



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

Exercices et solutions 2016 Années HarmoS 13/14/15

<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//sociétésuissepourl'infor
matique dansl'enseignement//societàsviz
zera per l'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. 43.



Préambule

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

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

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

Le Castor Informatique vise à motiver les élèves 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. Le code KIX	1
2. Le filtre médian	3
3. La poignée de main style zip	5
4. Les jeux de Nim	9
5. Le classement des numéros de maillot	11
6. Cassy, la tortue	13
7. Chaque palette en son temps !	17
8. Jeux de boule	21
9. Deux possibilités exclusives	25
10. Le carrefour des trams	27
11. Le codage des images numériques	29
12. Las de nouer !	33
13. Peinture récursive	35
14. Le jeu de formes géométriques	37
15. Les écureuils égoïstes	41
A. Auteurs des exercices	43
B. Sponsoring : Concours 2016	44
C. Offres ultérieures	46



1. 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 ?



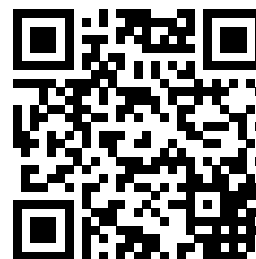
Solution

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>



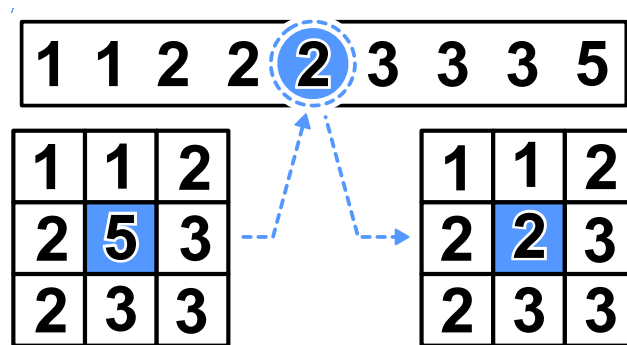
2. Le filtre médian

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.

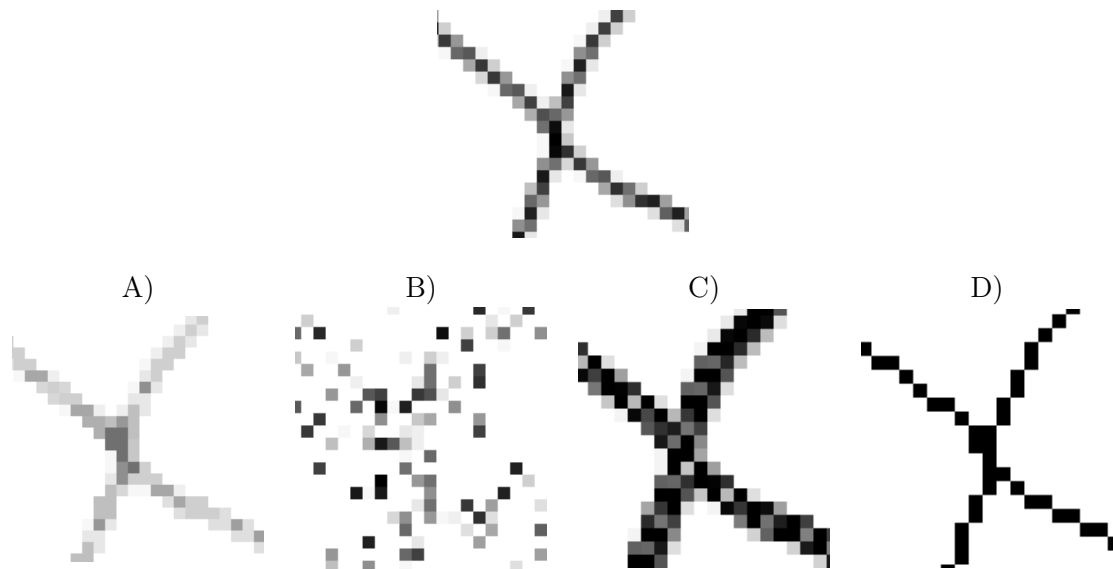
Un filtre appelé «filtre médian» influence le ton du gris de chaque pixel de l'image de sorte que...

- ... 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 ?





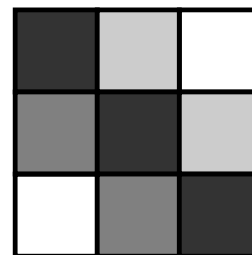
Solution

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

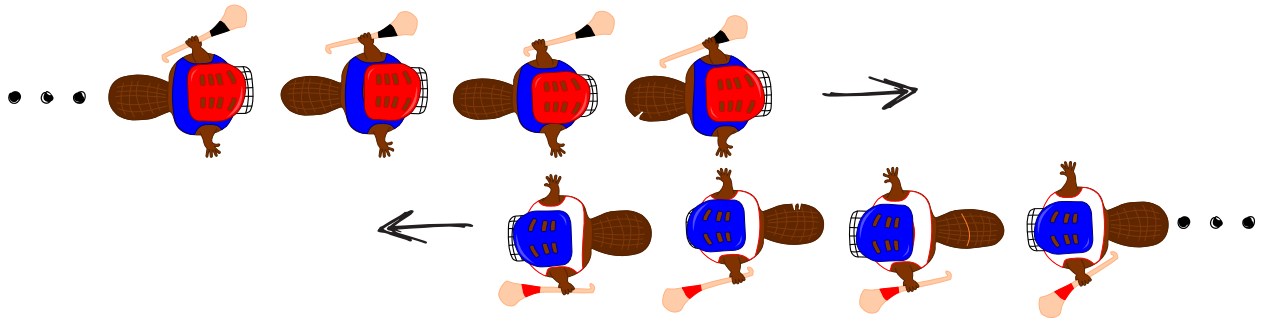
traitement des images, filtre médian, nuances de gris

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



3. 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.

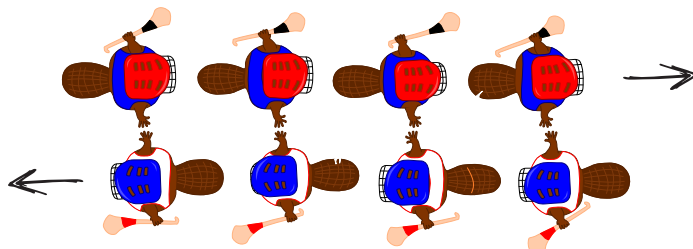
Combien de secondes faut-il au total pour que les deux équipes aient terminé le rituel de la poignée de main style zip ?



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

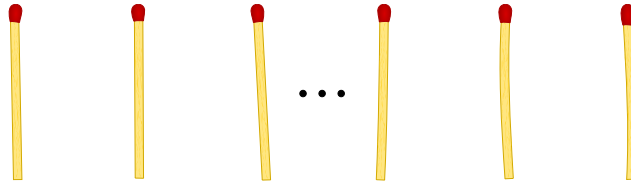
— https://fr.wikipedia.org/wiki/Comparaison_asymptotique





4. Les jeux de Nim

Beat et son ami jouent aux jeux de Nim. Sur la table, il y a 13 allumettes. À tour de rôle, les deux joueurs enlèvent soit 1 soit 2 soit 3 allumettes. Le joueur qui enlève la dernière allumette a gagné la partie.



Remarque : S'il n'y a que quatre allumettes sur la table, Beat ne peut plus gagner. C'est exactement cette situation qu'il vise à éviter.

C'est le tour de Beat. Combien d'allumettes doit-il enlever pour gagner cette partie ?

- A) 1
- B) 2
- C) 3
- D) Peu importe combien.



Solution

La réponse A) est correcte. Ainsi, il ne reste que 12 allumettes. Son ami peut maintenant enlever 1, 2 ou 3 allumettes et Beat en prend autant pour qu'il n'en reste que 8. Ensuite, c'est le tour de son ami qui a le choix d'enlever soit 1, 2 ou 3 allumettes. Finalement, Beat n'a qu'à enlever autant pour qu'il n'en reste que 4. Son ami n'aura donc plus de chances de gagner la partie.

Si Beat enlevait 2 ou 3 allumettes au début de la partie, son ami enlèverait, en toute logique, autant d'allumettes pour qu'il n'en reste qu'un multiple du nombre 4 ... et Beat serait le perdant.

C'est de l'informatique !

Dans la théorie des jeux, les jeux comme les jeux de Nim ou le dilemme du prisonnier servent de modèle pour analyser des problèmes stratégiques réels afin d'en déduire une méthode de résolution. Dans le domaine de l'économie de marché, par exemple, les conclusions de ces jeux de réflexion peuvent servir à fixer les prix de façon optimale. D'une part, il est un fait bien connu que des baisses de prix peuvent augmenter les ventes. D'autre part, une telle décision peut affecter considérablement la marge bénéficiaire de chaque produit vendu. À l'inverse, une augmentation de prix permettra d'augmenter la marge bénéficiaire de chaque produit vendu. Pourtant, une telle décision peut affecter les ventes et, en conséquence logique, le bénéfice total de la société. Ainsi, les modèles appliqués dans la théorie des jeux peuvent aider à prédire de possibles réactions des acheteurs aux ajustements des prix. On ne comprend l'immense valeur des modèles de la théorie des jeux pour l'économie que quand on réalise qu'il existe déjà plusieurs travaux scientifiques sur des modèles de la théorie des jeux couronnés du Prix Nobel.

Sites web et mots clés

jeux de Nim, théorie des jeux, arbre de décision

- <http://www.mathematische-basteleien.de/nimspiel.html>
- <http://scienceblogs.de/zoonpolitikon/2008/04/22/spieltheorie-einfach-erklart-i-einleitung-und-gefangenendilemma/>



5. Le classement des numéros de maillot

Les images suivantes montrent deux équipes de 15 joueurs. Les joueurs des deux équipes portent des maillots imprimés avec un numéro individuel. Ceux de la première équipe se tiennent debout, l'un à côté de l'autre et sont classés par numéro. Ceux de la seconde équipe se tiennent également debout mais ils ne sont pas classés par numéro.

L'équipe no. 1 :



L'équipe no. 2 :



Comment peut-on savoir le plus vite possible quels numéros sont représentés tant dans l'équipe no. 1 que dans l'équipe no. 2 ?

- A) On parcourt les numéros de l'équipe no. 1 (1, 4, 5, ...) et on vérifie si ces numéros existent également dans l'équipe no. 2.
- B) On parcourt les numéros de l'équipe no. 2 (8, 28, 12, ...) et on vérifie si ces numéros existent également dans l'équipe no. 1.
- C) Peu importe avec quelle équipe vous commencez. La durée pour vérifier tous les nombres est la même pour les deux équipes.
- D) Il s'avère judicieux de vérifier d'abord combien de nombres n'existent pas dans les deux équipes. En soustrayant ce nombre du nombre total de 15, on obtient le nombre recherché.



Solution

La réponse correcte est B). Dans une série ordonnée, on peut déterminer une certaine valeur, comme par exemple les nombres, plus rapidement que dans une série non-ordonnée. Un ordinateur aurait besoin de seulement $\log_2(n)$ étapes de recherche au lieu de n étapes. Avec le procédé décrit dans la réponse B), il aurait donc besoin au pire des cas de $n \cdot \log_2(n)$ étapes de recherche.

Avec le procédé décrit dans la réponse A), on ne pourrait pas parcourir la liste des numéros de manière rapide. Bien au contraire, il faudrait que l'on parcoure tous les nombres un par un. Un tel procédé demanderait au pire des cas n^2 étapes de recherche, ce qui reviendra donc à plus de $n \cdot \log_2(n)$ étapes de recherche.

Par conséquent, la réponse C) ne peut pas être juste car n^2 n'est pas égal à $n \cdot \log_2(n)$.

Le procédé décrit dans la réponse D) n'aboutit en général pas au résultat souhaité. Voilà pourquoi cette réponse sera rejetée.

C'est de l'informatique !

Il est beaucoup plus facile de fouiller une série ordonnée qu'une série non-ordonnée. Regardons de plus près comment tu as cherché par exemple le numéro 9 dans l'équipe no. 1. D'abord, tu regardes le joueur situé au milieu de l'équipe qui porte le numéro 17. Le numéro recherché est plus petit que le numéro 17, voilà pourquoi tu continues à chercher le numéro 9 dans la partie gauche de la série ordonnée. Là, tu fixes d'abord la partie centrale. Tu y découvres le numéro 7. Finalement, tu regardes à droite et tu y découvres immédiatement le numéro 9. L'astuce est que tu divises la zone de recherche par deux à chaque étape de ta recherche et que, de cette manière, tu la réduis à une taille beaucoup plus transparente. En informatique on parle de «recherche binaire». Ce mot d'origine latine comprend le terme *bis* qui, lui, signifie «deux fois». Le classement prend une place importante en informatique car la recherche binaire permet des opérations beaucoup plus rapides à l'intérieur des ensembles de données ordonnées. Ainsi, dans le cas d'une recherche de n nombres, la recherche binaire n'aura besoin que d'environ $\log_2(n)$ étapes de recherche.

Sites web et mots clés

classement, recherche binaire

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



6. 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.

À répéter cinq fois!

instructions

bloc d'instructions

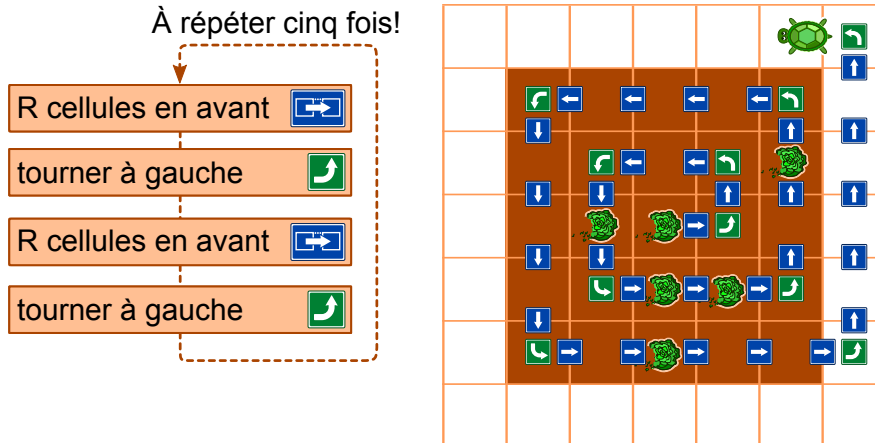
tester



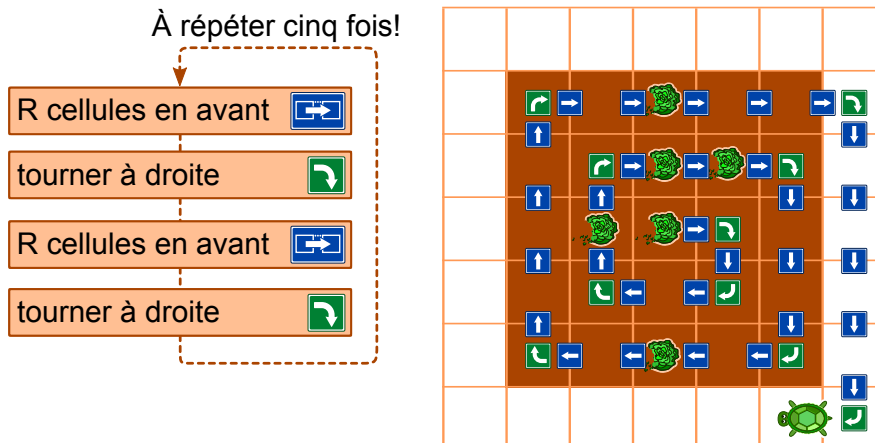
Solution

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.

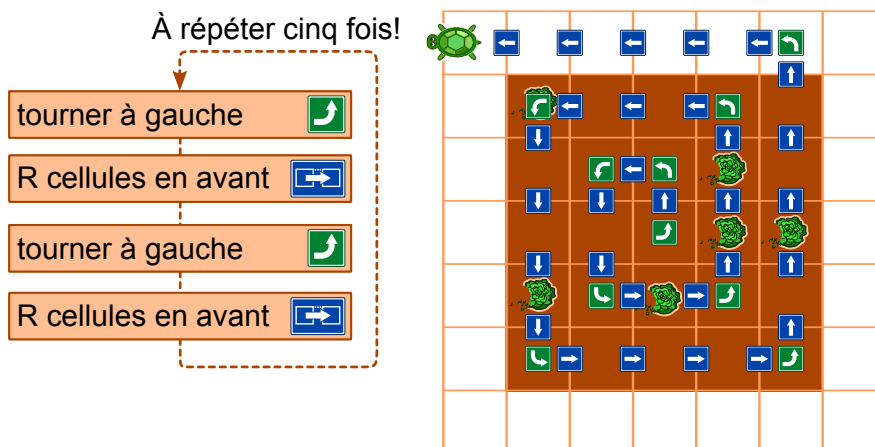
- R cellules en avant, tourner à gauche, R cellules en avant, tourner à gauche



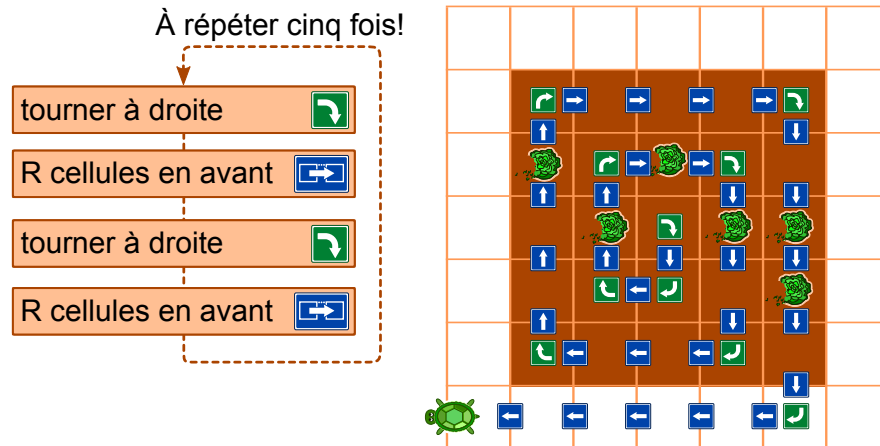
- R cellules en avant, tourner à droite, R cellules en avant, tourner à droite



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



- 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>

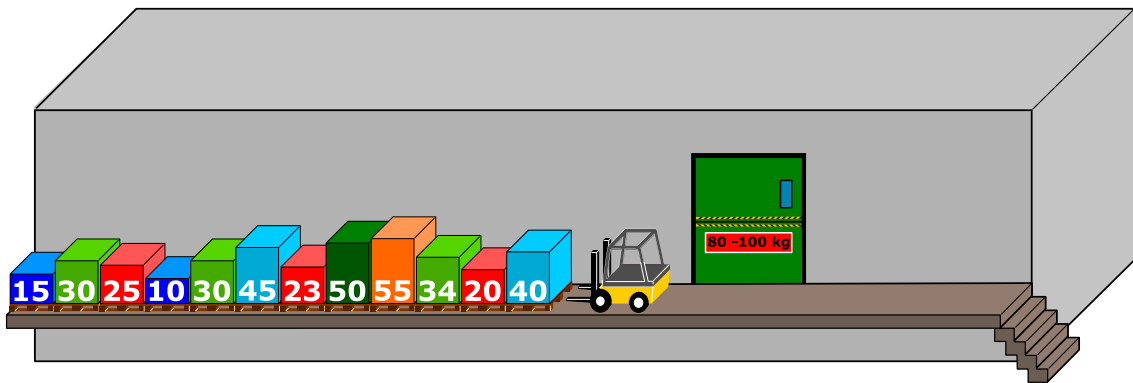




7. Chaque palette en son temps !

Sur une plate-forme de chargement très étroite, il y a des palettes placées l'une à côté de l'autre. Chaque palette présente un poids différent (de gauche à droite) : 15 kg, 30 kg, 25 kg, 10 kg, 30 kg, 45 kg, 23 kg, 50 kg, 55 kg, 34 kg, 20 kg, 40 kg. La plate-forme est étroite au point que l'on ne peut pas déplacer les palettes en les soulevant l'une au-dessus de l'autre.

Afin de transporter les palettes dans un entrepôt, il faut les mettre dans un ascenseur de charge. Cet ascenseur transporte les palettes à l'entrepôt quand la charge comporte au moins 80 kg. Pourtant, la charge ne doit pas dépasser les 100 kg. Au moment où l'ascenseur est déchargé, il repart vide vers le niveau de la plate-forme.



L'élévateur de palettes déplace généralement d'abord la palette qui se trouve le plus près du monte-charge. Au cas où le poids total dépasse les 100 kg lors du chargement de l'ascenseur avec la dernière palette, l'élévateur de palettes dépose cette dernière à l'autre bout de la plate-forme (à droite). Si le poids total ne dépasse pas les 100 kg, la palette restera dans le monte-charge.

Quand toutes les palettes du côté gauche de la plate-forme se trouveront dans l'entrepôt, l'élévateur de palettes déplacera de la même manière toutes les autres palettes du côté droit de la plate-forme afin de les déposer dans le monte-charge.

Laquelle des affirmations ci-dessous est correcte ?

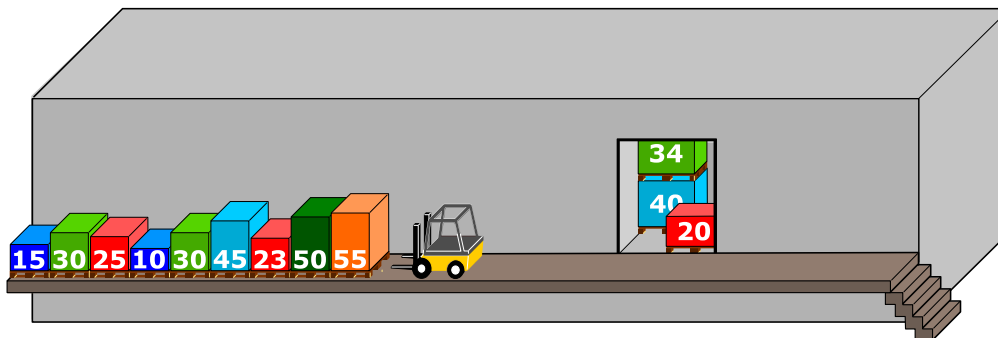
- A) La deuxième charge de l'ascenseur s'élève à 98 kg.
- B) On n'utilise pas les palettes de l'autre bout de la plate-forme.
- C) Une charge du monte-charge s'élève à 100 kg.
- D) Le monte-charge descend et monte cinq fois au total.
- E) La méthode pour déplacer les palettes décrite ci-dessus ne permet pas de transporter les palettes dans l'entrepôt.



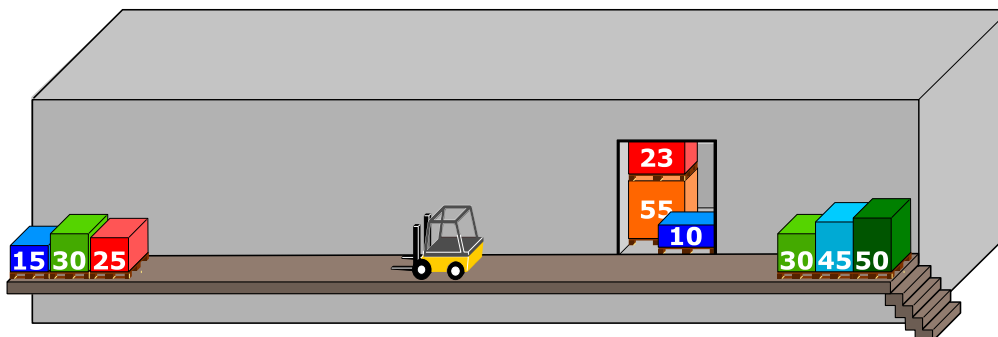
Solution

La réponse correcte est C) :

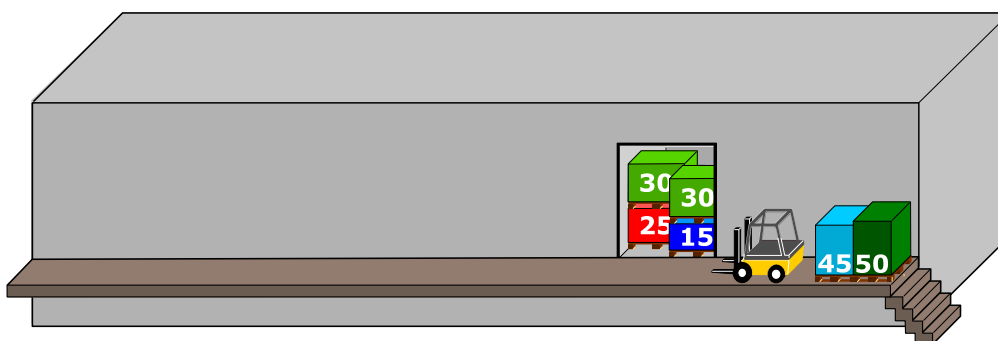
- La première charge comporte les trois premières palettes. Leur poids s'élève à : $40 \text{ kg} + 20 \text{ kg} + 34 \text{ kg} = 94 \text{ kg}$.



- La deuxième charge ne comporte qu'une seule palette (55 kg). Si on y ajoutait la palette suivante (50 kg), la charge dépasserait les 100 kg, c'est la raison pour laquelle l'élévateur de palettes la dépose au côté droit de la plate-forme de chargement. Ensuite, il charge l'ascenseur avec la palette suivante (23 kg) ; cette charge sera encore trop légère : le poids total ne s'élèvera qu'à $55 \text{ kg} + 23 \text{ kg} = 78 \text{ kg}$. Avec la palette suivante (45 kg), par contre, la charge dépasserait de nouveau le poids maximal de 100 kg. Voilà pourquoi l'élévateur de palettes déposera la palette de 45 kg au côté droit de la plate-forme, juste à côté de la palette à 50 kg. On procédera de la même manière avec la palette suivante (30 kg). Finalement, il est possible de charger l'ascenseur avec la palette dont le poids s'élève à 10 kg seulement. Ainsi, l'ascenseur repartira vers l'entrepôt avec la charge de $55 \text{ kg} + 23 \text{ kg} + 10 \text{ kg} = 88 \text{ kg}$.



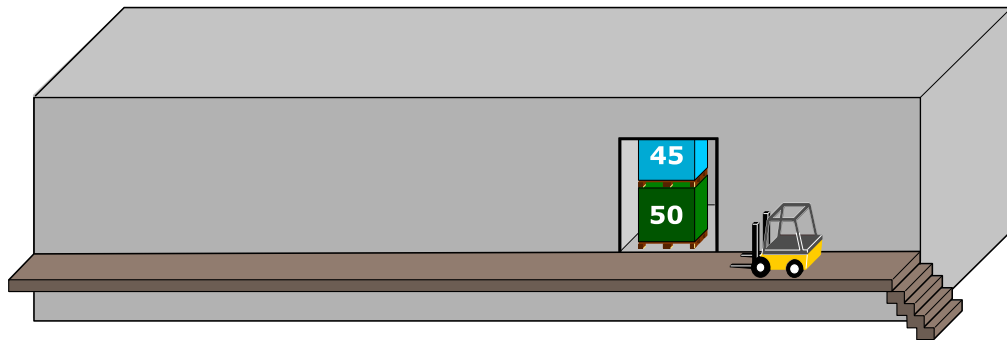
- L'élévateur de palettes déplace ensuite les trois dernières palettes ($25 \text{ kg} + 30 \text{ kg} + 15 \text{ kg} = 70 \text{ kg}$) et les dépose dans le monte-charge. Il ajoutera à cette troisième charge la palette de 30 kg se trouvant actuellement comme première dans la rangée des palettes au côté droit de la plate-forme. Ainsi, la charge de l'ascenseur s'élèvera exactement à 100 kg.



- Finalement, il ne restent que les deux palettes du côté gauche de la plate-forme :



45 kg + 50 kg = 95 kg. Elles constitueront la quatrième et dernière charge que l'élevateur de palette déposera dans le monte-charge.



Prenant en compte les explications ci-dessus, les réponses A), B), D) et E) ne sont pas correctes.

C'est de l'informatique !

Comme la plate-forme est si étroite, l'élevateur ne peut déplacer que la première palette de la rangée avant de traiter les suivantes. Ce traitement est comparable au principe de l'empilage : on ne peut enlever que l'objet qui se trouve tout en haut de la pile et, de manière analogue, on ne peut ajouter un autre objet qu'en le déposant sur l'objet qui se trouve tout en haut de la pile. En informatique, on nomme une telle structure de données également «une pile». Dans la présente tâche, nous avons trois piles : la rangée de palettes de gauche (avec un premier élément à l'extrême droite), la rangée de palettes à droite (avec un premier élément à l'extrême gauche) et le monte-charge.

Au quotidien, nous connaissons aussi des piles : quand un enfant construit une tour avec des blocs, il ne peut enlever ou ajouter un bloc si ce-dernier se trouve tout en haut de la pile. Il en est de même pour un cornet de glace à plusieurs boules : la boule ajoutée en dernier sera mangée en premier. Le même principe s'applique à la navigation sur Internet, car la sauvegarde des pages visitées sur Internet suit le principe de l'empilage : en cliquant sur «retour», la page visitée en dernier sera affichée en premier.

Ce principe est également appelé LIFO : „last in – first out“ («dernier arrivé, premier sorti»). Il s'oppose au principe FIFO : „first in – first out“ («premier arrivé, premier sorti»). Le principe FIFO est un mode de traitement que l'on applique dans le cas d'une queue au guichet ou dans un cabinet médical (bien que les cas d'urgence puissent heureusement être traités en priorité).

Sites web et mots clés

structure de données, pile, algorithme, principe LIFO

- [https://fr.wikipedia.org/wiki/File_\(structure_de_données\)](https://fr.wikipedia.org/wiki/File_(structure_de_données))
- https://fr.wikipedia.org/wiki/Last_in,_first_out





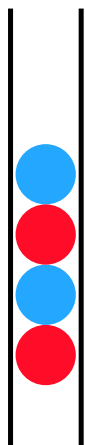
8. Jeux de boule

Emil s’amuse à jouer à un nouveau jeu vidéo sur son ordinateur. Le jeu démarre avec une pile d’au moins trois boules colorées (en rouge ou en bleu). Ces boules se trouvent à l’intérieur d’un tube. En cliquant sur un bouton, les deux boules qui se trouvent en bas de la pile sont poussées en avant et tombent du tube. En plus, au moment où elles tombent, de nouvelles boules tombent d’en haut pour se rajouter à la pile de boules dans le tube. En fonction de la couleur des boules qui se trouvent en bas de la pile, ce processus peut avoir deux conséquences :

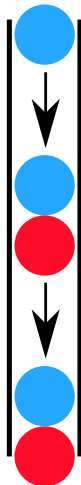
Si la boule qui se trouve en bas de la pile est rouge, une boule bleue tombe d’en haut et rejoint les boules empilées dans le tube :

Si la boule qui se trouve en bas de la pile est bleue, trois boules aux couleurs rouge, bleu et rouge rejoignent les boules empilées dans le tube :

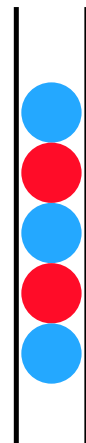
avant



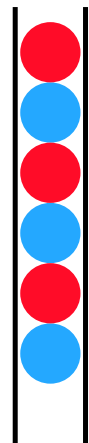
après



avant



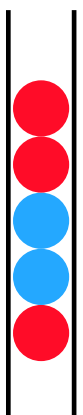
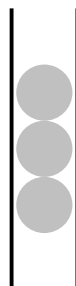
après



Emil répète le clic sur le bouton jusqu’à ce qu’il reste au moins trois boules dans le tube. Le jeu se termine au moment où il y a moins de trois boules dans le tube.

Si Emil commence avec une pile comme elle est représentée dans la figure de droite, le jeu sera bientôt terminé car après cinq clics, il ne restera que deux boules dans le tube.

Choisis des boules colorées de la pile de droite pour les insérer dans les places d’empilement libres afin que tu aies une pile de départ qui permettra un jeu infini.

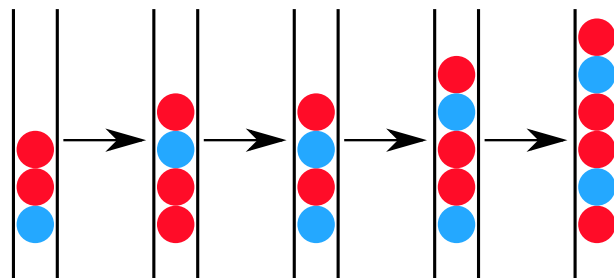
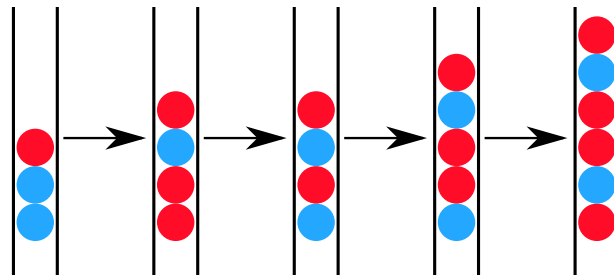
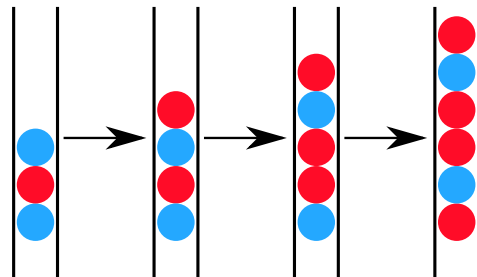
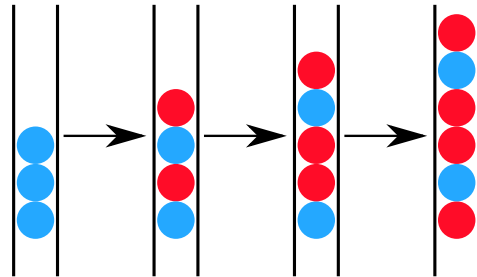




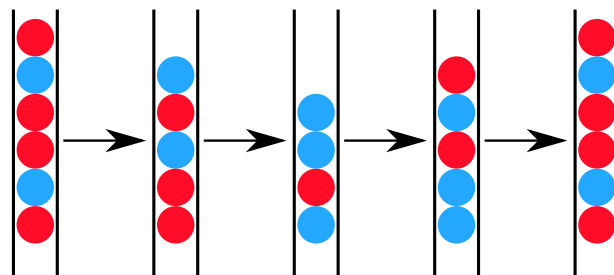
Solution

Si la boule qui se trouve tout en bas de la pile est rouge, le jeu se terminera tout de suite après le premier clic sur le bouton parce que après ce premier clic, il ne restera que deux boules dans le tube.

Mais, si la boule qui se trouve tout en bas de la pile est bleue, le jeu continuera à l'infini. Avec au maximum quatre clics sur le bouton, on obtient de nouveau la pile à six boules avec la séquence rouge-bleu-rouge-rouge-bleu-rouge :



Par la suite, le jeu continue selon un cycle à quatre étapes en tant que cycle infini :





C'est de l'informatique !

Le jeu présenté dans cette tâche a servi comme exemple lors d'une démonstration de la part d'Emil Leon Post. Avec cet exemple, il voulait montrer qu'il est possible de créer des processus infinis durant la réécriture des chaînes de caractères («strings»). Emil Leon Post (1897-1954) était un mathématicien et logicien polonais qui a publié beaucoup d'articles relatifs à l'informatique théorique notamment dans le domaine de la logique propositionnelle. On peut décrire un tel modèle de réécriture à l'aide de la grammaire formelle. À partir de l'alphabet d'entrée, on définit les règles selon lesquelles on substituera des caractères. Dans notre exemple, la grammaire connaîtra quatre règles (où X représentera une quelconque combinaison de boules bleues et rouges) :

$$Xbb \rightarrow rbrX$$

$$Xrb \rightarrow rbrX$$

$$Xbr \rightarrow bX$$

$$Xrr \rightarrow bX$$

Sites web et mots clés

modèle de calcul, langue formelle, règle de production, chaîne de caractères, système de traitement de données

- http://esolangs.org/wiki/Post_canonical_system
- https://en.wikipedia.org/wiki/Tag_system
- https://en.wikipedia.org/wiki/Post_canonical_system



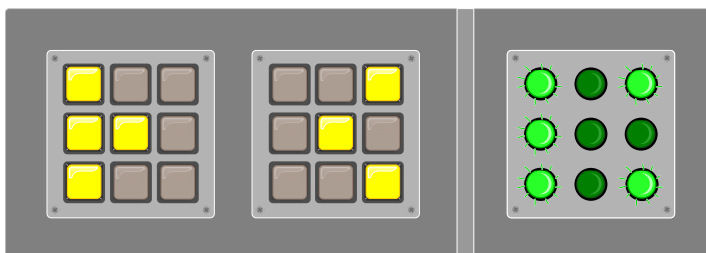


9. Deux possibilités exclusives

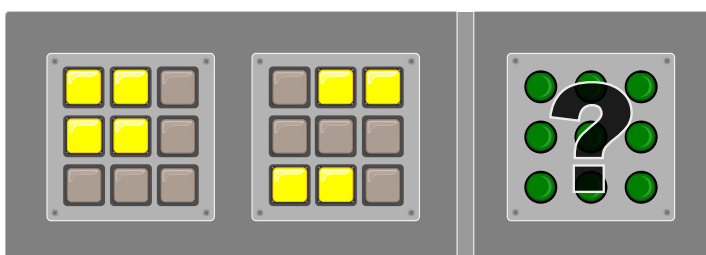
Erich a trouvé un vieil appareil électronique. Sur le côté gauche, il y a deux champs et chacun dispose de 9 touches sur lesquelles on peut appuyer. Sur le côté droit, il y a un champ avec 9 lampes. En fonction de la combinaison des touches, les lampes s'allument ou s'éteignent.

Erich découvre que la position d'une lampe qui s'allume et s'éteint correspond à une combinaison de touches particulière dans les deux champs.

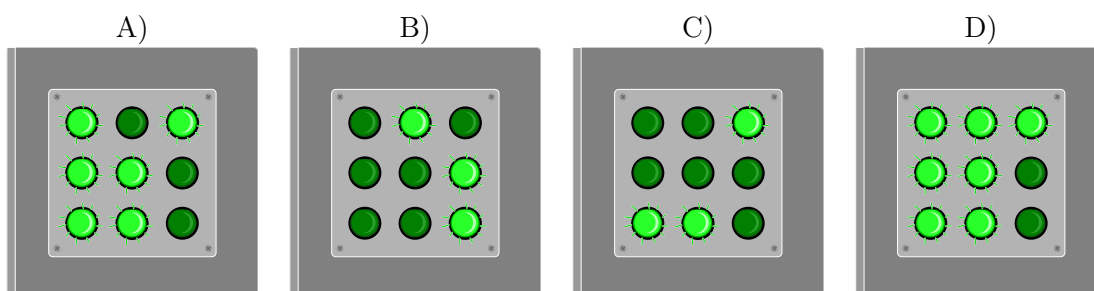
Actuellement, les lampes sont allumées comme suit :



Maintenant, Erich a envie de changer la combinaison des touches. Voilà le résultat :



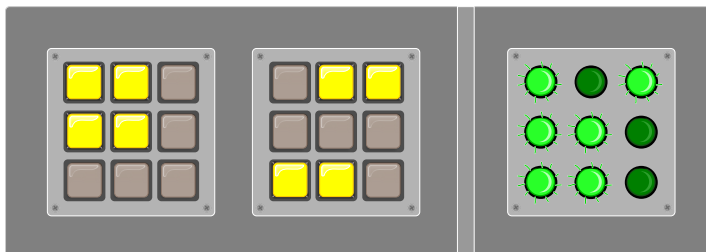
Lesquelles des lampes sont allumées ?





Solution

La réponse correcte est A). La règle est : si les touches de la même position dans les deux champs à gauche ne sont allumées ou éteintes qu'une seule fois, la lampe correspondante est allumée, sinon elle est éteinte.



C'est de l'informatique !

Dans tous les domaines de l'informatique, on a volontiers recours à la logique binaire pour vérifier dans une situation d'application particulière si une affirmation est vraie ou fausse. Ceci n'est possible que s'il n'existe que deux possibilités exclusives (deux valeurs). Une troisième possibilité sera ainsi exclue («*tertium non datur*»). Les fonctions logiques peuvent donc fournir deux valeurs logiques : soit «faux» soit «vrai».

Dans notre exemple, nous prenons recours à la fonction XOR (de l'anglais : eXclusive OR, «ou exclusif»). XOR fonctionne comme suit : partant de deux affirmations dont une seule est exclusivement «vraie», XOR livre la valeur logique «vrai». Pour tous les autres cas, XOR livre la valeur logique «faux». Dans l'exemple de notre tâche, la fonction logique XOR est appliquée neuf fois (touches) et elle livre deux affirmations (lampes).

Sites web et mots clés

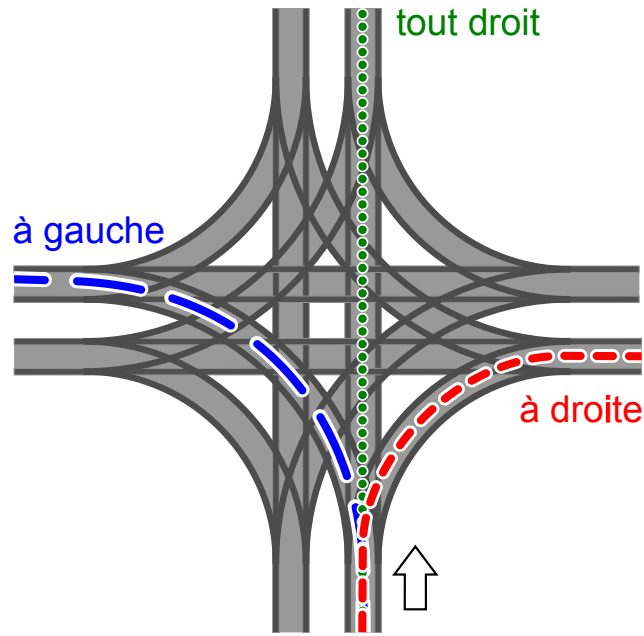
logique, binaire, fonction XOR

- https://fr.wikipedia.org/wiki/Fonction_OU_exclusif
- https://en.wikipedia.org/wiki/XOR_gate



10. Le carrefour des trams

Dans les rues de Saint-Petersbourg circulent des trams. Au centre-ville, il y a un carrefour qui leur permet de continuer dans toutes les directions : Quelle que soit la direction d'où ils arrivent, ils peuvent tourner à gauche, à droite ou continuer tout droit.



La direction dans laquelle les trams partent est déterminée par la position des aiguilles. Ces positions sont décrites à l'aide de la combinaison des mots suivants : TOUT DROIT, À GAUCHE, À DROITE. La combinaison À GAUCHE-TOUT DROIT-À GAUCHE-À DROITE, par exemple, signifie que la position des aiguilles est telle que le premier tram qui arrive tourne à gauche, le suivant (en sens horaire) continue tout droit, le succédant (toujours en sens horaire) tourne à gauche et le quatrième tram tourne à droite.

Il est également possible que les trams arrivent en même temps au carrefour tout en provenant de toutes les directions possibles.

Quelles positions des aiguilles risquent de provoquer une collision entre les trams ?

- A) À DROITE-À DROITE-À DROITE-À DROITE
- B) À DROITE-À DROITE-À GAUCHE- À GAUCHE
- C) À GAUCHE-À DROITE-À GAUCHE-À DROITE
- D) À DROITE-À GAUCHE- À DROITE- À GAUCHE



Solution

La réponse correcte est B). Le premier tram (qui tourne à droite) heurtera le troisième tram (qui tourne à gauche). De même, le deuxième tram (qui tourne à droite) heurtera le quatrième tram (qui tourne à gauche).

Toutes les autres solutions permettront aux trams de passer sans aucun incident : dans la solution A), tous les trams se trouvent sur la chaussée à l'extrême droite, dans la solution C) et D), il y a deux trams qui passent tranquillement l'un à côté de l'autre. Ici, il est important de noter que les chaussées des trams au centre ne se touchent pas. On appelle une chaussée où deux trams tournent à gauche sur deux chaussées séparées et parallèles une chaussée «tangentielle» ou «américaine».

C'est de l'informatique !

Les chaussées des trams – surtout en matière de croisements – représentent en fait des ressources auxquelles de différents acteurs ont recours et qu'ils partagent conjointement. Dans notre exemple, afin d'éviter une collision, il est pourtant important qu'il n'y ait qu'un seul acteur (dans notre cas il s'agit d'un seul tram) par endroit. En informatique, la coordination et l'optimisation du partage des ressources de différents processeurs en cours est un domaine important. Tout comme dans le cas de la chaussée des trams, la technologie digitale requiert que l'on se rende compte des risques et que l'on respecte des règles qui garantissent par exemple l'accès sécurisé aux bases de données ou bien aux ordinateurs publics ou partagés. Dans le cas contraire, on risque des pertes de données ou la destruction matérielle de composants informatiques.

Dans le cas du carrefour des trams, il est judicieux de n'approuver que des positions d'aiguilles qui permettent un passage sécurisé des trams. Une solution tout à fait valable serait d'installer des signaux lumineux. L'informatique, elle aussi, connaît des solutions qui ont recours à des signaux appelés «sémaphores» (de l'anglais pour «signal»). Leur fonction est d'indiquer au processeur quelles ressources sont en train d'être utilisées afin qu'il attende son propre tour.

Sites web et mots clés

carrefour des trams, synchronisation des processus, accès aux ressources restreintes

- [https://fr.wikipedia.org/wiki/Synchronisation_\(multitâches\)](https://fr.wikipedia.org/wiki/Synchronisation_(multitâches))
- <http://www.swisseduc.ch/informatik/infotraffic/logictraffic/>



11. Le codage des images numériques

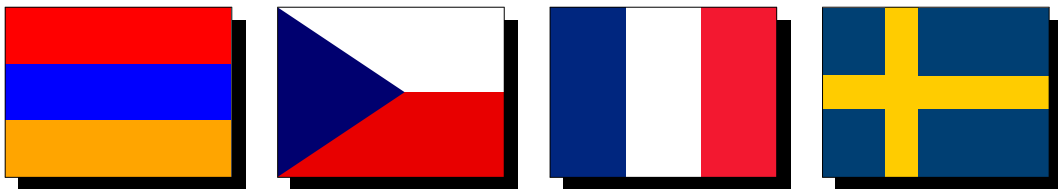
Les images numériques sont constituées d'un ensemble de points appelés « pixels ». Quand on veut enregistrer des images numériques en tant que données, dans le cas le plus simple, on décrit la couleur de chaque pixel individuellement. Le format de fichier GIW (imaginaire) permet de compresser les images numériques, c'est-à-dire de les enregistrer avec une taille réduite. Voilà comment ce procédé fonctionne :

- On code chaque ligne de pixels séparément.
- On code chaque couleur en lui attribuant une abréviation comprenant trois lettres.
- Une suite de pixels de la même couleur est codée par une paire de parenthèses comprenant l'abréviation des couleurs correspondantes ainsi que le nombre des pixels de la même couleur.

Exemple : la ligne de pixels codée par les paires de parenthèses (ver, 20) (blc, 13) comprend d'abord 20 pixels verts, puis 13 pixels blancs.

Voici quatre images numérique représentant quatre drapeaux. Le nombre de lignes de pixels est égal pour chaque image tout comme le nombre de pixels. Pour enregistrer les images, on choisit le format GIW.

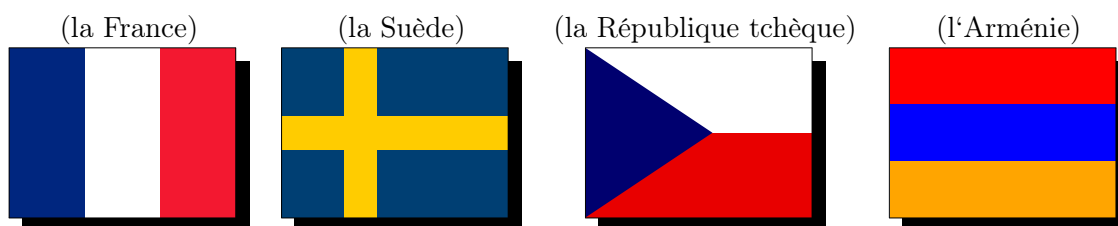
Classe les images selon la taille de leur fichier GIW !





Solution

Le bon ordre est le suivant :



Au cas où une ligne de pixels entière ne comporte qu'une seule couleur, il ne faut la coder que par une paire de parenthèses. Chaque changement de couleur entraîne une nouvelle paire de parenthèses. Dans le drapeau de l'Arménie, par exemple, chaque ligne de pixels comporte la même couleur. C'est ainsi que l'on peut coder chaque ligne de pixels par une seule paire de parenthèses.

Dans l'exemple du drapeau de la République tchèque, par contre, la couleur change à l'intérieur de chaque ligne de pixels et ceci exactement une fois : du bleu au blanc ou du bleu au rouge. Le codage pour un fichier comprimé de ce drapeau requiert donc pour chaque ligne de pixels deux paires de parenthèses. Pour le codage du fichier comprimé du drapeau français, on a besoin de trois paires de parenthèses car la couleur change dans ce cas-là deux fois à l'intérieur de chaque ligne de pixels.

Le drapeau suédois comporte une bande jaune horizontale. Pour ce secteur, on a donc besoin d'une paire de parenthèses par ligne de pixels. Dans toutes les autres lignes de pixels, la couleur change deux fois et on a besoin de trois paires de parenthèses. Il s'ensuit que le codage de l'image du drapeau suédois nécessite plus de paires de parenthèses que celui du drapeau arménien, mais moins que celui du drapeau français.

Finalement, il nous faut comparer le codage de l'image du drapeau suédois avec celui du drapeau tchèque. Si le drapeau suédois avait autant de lignes de pixels avec une couleur qu'avec trois couleurs, il faudrait en moyenne exactement 2 paires de parenthèses pour le codage. Comme la bande dans la zone moyenne du drapeau suédois est plus étroite que les autres parties du drapeau, le codage nécessite plus que deux paires de parenthèses. Il en résulte que, dans son ensemble, le codage du drapeau suédois a besoin de plus de paires de parenthèse que le codage du drapeau tchèque.

C'est de l'informatique !

La compression des données est un domaine important en informatique. Cette opération informatique consiste à réduire la taille de transmission ou de stockage des données afin de pouvoir les transmettre plus vite via le réseau. Les algorithmes de compression peuvent accélérer de manière considérable la transmission des données à l'intérieur d'un réseau. Si, par exemple, on diffusait la musique d'une webradio sans compression, on aurait besoin d'une bande passante capable de gérer un volume de données 10 fois supérieur au même volume de données compressé de façon courante aujourd'hui. C'est la raison pour laquelle on mise sur la recherche intensive des nouveaux algorithmes de compression capables de stocker les fichiers des photos, de la musique ou des vidéos de façon encore plus efficace. Le procédé de compression présenté dans la tâche appartient au type «encodage à double longueur de plage de données numériques» (RLE = Run-length encoding). Pour plus d'informations sur ce procédé, consulter : https://www.youtube.com/watch?v=yPdNscvym_E.

Sites web et mots clés

codage, compression, algorithme de compression, fichier graphique bitmap



- https://www.youtube.com/watch?v=ypdNscvym_E
- https://fr.wikipedia.org/wiki/Run-length_encoding





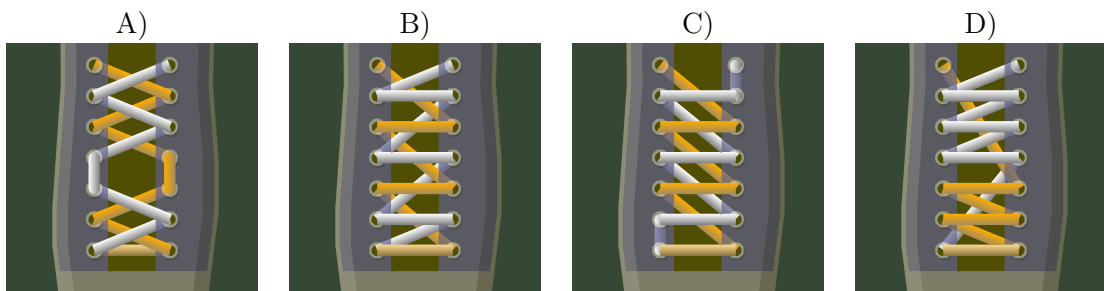
12. Las de nouer !

Les castors adorent les lacets chics. Mais comme ils sont las de les nouer, ils programment une machine qui les noue pour eux. Cette machine exécute les commandes suivantes :

Befehl	Bedeutung
{ ... }	Le contenu entre accolades est répété le plus souvent possible.
$n\{ \dots \}$	Le contenu entre accolades est répété une $n^{\text{ième}}$ fois.
orange:	Les commandes suivantes sont valides pour les lacets oranges.
blanc:	Les commandes suivantes sont valides pour les lacets blancs.
devant:	Le lacet passe devant l'œillet.
derrière:	Le lacet passe derrière l'œillet.
vers-le-haut:	Le lacet est mené vers le haut (vers le prochain œillet).
vers-le-bas:	Le lacet est mené vers le bas (vers le prochain œillet).
changement:	Le lacet est mené de gauche à droite ou l'inverse ; le lacet est ramené de manière correspondante.

Quel laçage les castors obtiennent-ils avec le programme suivant ?

```
orange: devant
blanc: devant
2{
    orange: vers-le-haut changement devant
    blanc: vers-le-haut changement devant
}
orange: vers-le-haut derrière
blanc: vers-le-haut derrière
{
    orange: vers-le-haut changement devant
    blanc: vers-le-haut changement devant
}
```





Solution

La solution correcte est la solution A). Le programme effectue le laçage des deux premiers œillets comme dans l'exemple. Ensuite, les lacets sont menés directement en haut. Finalement, les lacets changent de côté et le programme continue comme au départ.

C'est de l'informatique !

Le programme de la machine qui noue des lacets a recours à un langage de programmation simple. Ce dernier comprend des éléments caractéristiques du langage de programmation :

- Une série d'instructions effectuées les unes à la suite des autres (une *séquence*).
- Une répétition des blocs d'instructions (une *boucle*) d'un certain nombre de fois, et ceci, aussi bien en tant que boucle de comptage ainsi qu'en tant que boucle comportant des conditions d'arrêt.

À ces éléments caractéristiques du langage de programmation mentionnés ci-dessus s'ajoutent souvent deux autres éléments : l'exécution des blocs de base sous conditions particulières (*sélection*) et la sauvegarde des blocs de base fréquemment utilisés en tant que blocs séparés (*la programmation modulaire*), ce qui, par contre, n'est pas le cas dans notre exemple.

D'ailleurs, la machine qui noue des lacets peut être également programmée à travers un simulateur déjà existant. Pour ce faire, consultez le site : <http://www.klbg.com/lacing/>.

Sites web et mots clés

boucle, instructions, simulateur, lacets

- [https://fr.wikipedia.org/wiki/Interpr%C3%AAte_\(informatique\)](https://fr.wikipedia.org/wiki/Interpr%C3%AAte_(informatique))
- https://fr.wikipedia.org/wiki/Langage_de_programmation
- <http://www.klbg.com/lacing/>
- <http://www.fieggen.com/shoelace/lacingmethods.htm>

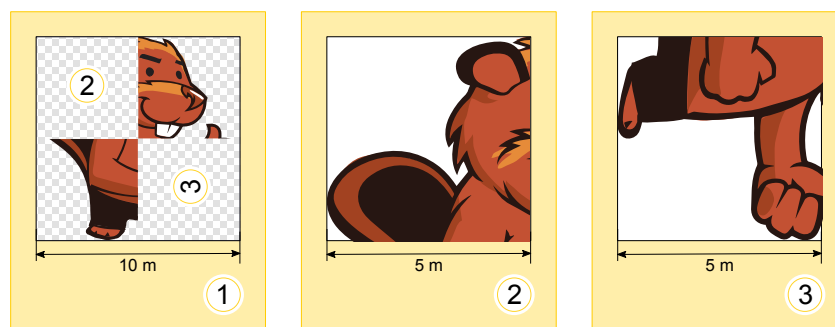


13. Peinture récursive

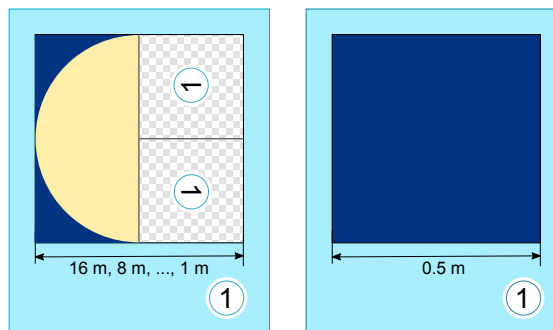
Tina et Ben sont en train de préparer une exposition temporaire au Musée de l'informatique. L'objectif est d'effectuer une peinture de 16 mètres sur 16 mètres sur le sol de l'espace de l'exposition temporaire.

De la part de l'artiste, Tina et Ben ont reçu un set de cartes. Ces cartes comportent des instructions formulées dans le langage pictural propre à l'artiste ainsi que des indications concernant les éléments de l'image, les mesures et les rotations nécessaires à effectuer. Quelques cartes comportent en outre des cases numérotées qui renvoient aux autres cartes du set.

Voici un exemple tiré d'un projet de peinture précédent. L'assemblage correct de tous les éléments montre l'image d'un castor. À noter en particulier : les longueurs différentes des cartes.

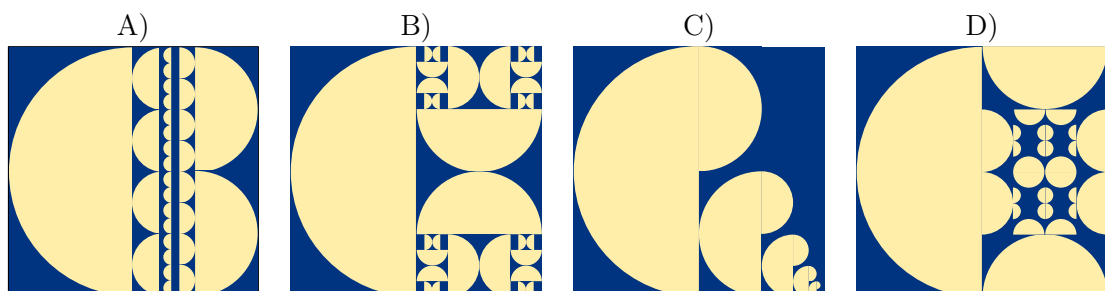


Afin de réaliser la peinture de l'exposition temporaire, Tina et Ben reçoivent deux cartes comportant les instructions suivantes :



Ben s'étonne : «Je ne vois pas comment réaliser ce projet. La carte de gauche se réfère à elle-même et puis, sur les deux cartes figure exactement le même numéro!». Tina éclate de rire : «On va y arriver, tu verras ! D'abord, nous ne nous servons que de la carte de gauche. La carte de droite nous indiquera ensuite le moment où nous devons achever l'œuvre».

Laquelle des quatre œuvres ci-dessous figurera sur le sol de l'espace de l'exposition ?





Solution

La carte de gauche nous indique qu'il faut peindre un demi-cercle sur la moitié gauche du sol. Le côté rond du demi-cercle est tourné vers la gauche. Pour peindre la moitié droite du sol, on se servira de la même carte deux fois, mais cette fois-ci, en format réduit. L'orientation des éléments de l'image correspondra à l'orientation des cases numérotées 1.

Sur la carte de gauche, la case numérotée 1 d'en haut montre le chiffre 1 tourné de 90 degrés vers la gauche. C'est la raison pour laquelle l'élément de l'image doit également être tourné de 90 degrés vers la gauche. Il s'ajoute que la rondeur du demi-cercle sera tournée vers le bas.

Sur la carte de gauche, la case numérotée 1 d'en bas montre le chiffre 1 tourné de 90 degrés vers la droite. C'est la raison pour laquelle l'élément de l'image doit également être tourné de 90 degrés vers la droite. La rondeur du demi-cercle sera tournée, cette fois-ci, vers le haut.

Toutes ces prémisses ne correspondent qu'à l'image B). C'est pourquoi B) est l'unique réponse possible.

C'est de l'informatique !

Dans le domaine de l'informatique, on nomme un programme informatique «récursif» quand les instructions s'appellent mutuellement afin que le programme puisse faire référence à lui-même et demander sa propre exécution au cours de son déroulement. Le mot «récursif» trouve son origine dans le terme latin *recurrere* (ce qui signifie «revenir en arrière», ici : «faire référence à soi-même»). Le concept récursif est très puissant. Il est possible d'accomplir certaines tâches complexes à l'aide d'un tel programme, et ceci de manière très efficace et intelligible. Pourtant, certains programmes récursifs ne sont pas aussi évidents et plutôt difficiles à comprendre.

Une instruction à fonction récursive comporte non seulement l'appel à la fonction récursive mais aussi une condition qui détermine le moment de l'interruption de cette fonction. Sinon il se peut que le programme récursif répète les instructions à l'infinie (ou jusqu'à l'épuisement d'une des ressources comme de la mémoire vive (RAM)).

Sites web et mots clés

programmer, programme récursif, conditions d'arrêt
— <https://fr.wikipedia.org/wiki/Récurtivité>



14. Le jeu de formes géométriques

Alicia joue à un jeu de formes géométriques. À chaque nouvelle partie de jeu, elle remplace les formes selon de nouvelles règles qu'elle s'est imposées. Elle commence toujours avec une seule forme. Voici un exemple d'un ensemble des règles de remplacement qu'elle va appliquer :



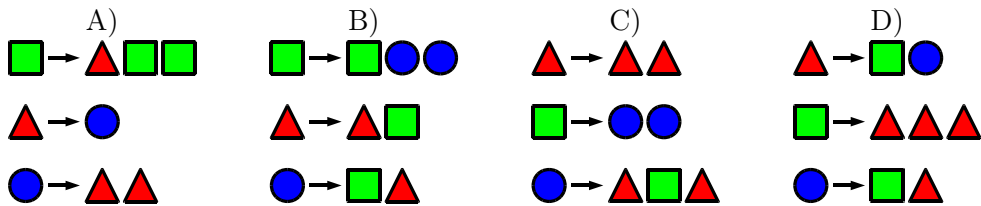
Si Alicia commence avec un carré et qu'elle respecte les deux règles ci-dessus, elle devrait procéder comme suit :



Dans la partie de jeu suivante, elle arrange les formes de cette manière :



Choisis l'ensemble des règles de remplacement qu'elle a appliqué pour arriver à l'arrangement des formes ci-dessus.





Solution

La réponse B) est correcte. Si Alicia commence avec un triangle, elle effectuera les remplacements suivants :



Les autres réponses ne sont pas correctes. Voici pourquoi :

Prenons la réponse A) : si Alicia commence avec un triangle ou un cercle, elle n'arrivera jamais à les remplacer par des carrés. Si, par contre, elle commence avec un carré, elle produira l'arrangement suivant :



Comme tous les remplacements qui s'ensuivent deviennent de plus en plus longs, cet ensemble de règles de remplacement n'aboutira pas au résultat souhaité.

Si elle applique l'ensemble des règles de remplacement C) et qu'elle commence avec un triangle, elle ne peut pas le remplacer par un carré ou un cercle. Si elle commence avec un carré, elle obtient l'arrangement suivant :



Malheureusement, elle n'arrivera jamais à remplacer les deux premiers triangles initiaux par un triangle et un carré, ce qui, pourtant, sera exigé. Voyons à quel arrangement elle arrive, si elle commence avec un cercle :






Alicia rencontrera le même problème irrésolu. Elle ne pourra jamais remplacer les deux triangles initiaux par un triangle et un carré.

Si elle applique l'ensemble des règles de remplacement D), Alicia n'arrivera pas à poser deux cercles l'un à côté de l'autre. Il s'ajoute qu'il ne sera même pas possible de reproduire plus qu'un seul cercle car chaque cercle reproduit sera tout de suite remplacé par un carré et un triangle. Cependant, pour arriver à l'arrangement souhaité, Alicia aurait besoin de deux cercles posés l'un à côté de l'autre.

C'est de l'informatique !

Les règles de ce jeu représentent dans le domaine de l'informatique théorique un système de règles avec le but de transformer des mots. Plus concrètement, on pourrait dire qu'il s'agit d'un système grammatical indépendant du contexte dans lequel on remplace un mot (dans notre exemple : un symbole géométrique) par un autre ou plusieurs mots (symboles géométriques).

Le système grammatical indépendant du contexte se compose :

- d'un ensemble de mots (dans notre exemple, il s'agit de symboles ,  et  ; en outre, on fait la distinction entre les mots que l'on remplacera plus tard et ceux avec lesquels on n'a pas le droit de les remplacer.
- d'un symbole initial (qui peut être choisi indépendamment de l'ensemble de mots).
- d'un ensemble de productions (dans notre exemple, il s'agit de l'ensemble des règles de remplacement).



Dans le système de la grammaire indépendante du contexte et contrairement à la démarche d'Alicia, on n'est pas contraint d'appliquer tous les ensembles de productions possibles en même temps. Généralement, on choisit le moment où l'on veut appliquer une des productions.

On met en pratique le système de la grammaire indépendante dans de différents domaines : pour des langages formels comme le langage de programmation ou pour la description des langages naturels ou, par exemple, pour décrire le processus de croissance des plantes.

Pour résoudre le problème, il faut donc, d'une part, comprendre les règles imposées et d'autre part, savoir analyser des chaînes de symboles. Dans le domaine de l'informatique, ce concept est nommé «parsing», c'est-à-dire un concept de l'analyse syntaxique. Le *parsing* s'applique pour traduire un programme d'ordinateur, écrit dans un langage de programmation, de manière que l'ordinateur puisse l'exécuter.

Sites web et mots clés

grammaire indépendante du contexte, analyse syntaxique

- https://fr.wikipedia.org/wiki/Grammaire_non_contextuelle
- <https://fr.wikipedia.org/wiki/L-Systeme>
- https://fr.wikipedia.org/wiki/Analyse_syntaxique

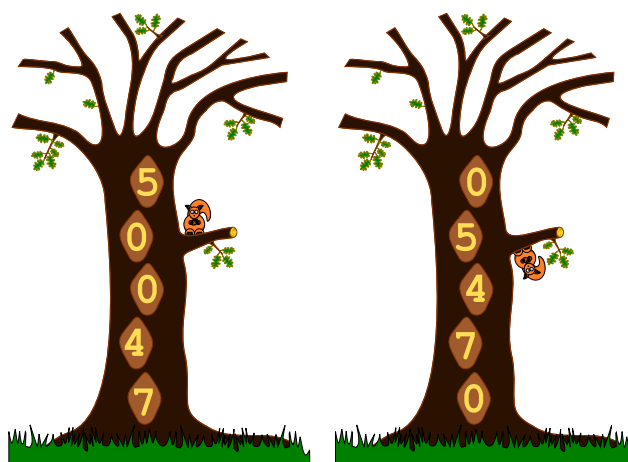




15. Les écureuils égoïstes

Les écureuils vivent dans les cavités des arbres. Un arbre a cinq cavités disposées les unes au-dessus des autres. Ces cavités sont peuplées de seize écureuils ce qui revient à dire que ces seize écureuils habitent ensemble les cinq cavités d'arbre.

Chaque matin, tous les écureuils vérifient les cavités pour savoir dans lesquelles se trouvait le nombre le plus restreint des écureuils la nuit précédente. Plus exactement, les écureuils contrôlent le nombre de voisins qui demeuraient au-dessus ou au-dessous de leur propre cavité. Pour la nuit suivante, chaque écureuil choisit en



clandestin la cavité avoisinée où il y avait le plus petit nombre d'écureuils pendant la nuit précédente. Au cas où chaque cavité hébergeait le même nombre d'écureuils, les écureuils préfèrent passer la nuit plutôt dans leur propre cavité que dans la cavité située au-dessus de cette-dernière, et s'ils ne passent pas la nuit dans leur propre cavité, ils préfèrent la cavité située au-dessus à celle située au-dessous de leur propre cavité.

Si, par exemple, pendant la nuit précédente se trouvaient dans les cavités de haut en bas 5, 0, 0, 4 et 7 écureuils, le jour suivant, la situation changera comme suit : Tous les cinq écureuils qui ont passé la nuit dans la cavité située tout en haut se déplaceront pour passer la nuit dans la cavité située directement au-dessous de leur propre cavité (car ils préfèrent avoir 0 à 4 voisins). Les 7 écureuils qui ont passé la nuit tout en bas de la séquence des cavités se déplaceront pour passer la nuit dans la cavité située directement au-dessus de leur propre cavité (car ils préfèrent avoir 4 à 6 voisins) et les 4 écureuils qui ont passé la nuit dans la cavité au-dessus de celle des 7 écureuils se déplaceront pour passer la nuit dans la cavité située directement au-dessus de leur propre cavité (car ils préfèrent avoir 0 à 3 voisins).

Ce matin, les cavités étaient peuplées comme suit : de haut en bas, il y avait 6, 3, 3, 0 et 4 écureuils. Combien de jours faudra-t-il pour que tous les écureuils se retrouvent de nouveau exactement dans les mêmes cavités qu'aujourd'hui ?

- A) Deux jours.
- B) Trois jours.
- C) Quatre jours.
- D) Pas de chance, ils ne se retrouveront plus jamais dans la même cavité.



Solution

La réponse correcte est : trois jours.

Aujourd'hui : 6, 3, 3, 0, 4
Le jour suivant : 0, 9, 0, 7, 0
Après 2 jours : 9, 0, 7, 0, 0
Après 3 jours : 0, 16, 0, 0, 0

C'est de l'informatique !

Le problème en question est un exemple de l'intelligence collective. L'idée de base de ces algorithmes est que l'on puisse résoudre des problèmes complexes avec des éléments très simples à condition que l'on en dispose de beaucoup. Les fourmis, par exemple, se comportent selon des règles simples et indépendamment les unes des autres. Cependant, quand il y en a beaucoup, elles sont capables d'accomplir des tâches exigeantes comme par exemple construire une fourmilière et y aménager des dômes, trouver le chemin le plus favorable afin d'atteindre une ressource ou couper des feuilles.

Dans la tâche présente, il y a également une grande «quantité» d'éléments (dans notre cas, elle est représentée par des écureuils) qui se comportent selon des règles simples. Malheureusement, leur comportement collectif est loin d'être perçu comme «intelligent». Ils revendiquent tous l'espace le plus grand possible, pourtant, à la fin, ils se retrouvent tous dans la même cavité. Il s'ensuit donc que l'on ne peut pas attribuer le comportement des fourmis à un algorithme de colonies de fourmis à 100% : parfois on s'aperçoit toutefois qu'il est mieux de coopérer au lieu de se comporter de manière égoïste.

Sites web et mots clés

intelligence collective, algorithme de colonies des fourmis

- https://fr.wikipedia.org/wiki/Intelligence_distribuée
- https://fr.wikipedia.org/wiki/Intelligence_collective
- https://fr.wikipedia.org/wiki/Algorithme_de_colonies_de_fourmis



A. Auteurs des exercices

 Nursultan Akhmetov
 Wilfried Baumann
 Andrea Brabcová
 Eugenio Bravo
 Nicolas Brunner
 Valentina Dagiené
 Christian Datzko
 Susanne Datzko
 Olivier Ens
 Hanspeter Erni
 Gerald Futschek
 Haris Gavranovic
 Martin Guggisberg

 Hans-Werner Hein
 Filiz Kalelioğlu
 Akiko Kikui
 Hiroki Manabe
 Khairul A. Mohamad Zaki
 Hamed Mohebbi
 Anna Morpurgo
 Tom Naughton
 Erkulan Nurtazanov
 Henry Ong
 Serena Pedrocchi
 Wolfgang Pohl
 Ilya Posov

 Sergei Pozdniakov
 J. P. Pretti
 Lorenzo Repetto
 Kirsten Schlüter
 Eljakim Schrijvers
 Seiichi Tani
 Willem van der Vegt
 Jiří Vaníček
 Troy Vasiga
 Lina Vinikienė
 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
0100000100101110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SS!E

www.svia-ssie-ssii.ch
schweizerischervereinfürinformatikind
erausbildung//sociétésuissepourl'infor
matique 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.