où un ordinateur doit déchiffrer un prénom spirituel, l'informaticien doit être conscient du problème des préfixes afin que ses réflexions ne conduisent pas à des conclusions erronées.

D'ailleurs, toutes les syllabes que tu trouves dans le tableau de notre tâche proviennent du langage japonais. C'est la raison pour laquelle tous les prénoms traduits selon cette méthode sonnent fortement comme de vrais noms japonais.

Sites web et mots clés

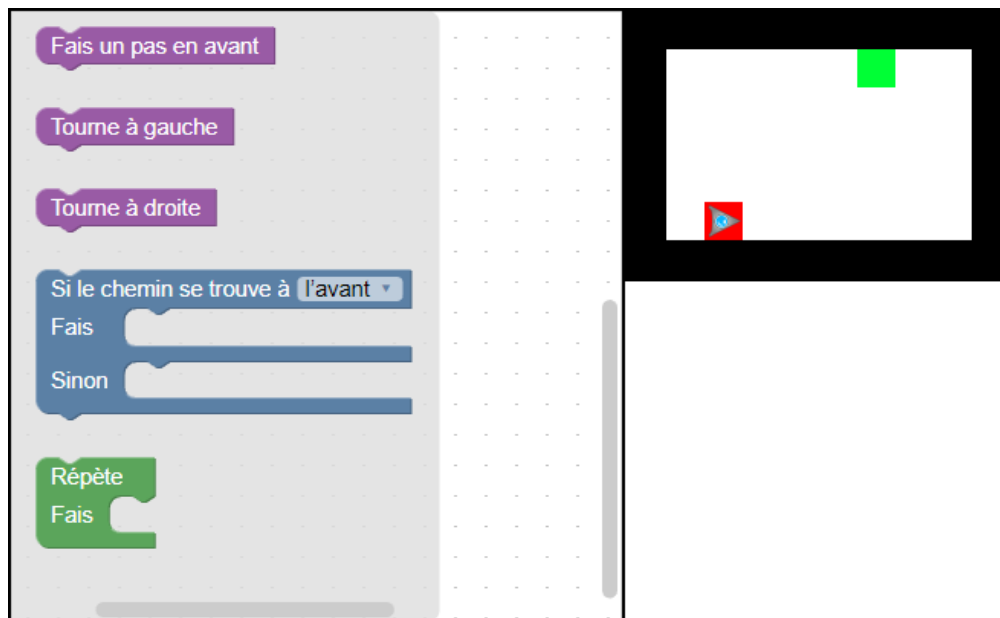
langage congruentiel, système de réécriture, système de semi-Thue, chiffrement monoalphabétique.

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



10. Un programme court

Un robot de forme triangulaire doit parcourir le chemin suivant : il commence tout en bas dans la zone marquée en rouge et se dirige vers la cible, qui est la zone marquée en vert. Malheureusement, il n'arrive à lire que des programmes très courts.

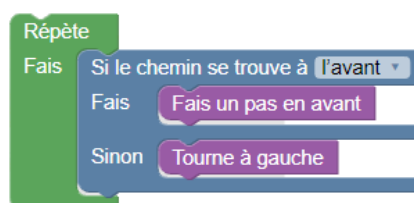


Pour créer un tel programme, déplace les instructions et insère-les dans l'organigramme de programmation afin qu'elles suivent un ordre logique. Seule règle : tu ne dois pas utiliser plus de 4 instructions.



Solution

Voilà une solution possible :



L'idée de base de ce programme est de faire avancer le robot le long du mur et de le faire tourner à gauche à chaque angle. Ainsi, il arrivera automatiquement à destination.

C'est de l'informatique !

Dans le champ formé de 8×5 pas, le robot devrait au moins effectuer neuf mouvements pour arriver à destination : tout droit – tout droit – tout droit – tout droit – à gauche – tout droit – tout droit – tout droit – tout droit.

Pourtant, le programme ne comprend que quatre instructions, c'est la raison pour laquelle nous avons besoin d'une boucle (loop) qui permettra au robot de faire autant de pas en avant et autant de virages à gauche que nécessaire jusqu'à ce qu'il arrive à destination. En outre, pour pouvoir contrôler les mouvements du robot, surtout celui où il faut tourner à gauche sans que le robot entre en collision avec le mur, il faut introduire une instruction conditionnelle.

Malheureusement, ce programme de courte durée ne permet pas de trouver le chemin le plus court et puis, il ne fonctionne pas dans toutes les situations non plus. Si, par exemple, notre cible ne se trouvait pas le long d'un mur, le robot ne la trouverait en aucun cas.

Les séquences d'instructions, les boucles ainsi que les instructions conditionnelles constituent la base de chaque algorithme.

Sites web et mots clés

programmer, boucle, condition, instruction

- <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/>

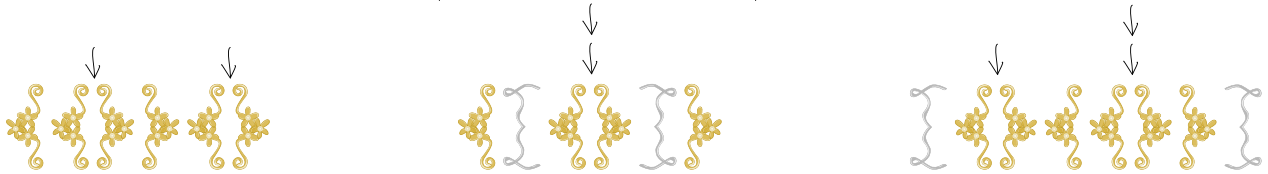


11. Orner des pendentifs médiévaux

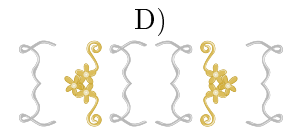
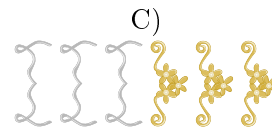
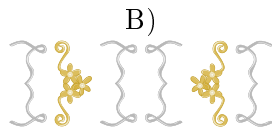
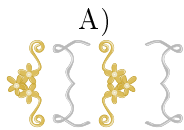
Pour une fête médiévale, la famille Castor produit des bijoux, plus précisément des pendentifs. La décoration de ces derniers consiste à placer des ornements qui ressemblent à des accolades. Chaque accolade est utilisée par paires. Pour décorer le bijou, les Castor appliquent une méthode particulière : on commence par une des deux paires d'accolades suivantes :



Ensuite, on ajoute une autre paire d'accolades à plusieurs reprises et à un endroit quelconque, à la seule condition que les paires supplémentaires soient placées entre deux accolades d'une paire d'accolades placée préalablement (voir l'exemple ci-dessous) :



Lequel des pendentifs a été décoré selon la méthode décrite ci-dessus ?





Solution

La réponse D) est correcte. Entre les deux accolades de l'ornement initial, on a placé une paire d'accolades supplémentaire, entre les accolades de cette dernière, on a placé une autre paire d'accolades. Tous les autres pendentifs n'ont pas été décorés selon la méthode développée par la famille Castor :

- A) Partant de la gauche, la première erreur s'est produite à la position 3 : on a fermé la première paire d'accolades avant que l'on ait pu placer la seconde accolade de la deuxième paire (position 2).
- B) Partant de la gauche, la première erreur s'est produite à la position 1 : on a placé la seconde accolade de l'ornement initial sans avoir placé la première accolade.
- C) Partant de la gauche, la première erreur s'est produite à la position 4 : on a fermé une paire d'accolades avant même de l'avoir ouverte.

C'est de l'informatique !

Les règles de l'ornement des pendentifs correspondent tout à fait à celles utilisées en mathématiques ou en informatique. Si une entité lexicale d'un langage informatique est correcte, on l'appelle «bien formée». Une entité bien formée est également appelée *syntactiquement correcte* parce qu'elle respecte la grammaire formelle du langage en question, c'est-à-dire les règles d'agencement des entités lexicales de ce langage. En règle générale, il est plus facile de corriger une suite d'entités lexicales d'un langage qui comporte des erreurs de syntaxe qu'une suite d'entités qui comporte des «erreurs sémantiques». Ces dernières sont plus subtiles et peuvent comprendre des erreurs de raisonnement.

Sites web et mots clés

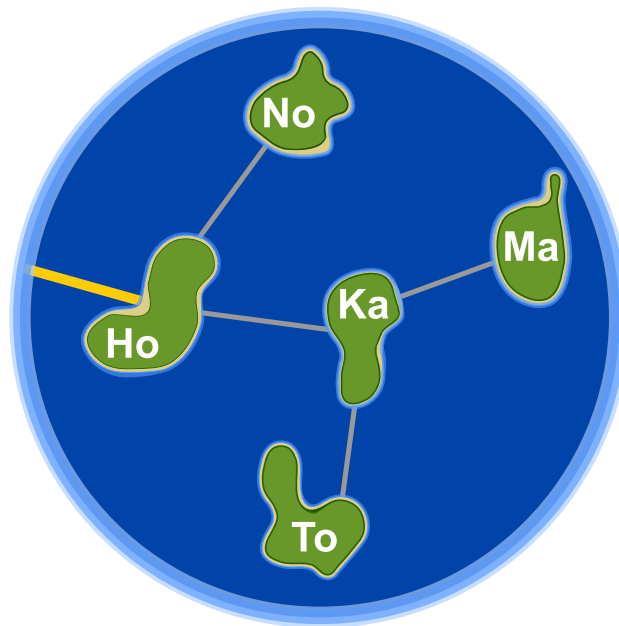
bien formée, syntaxe, sémantique

- <https://fr.wikipedia.org/wiki/Syntaxe>
- <https://fr.wikipedia.org/wiki/Sémantique>



12. Honomakato

L'archipel Honomakato est formé de cinq îles Ho, No, Ma, Ka et To. L'île principale Ho est connectée à Internet par un câble. En outre, quelques câbles parcourent les îles Ho et No, Ho et Ka, Ka et Ma ainsi que Ka et To. Toutes les îles sont donc connectées à l'île principale Ho et par conséquent à Internet.



Les habitants de Honomakato demandent une connexion fiable à Internet pour toutes les îles : cela veut dire que même si un des câbles est endommagé, chacune des îles doit avoir accès à Internet. *Fais en sorte que l'archipel Honomakato obtienne une connexion fiable à Internet. Pose deux autres câbles entre les îles. Il existe plusieurs solutions possibles.*

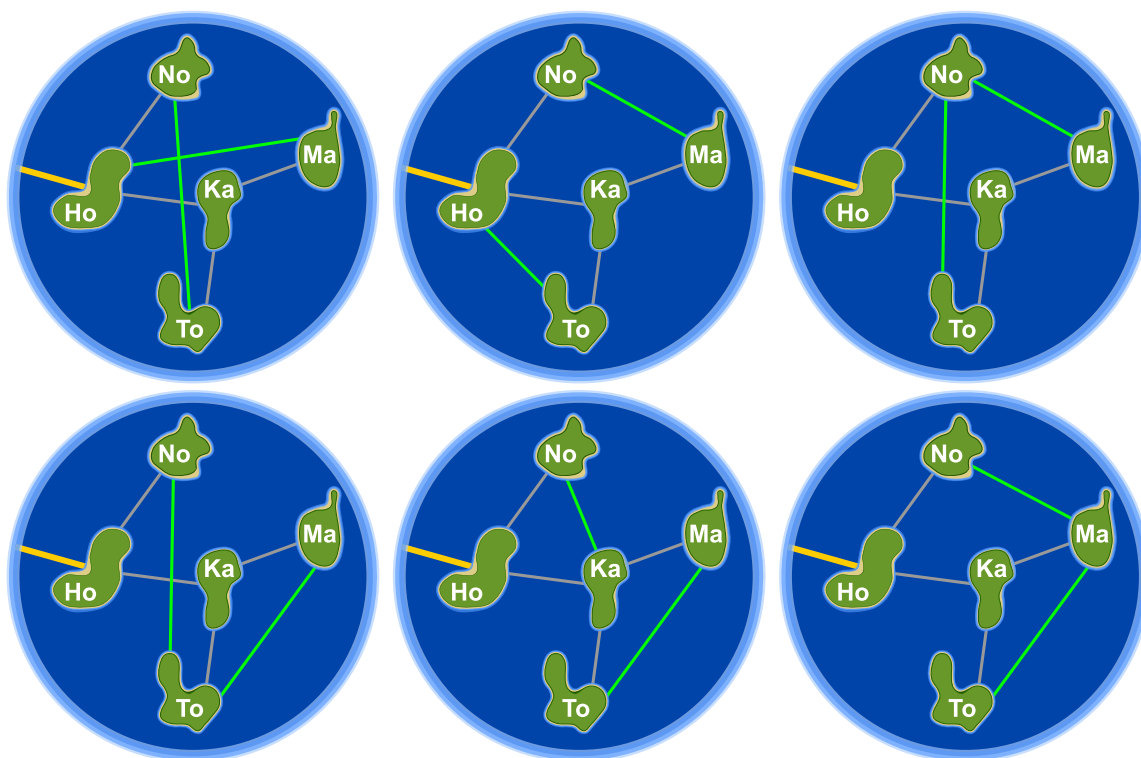


Solution

En posant deux autres câbles, l'archipel Honomakato obtient une connexion fiable à Internet. Pour résoudre notre problème, il existe six possibilités différentes. Les câbles posés préservent les îles du moment inopportun où elles n'auraient plus accès à Internet si un câble était endommagé.

1. Ho-Ma et No-To : Ho-Ma préserve Ma und Ka, No-To préserve No und To.
2. Ho-To et No-Ma : Ho-To préserve To et Ka, No-Ma préserve Ma, No, et Ka.
3. No-To et No-Ma : No-To préserve No, To et Ka, No-Ma préserve Ma, No, et Ka.
4. No-To et Ma-To : No-To préserve No, To et Ka, Ma-To préserve Ma et To.
5. No-Ka et Ma-To : No-Ka préserve No et Ka, Ma-To préserve Ma et To.
6. No-Ma et Ma-To : No-Ma préserve Ma, No, et Ka, Ma-To préserve Ma et To.

En général, pour chaque solution, les règles suivantes doivent être respectées : (1) chaque île est équipée d'au moins deux connexions et (2) il n'est pas possible de diviser l'archipel Honomakato en deux groupes qui n'ont qu'une seule connexion.



C'est de l'informatique !

D'une part, le réseau de câbles avec lequel l'archipel Honomakato est connecté à Internet ne représente qu'une petite partie du réseau global. D'autre part, il sert aussi d'exemple pour montrer comment le réseau global est construit. Les routeurs, les serveurs et les autres dispositifs télématiques pourvus d'une propre adresse Internet sont des nœuds d'un grand réseau global appelé «Internet» ; dans notre tâche, les îles de l'archipel Honomakato représentent justement ces nœuds.

Internet a été conçu dans les années 1960 comme un réseau robuste (appelé aussi «fiable»). Un des objectifs était qu'une panne de connexion entre les nœuds de réseau ne provoquerait pas une panne dans le réseau entier. C'est la raison pour laquelle on connecte les nœuds de manière multiple et on les



configure de façon à ce que, en cas de défaillance ou congestion d'une connexion, on ait la possibilité de recourir à une autre connexion. Cette précaution est également importante pour d'autres réseaux, comme par exemple pour les réseaux de transport ou les réseaux d'approvisionnement.

En informatique, on utilise la théorie des graphes pour effectuer des calculs concernant ces types de réseaux. Un graphe (en théorie des graphes) est un réseau qui se compose de «points» appelés des *nœuds* ou *sommets* et de «liens» entre les nœuds appelés des *arêtes*. Un graphe est appelé «connexe» lorsque pour chaque paire de nœuds A et B, B est connecté avec A à travers au moins une arête. Une seule arête est donc nécessaire pour qu'un graphe soit connecté et dans ce cas-là, l'arête assume la fonction d'un *pont*. En informatique, on développe des algorithmes qui sont capables de repérer ces ponts à l'intérieur des graphes. Robert Tarjan a développé un de ces algorithmes (efficaces).

Sites web et mots clés

structure de données dynamique, graphe, pont (informatique)

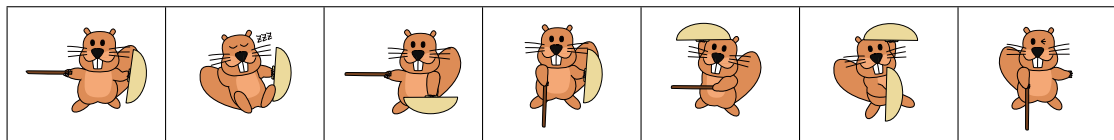
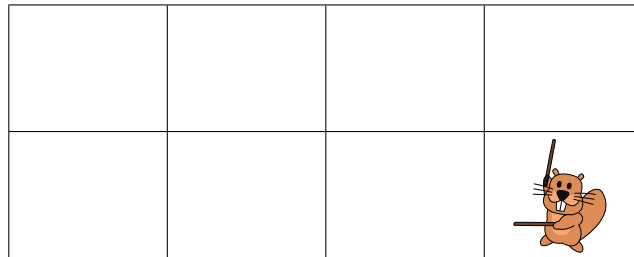
— [https://fr.wikipedia.org/wiki/Séparateur_\(théorie_des_graphes\)](https://fr.wikipedia.org/wiki/Séparateur_(théorie_des_graphes))





13. Un art martial japonais

Lucia et ses amis sont membres d'un club d'art martial japonais qui enseigne le maniement du bâton. Pour une photo dans la cour de récréation, ils aimeraient bien se mettre en place afin que chaque bâton vise un bouclier. Pour que chacun puisse se mettre correctement en place, on a dessiné quelques cases sur le sol de la cour de récréation. Lucia a déjà choisi une case et elle montre sa pose préférée. En dessous, tu peux voir tous ses amis qui présentent leur propre pose préférée :

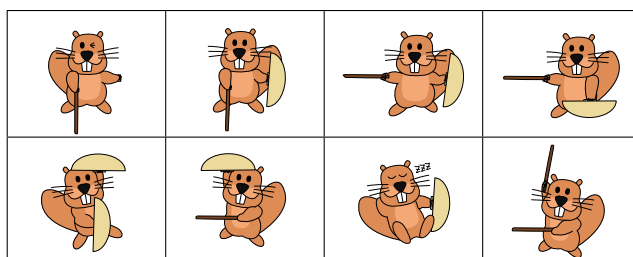


Déplace les images des amis dans les cases dessinées sur le sol de la cour de récréation pour que chaque bâton vise un bouclier.



Solution

La réponse correcte est :



Si on déplace les images des amis comme indiqué ci-dessus, chaque bâton visera un bouclier. Pour résoudre ce problème, il n'existe pas d'autre solution.

C'est de l'informatique !

Nous avons sept images qu'il faut déplacer et mettre au bon endroit. Celui qui tente de résoudre la tâche par essai-erreur aura besoin de beaucoup de temps : il existe en fait $1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 = 7! = 5040$ arrangements possibles et la plus grande partie d'entre eux est naturellement fausse. Par la logique, tu pourras trouver la solution plus rapidement :

1. Tous les castors dont le bâton ou le bouclier montre le haut doivent être placés dans la rangée inférieure.
2. Tous les castors dont le bâton ou le bouclier montre le bas doivent être placés dans la rangée supérieure.
3. Il n'y a qu'un seul castor dont le bouclier montre vers le bas. C'est la raison pour laquelle il doit être placé au-dessus de Lucia.

Ces quelques règles aident à limiter l'espace de recherche des diverses solutions possibles et finalement, à trouver la bonne. Le fait de tester systématiquement toutes les options possibles par essai-erreur est appelé le *backtracking*. Mais une telle méthode n'est raisonnable que si l'on a restreint l'espace de recherche auparavant. Voilà pourquoi la délimitation de l'espace de recherche par des règles logique est si importante.

Sites web et mots clés

raisonnement logique, conclusion

- https://fr.wikipedia.org/wiki/Retour_sur_trace
- http://www.inf-schule.de/grenzen/komplexitaet/affenpuzzle/einstieg_affenpuzzle



14. La confiture de grand-père

Anna, Pierre et Lisa aident leur grand-père à mettre de la confiture en pots. Voilà les étapes de travail qu'il faut faire – et ceci impérativement dans l'ordre suivant :



Rincer un pot de confiture prend 3 minutes.



Mettre de la confiture dans un pot prend 2 minutes.



Fermer un pot de confiture prend 1 minute.

Anna, Pierre et Lisa aimeraient bien se répartir ces différentes étapes de travail et pour cela, ils établissent un plan. Ils doivent respecter les règles suivantes : une tâche doit être complètement accomplie avant qu'ils n'entament la prochaine. Ainsi, un pot de confiture ne peut être fermé avant qu'il ait été d'abord rincé, puis rempli de confiture.

Ce plan, par exemple, ne respecte pas les règles préalablement déterminées :



ANNA										
PIERRE										
LISA										

Anna, Pierre et Lisa aimeraient bien remplir un maximum de pots en 10 minutes. Aide-les à établir un plan qui respecte toutes les règles.

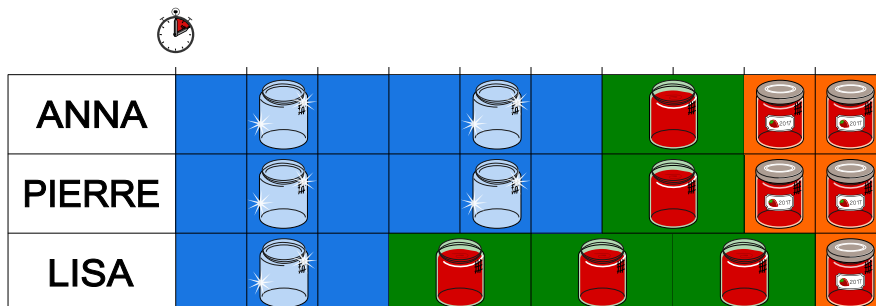


ANNA										
PIERRE										
LISA										



Solution

Voilà un plan selon lequel Anna, Pierre et Lisa arrivent à mettre de la confiture en pots en 10 minutes. Il existe naturellement bien d'autres plans selon lesquels ils arriveraient à atteindre leur objectif. Par exemple, ils pourraient tout simplement conserver les mêmes agencements de tâches, mais en se les répartissant différemment entre eux.



Mais ils n'arriveront jamais à remplir plus de 5 pots en 10 minutes. Remplir un pot prend 6 minutes ($3 + 2 + 1 = 6$). Chacun des trois enfants ne peut travailler que pendant 10 minutes. Le temps de travail au total est donc de 30 minutes. Pendant ces 30 minutes, ils arrivent à remplir $30 : 6 = 5$ pots au maximum.

C'est de l'informatique !

Dans cette tâche, les trois enfants Anna, Pierre et Lisa travaillent en même temps, c'est-à-dire qu'ils effectuent en parallèle les différentes étapes de travail. En informatique, le traitement en parallèle est un domaine important. Les centres de calculs traitent à l'aide de beaucoup d'ordinateurs de grandes quantités de données en parallèle. Pour ce faire, on doit distribuer les différentes opérations à différents ordinateurs de manière logique, comme c'était le cas dans notre tâche quand il s'agissait de répartir les différentes étapes de travail aux trois enfants. Les ordinateurs individuels ou les smartphones, eux aussi, possèdent plusieurs processeurs principaux (multiprocesseurs) capables de traitements parallèles. En informatique, on appelle la planification du travail *scheduling*.

En dehors de l'informatique, le scheduling est très fréquent aussi. Dans la gestion de grands projets, par exemple, pour planifier les différentes étapes, on a souvent recours à la *méthode des antécédents*, appelée aussi *réseau des antécédents*. Cet outil de planification de projets détermine l'ordre des tâches ainsi que leur durée nécessaire. La mise en oeuvre d'un plan de réseau fonctionne comme le scheduling. Ainsi, les connaissances en informatique peuvent toujours être utiles, même si on ne travaille pas comme informaticienne ou informaticien.

Sites web et mots clés

scheduling, traitement parallèle, méthode des antécédents

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

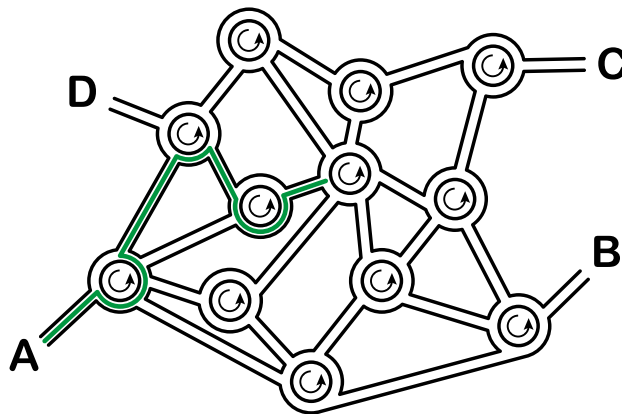


15. La ville riche en ronds-points

Dans la ville des castors, tous les carrefours ont la forme de ronds-points. Quand les habitants de la ville expliquent le chemin à un touriste, ils lui disent simplement :

- Au prochain ronds-point, prends la quatrième sortie.
- Au prochain ronds-point, prends la première sortie.
- Au prochain ronds-point, prends la deuxième sortie.

Si la personne connaît déjà assez bien la ville mais qu'elle cherche un endroit particulier, les castors ne lui indiquent qu'une suite de chiffres comme par exemple «4 1 2». Cette personne comprendra donc tout de suite qu'il faudra prendre l'itinéraire suivant :



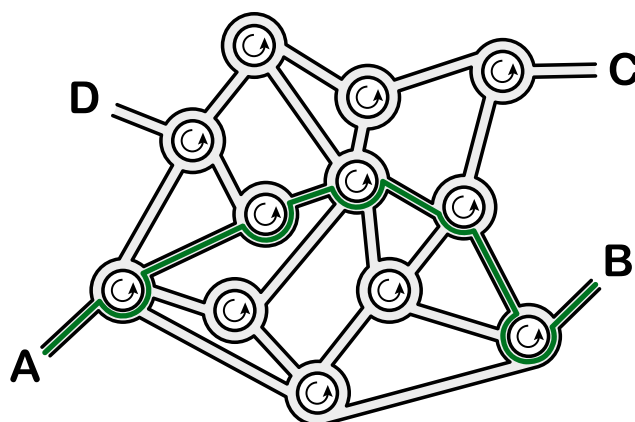
Quand un touriste part du point A, à quel endroit l'indication «3 1 3 2 3» le mènera-t-elle ?

- A) L'indication le mènera au point A.
- B) L'indication le mènera au point B.
- C) L'indication le mènera au point C.
- D) L'indication le mènera au point D.



Solution

L'indication «3 1 3 2 3» le mènera au point B :



C'est de l'informatique !

Cette tâche est un bon exemple pour illustrer «l'information structurée». Un ordinateur comprend difficilement un itinéraire formulé en détail dans notre langue. En transformant notre langue, par exemple, en une suite de chiffres, comme c'est le cas dans notre exercice, l'ordinateur sera tout de suite capable d'interpréter ces informations comme une suite d'instructions en langage machine. On appelle une telle suite une *séquence*. La séquence d'instructions est à la base de beaucoup de langages de programmation.

Dans notre exemple, il est utile que le réseau routier soit uniforme : tous les carrefours se présentent sous forme de ronds-points. On appelle de telles structures uniformes «homogènes» contrairement à des structures variées appelées «hétérogènes». En informatique, on préfère les structures homogènes aux structures hétérogènes car elles peuvent être traitées de manière automatique et efficace par un logiciel, donc beaucoup plus vite que les structures hétérogènes.


Sites web et mots clés


séquences, exécution de programmes, langage formel


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




A. Auteurs des exercices


 Nursultan Akhmetov

 Adil Aliyev


 Haim Averbuch

 Khuyagbaatar Batsuren


 Wilfried Baumann


 Daphne Blokhuis


 Eugenio Bravo

 Andrej Brodnik

 Carmen Bruni

 Anton Chukhnov

 Darija Dasović Rakijašić

 Christian Datzko

 Susanne Datzko

 Janez Demšar

 Olivier Ens


 Hanspeter Erni


 Arnheiður Guðmundsdóttir

 Martin Guggisberg


 Hans-Werner Hein


 Fredrik Heintz


 Jia-Ling Koh

 Hiroki Manabe


 Wolfgang Pohl

 Sergei Pozdniakov


 J.P. Pretti


 Daniel Rakijašić

 Chris Roffey

 Frances Rosamond

 Kirsten Schlüter


 Eljakim Schrijvers

 Maiko Shimabuku

 Monika Tomcsányiová

 Peter Tomcsányi


 Ahto Truu

 Willem van der Vegt

 Troy Vasiga

 Michael Weigend

 Hongjin Yeh

 Momo Yokoyama

 Khairul A. Mohamad Zaki



B. Sponsoring : Concours 2017

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



<http://www.roborobo.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
schweizerischervereinfürinformatikind
erausbildung//sociétésuisse del'inform
atique dans l'enseignement//societàsviz
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.