



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

Exercices et solutions 2016 Années HarmoS 9/10

<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
erusbildung//sociétésuissepourlinfor
matique dansl'enseignement//societàsviz
zera perl'informaticanell'insegnamento



Ont collaboré au Castor Informatique 2016

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

Nous adressons nos remerciements à :

Juraj Hromkovič, Giovanni Serafini, Urs Hauser, Tobias Kohn, Ivana Kosírová, Serena Pedrocchi, Björn Steffen : ETHZ

Valentina Dagienė : Bebras.org

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

Anna Morpurgo, Violetta Lonati, Mattia Monga : Italie

Gerald Futschek : Austrian Computer Society, Austria

Zsuzsa Pluhár : ELTE Informatikai Kar, Hongrie

Eljakim Schrijvers, Daphne Blokhuis, Marissa Engels : Eljakim Information Technology by, 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 2016 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 2016. 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. 38.



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 HarmoS 5 et 6) a été organisé pour la première fois en 2012.

Le Castor Informatique vise à motiver les élèves pour 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 dans l'utilisation des ordinateurs, sauf savoir «surfer» sur Internet, car le concours s'effectue en ligne sur un PC. 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 2016 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, dont 5 de degré de difficulté facile, 5 de degré moyen et 5 de degré difficile.

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>



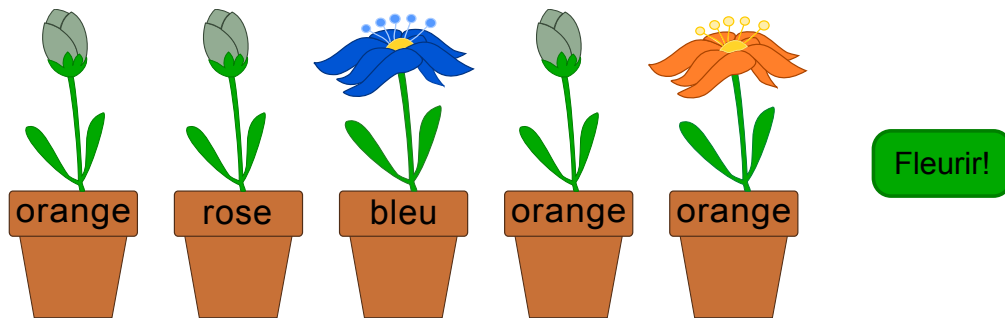
Table des matières

Ont collaboré au Castor Informatique 2016	i
Préambule	ii
1. Que les fleurs fleurissent !	1
2. Un anniversaire à deux valeurs	3
3. Tous en même temps	7
4. Le parcours de la boule	11
5. Des fleurs et des soleils	13
6. Le tchat des castors	15
7. Quatre courses à faire	17
8. Des impasses	19
9. Vive la hiérarchie !	21
10. Plongeons-nous dans la construction des ponts !	23
11. Des bonbonnières	25
12. Le code KIX	27
13. Le filtre médian	29
14. La poignée de main style zip	31
15. Cassy, la tortue	35
A. Auteurs des exercices	38
B. Sponsoring : Concours 2016	39
C. Offres ultérieures	41

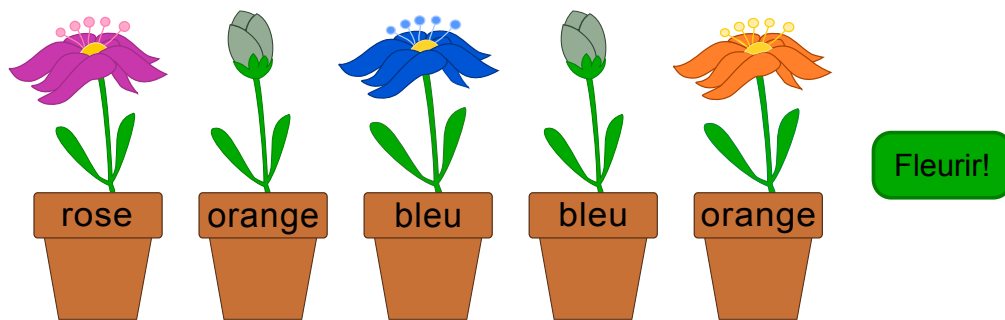


1. Que les fleurs fleurissent !

Jana joue à un jeu vidéo sur ordinateur. Secrètement, l'ordinateur a choisi les couleurs des 5 bourgeons. Les couleurs disponibles sont le bleu, l'orange et le rose. La gamme des fleurs ne peut pas être changée. Jana a choisi la couleur pour chaque bourgeon et elle a cliqué sur «Fleurir!». Seules les fleurs dont Jana a deviné la juste couleur fleuriront, les autres ne fleuriront pas.



Ensuite, Jana change les couleurs de quelques fleurs. Le résultat est le suivant :



Choisis la couleur pour chacune des fleurs.



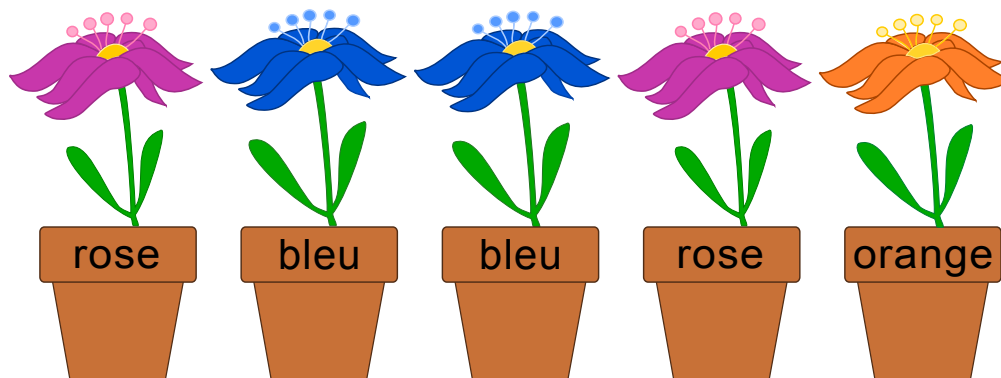
Solution

La réponse correcte est : «rose», «bleu», «bleu», «rose» et «orange».

Après avoir essayé de deviner deux fois la couleur juste, il n'y a eu que trois bourgeons à avoir fleuri. C'est la raison pour laquelle nous pouvons déjà déterminer les couleurs choisies par l'ordinateur pour le premier, le troisième et le cinquième bourgeon.

Lors du premier essai, Jana a choisi pour le deuxième bourgeons la couleur «rose» avec le résultat qu'il n'a pas fleuri. Lors du deuxième essai, elle a choisi «orange» et le bourgeon n'a toujours pas fleuri. Puisqu'il n'existe que trois couleurs, il s'ensuit que le deuxième bourgeon doit être «bleu».

Ensuite, Jana a choisi «orange» et «bleu» pour le bourgeon de la quatrième fleur et cette dernière n'a pas fleuri. Elle en déduit correctement que le quatrième bourgeon doit être «rose».



C'est de l'informatique !

Une des capacités les plus importantes dans la résolution des problèmes est de savoir tirer les bonnes conclusions des événements qui ont eu lieu ou qui, par contre, n'ont pas eu lieu. La tâche en question est une version simplifiée d'un jeu de logique très populaire. Le jeu a été simplifié dans le sens que le joueur obtient toutes les informations dont il a besoin pour trouver la solution après avoir essayé de deviner la bonne solution lui-même. Au troisième essai, au plus tard, le joueur connaîtra la couleur juste de la fleur particulière... à moins qu'il ait bien fait attention auparavant.

Sites web et mots clés

logique, jeu de stratégie, algorithme

— <https://fr.wikipedia.org/wiki/Mastermind>



2. Un anniversaire à deux valeurs

C'est l'anniversaire de Benno : il va avoir onze ans. Malheureusement, sa mère ne dispose que de cinq bougies pour le gâteau d'anniversaire. Pourtant, elle a déjà une petite idée comment représenter le nombre onze avec cinq bougies. Son plan est de planter une bougie à côté de l'autre tout en définissant les conditions suivantes :

- La bougie à l'extrême droite prend la valeur numérique 1.
- Toutes les autres bougies prennent la valeur numérique double de la bougie à leur droite.
- On n'additionne que les valeurs numériques des bougies allumées.

Voici quelques exemples :



Quelles bougies sa mère va-t-elle donc allumer pour représenter le nombre 11 ?





Solution

La réponse correcte est A) : A (01011) : la mère va allumer les bougies numéro 8, 2 et 1. Ainsi, la valeur numérique totale des bougies sera $0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 11$.



Voici pourquoi les réponses B), C), D) et E) sont incorrectes :

B) (01110), car si la mère allume les bougies qui correspondent aux valeurs numériques 8, 4 et 2, la valeur numérique totale sera 14 : $0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 14$.



C) (10000), car si elle n'allume que la bougie à l'extrême droite, la valeur numérique totale sera 16 : $1 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 = 16$.



D) (11010), car si elle allume les bougies qui correspondent aux valeurs numériques 16, 8, 2, le total sera 26 : $1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 26$.



E) (11111), car si elle allume toutes les bougies, la valeur numérique totale sera 31 : $1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 31$.



C'est de l'informatique !

Grâce au système de numération binaire, on peut créer un nombre indéterminé de nombres. À l'aide de la formule «bougie éteinte» ou «bougie allumée», on définit si la valeur numérique sera additionnée ou non. L'emplacement, lui, détermine le montant de la valeur numérique. Si on remplaçait les bougies par les caractères 0 (bougies éteinte) et 1 (bougie allumée), on arriverait aux mêmes résultats. En effet, pour représenter les données, presque tous les processeurs des ordinateurs actuels utilisent le système de numération binaire (également appelé système de numération utilisant la base 2). Ceci pour des raisons pratiques : il est plus facile de réaliser des circuits logiques pour le système de numération binaire que, par exemple, pour le système décimal.



Sites web et mots clés

Systeme de numération binaire, représentation de l'information en binaire, système de numération utilisant la base 2.

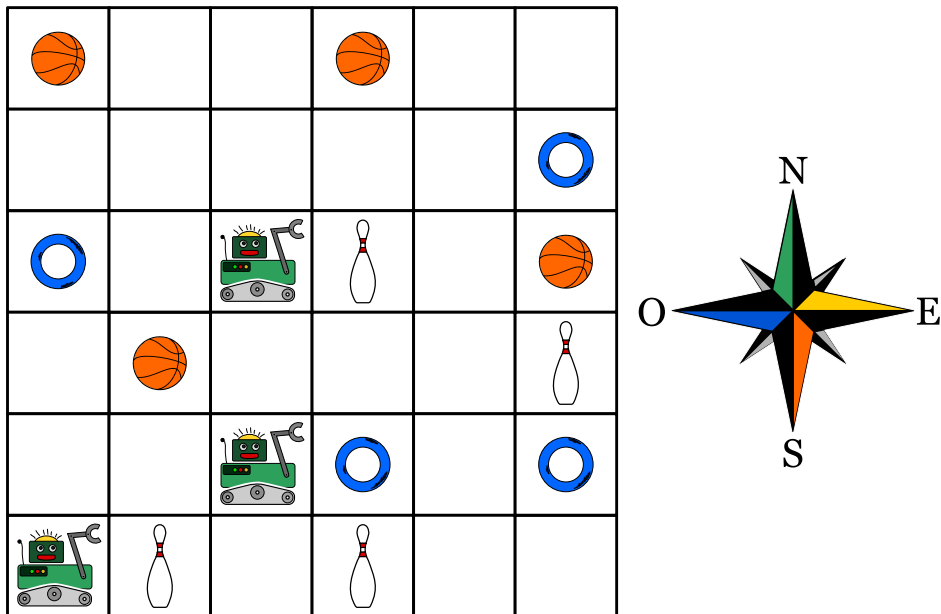
— https://fr.wikipedia.org/wiki/Systeme_binaire





3. Tous en même temps

Trois robots travaillent en équipe. Tu peux piloter cette équipe à l'aide des commandes de démarrage et de sens : vers le nord (N), le sud (S), l'est (E) ou l'ouest (O). En utilisant la commande de démarrage et de sens, les trois robots bougent non seulement en même temps mais aussi dans la même direction : ils avancent donc tous ensemble d'une case, et ceci dans la direction souhaitée. L'objectif est de piloter les trois robots dans la direction des objets qu'ils doivent finalement saisir. Pour qu'ils ne saisissent pas des objets non désirés, tu dois les piloter de sorte qu'ils les évitent. Exemple : Si tu pilotes les robots avec les commandes N, N, S, S, E, ils saisiront à la fin deux quilles et un anneau.



Les robots doivent saisir un ballon, un anneau et une quille.

Laquelle des quatre séries de commandes permettra aux trois robots de saisir en même temps les trois objets souhaités ?

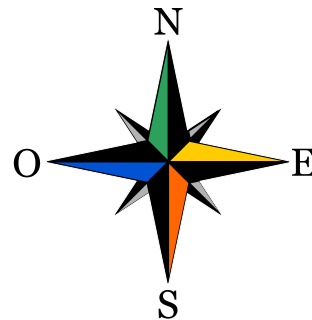
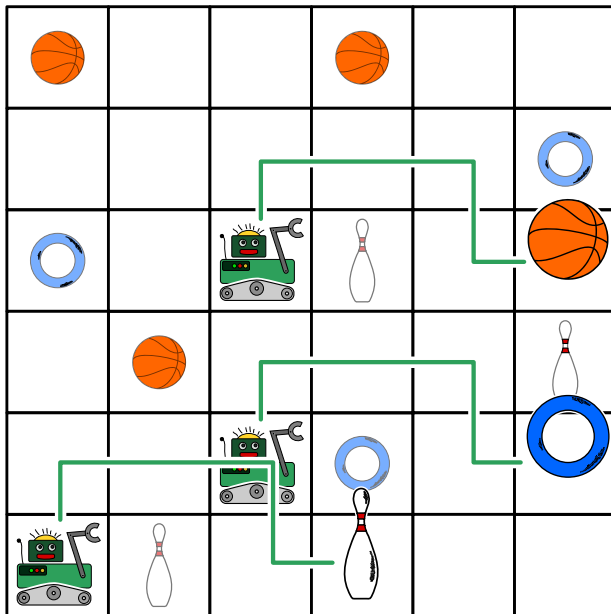
- A) N, E, E, E
- B) N, E, E, S, E
- C) N, N, S, E, N
- D) N, E, E, S, O



Solution

La réponse B) est correcte :

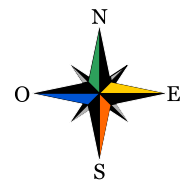
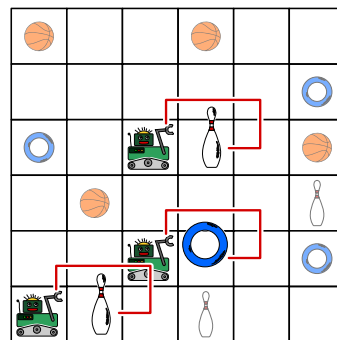
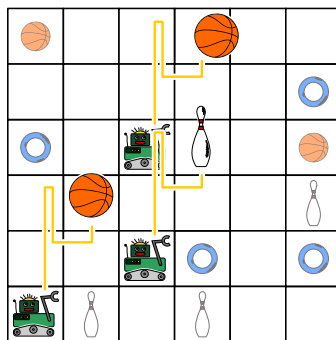
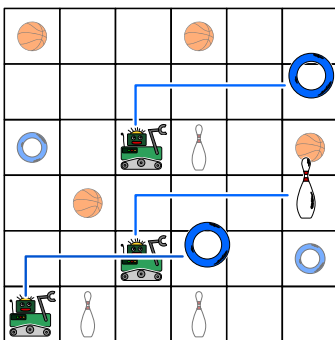
Avec les commandes N, E, E, S, E, tu pilotes les robots le long des lignes vertes. À la fin de leur parcours, les robots saisissent en effet un ballon, un anneau et une quille.



La réponse A) n'est pas correcte : avec les commandes N, E, E,E, tu pilotes les robots le long des lignes bleues. À la fin, ils saisissent deux anneaux et une quille.

La réponse C) n'est pas correcte : Avec les commandes N, N, S, E, N, tu pilotes les robots le long des lignes jaunes. À la fin, ils saisissent deux ballons et une quille.

La réponse D) n'est pas correcte : Avec les commandes N, E, E, S, O, tu pilotes les robots le long des lignes rouges. À la fin, ils saisissent deux quilles et un anneau.



C'est de l'informatique !

Dans la plupart des cas, on écrit des programmes informatiques de façon que les opérations seront exécutées successivement. De même, la plupart des langages de programmation ainsi que beaucoup de gens qui savent programmer fonctionnent de la même manière, c'est-à-dire ils exécutent une opération après l'autre.

Depuis plusieurs années déjà, le développement des processeurs touche à ses limites : il est de plus en plus difficile d'améliorer la performance des processeurs qui exécutent des programmes successivement. En même temps, on se rend compte qu'il est plus économique d'incorporer des plate-formes



de traitement multi-cœur dans un seul ordinateur. En effet, les ordinateurs actuels réunissent en règle générale 2, 4 ou plus de cœurs de processeur qui sont capables d'exécuter plusieurs opérations en même temps. Dans le cas des cartes graphiques, cette pratique est encore plus extrême : elles possèdent souvent de nombreux cœurs dont chacun d'entre eux ne satisferait jamais à la performance attendue, mais qui, dans l'ensemble, sont très performants. À part des calculs traitant la représentation des images, certains processeurs de cartes graphiques sont mêmes capables d'exécuter n'importe quelle opération demandée.

Ceci implique pourtant un changement dans l'approche de la programmation : il faut veiller à ce que les processeurs qui effectuent en parallèle différentes opérations ne se compromettent pas l'un ou l'autre, par exemple qu'ils n'attendent pas jusqu'à ce que l'autre processeur ait fini son opération, afin qu'ils terminent leur tâche ensemble tout en présentant un résultat. C'est exactement cet aspect-là qu'il faut prendre en considération dans la tâche présente.

Sites web et mots clés

programmation parallèle

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

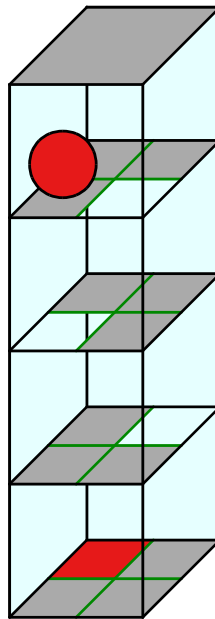
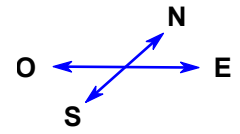




4. Le parcours de la boule

Un labyrinthe en trois dimensions se base sur quatre niveaux comptant chacun quatre champs. Une boule est placée sur le niveau le plus haut. Au niveau le plus bas se trouve la zone cible : le champ marqué en rouge.

Tu peux piloter la boule à l'aide des commandes de démarrage et de sens N (vers le nord), E (vers l'est), S (vers le sud) et O (vers l'ouest). Si la boule entre en contact avec un champ transparent, elle tombe sur le niveau inférieur. Le labyrinthe est un espace fermé ; tu ne peux donc pas piloter la boule à travers les murs.



Pilote la boule jusqu'à la zone cible !

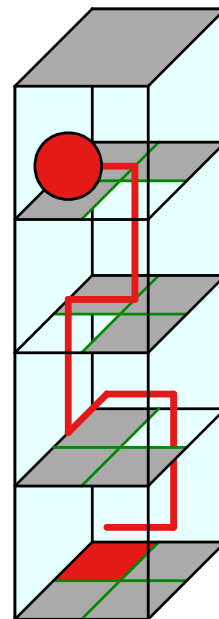


Solution

Avec la séquence de commandes comme «EONEO» (voir la ligne rouge dans la figure de droite) ou «EOENO» tu pilotes la boule jusqu'à la zone cible. Pourtant, il existe bien d'autres possibilités pour piloter la boule jusqu'à la zone cible et nul n'interdit de faire des détours ou des parcours plus longs.

C'est de l'informatique !

La séquence de commandes avec laquelle la boule est pilotée à travers le labyrinthe est en vérité un programme informatique court. Le langage y correspondant ne connaît que quatre commandes : N, O, S et E. En informatique, on parle également d'*instructions*. Un programme qui prend recours à ce langage se base sur une séquence d'instructions. Ces instructions sont exécutées de manière successive (en informatique on dit aussi de manière *séquentielle*). Les langages de programmation professionnelle prennent également recours aux séquences d'instructions. Ces dernières font partie des éléments fondamentaux pour créer un programme qui comporte de différentes instructions. Voici d'autres éléments fondamentaux pour structurer un programme informatique : la répétition (cycles), l'instruction conditionnelle (sélection) ainsi que l'appel à des parties du programme fréquemment utilisées à l'aide des sous-programmes. Et ce sera tout ; en effet, les programmes informatiques les plus complexes sont basés sur de telles structures apparemment très simples.



Sites web et mots clés

programme, séquence

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



5. Des fleurs et des soleils

Barbara a reçu deux tampons à encre. Avec l'un des deux, elle peut créer l'empreinte d'une fleur, avec l'autre, l'empreinte d'un soleil. Maintenant, elle réfléchit comment elle pourrait créer son nom à l'aide de ces deux motifs.

Pour représenter les différentes lettres, elle choisit des suites différentes de fleurs et de soleils.

lettre	B	A	R	E	Y
suite					

À l'aide de ses suites, elle peut représenter son propre nom, «Barbara», comme suit :



Puis, elle aimerait bien représenter le nom d'un de ses amis :



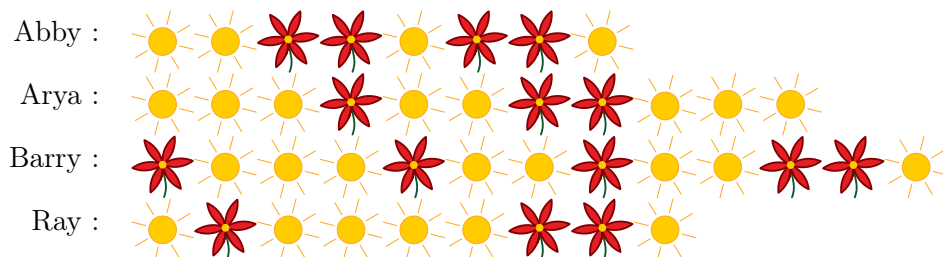
Lequel des noms ci-dessous est-ce que Barbara a représenté ?

- A) Abby
- B) Arya
- C) Barry
- D) Ray



Solution

La réponse correcte est Abby. Voici tous les codes des noms de ses amis :



C'est de l'informatique !

Il existe de différentes manières pour coder des données. Une pratique courante est par exemple d'enregistrer les caractères que l'on a tapés sur le clavier en UTF-8. L'UTF-8 est une variante de l'unicode. Pour enregistrer les caractères fréquents, on a besoin d'un espace d'un octet ce qui revient à plus de 250 caractères différents. Pour des caractères peu fréquents, il faut un espace de quatre octets ; ainsi, on peut réaliser plusieurs millions de caractères différents et ceci même dans de différentes langues.

Ce système fonctionne déjà assez bien, pourtant, il s'avère que parmi les caractères fréquents, il y en a ceux qui sont utilisés de manière plus fréquente que d'autres : le «E» ou le «N» sont par exemple plus souvent utilisés que le «X» ou le «Ö». Dans ce cas, on a recours à des codes utiles qui fonctionnent avec une longueur de symboles entièrement variable.

Dans le cas des codes à longueur variable, il est judicieux d'utiliser des débuts de codes distinctifs pour chaque symbole de la source afin de pouvoir la décoder de manière rapide et sûre. On appelle de tels codes des codes préfixe. Un des codes préfixe les plus connus est le code Morse.

Pour obtenir un code dont les données sont comprimées au maximum, il faut que l'on recherche d'abord le nombre d'occurrences de chaque caractère. Ensuite, à l'aide du codage dit codage de Huffman, on arrive à calculer un code dont les données sont comprimées au maximum. Chaque code de Huffman est également un code préfixe.

Sites web et mots clés

code préfixe, code de Huffman, compression de données

- https://fr.wikipedia.org/wiki/Code_préfixe
- https://fr.wikipedia.org/wiki/Codage_de_Huffman




6. Le tchat des castors

Le tchat nommé «Tchat des castors» peut être utilisé gratuitement et sa diffusion est financée par la publicité. L'agence de voyage «Sunshine Travel» profite de cette plate-forme et elle y est présente avec de différentes images publicitaires afin d'atteindre de différents groupes cibles. Pour choisir les images, elle profite des résultats de l'analyse des messages du tchat. Au centre de ses intérêts se trouvent des mots particuliers auxquels le système attribue automatiquement des points positifs ou négatifs :

- Les formules de salutation comme «Chère(s)» et «Cher(s)» sont particulièrement utilisés par des «castors» âgés. Le système les pénalise de deux points négatifs.
- Les formules de salutation comme «Hi», «Hey» ou «Arriba» sont très populaires parmi les jeunes castors. Le système attribue à ces messages-là deux points positifs.
- Aux messages comprenant les abréviations telles que «bf» (meilleur(e)s ami(e)s), «gr8» (merveilleux, fantastique), «np» (aucun problème) ou «thx» (merci), le système attribue un point positif.
- Chaque message avec des mots composés de 10 lettres ou plus est sanctionné par un point négatif.

Ainsi, chaque castor sera assigné à un groupe cible selon le total des points qui ont été attribués à ses messages.

nombre de points	groupe cible	image affichée
inférieur à 0	des personnes âgées	
supérieur à 0	des jeunes	
0	aucun des deux groupes cible	

Lisez les messages suivants. Quelles images l'agence de voyages choisira-t-elle pour chacun des messages ?

- A) Chers amis, l'été approche et je cherche un logement sympa tout près du Rhin. Merci de vos suggestions, Richie.
- B) Arriba! Y a-t-il quelqu'un ?
- C) @Mia : <3 <3 <3
- D) d'acc. gr8. Thx



Solution

Message A) : Sunshine Travel affiche l'image de la plage. Le message comporte la formule de salutation «Chers» et le mot «suggestions» qui, lui, est composé de plus de 10 lettres. Le système attribue donc à ce message un point négatif.

Message B) : Sunshine Travel affiche l'image avec les planches de surf. Le message comporte la formule de salutation très populaire auprès des jeunes castors.

Message C) : Sunshine Travel affiche l'image de la tour Eiffel, le synonyme pour des voyages à la découverte d'une ville. Le système attribue à ce message zéro points car aucune des règles préalablement établies ne peut y être appliquée.

Message D) : Sunshine Travel affiche l'image avec les planches de surf. Le message comporte les deux abréviations «gr8» et «thx» et le système lui attribue un point positif.

C'est de l'informatique !

La valorisation des textes selon des règles préalablement établies peut être facilement exécutée à l'aide de programmes informatiques. La recherche d'éléments particuliers dans un texte est un exemple simple de recherche basée sur le modèle appelé «pattern matching» ou filtrage par motif. Ce procédé de filtrage trouve son application aussi bien dans le traitement de texte que dans d'autres domaines comme par exemple dans le traitement des images ou encore dans d'autres programmes informatiques.

De nombreuses entreprises ont recours à l'évaluation automatisée des profils des utilisateurs sur Internet afin de créer des offres adaptées au type de client et à ses besoins. En tant qu'internaute, il s'avère donc judicieux de prendre conscience des procédés décrits ci-dessus et de gérer ses données personnelles de manière responsable. Du coup, les informaticiens se retrouvent face à un dilemme : d'une part, ils doivent aider à aiguïser la conscience du public envers de telles démarches, d'autre part, ce sont justement ces démarches qui créent de nouveaux emplois dans le secteur de l'informatique. Pour les informaticiens, les fonctions de valorisation des textes peuvent être également très utiles dans d'autres domaines, comme par exemple pour afficher les résultats de recherche selon leur pertinence.

Sites web et mots clés

profilage des utilisateurs, publicité appropriée aux groupes cible

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







7. Quatre courses à faire

Alexandra aimerait bien faire les courses pendant la pause de midi (de 12h 00 jusqu'à 13h 00) :

- aller acheter un livre à la librairie ;
- aller acheter une bouteille de lait à l'épicerie ;
- envoyer le livre qu'elle vient d'acheter à la librairie par poste ;
- aller boire un café au restaurant.

Alexandra a calculé le temps dont elle a besoin pour effectuer chaque achat. Les durées listées ci-dessous ne comptent que pour des situations en dehors des heures de pointe. C'est la raison pour laquelle Alexandra essaie de les éviter à tout prix.

	Lieu	Durée	Heures de pointe
	Librairie	15 minutes	12h 40 – 13h 00
	Épicerie	10 minutes	12h 00 – 12h 40
	Poste	15 minutes	12h 00 – 12h 30
	Restaurant	20 minutes	12h 30 – 12h 50

Déplace les achats divers en une séquence correcte afin qu'Alexandra puisse éviter les heures de pointe.



Solution

La séquence correcte est : restaurant, librairie, poste, épicerie.

Ce problème a quelques limitations que l'on peut rendre évident à l'aide du tableau suivant : (en rouge foncé : les heures de pointe, en vert clair : les heures normales) :

Lieu	Durée	12.00-12.05	12.05-12.10	12.10-12.15	12.15-12.20	12.20-12.25	12.25-12.30	12.30-12.35	12.35-12.40	12.40-12.45	12.45-12.50	12.50-12.55	12.55-13.00
Librairie	15 minutes					X	X	X					
Épicerie	10 minutes											X	X
Poste	15 minutes								X	X	X		
Restaurant	20 minutes	X	X	X	X								

Alexandra doit avoir quitté la librairie avant 12h 40 et elle ne peut aller à l'épicerie qu'après 12h 40. Aller à la poste n'est possible qu'après avoir été à la librairie. Mais elle ne peut aller à la poste qu'après 12h 30 et elle doit avoir bu un café avant 12h 30 parce que sinon, à 12h 50, elle n'aura pas assez de temps pour sa pause de midi.

Voilà le seul plan possible pour qu'elle puisse faire tous les achats tout en évitant les heures de pointe (dans le tableau marqué par X) :

- aller boire un café au restaurant entre 12h 00 et 12h 20 ;
- aller acheter un livre à la librairie entre 12h 20 et 12h 35 ;
- aller à la poste pour envoyer le livre qu'elle vient d'acheter à la librairie entre 12h 35 et 12h 50 ;
- aller à l'épicerie entre 12h 50 et 13h 00.

C'est de l'informatique !

Quand on doit résoudre un problème, l'informatique poursuit un objectif important qui est de trouver des solutions qui répondent le mieux aux limitations (conditions) existantes. Une des conditions de notre problème est qu'Alexandra doit éviter les heures de pointe. Souvent, dans d'autres situations où il faut résoudre un problème, la question est en effet de savoir s'il existe réellement une solution, plus précisément si on peut trouver une solution qui réponde à toutes les conditions préalablement définies.

En informatique, ce type de questionnement s'appelle «scheduling» (planification). «Scheduling» signifie «planification de tâches informatiques», c'est-à-dire qu'on cherche à trouver la séquence la plus correcte et la plus favorable pour exécuter des opérations d'une tâche particulière. Le scheduling trouve son application, par exemple, dans le domaine de l'industrie, lors des projets vastes ou bien dans la production des pièces industrielles. On a également recours au scheduling pour les ordinateurs, par exemple, quand différents processeurs (ou CPU) dotés de plusieurs cœurs doivent exécuter différentes opérations en même temps.

Sites web et mots clés

planification de tâches informatiques, optimisation

- https://fr.wikipedia.org/wiki/Ordonnancement_de_tâches_informatiques



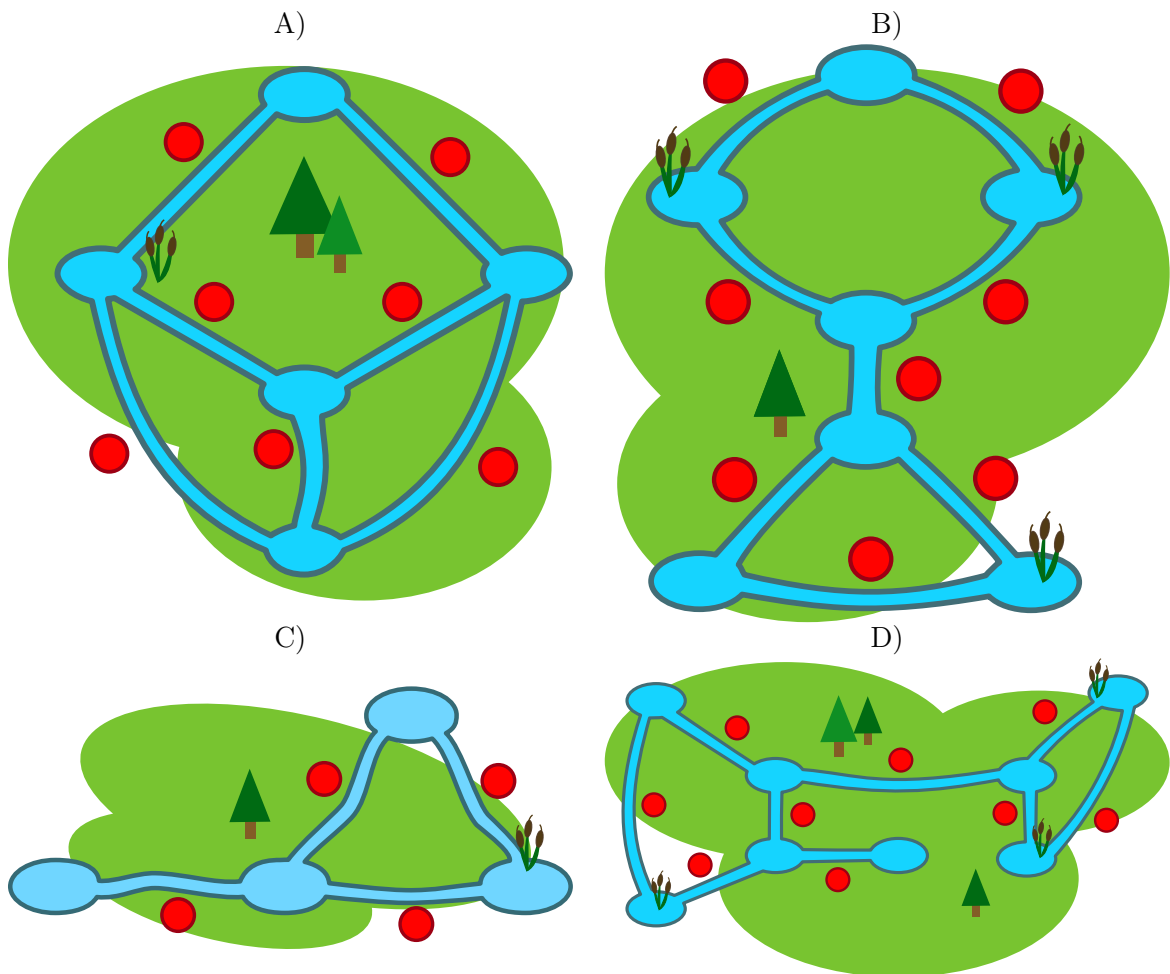
8. Des impasses

Les castors adorent construire des barrages. Mais quand ils nagent dans leurs canaux et qu'ils arrivent à un de ces barrages, ils sont obligés de faire un détour en sortant de l'eau. Voilà quelque chose qu'ils n'aiment pas du tout ! Au lieu de sortir de l'eau, ils préfèrent rebrousser chemin et choisir un autre canal afin d'atteindre leur destination finale.

Les castors appellent «impasses» les canaux qu'ils doivent absolument parcourir parce qu'ils n'ont pas d'autre choix pour arriver à leur destination finale. Ils décident donc de ne pas construire de barrages dans les impasses.

Sur les images ci-dessous, on aperçoit quelques territoires de castors. Les repères rouges marquent l'endroit approximatif des lieux de construction des barrages dans le canal.

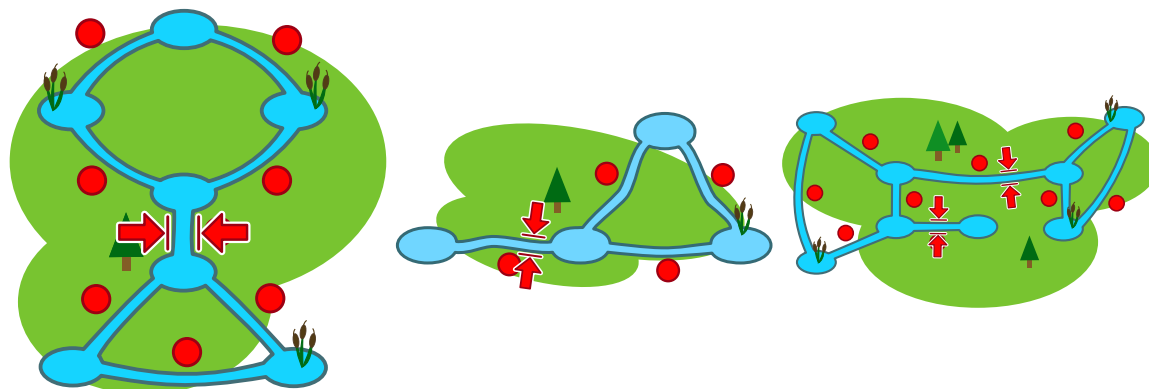
Il n'y a qu'un seul territoire qui n'a pas d'impasse. Il s'agit duquel ?





Solution

La réponse correcte est A). Dans ce territoire, il existe pour chaque canal un chemin alternatif. Dans le territoire B), l'impasse se trouve au centre. Dans le territoire C), l'impasse se trouve à l'extrême gauche et dans le territoire D) il y a deux impasses qui se trouvent au centre.



C'est de l'informatique !

Le territoire dans lequel vivent les castors est un réseau comprenant des canaux et des étangs. C'est tout à fait comparable à l'Internet : ainsi, les étangs représenteraient les ordinateurs, les portables, les téléviseurs etc. et les canaux représenteraient les lignes de transmissions ou les connexions sans fil. À l'origine, l'Internet a été inventé pour connecter les universités des États-Unis entre elles. Ses fondateurs déjà, veillaient à éviter les impasses car ils savaient qu'au cas où une connexion poserait des problèmes, il n'existerait pas d'alternative.

Pour observer des problèmes de réseau, l'informatique se sert de la théorie des graphes. Les graphes sont définis comme des systèmes de nœuds (étangs) et des arcs (canaux). À l'aide des graphes, il est possible de modéliser de différents types de réseaux comme par exemple un réseau de transport ou de communication. Pour analyser des problèmes de réseau, on a développé beaucoup d'algorithmes. Un problème bien connu est celui de trouver des «ponts» à l'intérieur des graphes, c'est-à-dire de trouver les impasses indésirables à l'intérieur du réseau – tout comme il fallait le faire dans la tâche présente.

Sites web et mots clés

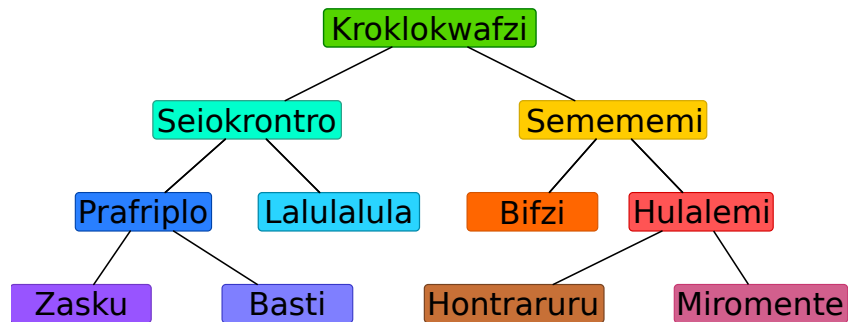
graphes, ponts, impasses

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



9. Vive la hiérarchie !

L'image ci-dessous représente les relations entre les espèces vivant sur la planète Morgenstern. On interprète la ligne qui relie deux espèces de telle manière que tous les animaux de l'espèce inférieure font également partie de l'espèce notée au-dessus.



Exemple : tous les «Hulalemi» font partie de l'espèce des «Semememi». Certains «Seiokron», par contre, ne font pas partie de l'espèce «Basti».

Des affirmations ci-dessous, il n'y en a qu'une qui est juste. Laquelle ?

- A) Tous les Basti font partie de l'espèce Seiokron.
- B) Certains Hontraruru ne font pas partie des Semememi.
- C) Tous les Zasku font également partie des Bifzi.
- D) Tous les Prafrplo font également partie des Basti.



Solution

La réponse A) est correcte.

A) : Tous les animaux de l'espèce Basti sont également des animaux de l'espèce Prafriple. Comme tous les animaux des Prafriple font partie de l'espèce Seiokrontro, tous les animaux des Basti font également partie de l'espèce Seiokrontro.

B) : Les Hontraruru sont des animaux de l'espèce Hulalemi. Ces-derniers sont des animaux de l'espèce Semememi. C'est la raison pour laquelle tous les animaux de l'espèce Hontraruru font partie de l'espèce Semememi (et non de l'espèce Seiokrontro).

C) : Les Zasku sont des animaux de l'espèce Prafriple. Ils ne font pas partie de l'espèce Bifzi.

D) : Les animaux de l'espèce Basti font partie de l'espèce Prafriple, et non l'inverse.

C'est de l'informatique !

Les relations entre les espèces sont représentées selon un principe que l'on appelle en informatique «arbre binaire». Les biologistes, eux, utilisent l'arbre «phylogénétique» pour illustrer les relations entre les espèces diverses.

En informatique, on a souvent recours à de tels arbres pour des représentations graphiques des relations. Un arbre généalogique représente, par exemple, la relation entre les enfants, les parents et les grands-parents. Si on utilise des arbres pour la représentation des relations, on peut rapidement comprendre quelles relations existent entre les différents éléments de l'arbre appelés des «nœuds».

Les arbres sont en outre un excellent moyen pour enregistrer des données de manière ordonnée et donc pour les retrouver le plus rapidement possible... ainsi, il ne faudra que peu d'étapes pour pouvoir accéder à un énorme volume de données.

Sites web et mots clés

arbres, spécialisation, généralisation

- [https://fr.wikipedia.org/wiki/Arbre_\(graphe\)](https://fr.wikipedia.org/wiki/Arbre_(graphe))
- https://fr.wikipedia.org/wiki/Arbre_phylogénétique



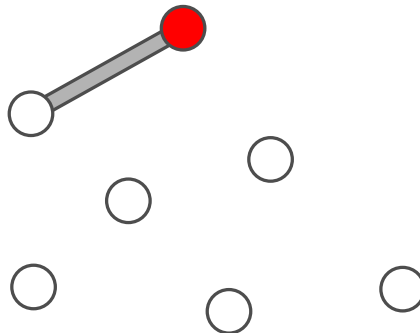
10. Plongeon-nous dans la construction des ponts !

Le grand-papa des castors est devenu un peu hydrophobe. Afin de pouvoir éviter l'eau le plus souvent possible, il imagine de relier sa hutte de castor avec toutes les autres huttes de sa famille par des ponts. Les autres castors apprécient son idée et ils décident de l'aider à condition que l'on tienne compte des points suivants :

- Pour atteindre chacune des autres huttes, grand-papa ne doit traverser que deux ponts au maximum.
- À part le pont que l'on construit pour atteindre une hutte, on ne construira que deux autres ponts au maximum pour atteindre une des autres huttes.

Avant de commencer la construction, les castors dessinent un plan. Ils décident de représenter les huttes sur le plan par des cercles. La hutte de grand-papa est représentée par un cercle rempli de couleur rouge. Quand les castors commencent, ils arrivent à dessiner un premier pont qui prend son départ de la hutte de grand-papa. Malheureusement, à partir de ce moment-là, ils ne savent plus comment continuer.

Complète le plan de sorte que toutes les conditions requises seront réunies. Il existe plusieurs possibilités pour accomplir cette tâche. Ce qui est clair, c'est que l'on a besoin de cinq autres ponts pour relier toutes les huttes.

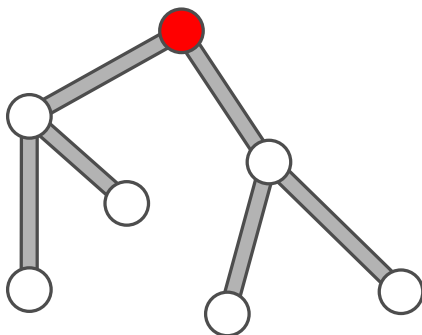




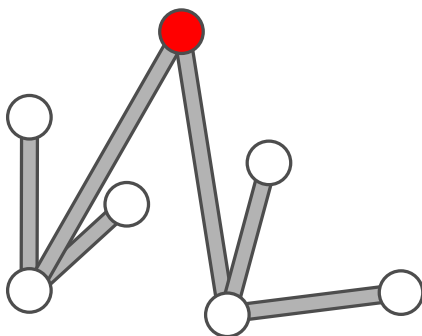
Solution

Le plan ci-dessous réunit toutes les conditions requises car :

- partant du cercle rouge, tous les autres cercles peuvent être atteints et ceci, en parcourant au maximum deux liens ;
- de chaque cercle il n'y a que trois liens qui sortent.



Bien entendu, il existe encore d'autres plans qui répondent aux conditions requises, comme par exemple le plan ci-dessous :



C'est de l'informatique !

Bien qu'il existe beaucoup de solutions possibles, chacune d'entre elles présente la même structure : de la hutte du grand-père sortent deux ponts qui la relient avec deux autres huttes. De ces dernières, deux autres ponts par hutte sortent pour en relier les quatre huttes restantes. S'il y avait une hutte de plus, il ne serait pas possible de l'atteindre tout en répondant aux conditions requises.

Avec leur plan, les castors créent une structure de données sous la forme d'une hiérarchie appelée *arbre* : tous les *nœuds* (les cercles/les huttes) peuvent être atteints en parcourant les *arcs* (les liens/les ponts). Les deux conditions préalablement définies garantissent que les castors créent un arbre particulier : étant donné qu'il n'y a que deux arcs qui doivent sortir d'un seul nœud, les castors génèrent avec leur plan un *arbre binaire*. Le fait que l'on ne doit parcourir que deux arcs au maximum pour arriver aux prochains nœuds garantit que l'arbre est *minimal*.

Sites web et mots clés

arbre, huttes

- [https://fr.wikipedia.org/wiki/Arbre_\(graphe\)](https://fr.wikipedia.org/wiki/Arbre_(graphe))
- https://fr.wikipedia.org/wiki/Arbre_binaire

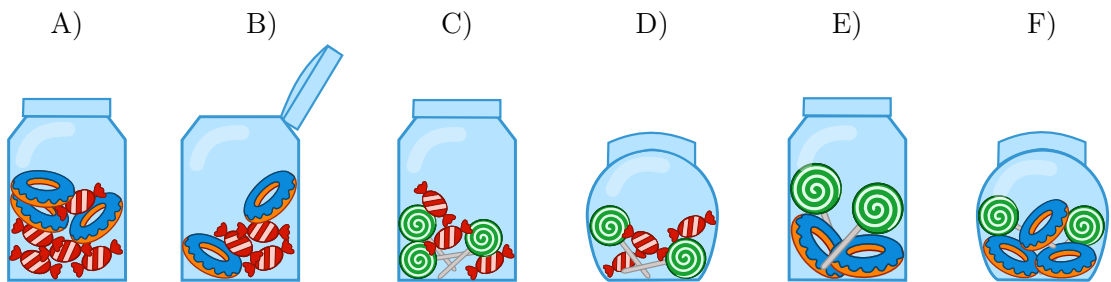


11. Des bonbonnières

Une bonbonnière est un récipient en verre utilisé pour conserver des friandises. Carl et Judy possèdent trois bonbonnières. Les bonbonnières A), B) et C) appartiennent à Carl tandis que les bonbonnières D), E) et F) sont à Judy. Chaque bonbonnière présente les cinq caractéristiques suivantes :

- La bonbonnière est soit ouverte soit fermée.
- La bonbonnière contient des bonbons à rayures blancs et rouges ou pas.
- La bonbonnière contient des bonbons-cercles bleus au sucre ou pas.
- La bonbonnière contient des sucettes en spirales et de couleur verte ou pas.
- La bonbonnière est soit ronde soit carrée.

Choisis la bonbonnière qui réponde aussi bien aux caractéristiques communes des bonbonnières de Carl qu'aux caractéristiques communes des bonbonnières de Judy.





Solution

Les bonbons-cercles bleus au sucre ne doivent pas être pris en considération parce qu'ils ne constituent pas une caractéristique commune des bonbonnières de Carl et de Judy : ils n'existent simplement pas dans les bonbonnières C) et D).

Les deux caractéristiques communes des bonbonnières A), B) et C) de Carl sont les suivantes :

- chaque bonbonnière est carrée ;
- chaque bonbonnière contient des bonbons à rayures blanches et rouges.

Les deux caractéristiques communes des bonbonnières D), E) et F) de Judy sont les suivantes :

- chaque bonbonnière est fermée ;
- chaque bonbonnière contient des sucettes en spirales et de couleur verte.

Seule la bonbonnière C) réunit toutes les quatre caractéristiques : elle est carrée, elle contient des bonbons à rayures blanches et rouges ainsi que des sucettes en spirales et de couleur verte et elle est fermée.

C'est de l'informatique !

Dans leurs modèles de données, les informaticiens préfèrent regrouper les objets selon leurs caractéristiques (le terme le plus approprié est : «propriété» ou «attribut»). La présente tâche comprend cinq propriétés et deux groupes. L'objectif est de chercher des objets qui réunissent toutes les propriétés communes aux deux groupes. En relation avec des bases de données relationnelles, une telle opération est aussi appelée : «calculer l'intersection de deux ensembles».

Dans beaucoup de bases de données relationnelles, on peut identifier («filtrer») des objets à l'aide de propriétés déterminées et ceci, même parmi des quantités énormes de données («construire un sous-ensemble»). De cette manière, il est possible d'entamer une recherche ciblée sur des boutiques en ligne, par exemple. Les propriétés d'un smartphone telles que la durée de vie de la batterie, la taille d'écran et d'autres encore sont enregistrées dans une base de données à laquelle le moteur de recherche peut accéder pour permettre une recherche rapide.

Lors de la création d'une base de données, il est très important d'être particulièrement attentif aux propriétés utilisées dans le modèle de données. Si, par exemple, on oublie quelques propriétés importantes, toute recherche ultérieure sera moins précise. Si, par contre, on introduit des propriétés redondantes, l'entrée des données ultérieures sera coûteuse sans que la valeur de la base de données augmente.

Sites web et mots clés

orientation objet, attribut, logique

- [https://fr.wikipedia.org/wiki/Intersection_\(mathématiques\)](https://fr.wikipedia.org/wiki/Intersection_(mathématiques))
- [https://fr.wikipedia.org/wiki/Union_\(mathématiques\)](https://fr.wikipedia.org/wiki/Union_(mathématiques))
- https://fr.wikipedia.org/wiki/Théorie_des_ensembles
- https://fr.wikipedia.org/wiki/Base_de_données



12. Le code KIX

Aux Pays-Bas, les codes postaux se composent de quatre caractères. Ces-derniers peuvent être des lettres ou des chiffres. Il y a même un code-barres propre aux codes postaux appelé code KIX. Chaque symbole du code KIX est subdivisé en deux parties : une partie supérieure (deux barres longues et deux barres courtes) et une partie inférieure (également deux barres longues et deux barres courtes). Dans la zone centrale du symbole, les deux barres courtes se superposent. Le tableau montre le code KIX composé pour les chiffres 0, 7, G, Y :

	0	1	2	3	4	5
	6	7	8	9	A	B
	C	D	E	F	G	H
	I	J	K	L	M	N
	O	P	Q	R	S	T
	U	V	W	X	Y	Z

Le code KIX du code postal G7Y0 est donc :

À quel code postal le code KIX correspond-il ?



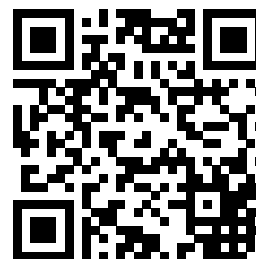
Solution

La réponse correcte est BC16 :

	0	1	2	3	4	5
	6	7	8	9	A	B
	C	D	E	F	G	H
	I	J	K	L	M	N
	O	P	Q	R	S	T
	U	V	W	X	Y	Z

C'est de l'informatique !

Aux Pays-Bas, on applique en effet le code KIX pour les codes postaux. Des codes sous forme lisible par machine permettent, par exemple, de trier des lettres et des colis de manière automatique. De tels codes sont en outre souvent utilisés dans la vie quotidienne, par exemple, sous forme de codes à barres («bar» en anglais comme «barre» en français) qui sont décodés par des lecteurs de code-barres des caisses des supermarchés ou bien sous forme de code QR (pour «quick response») qui a été inventé de l'industrie automobile pour étiqueter des pièces automobiles. Entre-temps, ces types de codes, notamment le code QR, trouvent leur application surtout dans la publicité et il existe beaucoup d'applications pour le smartphone qui permettent leur décodage rapide. Dépêche-toi, scanne ce code QR pour décodé son contenu!



Sites web et mots clés

code KIX, code à barres, code QR

- <https://nl.wikipedia.org/wiki/KIX-code>
- <https://fr.wikipedia.org/wiki/Code-barres>
- <https://fr.wikipedia.org/wiki/RM4SCC>



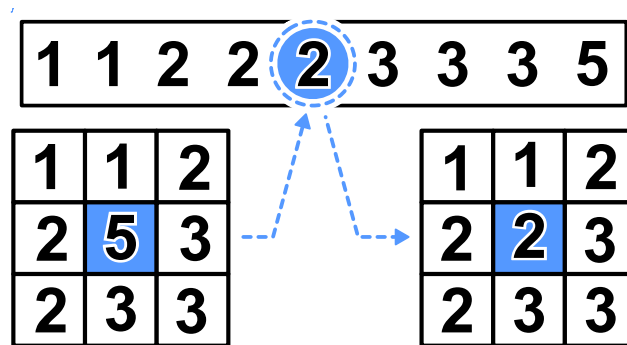
13. Le filtre médian

Une image est sauvegardée comme un tableau indiquant les valeurs de luminosité (entre 1 et 5) pour chaque pixel. La valeur 1 correspond à la couleur noire, la valeur 5 à la couleur blanche. Les valeurs 2 à 4 correspondent aux tons du gris de plus en plus clair.

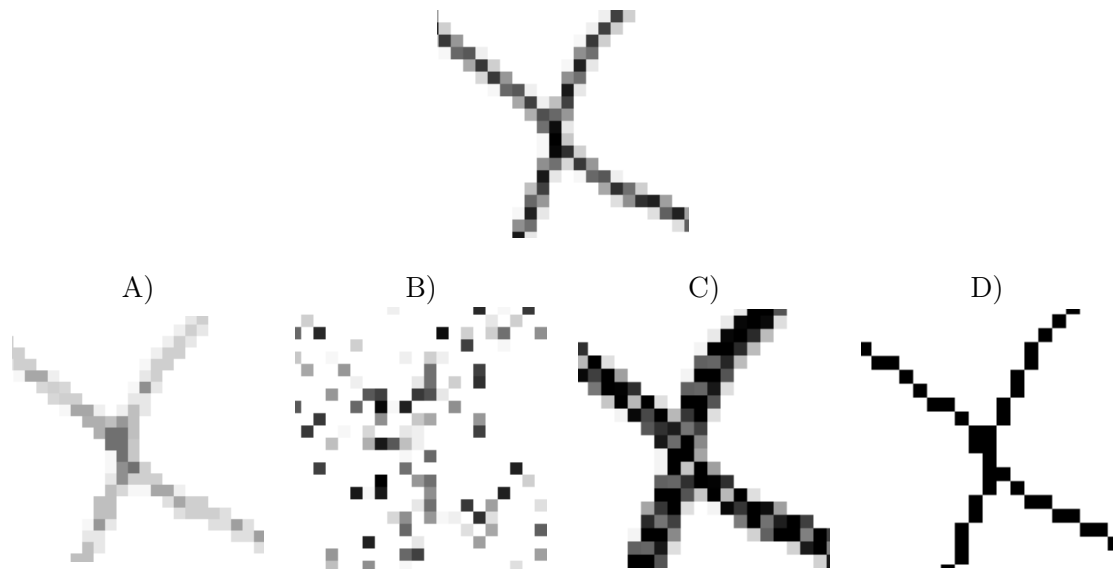
Un filtre appelé «filtre médian» influence le ton du gris de chaque pixel de l'image de sorte que...

- ... la valeur du pixel ainsi que les valeurs avoisinantes seront notées selon une séquence ordonnée croissante...
- ... le pixel assumera la cinquième valeur, soit la valeur médiane de la séquence, en tant que nouveau ton gris.

L'exemple ci-dessous montre que le pixel qui se trouve au centre passe de la valeur 5 à la valeur 2 :



Quel aspect cette image aura-t-elle au moment où on aura appliqué le filtre médian ?





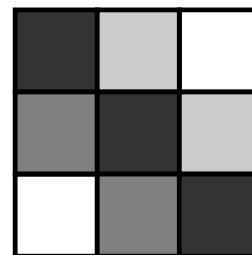
Solution

La réponse A) est correcte :

À travers le filtre médian choisi, des fragments d'image carrés sont traités avec neuf pixels. Le pixel noir au centre obtient une nouvelle valeur. Quand on regarde l'exemple à droite de plus près, on se rend compte qu'il n'y a que trois pixels qui sont noirs. Le médian de la séquence des tons gris filtrés est donc plus clair que l'original non filtré. Il s'ensuit que le noir changera en un ton gris.

Sur l'image originale de notre tâche, on voit que dans tous les fragments carrés à 9 pixels, les pixels noirs sont déjà minoritaires. C'est la raison pour laquelle l'image où on a appliqué le filtre médian ne comporte plus de pixels noirs.

Ceci n'est le cas que dans l'image A).



C'est de l'informatique !

Dans le traitement des photos, on aimerait bien réaliser rapidement certains effets visuels comme par exemple une plus grande netteté ou des couleurs plus lumineuses. Parfois, on aimerait bien obtenir des effets artistiques afin de personnaliser une photo. Ce sont justement de tels effets que l'on peut réaliser à l'aide des filtres d'image.

Le filtre médian fait partie de tels filtres. On applique le filtre médian pour corriger, par exemple, des erreurs de pixel qui se sont produites à cause d'un défaut du capteur d'images. Il en résultera donc que l'image semblera comme lissée et que chaque pixel sera uniformisé. On arrive ainsi aussi à réduire ou à atténuer certaines formes de bruits.

Sites web et mots clés

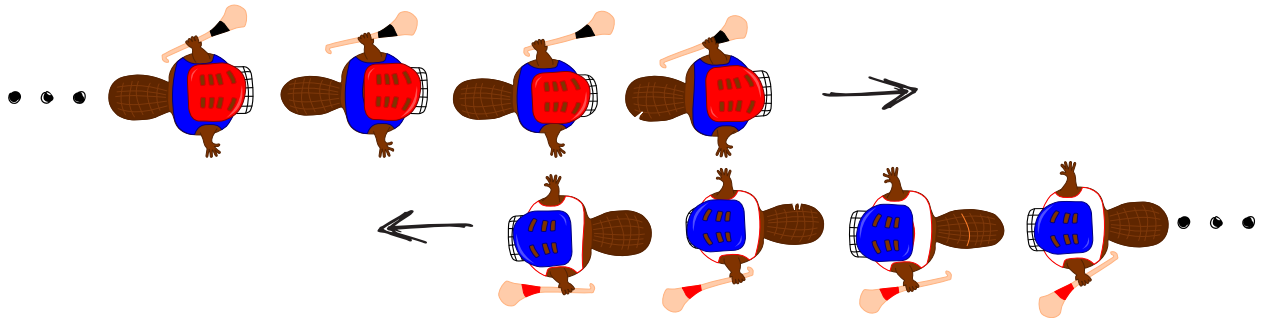
traitement des images, filtre médian, nuances de gris

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



14. La poignée de main style zip

Les castors adorent jouer au jeu irlandais appelé «Hurling». À la fin d'une partie, les joueurs des deux équipes se mettent en rang, l'un derrière l'autre. Ensuite, les deux équipes se croisent, toujours en rang, et les joueurs se serrent mutuellement la main pour remercier chaque joueur du jeu.



Plus exactement, ce rituel se déroule comme suit : D'abord, les deux premiers joueurs des deux équipes se serrent la main. Ensuite, les deux premiers joueurs serrent la main du deuxième joueur de l'équipe adverse (voir image ci-dessus). Et ainsi de suite, jusqu'à ce que les deux derniers joueurs des deux équipes se soient serré la main.

Le «Hurling» exige 15 joueurs par équipe. Le temps dont chaque joueur a besoin pour serrer la main du joueur de l'équipe adverse et pour avancer vers le prochain joueur est à une seconde.

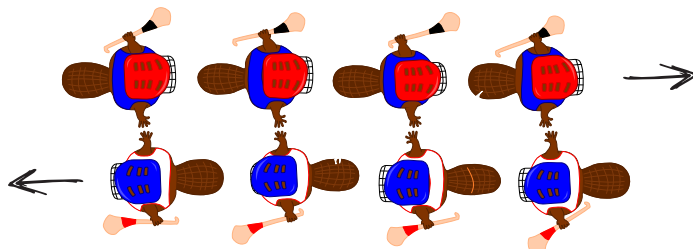
Combien de secondes faut-il au total pour que les deux équipes aient terminé le rituel de la poignée de main style zip ?



Solution

La réponse correcte est 29.

Le rituel de la poignée de main se déroule comme suit : le premier joueur de chaque équipe serre la main de chaque joueur de l'équipe adverse. Après quatre joueurs, on sera à quatre secondes et la situation se présentera ainsi :



En même temps, le dernier joueur de chaque équipe serre d'abord la main à un joueur de l'équipe adverse. Ensuite, il doit serrer la main à tous les autres joueurs. Si on suppose quatre joueurs, ceci prendra trois secondes de plus.

Il est donc possible de généraliser et d'affirmer que si on suppose n joueurs, la durée peut être calculée comme suit : d'abord on a besoin de n secondes, ensuite encore une fois de $n - 1$ secondes. Finalement, on arrive à l'équation $n + n - 1 = 2n - 1$ secondes. Dans le cas des 15 joueurs, le rituel de la poignée de main style zip durera donc $15 + 15 - 1 = 29$ secondes.

C'est de l'informatique !

Pour les équipes du jeu Hurling avec 15 joueurs, nous avons donc été capables de calculer exactement la durée de la poignée de main. On imagine que les spectateurs arrivent à bien supporter d'attendre ces 29 secondes jusqu'à ce que ce rituel soit terminé. Cependant, prenons une équipe de hockey qui présente 22 joueurs ? Peut-on toujours recourir à l'algorithme appliqué aux équipes Hurling ou est-ce que cette procédure durera trop longtemps ? Ce serait en effet très utile de disposer d'un algorithme capable d'évaluer le temps d'exécution sans devoir tout calculer à nouveau en détail.

L'informatique se consacre à une recherche intensive des algorithmes capables d'analyser le temps d'exécution. Ces analyses fournissent une expression mathématique comprenant une variable n qui représente la taille des données saisies. Dans le cas de la poignée de main du jeu Hurling, nous obtenons une telle expression si nous remplaçons la deuxième phrase de la réponse : on remplace par n la partie de la phrase «le nombre des joueurs d'une équipe» (15) par $2n - 1$. Il est ainsi possible de calculer exactement le temps d'exécution aussi pour d'autres nombres de joueurs : pour 22 joueurs, par exemple, le temps d'exécution sera à 43 secondes ($2n - 1 = 2 \cdot 22 - 1 = 43$ secondes), pour 40 joueurs, le temps d'exécution sera à 79 secondes et ainsi de suite.

L'expression du temps d'exécution $2n - 1$ est en fait une fonction linéaire. L'algorithme de «la poignée de main» appartient donc à la classe des algorithmes à fonction linéaire appelée aussi $O(n)$. Imaginons encore, si on serrait la main d'une autre manière. Quelles en seraient les conséquences, par exemple, si chacun serrait la main de l'autre individuellement (et non en même temps) ? Dans ce cas-là, l'algorithme appartiendrait à la classe $O(n^2)$ et les deux équipes du jeu Hurling se serreraient la main mutuellement pendant $15^2 = 225$ secondes, soit environ quatre minutes. Si l'algorithme avait un temps d'exécution exponentiel, il se trouverait dans la classe $O(2^n)$ et les pauvres joueurs devraient se serrer la main mutuellement pendant environ $2^{15} = 32768$ secondes, ce qui reviendrait à un temps d'exécution de 9 heures ! Les spectateurs, eux, ils auraient sans aucun doute quitté les rangs depuis longtemps pour aller se coucher... Il s'avère donc judicieux de réfléchir au fait de pouvoir exécuter quelques opérations en même temps pour épargner un peu de temps.



Sites web et mots clés

complexité du temps d'exécution, analyse du temps d'exécution

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





15. Cassy, la tortue

La tortue appelée Cassy habite un pays dont le sol est quadrillé. Le champ sur lequel elle bouge est assez restreint : elle ne vit que sur une parcelle de cinq fois cinq cellules de grille. Elle adore plus que tout manger de la salade fraîche. Quelle chance : tous les matins, les salades poussent de nouveau sur le champ. Cassy ne sait jamais où elles repousseront exactement, cependant, ce qu'elle sait très bien c'est qu'elle veut les manger toutes ! Pour ce faire, Cassy démarre chaque matin au milieu du champ et suit les instructions selon le bloc d'instructions.

Procède en sorte que Cassy parcourt toutes les cellules de grille de son champ à salades.

- R est une variable de comptage. Quand le bloc d'instructions est exécuté la première fois R est égal à 1, à la deuxième exécution, R est égal à 2, et ainsi de suite.
- Cassy peut quitter le champ, mais elle n'est pas autorisée à quitter son pays.
- En cliquant sur le bouton «tester», tu peux tester ton programme.





Solution


Il est possible d'insérer quatre instructions au maximum par bloc d'instructions et chaque bloc est répété cinq fois. Ceci signifie que le chemin de la tortue ressemblera à une spirale. Il existe quatre séquences d'instructions que Cassy pourrait suivre afin de parcourir le champ entier et en spirale.


- R cellules en avant, tourner à gauche, R cellules en avant, tourner à gauche

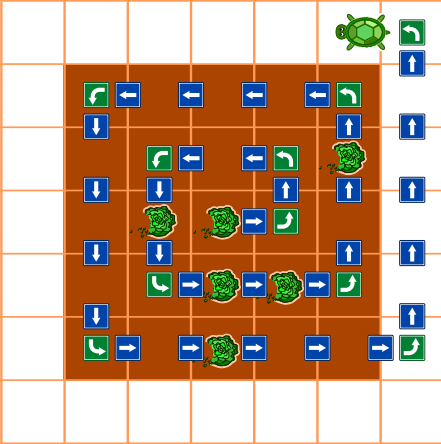
À répéter cinq fois!

R cellules en avant 

tourner à gauche 


R cellules en avant 


tourner à gauche 





- R cellules en avant, tourner à droite, R cellules en avant, tourner à droite

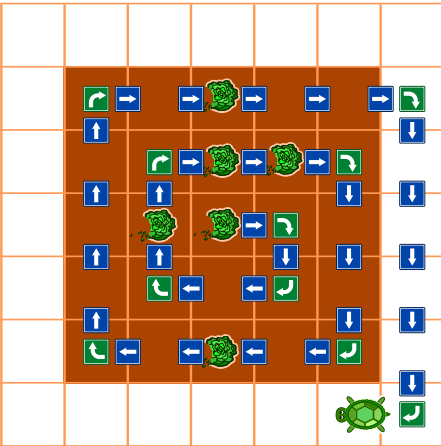
À répéter cinq fois!

R cellules en avant 

tourner à droite 


R cellules en avant 


tourner à droite 





- tourner à gauche, R cellules en avant, tourner à gauche, R cellules en avant

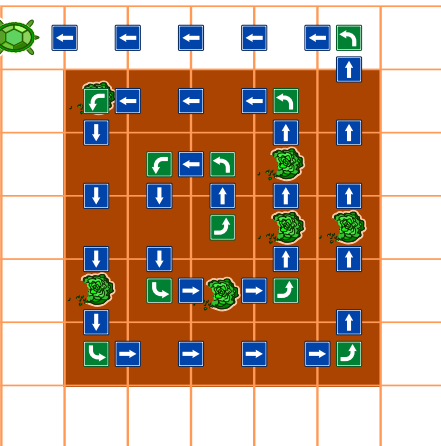
À répéter cinq fois!

tourner à gauche 

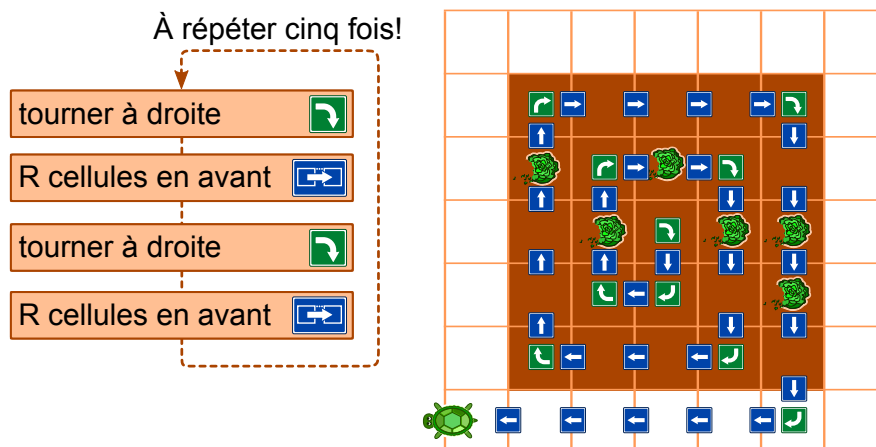
R cellules en avant 

tourner à gauche 

R cellules en avant 



- tourner à droite, R cellules en avant, tourner à droite, R cellules en avant



C'est de l'informatique !

Cette tâche nécessite une programmation informatique. Le programme comportera une séquence d'instructions (un bloc d'instructions) qui sera répétée cinq fois. Ce type de répétition fait partie des répétitions cycliques rassemblées sous le terme «boucle informatique». Dans le cas présent, on parlera d'une boucle de comptage car R passe de 1 à 5. Le processeur d'un ordinateur exécute ces instructions dans l'ordre. Tous les langages de programmation comprennent des boucles – ainsi que d'autres instructions qui peuvent contrôler le déroulement du programme, comme par exemple les branchements et les appels aux sous-programmes.

Si le programme fonctionne correctement, l'ordinateur se comportera exactement comme tu l'avais prévu. Si le programme ne fonctionne pas correctement, l'ordinateur procédera à l'exécution des instructions. Pourtant, Cassy la tortue ne parcourra pas les champs comme tu l'avais prévu. Habituellement, un ordinateur n'est pas capable de reconnaître si un programme créé et exécuté aboutira à ses – disons à tes – fins!

Sites web et mots clés

la fenêtre graphique Turtle

- <http://www.turtlegrafik.ch/>
- <http://primalogo.ch/fr>



A. Auteurs des exercices

 Daphne Blokhuis
 Valentina Dagienė
 Christian Datzko
 Susanne Datzko
 Olivier Ens
 Jürgen Frühwirth
 Gerald Futschek
 Peter Garscha
 Yasemin Gülbahar
 Martin Guggisberg
 Urs Hauser
 Hans-Werner Hein
 Mathias Hiron
 Juraj Hromkovič

 Yukio Idosaka
 Mile Jovanov
 Martina Kabátová
 Joseph Kaperst
 Tobias Kohn
 Ivana Kosírová
 Khairul A. Mohamad Zaki
 Hamed Mohebbi
 Tom Naughton
 Serena Pedrocchi
 Wolfgang Pohl
 Sergei Pozdniakov
 Dániel Pressing
 J. P. Pretti


 Kirsten Schlüter
 Eljakim Schrijvers
 Maiko Shimabuku
 Emil Stankov
 Björn Steffen
 Gabrielė Stupurienė
 Peter Tomcsányi
 Monika Tomcsányiová
 Willem van der Vegt
 Jiří Vaníček
 Troy Vasiga
 Michael Weigend



B. Sponsoring : Concours 2016

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


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

 **digitec.ch** <http://www.digitec.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

 <http://www.verkehrshaus.ch/>
i-factory (Musée des transports, Lucerne)

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

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

 <http://www.presentex.ch/>
PRESENTEX
Das Geschenk - die gute Werbung



ITgirls@hslu

<https://www.hslu.ch/de-ch/informatik/agenda/veranstaltungen/fuer-schulen/itgirls/>
HLSU, Lucerne University of Applied Sciences and Arts
Engineering & Architecture

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.



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.