



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

## Exercices et solutions 2016 Années scolaires 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](http://www.svia-ssie-ssii.ch)  
schweizerischerverein für informatik  
erziehung und ausbildung // société suisse de l'inform  
atique dans l'enseignement // società sviz  
zera per l'informatica nell'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 Dagiene : 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

Tout lien a été vérifié le 1 novembre 2016. Ce cahier d'exercice a été produit le 13 novembre 2016 avec le logiciel de mise en page L<sup>A</sup>T<sub>E</sub>X.



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. 40.



# 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 de 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 scolaire 3 et 4) 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 scolaires 3 et 4 (Petit Castor)
- Années scolaires 5 et 6
- Années scolaires 7 et 8
- Années scolaires 9 et 10
- Années scolaires 11 à 13

Les élèves des années scolaires 3 et 4 avaient 9 exercices à résoudre (3 faciles, 3 moyens, 3 difficiles). Chaque autre tranche d'âge devait résoudre 15 exercices, 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](mailto:castor@castor-informatique.ch)

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



---

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



# Table des matières

<b>Ont collaboré au Castor Informatique 2016</b>	<b>i</b>
<b>Préambule</b>	<b>ii</b>
<b>1. Des messages secrets</b>	<b>1</b>
<b>2. Vive la hiérarchie !</b>	<b>3</b>
<b>3. Plongeons-nous dans la construction des ponts !</b>	<b>5</b>
<b>4. Des cônes servant de cachette</b>	<b>7</b>
<b>5. Le travail en groupe</b>	<b>9</b>
<b>6. Des bonbonnières</b>	<b>13</b>
<b>7. 60 minutes... Zut !</b>	<b>15</b>
<b>8. La poignée de main style zip</b>	<b>17</b>
<b>9. Les jeux de Nim</b>	<b>21</b>
<b>10. Le classement des numéros de maillot</b>	<b>23</b>
<b>11. Chaque palette en son temps !</b>	<b>25</b>
<b>12. Jeux de boule</b>	<b>29</b>
<b>13. Deux possibilités exclusives</b>	<b>33</b>
<b>14. Le carrefour des trams</b>	<b>35</b>
<b>15. Le codage des images numériques</b>	<b>37</b>
<b>A. Auteurs des exercices</b>	<b>40</b>
<b>B. Sponsoring : Concours 2016</b>	<b>41</b>
<b>C. Offres ultérieures</b>	<b>43</b>



# 1. Des messages secrets

Les agents Boris et Bertha communiquent à travers des messages secrets que personne n'est censé comprendre. Boris aimerait bien envoyer à Bertha le message secret suivant :

RENCONTRERBILLYA6H

Il écrit les lettres du texte successivement dans un tableau à quatre colonnes et cinq rangées. Il commence à gauche en haut et continue, case après case, rangée après rangée jusqu'à ce qu'il arrive en bas. Si jamais il reste à la fin quelques champs libres, il insère un astérisque par champ libre. Voici le résultat :

R	E	N	C
O	N	T	R
E	R	B	I
L	L	Y	A
6	H	*	*

Ensuite, il crée le message secret. Sur une feuille vierge, il réécrit les lettres du tableau ci-dessus de haut en bas, case après case, colonne après colonne, de gauche à droite.

ROEL6ENRLHNTBY\*CRIA\*

Bertha reprend cette méthode pour créer sa réponse. Elle lui envoie le message secret suivant :

OVD AIKIRU\*JEAS\*ENIS\*

*Choisis le message secret que Bertha a envoyé à Boris.*

- A) OKJEVERR AISIJEPEUX
- B) OKJEVIENDRAIAUSSI
- C) OKJESERAILAAUSSI
- D) OKJENEPEUXPASVENIR



## Solution

Voilà comment on peut reconstruire le texte original : on insère à nouveau le message secret reçu dans un tableau à quatre colonnes et cinq rangées, en commençant à gauche, tout en haut, pourtant cette fois-ci, on continue de haut en bas, colonne par colonne.

O	K	J	E
V	I	E	N
D	R	A	I
A	U	S	S
I	*	*	*

Quand on lit le texte de gauche à droite, rangée par rangée on obtient le message suivant :

OKJEVIENDRAIAUSSI

Les astérisques à la fin du message ne font plus partie du message.

## C'est de l'informatique !

Les messages que nous transmettons à travers un réseau informatique peuvent être facilement interceptés. Quand ces messages contiennent des données sensibles comme des mots de passe ou des informations privées, nous préférons qu'il n'y ait que le destinataire qui puisse lire le texte. Dans ce cas-là, on peut coder le message (le texte en langage clair) afin qu'il soit transformé en un message secret. Seul le destinataire saura comment déchiffrer ce message pour obtenir le message original.

Dans la pratique, il existe de différentes méthodes de cryptage. La méthode que l'on vient d'utiliser dans cette tâche est appelée transposition et on l'applique depuis environ 2'400 ans. En appliquant cette méthode, on maintient toutes les lettres du texte clair. On ne change que leur ordre. Cette méthode est malheureusement facile à déchiffrer, en fait, il ne s'agit même pas d'une véritable méthode de cryptage. On parlerait plutôt d'un camouflage de l'information.

La cryptographie est une technique d'écriture en langage chiffré ou codé. En tant que science, elle étudie les méthodes et technologies de cryptage et représente un domaine important de l'informatique. Toutes les entreprises commerciales ainsi que les banques qui sont présentes sur Internet dépendent des systèmes de chiffrement hautement sécurisés. Les chiffrements modernes sont exécutés par ordinateur et se basent sur des méthodes mathématiques qui rendent pratiquement impossible de déchiffrer un message si on ne connaît pas la clé.

## Sites web et mots clés

chiffrement, cryptographie, transposition, protection des données

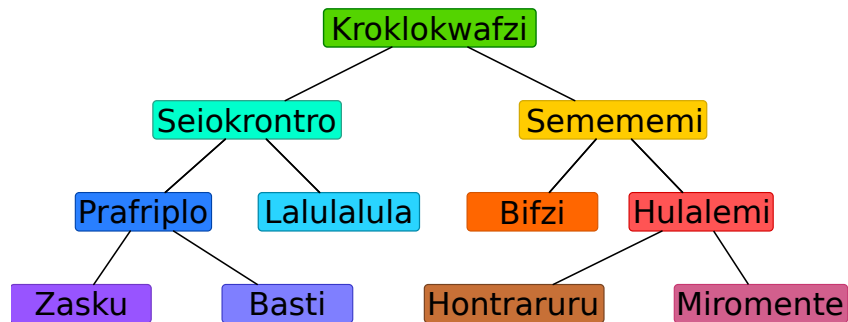
— [https://fr.wikipedia.org/wiki/Chiffrement\\_par\\_transposition](https://fr.wikipedia.org/wiki/Chiffrement_par_transposition)





## 2. 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 «Seiokronro», 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 Seiokronro.
- B) Certains Hontraruru ne font pas partie des Semememi.
- C) Tous les Zasku font également partie des Bifzi.
- D) Tous les Prafriplo 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](https://fr.wikipedia.org/wiki/Arbre_phylogénétique)

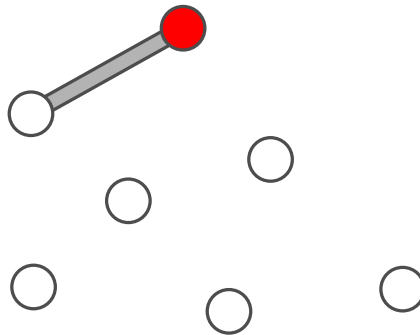
### 3. Plongeons-nous dans la construction des ponts !

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.*

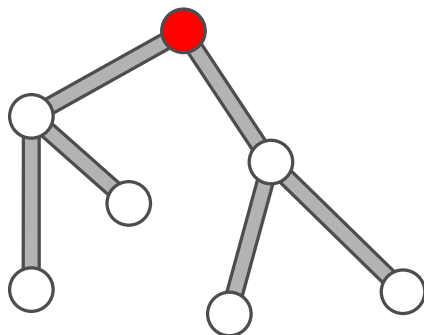




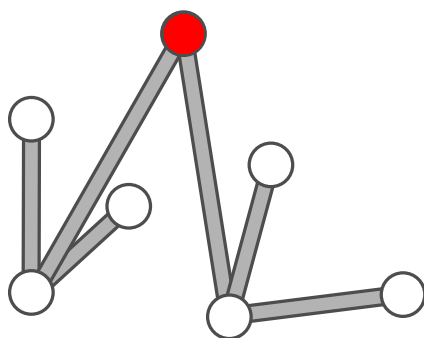
## Solution

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

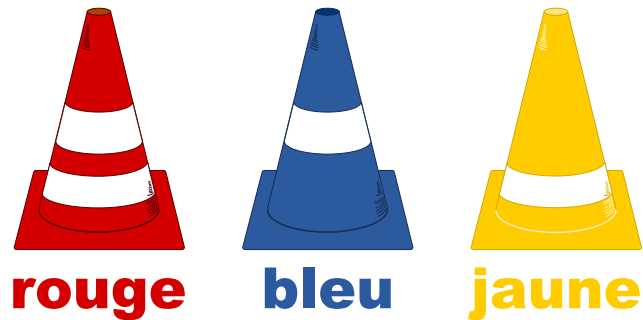
arbre, huttes

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



## 4. Des cônes servant de cachette

Vreni cache des cartes à jouer sous trois cônes de circulation. Elle dépose toujours une carte sous un cône particulier. S'il y a déjà une carte, elle met la nouvelle carte au-dessus de la précédente.



Pour mémoriser laquelle des cartes à jouer se trouve sous quel cône de circulation, elle note comme suit :

rouge + 5

rouge + 3



Vreni a noté :

rouge + 3

jaune + 5

rouge + 6

jaune + 8

bleu + 1

jaune + 3

*Selon les notes de Vreni, quelle carte à jouer se trouvera tout en haut de la pile sous le cône de circulation correspondant ?*

- A) rouge : 3, bleu : 1, jaune : 5
- B) rouge : 9, bleu : 1, jaune : 16
- C) rouge : 6, bleu : 1, jaune : 3
- D) rouge : 8, bleu : 1, jaune : 3



## Solution

Au début, il n'y a aucun cône de circulation qui cache une carte. Le tableau ci-dessous montre quelles cartes se trouvent sous quel cône tout en haut de la pile après chaque étape.

Note	rouge	bleu	jaune
	-	-	-
rouge ← 5	5	-	-
jaune ← 5	5	-	5
rouge ← 6	6	-	5
jaune ← 8	6	-	8
bleu ← 1	6	1	8
jaune ← 3	6	1	3

Ainsi, il est évident que la réponse C) est juste.

## C'est de l'informatique !

Un cône peut être interprété comme une variable dont le nom est «rouge», «bleu» ou «jaune». En informatique, une variable est un espace de stockage où l'ordinateur peut sauvegarder des valeurs comme dans notre exemple des nombres entiers. Chaque commande est en effet une affectation (informatique). Cela signifie que la valeur de la variable utilisée auparavant sera substituée par une nouvelle valeur.

Dans un langage de programmation, on note les affectations comme suit : «=» ou «:=». Souvent, ces annotations peuvent être sources de raisonnements erronés : pendant qu'en mathématiques l'équation « $x = x + 1$ » est simplement fausse, en informatique, cette annotation signifie que la valeur de la variable  $x$  est augmentée de la valeur 1 et que le résultat sera sauvegardé en tant qu'une nouvelle valeur de la variable  $x$  égal à « $x + 1$ ». En dehors des langages de programmation et pour mieux différencier les affectations, on a recours à la flèche «←». Ainsi, l'annotation en question sera immédiatement comprise et par les mathématiciens et par les informaticiens.

## Sites web et mots clés

variable, affectation (informatique)

- [https://fr.wikipedia.org/wiki/Variable\\_\(informatique\)](https://fr.wikipedia.org/wiki/Variable_(informatique))
- [https://fr.wikipedia.org/wiki/Affectation\\_\(informatique\)](https://fr.wikipedia.org/wiki/Affectation_(informatique))

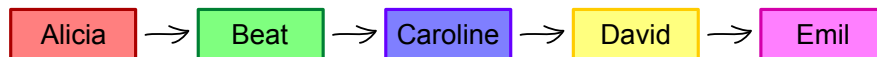


## 5. Le travail en groupe

Sarah et ses camarades de classe, Alicia, Beat, Caroline, David et Emil doivent faire un travail en groupe. Ils répartissent les tâches et le groupe confie à Sarah la tâche de rassembler les résultats de tous les membres du groupe. Elle n'a aucun problème à obtenir les résultats d'Emil car il peut les lui donner immédiatement. Pour obtenir les résultats des autres camarades, par contre, elle doit respecter les conditions suivantes :

- Pour obtenir les résultats de David, Sarah doit lui présenter d'abord les résultats d'Alicia.
- Pour obtenir les résultats de Beat, elle doit lui présenter d'abord les résultats d'Emil.
- Pour obtenir les résultats de Caroline, elle doit lui présenter d'abord les résultats de Beat et de David.
- Pour obtenir les résultats d'Alicia, elle doit lui présenter d'abord les résultats de Beat et d'Emil.

*Déplace les noms afin de les classer dans un ordre logique qui permettra à Sarah d'obtenir les résultats de tous les membres du groupe.*





## Solution

La réponse correcte est :

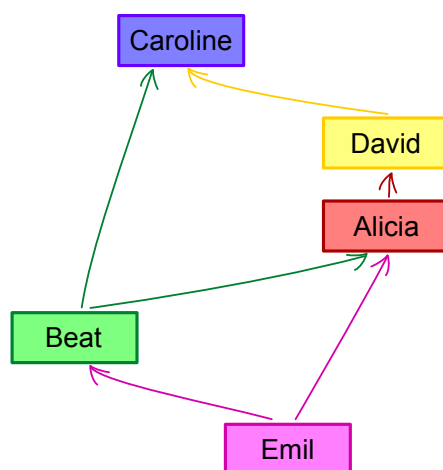


Emil est le seul qui ne dépend pas des résultats des autres. C'est la raison pour laquelle Sarah doit l'interroger en premier. Beat est le seul qui a besoin des résultats d'Emil afin de pouvoir terminer son travail. Pour qu'Alicia puisse rendre ses résultats, elle doit tenir compte des résultats d'Emil et de Beat. David ne peut terminer son travail que s'il jette un œil sur les résultats d'Emil, de Beat et d'Alicia (bien qu'en vérité, il ne s'intéresse qu'aux résultats d'Alicia). Finalement, il y a Caroline qui montrera volontiers ses résultats après avoir vu les résultats de Beat et de David.

## C'est de l'informatique !

Dans la vie quotidienne, il arrive souvent que l'on ne puisse terminer un travail avant que certaines conditions préalables soient remplies. L'informatique s'intéresse également aux opérations de planification des tâches («scheduling»). La mise en œuvre de tels programmes est très fréquente dans le domaine de l'industrie. Ainsi, il est possible de réaliser des opérations complexes comme par exemple le montage d'un véhicule le plus efficacement possible (sans, par exemple, devoir attendre trop longtemps pour pouvoir monter les pièces nécessaires). En optimisant les étapes différentes au sein d'une entreprise, on économise beaucoup d'argent car on n'aura par exemple plus besoin d'un entrepôt et les machines seront plus performantes.

Pour modeler une planification, on utilise des graphes dont les nœuds représentent les différentes opérations (dans le cas présent, l'action de demander les résultats auprès des camarades) ainsi que les arcs, ici, sous forme de flèches. Le fait d'avoir recours à des flèches signifie que les arcs sont orientés. Il s'ensuit qu'un tel graphe est appelé « graphe orienté ». Le graphe qui correspond à notre problème est représenté comme suit :



Une solution possible d'un plan de déroulement est un parcours à travers le graphe qui passe par tous les nœuds. Dans l'exemple de notre graphe, il n'y a qu'un seul parcours valable à savoir la réponse correcte à la question qui est : Emil → Beat → Alicia → David → Caroline.

Comme expliqué plus haut, la recherche d'une solution n'est qu'une partie de la solution du problème. Il existe bien d'autres situations complexes dans lesquelles les opérations se déroulent parallèlement. Il est en outre souvent important de trouver une solution qui permette de maintenir les coûts aussi bas que possible ou qui rendent l'exécution d'une opération particulièrement rapide. Dans de maintes





situations, il est préférable d'appliquer des algorithmes heuristiques (algorithme d'approximation) pour pouvoir s'approcher d'une solution optimale au lieu de simplement tester toutes les solutions possibles.

## Sites web et mots clés

planification des tâches, plan de déroulement, graphe orienté

— [https://fr.wikipedia.org/wiki/Ordonnancement\\_de\\_travaux\\_informatiques](https://fr.wikipedia.org/wiki/Ordonnancement_de_travaux_informatiques)



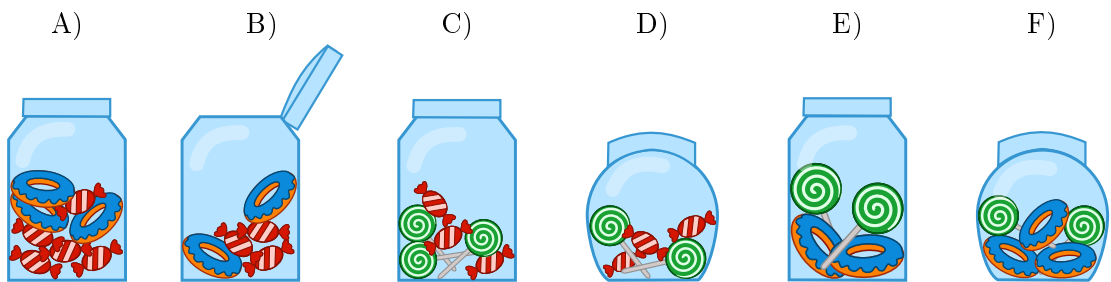


## 6. Des bonbonnières

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

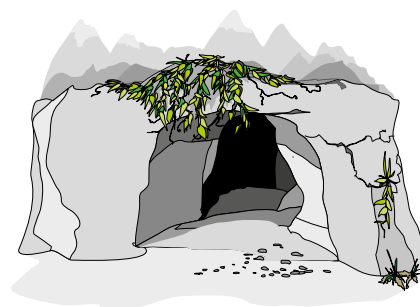
orientation objet, attribut, logique

- [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/Théorie_des_ensembles)
- [https://fr.wikipedia.org/wiki/Base\\_de\\_données](https://fr.wikipedia.org/wiki/Base_de_données)



## 7. 60 minutes... Zut!

Anna et Benno font une randonnée avec leurs parents. En route, il faut qu'ils passent par un tunnel. L'expérience leur a appris que chaque membre du groupe a besoin d'un laps de temps différent pour le traverser : Anna a besoin de 10 minutes tandis que Benno ne met que 5 minutes. La mère, elle, a besoin de 20 minutes et le père a besoin de 25 minutes.


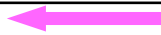
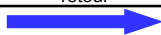
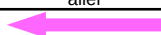
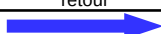


Le tunnel est sombre et très étroit, on doit le traverser soit seul soit à deux. Donc, pour que chaque membre du groupe arrive à l'autre bout du tunnel, ils doivent organiser leurs passages. Si deux personnes passent par le tunnel ensemble, le temps dont ils ont besoin correspondra au temps dont la personne la plus lente des deux aurait besoin pour le traverser seule. En plus, dans le tunnel, il faut que l'on se serve d'une lampe de poche.

Au moment où ils arrivent à l'entrée du tunnel, ils réalisent que la charge de la batterie de la lampe ne permet de l'utiliser que pendant 60 minutes. D'après toi, est-il possible que tous les membres du groupe arrivent à passer le tunnel en 60 minutes ?

Anna prétend que oui : «C'est tout à fait possible», et elle ajoute : «Pour passer par le tunnel, il nous faut cinq passages!»

*Déplace les noms dans les champs correspondants pour que le plan d'Anna puisse être mis en œuvre.*

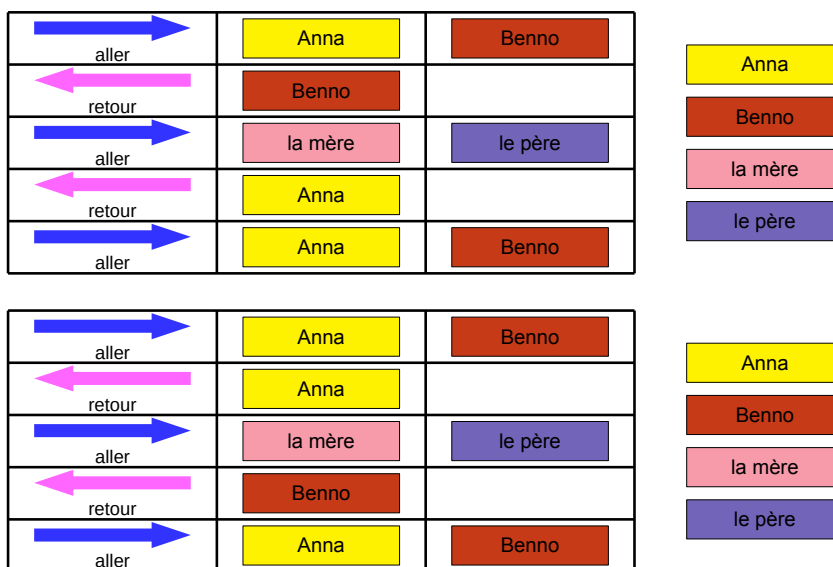
 aller		
 retour		
 aller		
 retour		
 aller		

Anna
Benno
la mère
le père



## Solution

Pour que tous puissent passer par le tunnel en 60 minutes au maximum, il faut que la mère et le père, soit les deux membres du groupe les plus lents, ne le traversent qu'une seule fois. Ceci n'est possible que si Anna et Benno traversent le tunnel en premier (passage 1) et que l'un des deux retourne avec la lampe de poche (passage 2). Ainsi, celui-ci peut la rendre à ses parents qui passeront ensuite par le tunnel à deux (passage 3). Celui qui les attend à la sortie du tunnel reprend la lampe et retourne une dernière fois (passage 4) pour effectuer ensuite le passage final à deux. Peu importe si c'est Anna ou Benno qui effectueront le passage 2, le passage 4 sera effectué par l'un ou par l'autre, ce qui reviendra à un laps de temps de 15 minutes au maximum.



## C'est de l'informatique !

Dans le domaine de l'informatique, on a souvent très peu de *ressources* pour résoudre des problèmes. En même temps, la solution d'un problème doit également répondre à certaines *contraintes secondaires*. Ainsi, pour résoudre un problème, il faut qu'on établisse un plan qui ne dépassera pas les ressources et qui répondra à toutes les contraintes secondaires.

Pour faire ceci, on prend souvent recours aux algorithmes d'*ordonnancement (scheduling)*. Comme les problèmes apparus dans la pratique se révèlent souvent très compliqués et que l'ordinateur aurait besoin de beaucoup de temps (voire probablement de plusieurs années) afin de trouver la solution la plus favorable, on se contente d'en trouver une qui mènera approximativement au résultat souhaité. Dans notre exemple, une telle solution est tout à fait convenable : on ne demande pas de trouver une solution pour que les membres du groupe puissent traverser le tunnel le plus vite possible. L'objectif est de ne pas dépasser les 60 minutes, c'est-à-dire, une solution qui répond à la contrainte qu'on ne puisse pas passer par le tunnel sans lampe. Quand on a trouvé la solution qui répond convenablement à cette contrainte, on a résolu le problème.

## Sites web et mots clés

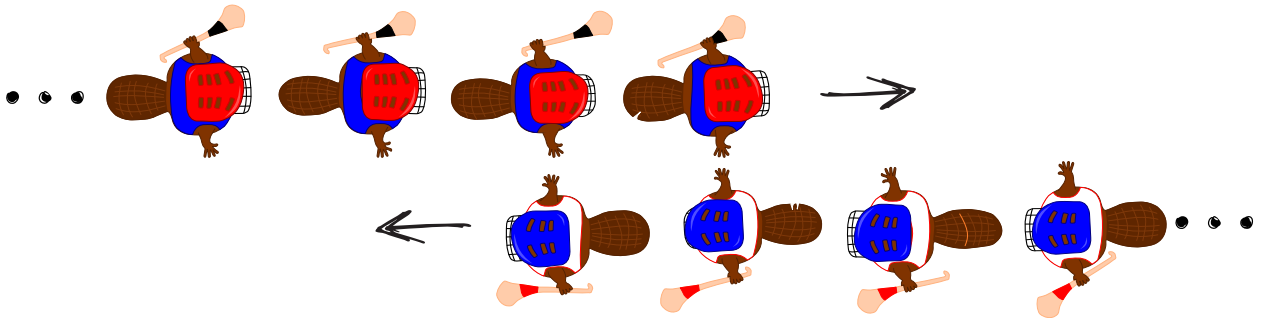
ordonnancement, contraintes secondaires, ressources

— [https://fr.wikipedia.org/wiki/Ordonnancement\\_de\\_tâches\\_informatiques](https://fr.wikipedia.org/wiki/Ordonnancement_de_tâches_informatiques)



## 8. 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.

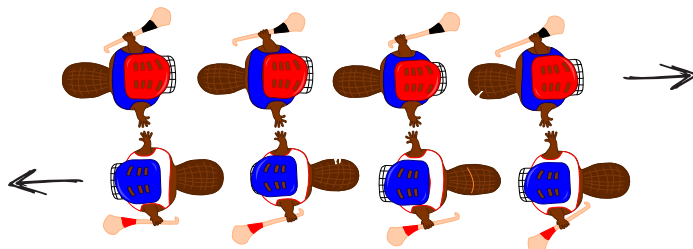
*Combien de secondes faut-il au total pour que les deux équipes aient terminé le rituel de la poignée de main style zip ?*



## 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

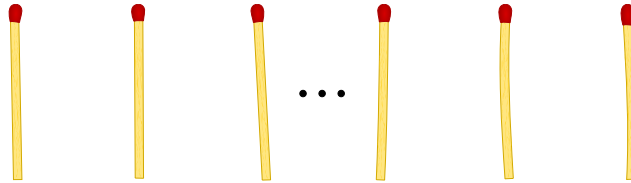
— [https://fr.wikipedia.org/wiki/Comparaison\\_asymptotique](https://fr.wikipedia.org/wiki/Comparaison_asymptotique)





## 9. 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 en 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/>

## 10. 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

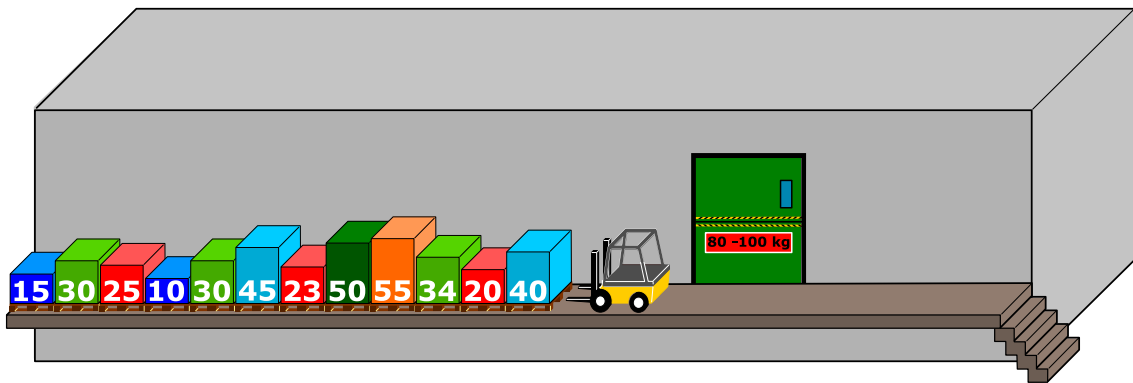
— [https://fr.wikipedia.org/wiki/Recherche\\_dichotomique](https://fr.wikipedia.org/wiki/Recherche_dichotomique)



## 11. 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 ?*

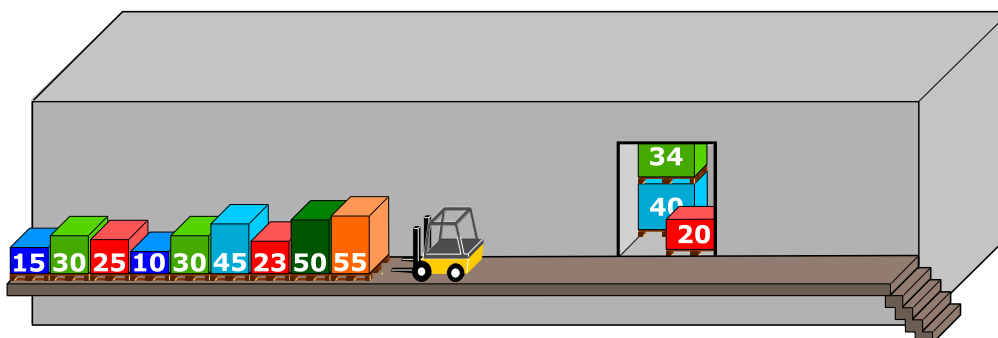
- A) La deuxième charge de l'ascenseur s'élève à 98 kg.
- B) On n'utilise pas les palettes de l'autre bout de la plate-forme.
- C) Une charge du monte-charge s'élève à 100 kg.
- D) Le monte-charge descend et monte cinq fois au total.
- E) La méthode pour déplacer les palettes décrite ci-dessus ne permet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



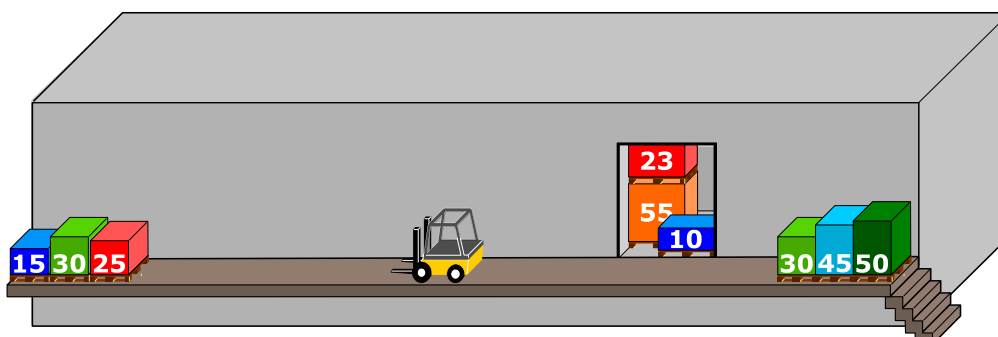
## Solution

La réponse correcte est C) :

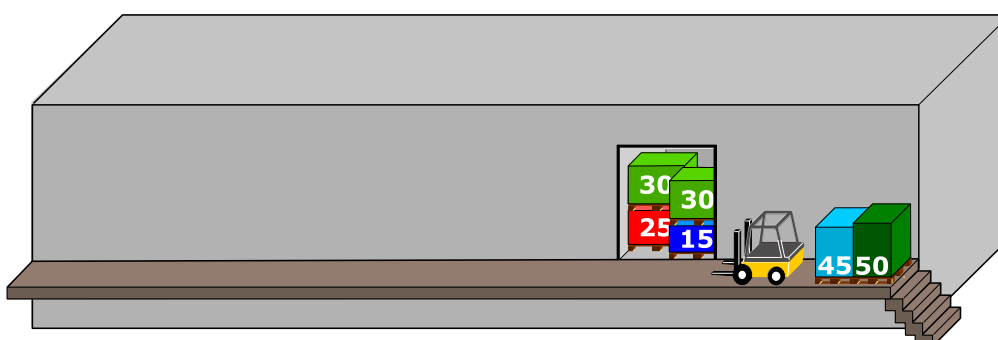
- La première charge comporte les trois premières palettes. Leur poids s'élève à :  $40 \text{ kg} + 20 \text{ kg} + 34 \text{ kg} = 94 \text{ kg}$ .



- 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.

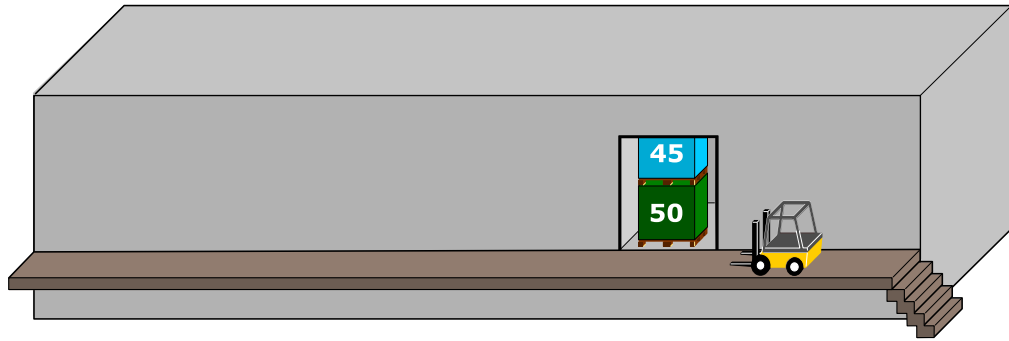


- Finalement, il ne restent que les deux palettes du côté gauche de la plate-forme :





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\\_\(structure\\_de\\_donn%C3%A9es\)](https://fr.wikipedia.org/wiki/File_(structure_de_donn%C3%A9es))
- [https://fr.wikipedia.org/wiki/Last\\_in,\\_first\\_out](https://fr.wikipedia.org/wiki/Last_in,_first_out)





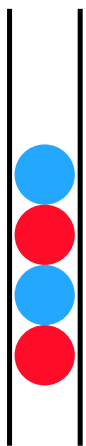
## 12. Jeux de boule

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 :

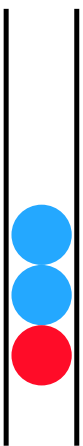
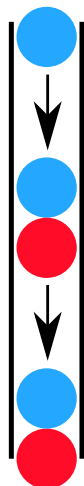
Si la boule qui se trouve en bas de la pile est rouge, une boule bleue tombe d’en haut et rejoint les boules empilées dans le tube :

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 :

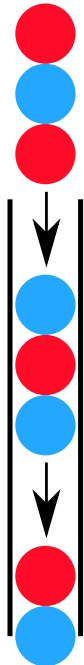
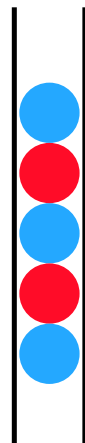
avant



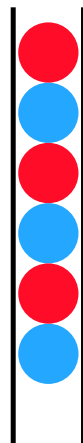
après



avant



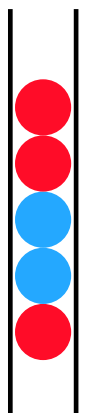
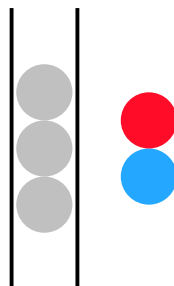
après



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.*

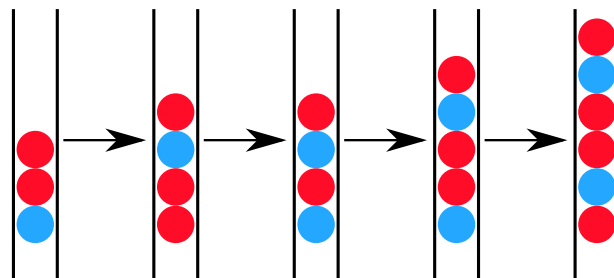
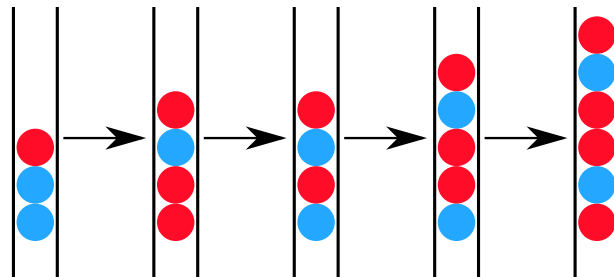
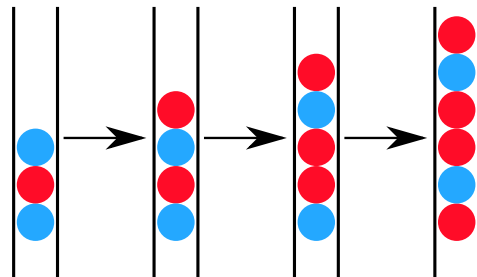
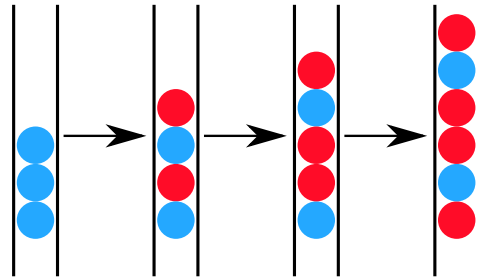




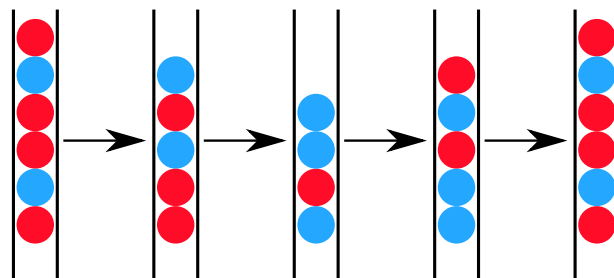
## Solution

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) :

$$\begin{aligned}Xbb &\rightarrow rbrX \\Xrb &\rightarrow rbrX \\Xbr &\rightarrow bX \\Xrr &\rightarrow bX\end{aligned}$$

## 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](http://esolangs.org/wiki/Post_canonical_system)
- [https://en.wikipedia.org/wiki/Tag\\_system](https://en.wikipedia.org/wiki/Tag_system)
- [https://en.wikipedia.org/wiki/Post\\_canonical\\_system](https://en.wikipedia.org/wiki/Post_canonical_system)



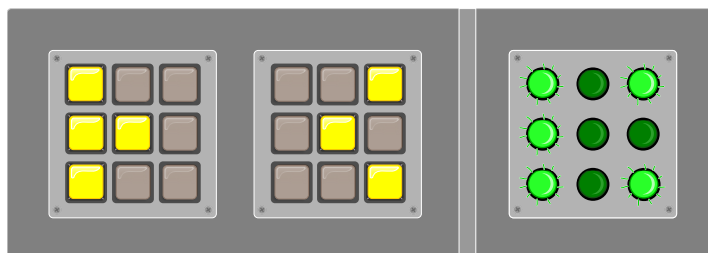


## 13. Deux possibilités exclusives

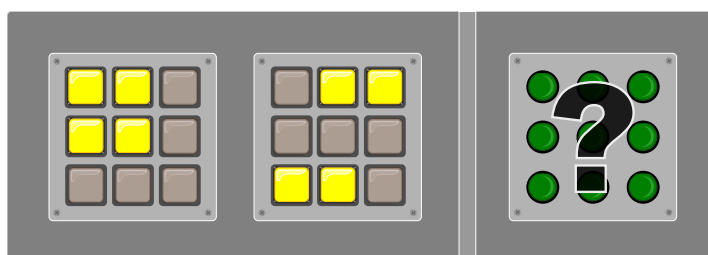
Erich a trouvé un vieil appareil électronique. Sur le côté gauche, il y a deux champs et chacun dispose de 9 touches sur lesquelles on peut appuyer. Sur le côté droit, il y a un champ avec 9 lampes. En fonction de la combinaison des touches, les lampes s'allument ou s'éteignent.

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.

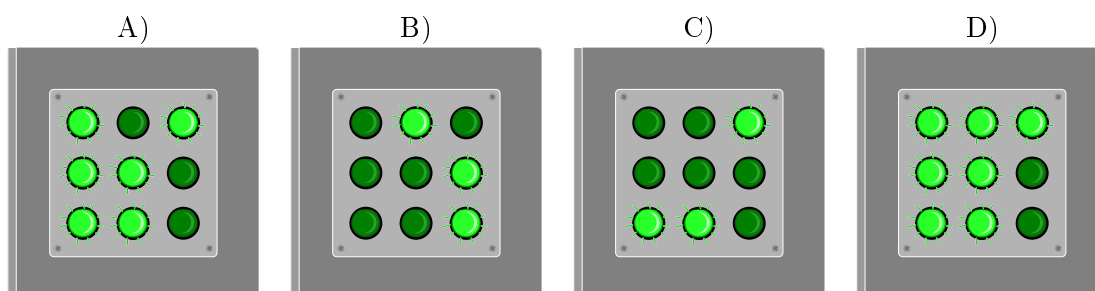
Actuellement, les lampes sont allumées comme suit :



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



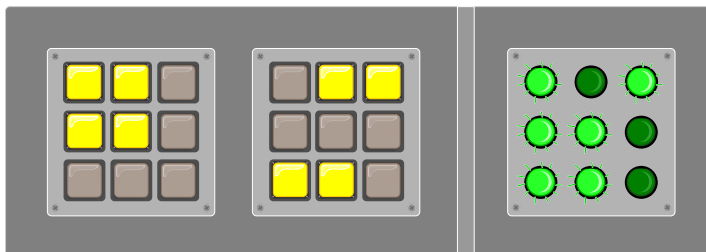
*Lesquelles des lampes sont allumées ?*





## 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

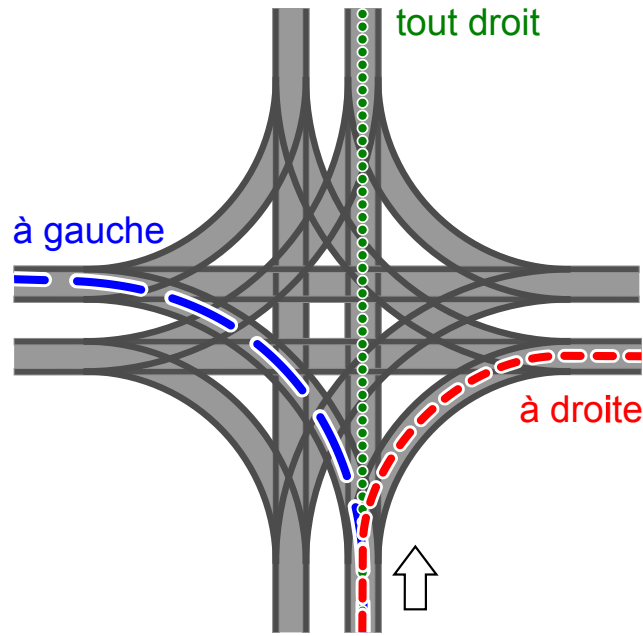
- [https://fr.wikipedia.org/wiki/Fonction\\_0U\\_exclusif](https://fr.wikipedia.org/wiki/Fonction_0U_exclusif)
- [https://en.wikipedia.org/wiki/XOR\\_gate](https://en.wikipedia.org/wiki/XOR_gate)





## 14. 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/>



## 15. Le codage des images numériques

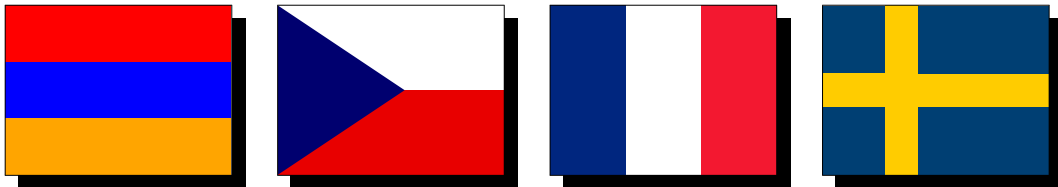
Les images numériques sont constituées d'un ensemble de points appelés « pixels ». Quand on veut enregistrer des images numériques en tant que données, dans le cas le plus simple, on décrit la couleur de chaque pixel individuellement. Le format de fichier GIW (imaginaire) permet de compresser les images 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.

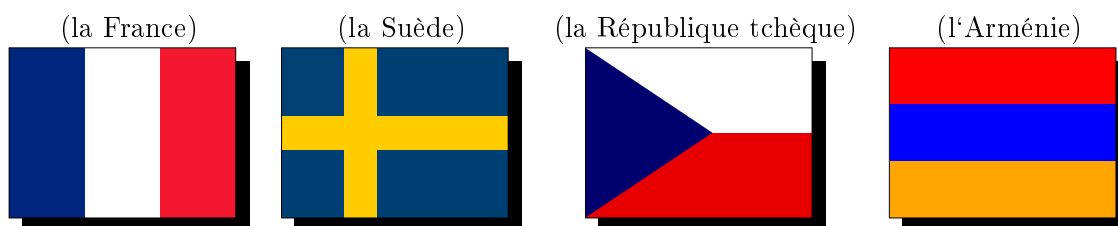
*Classe les images selon la taille de leur fichier GIW!*





## 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](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://www.youtube.com/watch?v=ypdNscvym_E)
- [https://fr.wikipedia.org/wiki/Run-length\\_encoding](https://fr.wikipedia.org/wiki/Run-length_encoding)



## A. Auteurs des exercices

 Guðjón Karl Arnarson  
 Wilfried Baumann  
 Andrea Brabcová  
 Eugenio Bravo  
 Nicolas Brunner  
 Christian Datzko  
 Susanne Datzko  
 Olivier Ens  
 Jürgen Frühwirth  
 Haris Gavranovic  
 Martin Guggisberg  
 Urs Hauser  
 Hans-Werner Hein

 Mathias Hiron  
 Juraj Hromkovič  
 Akiko Kikui  
 Tobias Kohn  
 Ivana Kosírová  
 Greg Lee  
 Hiroki Manabe  
 Tom Naughton  
 Erkulan Nurtazanov  
 Henry Ong  
 Serena Pedrocchi  
 Wolfgang Pohl  
 Ilya Posov


 Sergei Pozdniakov  
 Dániel Pressing  
 Lorenzo Repetto  
 Kirsten Schlüter  
 Eljakim Schrijvers  
 Sue Sentance  
 Björn Steffen  
 Seiichi Tani  
 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](http://www.svia-ssie-ssii.ch)  
schweizerischerverein für informatik und  
erausbildung // société suisse de l'inform  
atiquedans 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.