



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

## Exercices et solutions 2017 Années scolaires 7/8

<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 in d  
erausbildung // société suisse de l'inform  
atique dans l'enseignement // società sviz  
zera per l'informatica nell'insegnamento





# Ont collaboré au Castor Informatique 2017

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

Nous adressons nos remerciements à :

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

Valentina Dagiene : Bebras.org

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

Anna Morpurgo, Violetta Lonati, Mattia Monga : Italie

Gerald Futschek, Wilfried Baumann : Austrian Computer Society, Austria

Zsuzsa Pluhár : ELTE Informatikai Kar, Hongrie

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

Roman Hartmann : hartmannGestaltung (Flyer Castor Informatique Suisse)

Christoph Frei : Chragokyberneticks (Logo Castor Informatique Suisse)

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

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

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

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



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

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

## HASLERSTIFTUNG

Tout lien a été vérifié le 1 novembre 2017. Ce cahier d'exercice a été produit le 18 novembre 2017 avec avec le logiciel de mise en page L<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 pour l'Informatique dans l'Enseignement, et soutenu par la Fondation Hasler dans le cadre du programme d'encouragement «FIT in IT».

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

Le concours a été organisé pour la première fois en Suisse en 2010. Le Petit Castor (années scolaires 3 et 4) a été organisé pour la première fois en 2012.

Le Castor Informatique vise à motiver les élèves à apprendre l'informatique. Il souhaite lever les réticences et susciter l'intérêt quant à l'enseignement de l'informatique à l'école. Le concours ne suppose aucun prérequis quant à l'utilisation des ordinateurs, sauf de savoir naviguer sur Internet, car le concours s'effectue en ligne. Pour répondre, il faut structurer sa pensée, faire preuve de logique mais aussi de fantaisie. Les exercices sont expressément conçus pour développer un intérêt durable pour l'informatique, au-delà de la durée du concours.

Le concours Castor Informatique 2017 a été fait pour cinq tranches d'âge, basées sur les années scolaires :

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

Les élèves des années scolaires 3 et 4 avaient 9 exercices à résoudre (3 faciles, 3 moyens, 3 difficiles). Chaque autre tranche d'âge devait résoudre 15 exercices (5 faciles, 5 moyens et 5 difficiles).

Chaque réponse correcte donnait des points, chaque réponse fausse réduisait le total des points. Ne pas répondre à une question n'avait aucune incidence sur le nombre de points. Le nombre de points de chaque exercice était fixé en fonction de son degré de difficulté :

	Facile	Moyen	Difficile
Réponse correcte	6 points	9 points	12 points
Réponse fausse	-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

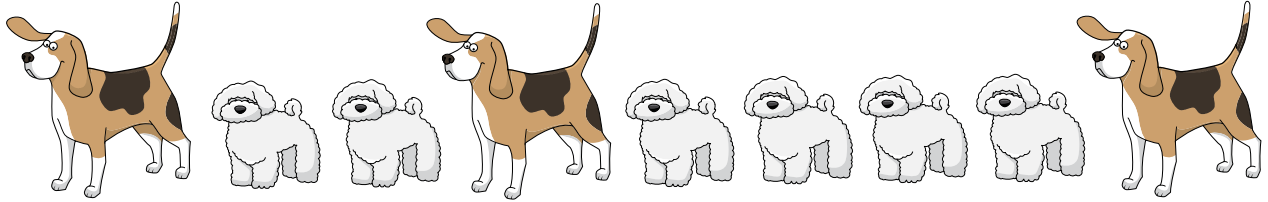
<b>Ont collaboré au Castor Informatique 2017</b>	<b>i</b>
<b>Préambule</b>	<b>ii</b>
<b>1. Déplacer des chiens</b>	<b>1</b>
<b>2. Le nom japonais</b>	<b>3</b>
<b>3. Un programme court</b>	<b>5</b>
<b>4. Orner des pendentifs médiévales</b>	<b>7</b>
<b>5. Le journal scolaire</b>	<b>9</b>
<b>6. Honomakato</b>	<b>13</b>
<b>7. Un art martial japonais</b>	<b>17</b>
<b>8. Pizzeria Castoria</b>	<b>19</b>
<b>9. Une commande chiffrée</b>	<b>23</b>
<b>10. Jeu des pièces</b>	<b>25</b>
<b>11. Bar à jus de fruits</b>	<b>27</b>
<b>12. Intrusion au musée</b>	<b>31</b>
<b>13. Des jeux de lumière</b>	<b>33</b>
<b>14. Substitutions</b>	<b>35</b>
<b>15. Jeu de billes</b>	<b>37</b>
<b>A. Auteurs des exercices</b>	<b>40</b>
<b>B. Sponsoring : Concours 2017</b>	<b>41</b>
<b>C. Offres ultérieures</b>	<b>43</b>



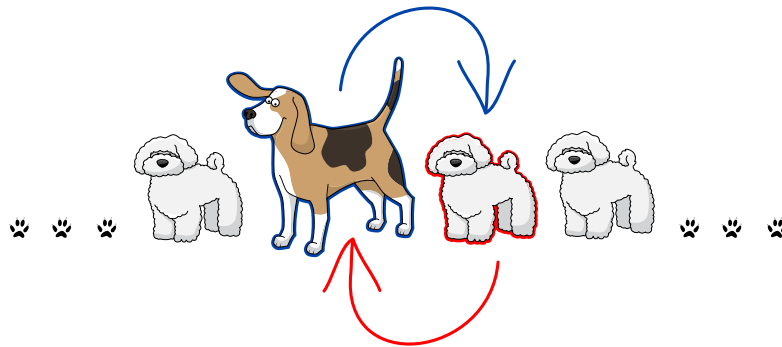


# 1. Déplacer des chiens

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



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



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

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

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



## Solution

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

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

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

## C'est de l'informatique !

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

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

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

## Sites web et mots clés

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

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





## 2. Le nom japonais

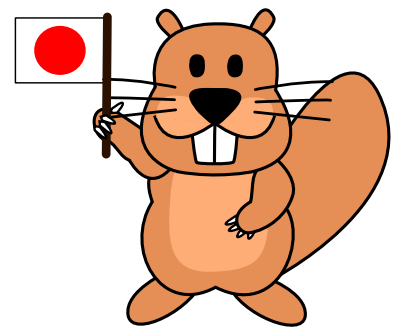
Une amie japonaise nous raconte que, selon une ancienne tradition, on peut transformer les lettres d'un prénom en un nom spirituel japonais. Il faut simplement remplacer chaque lettre par une syllabe qui lui est attribuée :

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

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

*Quel est le vrai nom de son ami croate ?*

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





## Solution

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

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

## C'est de l'informatique !

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

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

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

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

## Sites web et mots clés

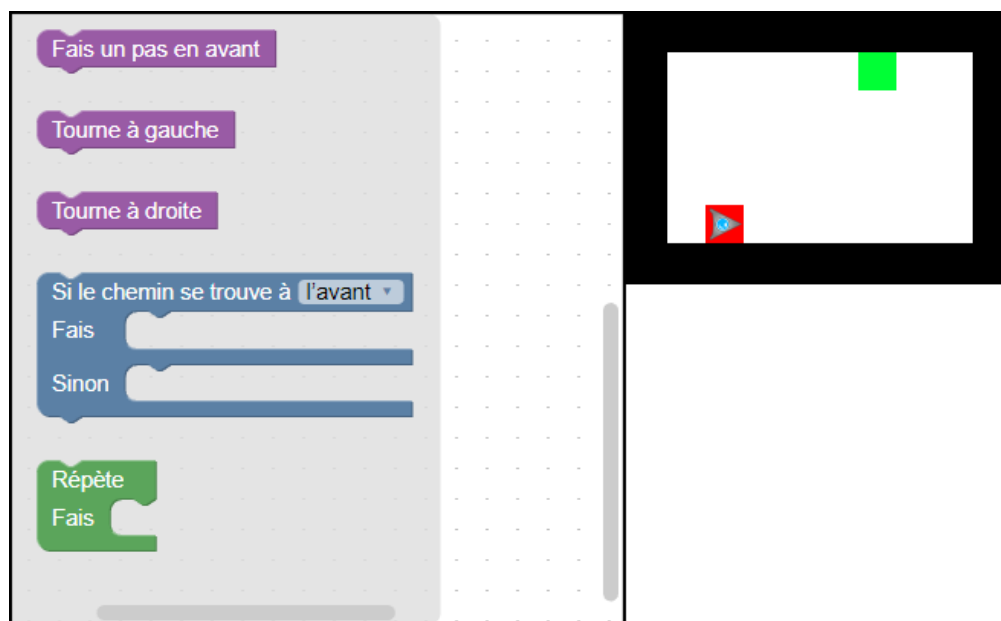
langage congruentiel, système de réécriture, système de semi-Thue, chiffrement monoalphabétique.

— [https://fr.wikipedia.org/wiki/Langage\\_congruentiel](https://fr.wikipedia.org/wiki/Langage_congruentiel)



### 3. Un programme court

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

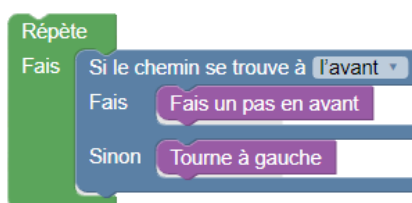


*Pour créer un tel programme, déplace les instructions et insère-les dans l'organigramme de programmation afin qu'elles suivent un ordre logique. Seule règle : tu ne dois pas utiliser plus de 4 instructions.*



## Solution

Voilà une solution possible :



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

## C'est de l'informatique !

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

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

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

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

## Sites web et mots clés

programmer, boucle, condition, instruction

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

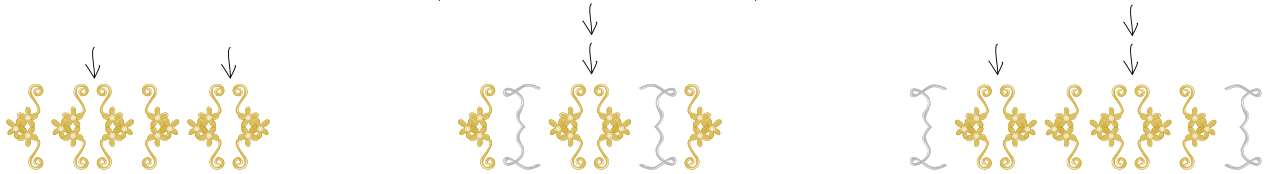


## 4. Orner des pendentifs médiévaux

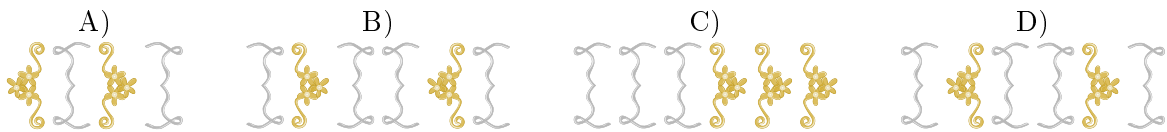
Pour une fête médiévale, la famille Castor produit des bijoux, plus précisément des pendentifs. La décoration de ces derniers consiste à placer des ornements qui ressemblent à des accolades. Chaque accolade est utilisée par paires. Pour décorer le bijou, les Castor appliquent une méthode particulière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



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



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





## Solution

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

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

## C'est de l'informatique !

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

## Sites web et mots clés

bien formée, syntaxe, sémantique

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



## 5. Le journal scolaire

L'équipe de rédaction du journal scolaire comprend 10 volontaires. Tous les vendredis, ils travaillent pendant leurs heures creuses pour rédiger des articles ou des reportages. L'emploi du temps ci-dessous nous montre les heures creuses individuelles des volontaires (en vert) pendant lesquelles chacun d'entre eux travaille pour le journal scolaire :

	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00
Anna								
Bea								
Celine								
David								
Emma								
Flo								
Gioa								
Hans								
Ida								
Jakob								

Comme les volontaires travaillent très bien, le proviseur a décidé de mettre à leur disposition de nouveaux ordinateurs portables.

*Combien de nouveaux ordinateurs portables l'école doit-elle acquérir pour que chaque volontaire puisse travailler sur un de ces nouveaux ordinateurs pendant ses heures creuses individuelles ?*

- A) quatre
- B) cinq
- C) sept
- D) dix



## Solution

La réponse A) n'est pas correcte. Entre 9h et 10h, les volontaires 1, 3, 5, 8 et 9 devraient travailler parallèlement. Ainsi, quatre ordinateurs portables ne seraient pas suffisants.

La réponse B) est correcte. De 9h à 10h et de 10h à 11h, il y a exactement 5 volontaires qui travaillent parallèlement. 5 est aussi le nombre maximum de volontaires qui travaillent en même temps. Ceci veut dire que pendant toutes les autres heures, il y a toujours moins de 5 volontaires qui travaillent en même temps. Il faut donc acquérir 5 ordinateurs portables au maximum.

	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00
Anna		3	3					
Bea			1	1	1	1		
Celine	1	1						
David					3	3	3	
Emma		2	2					
Flo				2	2			
Gioa			5	5		5	5	
Hans		5						
Ida	4	4	4					
Jakob						2	2	

Étant donné qu'il ne faut que 5 ordinateurs portables au maximum, les réponses C) et D) ne sont pas correctes.

## C'est de l'informatique !

Afin de mieux comprendre les interdépendances à l'intérieur d'une grande quantité de données, il s'avère utile de choisir une représentation des données appropriée comme, par exemple, un tableau, un diagramme ou un graphe. Selon le but visé, les différentes représentations sont plus ou moins avantageuses.

Pour notre tâche, il conviendrait de représenter les données comme suit :

	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00
1	Celine	Celine	Bea	Bea	Bea	Bea		
2		Emma	Emma	Flo	Flo	Jakob	Jakob	
3		Anna	Anna		David	David	David	
4	Ida	Ida	Ida					
5		Hans	Gioa	Gioa		Gioa	Gioa	

Cette représentation permet d'associer un ordinateur portable à l'heure ou aux heures de travail individuelles de chaque volontaire. Ainsi, on peut rapidement vérifier qu'il faut acquérir 5 ordinateurs portables au maximum.

Dans notre quotidien, on a souvent besoin de telles planifications surtout quand il s'agit d'exploiter les ressources disponibles de manière optimale. Chaque hôtel, par exemple, possède un système de réservation. D'une part, il est important que les réservations ne se recoupent pas, et d'autre part, il est souhaitable que l'hôtel affiche complet. Il en va de même pour les salles de classe d'une école. Là aussi, une planification optimale est souhaitable pour tous les acteurs concernés, même si le terme «optimal» peut être compris de manières diverses : serait-il par exemple préférable qu'une classe reste toute la journée dans la même salle ou, qu'au contraire, les professeurs ne changent pas de salle de classe ?





Il existe encore un autre type de représentation pour de telles planifications : c'est le graphe d'intervalles. Chaque intervalle de ce graphe représente l'allocation d'une ressource pendant un certain temps et les arêtes relient deux sommets lorsque deux intervalles correspondants se recoupent.

## Sites web et mots clés

planification, graphe d'intervalles

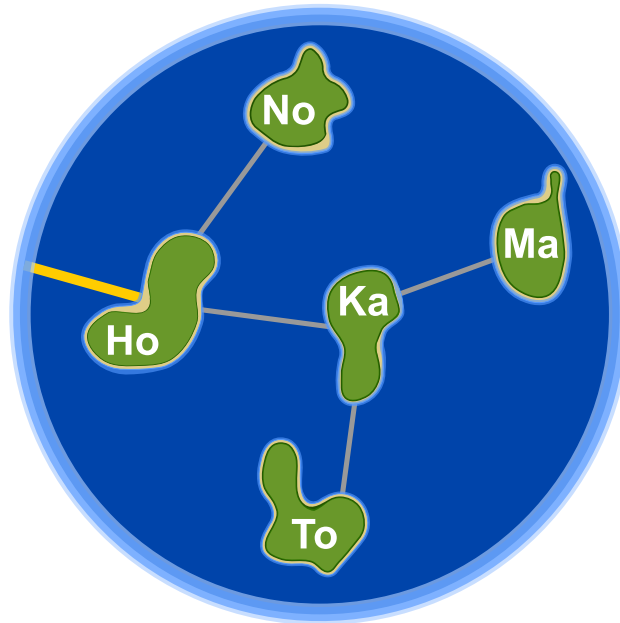
— [https://fr.wikipedia.org/wiki/Graphe\\_d'intervalles](https://fr.wikipedia.org/wiki/Graphe_d'intervalles)





## 6. Honomakato

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



Les habitants de Honomakato demandent une connexion fiable à Internet pour toutes les îles : cela veut dire que même si un des câbles est endommagé, chacune des îles doit avoir accès à Internet.

*Fais en sorte que l'archipel Honomakato obtienne une connexion fiable à Internet. Pose deux autres câbles entre les îles. Il existe plusieurs solutions possibles.*

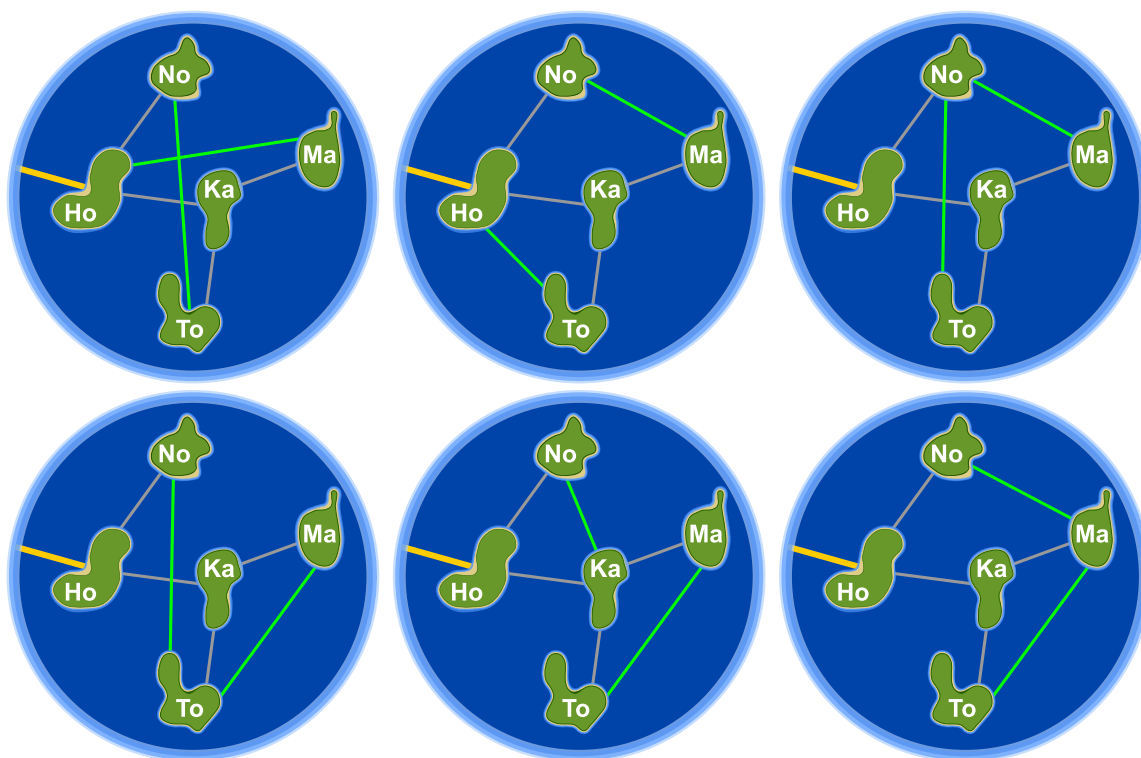


## Solution

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

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

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



## C'est de l'informatique !

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

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



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

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

## Sites web et mots clés

structure de données dynamique, graphe, pont (informatique)

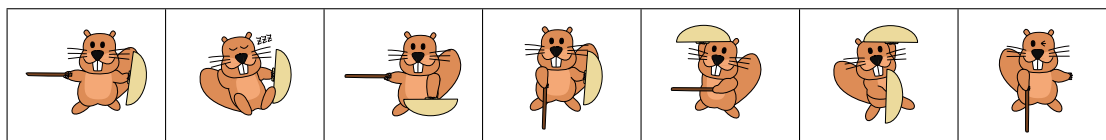
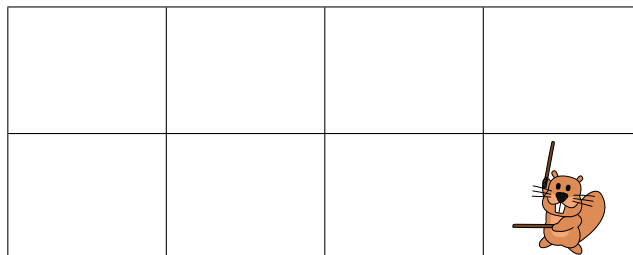
— [https://fr.wikipedia.org/wiki/Séparateur\\_\(théorie\\_des\\_graphes\)](https://fr.wikipedia.org/wiki/Séparateur_(théorie_des_graphes))





## 7. Un art martial japonais

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

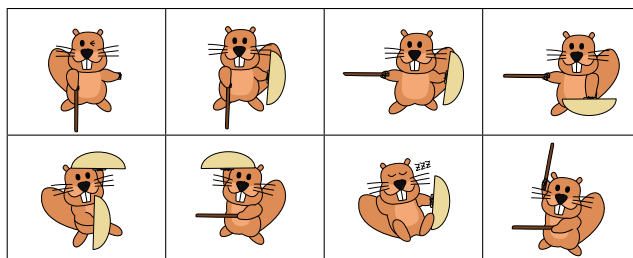


Déplace les images des amis dans les cases dessinées sur le sol de la cour de récréation pour que chaque bâton vise un bouclier.



## Solution

La réponse correcte est :



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

## C'est de l'informatique !

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

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

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

## Sites web et mots clés

raisonnement logique, conclusion

- [https://fr.wikipedia.org/wiki/Retour\\_sur\\_trace](https://fr.wikipedia.org/wiki/Retour_sur_trace)
- [http://www.inf-schule.de/grenzen/komplexitaet/affenpuzzle/einstieg\\_affenpuzzle](http://www.inf-schule.de/grenzen/komplexitaet/affenpuzzle/einstieg_affenpuzzle)





## 8. Pizzeria Castoria

La Pizzeria Castoria n'a qu'un seul four à pizza. C'est pourquoi le pizzaiolo ne peut cuire que peu de plats en même temps.

En tout, il y a trois combinaisons possibles :



Le temps de cuisson varie selon le type de repas : une petite pizza doit cuire pendant 10 minutes, une grande pizza pendant 15 minutes et la cuisson d'un pain ciabatta, par contre, nécessite 20 minutes. Ce qui facilite un peu la tâche du pizzaiolo est que même s'il est en train de cuire deux ou trois plats en même temps, il peut les mettre au four ou les sortir du four au bon moment, c'est-à-dire il peut les y mettre ou les en sortir successivement.

Aujourd'hui, le pizzaiolo a beaucoup de travail. Il doit préparer une petite pizza, deux grandes pizzas et quatre pains ciabatta. Les clients sont affamés et ils attendent leurs plats commandés le plus vite possible.

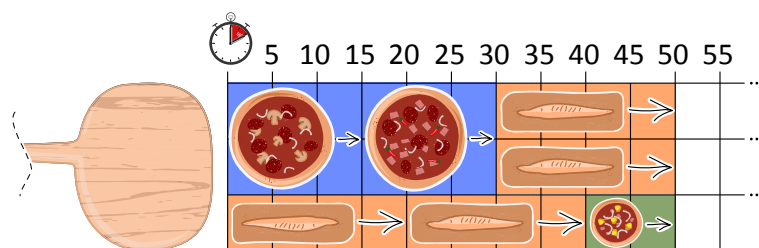
*Quel est le temps minimal pendant lequel le pizzaiolo arrive à faire cuire tous les plats commandés ?*





## Solution

Pour résoudre ce problème, il existe plusieurs solutions optimales. Une d'entre elles serait de cuire d'abord les deux grandes pizzas successivement tout en cuisant parallèlement deux pains ciabatta. Puis on mettra au four deux pains ciabatta et une petite pizza :

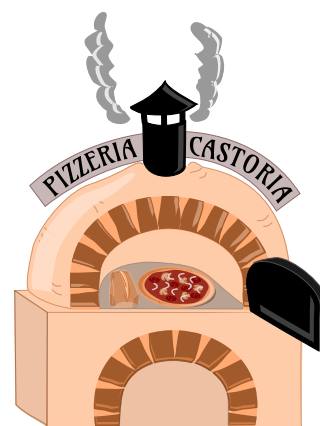


Ainsi, le temps de cuisson serait de 50 minutes au total. Il n'existe pas de cuisson plus rapide car avec cette solution, le four est chargé de manière maximale pendant tout le temps de la cuisson.

## C'est de l'informatique !

Quand on essaie de planifier l'ordre d'exécution des processus, dit d'ordonnancement (en anglais, *scheduling*), et ainsi de planifier les demandes d'accès à une ressource limitée (dans notre tâche, il s'agit du four ou plus exactement de la charge maximale du four), on cherche à trouver une solution optimale en ce qui concerne les coûts ou le temps d'exécution (voir notre problème à résoudre). Si la ressource est utilisée de manière optimale, le temps d'exécution des processus sera minimal.

Un algorithme d'ordonnancement très populaire est le *round-robin* (en anglais, *Round-Robin* qui, lui, est en fait un jeu de parcs, un tourniquet). L'idée de cet algorithme est que tous les processus qui doivent être exécutés sont insérés dans une file d'attente gérée comme une file circulaire : l'ordonnanceur accorde un temps processeur à chacun des processus. Si ce temps-là n'est pas suffisant, le processus sera réinséré dans la file d'attente jusqu'à ce que l'ordonnanceur lui accorde un nouvel intervalle de temps.



Cependant, le pizzaiolo de la pizzeria ne pourra pas se servir de cette stratégie car jamais il ne sortirait un plat du four à demi cuit pour l'y remettre un peu plus tard afin de terminer la cuisson. C'est pourquoi, en informatique, on recourra plutôt à une autre stratégie, appelée *greedy algorithm* (en français, *algorithme glouton* ou *gourmand*) : commençant par le processus qui demande le plus de ressources (par une grande pizza car celle-ci occupe la plus grande place dans le four), on continuera par des processus qui demandent moins de ressources et ainsi de suite. Étant donné que le four est assez grand pour faire cuire plusieurs plats en même temps (bien que ce processus soit soumis à des restrictions), le pizzaiolo peut sortir du four un plat dont la cuisson est terminée en le remplaçant par un autre plat qui est prêt à être cuit.

Pourtant, il reste à savoir si, en fin de compte, on a vraiment trouvé la solution optimale. Ceci n'est pas aussi évident car on ne peut pas exclure définitivement que l'on ne trouverait pas une meilleure solution à celle que l'on a trouvée. On pourrait, par exemple, commencer par trois pains ciabatta. Toutefois, quand on a réussi à utiliser de manière maximale toutes les ressources (dans notre tâche cela signifierait de charger le four au maximum), on peut être sûr que l'on ne trouvera pas de solution plus avantageuse.



## Sites web et mots clés

scheduling, l'ordonnancement, file d'attente, ressource, round-robin, greedy algorithm, algorithme glouton

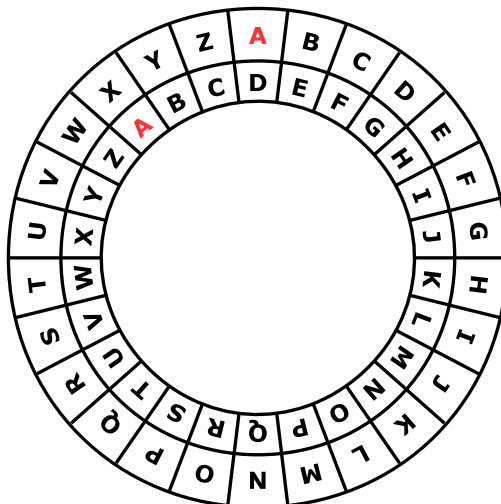
- [https://fr.wikipedia.org/wiki/Ordonnancement\\_dans\\_les\\_systèmes\\_d'exploitation](https://fr.wikipedia.org/wiki/Ordonnancement_dans_les_systèmes_d'exploitation)
- [https://fr.wikipedia.org/wiki/Round-robin\\_\(informatique\)](https://fr.wikipedia.org/wiki/Round-robin_(informatique))
- [https://fr.wikipedia.org/wiki/Algorithme\\_glouton](https://fr.wikipedia.org/wiki/Algorithme_glouton)





## 9. Une commande chiffrée

Anna passe ses commandes au restaurant à l'aide de messages chiffrés. Seul César, le cuisinier, sait les déchiffrer. Pour rédiger un message chiffré, elle utilise un disque particulier composé d'un anneau extérieur et d'un anneau intérieur mobile. Chaque anneau affiche les lettres de l'alphabet. Celles-ci sont ordonnées dans l'ordre de l'alphabet. Au début, les lettres des deux anneaux sont alignées : la lettre A (de l'anneau intérieur) se trouve exactement en dessous de la lettre A (de l'anneau extérieur), la lettre B se trouve en dessous de la lettre B, et ainsi de suite.



Pour rédiger un message chiffré, Anna procède comme suit : d'abord, elle note sa commande, par exemple une PIZZA. Ensuite, elle effectue les opérations suivantes :

1. Au-dessous de chaque lettre du plat commandé, elle note un chiffre au hasard. Celui-ci marque le nombre de rotations qu'il faut effectuer plus tard.
2. Pour chaque lettre du message original, elle met d'abord l'anneau intérieur à la position initiale, ensuite elle le tourne dans le sens inverse des aiguilles d'une montre, lettre par lettre. Le nombre de rotations correspondra au nombre de rotations propre à la lettre du message original.
3. Finalement, elle remplace la lettre originale par la lettre que l'anneau intérieur indique au-dessous de la lettre originale.

Si, par exemple, elle veut commander une PIZZA et qu'elle utilise les nombres de rotations 3, 1, 4, 1 et 5, elle rédige le message chiffré SJDAF.

commande	P	I	Z	Z	A
nombre de rotations	3	1	4	1	5
message chiffré	S	J	D	A	F

Pour une autre commande, Anna utilise les nombres de rotations 3, 1, 4, 1, 5, 9 et 2 et ensuite, elle rédige le message chiffré OBWBLWC.

Si nous regardons de plus près le message chiffré OBWBLWC, quelle commande Anna a-t-elle passée ?

commande						
nombre de rotations	3	1	4	1	5	9
message chiffré	O	B	W	B	L	W



## Solution

La réponse correcte est LASAGNA :

commande	L	A	S	A	G	N	A
nombre de rotations	3	1	4	1	5	9	2
message chiffré	O	B	W	B	L	W	C

Afin de déchiffrer le message chiffré, c'est-à-dire de savoir quelle commande Anna a passée, nous nous servons de son disque particulier. Pour trouver la lettre originale, nous tournons l'anneau intérieur selon le nombre de rotations qui correspond à cette lettre dans le sens des aiguilles d'une montre. Ensuite, on cherche la lettre chiffrée parmi les lettres de l'alphabet de l'anneau intérieur. La lettre qui se trouvera au-dessus de la lettre chiffrée sera une des lettres du message original. Donc, pour déchiffrer le message chiffré, on fait tout simplement l'inverse de ce que l'on fait quand on chiffre un message.

## C'est de l'informatique !

Anna chiffre son message afin qu'il ne soit lu que par son cuisinier préféré. Le *chiffrement* ou le *cryptage* est une méthode très ancienne. Les motivations de chiffrer un message sont évidentes : nous aimerions bien être sûrs que le message ne sera lu que par un destinataire particulier et non pas par un espion, par exemple. Il existe beaucoup de procédés différents de *cryptographie*, mais, en général, nous avons affaire à deux algorithmes principaux : un pour chiffrer le message et un pour le déchiffrer. Tous les deux dépendent d'une *clé de déchiffrement* pour exécuter leur travail.

Une des méthodes de chiffrement très simple date de l'époque de Jules César : dans ce cas-là, la clé de déchiffrement est un nombre, appelé la longueur du décalage. Celui-ci indique la distance entre les deux lettres, la lettre du message original et la lettre qui servira comme lettre chiffrée. La clé de déchiffrement 3, par exemple, signifiera un décalage de 3 vers la droite dans l'ordre de l'alphabet : ainsi, la lettre A du message original sera chiffrée par la lettre D, la lettre B du message original, par contre, sera chiffrée par la lettre E, etc. En même temps, la clé de déchiffrement indique l'inverse, notamment que la lettre D a substitué la lettre A et la lettre E a substitué la lettre B. Cette méthode de chiffrement est appelée *chiffrement par décalage*, aussi connu comme le *chiffre de César*. Tout comme dans notre tâche, cette méthode fonctionne à l'aide d'une superposition de deux alphabets.

Des méthodes de chiffrement qui ont pour base une seule clé de déchiffrement (par exemple une seule longueur de décalage) sont comparativement peu sûres. Apparemment, Anna en est parfaitement consciente car elle utilise pour chaque lettre une autre clé. Cette précaution est la base du *chiffrement de Vigenère*. Contrairement au chiffre César, qui, lui, est un *chiffrement monoalphabétique*, il s'agit ici d'un *chiffrement polyalphabétique*. Dans ce procédé, le nombre de rotations est répété surtout quand il s'agit de longs messages à chiffrer afin que la clé de déchiffrement ne soit pas trop longue. Malheureusement, quand le message est long et la clé est courte, le chiffrement est, en fin de compte, insuffisant et vulnérable aux attaques.

## Sites web et mots clés

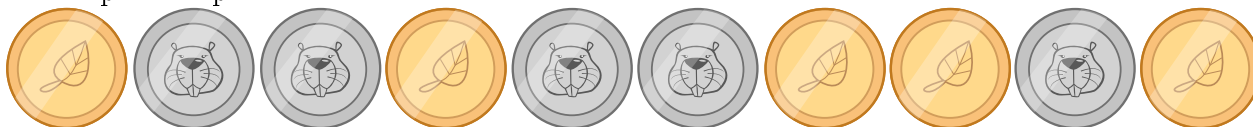
cryptographie, cryptage, chiffrement, déchiffrement, clé de déchiffrement, chiffrement monoalphabétique, chiffrement polyalphabétique, chiffre César, chiffrement de Vigenère

- [https://fr.wikipedia.org/wiki/Chiffrement\\_par\\_décalage](https://fr.wikipedia.org/wiki/Chiffrement_par_décalage)
- [https://fr.wikipedia.org/wiki/Chiffre\\_de\\_Vigenère](https://fr.wikipedia.org/wiki/Chiffre_de_Vigenère)



## 10. Jeu des pièces

Christine possède dix pièces de monnaie qui ont soit une face dorée (🍂) soit une face argentée (🐻). Elle dispose ces pièces sur la table de la manière suivante :



*Combien de fois doit-elle tourner un couple de deux pièces adjacentes pour qu'à la fin toutes les pièces montrent leur face dorée ? (Attention : il n'est possible de tourner que deux pièces de monnaie à la fois, ni plus, ni moins.)*

- A) 1
- B) 2
- C) 4
- D) 6
- E) 8
- F) Ce n'est pas possible.



## Solution

En effet, ce n'est pas possible.

Chaque fois qu'elle tourne deux pièces adjacentes, le nombre de pièces qui montrent la face argentée reste impair. On aura toujours soit deux faces argentées de plus, soit deux faces argentées de moins, soit le même nombre de faces argentées qu'avant : la parité des faces des pièces de monnaie reste la même.

La parité des faces argentées est au début impaire...et elle le restera jusqu'à la fin du jeu, car on n'arrivera jamais à tourner les pièces de sorte que le nombre de pièces avec une face argentée soit pair.

## C'est de l'informatique !

On peut calculer les parités de manière rapide et simple. Les parités sont utiles pour vérifier si une transmission de données a été faite correctement (comme par exemple la lecture d'un code-barre) ou si un nombre a été correctement introduit (comme par exemple le numéro de compte dans le contexte des opérations bancaires en ligne). Quand il faut faire des calculs plus complexes, on peut même corriger directement quelques erreurs pour que les données soient transmises à nouveau.

## Sites web et mots clés





parité, bit de parité








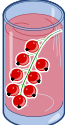



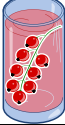



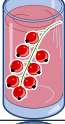
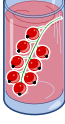



— [https://fr.wikipedia.org/wiki/Somme\\_de\\_contrôle#Exemple:\\_bit\\_de\\_parité](https://fr.wikipedia.org/wiki/Somme_de_contrôle#Exemple:_bit_de_parité)





## 11. Bar à jus de fruits

Sur leur route de vacances, quatre amis font une halte pour se rafraîchir dans un bar à jus de fruits. Chacun d'entre eux a ses propres préférences en ce qui concerne la saveur des jus. Celles-ci sont représentées dans le tableau ci-dessous. Plus il y a de cœurs, plus la personne en question préfère la saveur du jus indiquée. Anna préfère boire le jus  marqué par trois cœurs au jus  marqué par un seul cœur. Daniel, par contre, préfère boire le jus  marqué par quatre cœurs au jus  marqué par un cœur.

				
Anna				
Beat				
Christine				
Daniel				

Comme le bar à jus de fruits est très populaire, chacune des quatre saveurs ne peut être commandée qu'une seule fois.

*Choisis pour chaque ami un jus de fruits afin que le nombre total des cœurs soit aussi grand que possible.*



## Solution

Le nombre maximum de cœurs que l'on peut atteindre est 14. Voici une des solutions possibles :

Anna				
Beat				
Christine				
Daniel				

Pour trouver la solution, on tiendra d'abord compte des préférences de Daniel. Il adore boire le jus de fruits , marqué par quatre cœurs, que les autres n'aiment pas autant (car ils ne l'ont marqué que par un seul cœur). Ensuite, si on clique sur le jus soit pour Beat soit pour Christine, on pourra choisir pour les amis qui n'ont pas encore de jus de fruits (soit Anna et Christine, soit Anna et Beat) le jus de fruits marqué par trois cœurs.

Trois des quatre amis préfèrent la saveur aux autres saveurs. Comme chaque saveur ne peut être commandée qu'une seule fois, deux amis sur trois doivent forcément se contenter de leur deuxième choix. Donc, pour arriver à un maximum de cœurs, le calcul se présentera comme suit :  $3+3+4+4 = 14$ .

D'ailleurs, toute autre solution demanderait à au moins un des amis de choisir le jus de fruits placé au troisième rang. Dans ce cas, le nombre maximum de cœurs sera 13 ( $2 + 3 + 4 + 4 = 13$ ).

## C'est de l'informatique !

La tâche demande d'optimiser le nombre de cœurs (et ainsi, d'optimiser la satisfaction des quatre amis). En programmation informatique ainsi qu'en mathématiques, *l'optimisation* représente un domaine de recherche important car elle est demandée dans de maintes situations en relation avec l'amélioration de l'efficacité d'un programme informatique. Dans le cas présent, un algorithme simple qui cherche à trouver toutes les solutions possibles dans un nombre fini d'étapes (ainsi que celles dont le calcul n'aboutira à aucun résultat), aura besoin de calculer 65000 différentes solutions. Grâce à des réflexions habiles, on arrive à réduire considérablement ce chiffre énorme (en effet, il n'y a que 24 solutions possibles à calculer). Malheureusement, ces réflexions ne sont pas si évidentes.

Notre tâche comporte en fait une forme particulière du *problème de couplage* : à chacune des quatre personnes, on doit attribuer une seule boisson et de chaque saveur il ne reste qu'un seul verre. Puis, il s'ajoute une autre condition à savoir que la satisfaction du groupe (le nombre maximum de cœurs



au total) doit être la plus haute possible. Nous trouvons de tels problèmes également au quotidien : il suffit de penser à la liste d'attente pour une transplantation d'organes. Dans ce cas, il faut tenir compte des *conditions* préalablement définies avant d'attribuer un organe à un patient comme par exemple le temps d'attente, le degré d'urgence établi par les spécialistes ou la compatibilité génétique entre le donneur et le receveur.

## Sites web et mots clés

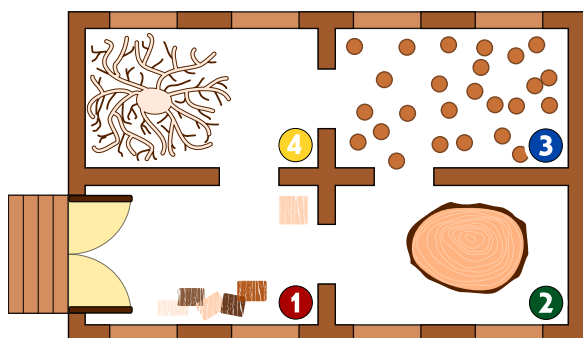
optimisation, couplage

- [https://fr.wikipedia.org/wiki/Optimisation\\_\(mathématiques\)](https://fr.wikipedia.org/wiki/Optimisation_(mathématiques))
- [https://fr.wikipedia.org/wiki/Séparation\\_et\\_évaluation](https://fr.wikipedia.org/wiki/Séparation_et_évaluation)
- [https://fr.wikipedia.org/wiki/Couplage\\_\(théorie\\_des\\_graphes\)](https://fr.wikipedia.org/wiki/Couplage_(théorie_des_graphes))





## 12. Intrusion au musée



Le *Musée de la sculpture moderne sur bois* possède un système de sécurité très intéressant. Étant donné que les sculptures modernes sont passionnantes, les visiteurs se déplacent très lentement de salle en salle. À chaque minute, le système compte le nombre de personnes qui se trouvent dans une salle et il enregistre les valeurs obtenues dans un tableau. En outre, en se basant sur ces données, il contrôle si un intrus a tenté d'entrer au musée. Un intrus est une personne qui s'est introduite clandestinement dans

un lieu, ici, dans le musée, sans qu'elle soit entrée par la porte principale donc sans autorisation.

Dès que le système enregistre une infraction, c'est-à-dire dès qu'un intrus se trouve dans le musée, il sonne l'alarme.

Le tableau à gauche montre les enregistrements du système de sécurité de 10 :01 jusqu'à 10 :07. L'image à droite montre le plan du musée avec les salles numérotées de 1 à 4.

heure	salle 1	salle 2	salle 3	salle 4
10 :01	2	0	0	0
10 :02	3	0	0	0
10 :03	2	1	0	0
10 :04	4	1	1	0
10 :05	2	2	3	0
10 :06	5	2	2	1
10 :07	4	1	2	2

À quelle minute le système de sécurité sonne-t-il l'alarme ?

- A) 10 :01
- B) 10 :02
- C) 10 :03
- D) 10 :04
- E) 10 :05
- F) 10 :06
- G) 10 :07



## Solution

Le système de sécurité sonne l'alarme à 10 :05. À cette minute-là, le système compte dans la salle 3 deux personnes donc une personne de plus que dans la minute d'avant. À 10 :04, dans la salle d'à côté, à savoir dans la salle 2, par contre, il ne compte qu'une seule personne. Par conséquent, une personne s'est introduite dans la salle 3 sans être passée par la salle 2 ni par la salle 4, d'ailleurs, (car de 10 :04 à 10 :05, il n'y avait personne encore) et donc, sans avoir passé la porte d'entrée.

Pendant les autres heures, aucune infraction n'a été enregistrée :

- 10 :01, il y a 2 personnes dans la salle 1 ; on peut supposer qu'elles sont entrées par la porte d'entrée du musée.
- Entre 10 :01 et 10 :02, une autre personne entre dans la salle 1 ; rien n'indique qu'elle ne soit entrée au musée par la porte d'entrée.
- Entre 10 :02 et 10 :03, une personne passe de la salle 1 à la salle 2.
- Entre 10 :03 et 10 :04, une personne passe apparemment de la salle 2 à la salle 3 et une autre personne passe de la salle 1 à la salle 2. En même temps, il y a 3 autres personnes qui entrent dans le musée par la porte d'entrée.
- Entre 10 :05 et 10 :06, une personne passe de la salle 3 à la salle 4 et 3 autres personnes entrent dans le musée par la porte d'entrée.
- Entre 10 :06 et 10 :07, une personne passe de la salle 1 à la salle 4 et une autre personne passe de la salle 2 à la salle 1. En outre, il est fort probable qu'une autre personne quitte le musée parce qu'elle a terminé sa visite.

## C'est de l'informatique !

Dans la vie quotidienne, on a recours à des systèmes de sécurité qui observent des personnes se trouvant dans des secteurs sensibles comme par exemple dans des aéroports. Des programmes informatiques exploitent les images de vidéo-surveillance en direct, notamment pour reconnaître une personne recherchée, ou ils analysent les données à d'autres fins précises. Aujourd'hui, ces programmes utilisent l'intelligence artificielle pour reconnaître des êtres humains, par exemple. Mais souvent, ils utilisent des règles simples comme dans la présente tâche pour détecter des problèmes de sécurité.

## Sites web et mots clés

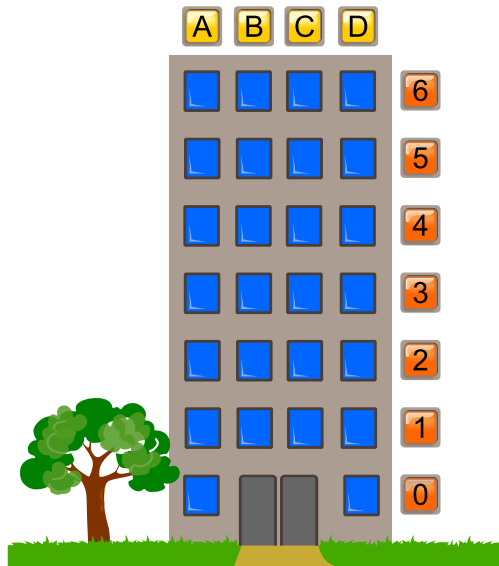
système de sécurité intelligent, règle, transition d'état

- [https://fr.wikipedia.org/wiki/Système\\_de\\_détection\\_d'intrusion](https://fr.wikipedia.org/wiki/Système_de_détection_d'intrusion)

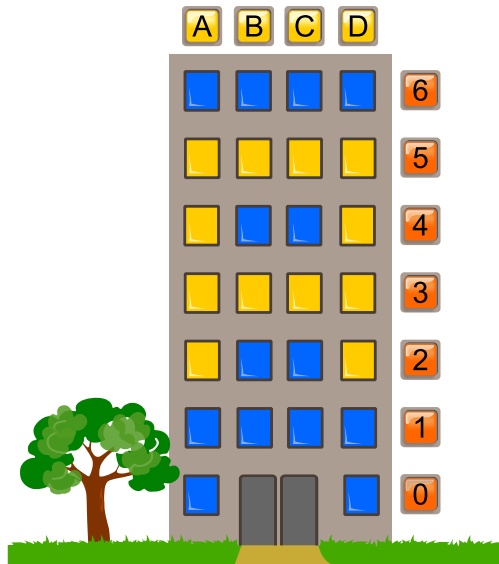


## 13. Des jeux de lumière

Dans la nouvelle tour de la ville, il y a un contrôle centralisé qui règle l'allumage des lumières. La tour comporte 26 fenêtres à travers lesquelles on peut voir si, à l'intérieur, les lumières sont allumées ou éteintes. Malheureusement, il n'est pas possible de régler les lumières séparément : soit on éclaire un étage entier, soit on éclaire une colonne de fenêtres entière.



Sur quels étages (à indiquer par leurs numéros) ou pour quelles colonnes (à indiquer par leurs lettres) faut-il allumer ou éteindre pour que les fenêtres s'allument comme montré ici ?





## Solution

On peut résoudre ce problème de manière simple : on allume d'abord les fenêtres des étages no 3 et no 5, ensuite, on éteint les fenêtres des colonnes A et D. Finalement, on doit éteindre les lumières des étages no 6, 1 et 0. Il existe évidemment beaucoup d'autres solutions possibles, mais toutes les solutions possibles ont recours à cette même séquence d'opérations.

## C'est de l'informatique !

On peut comparer les interrupteurs de cette tâche avec des instructions de n'importe quel système ou de n'importe quel appareil informatique. Même les programmes informatiques beaucoup plus complexes peuvent être compris comme une séquence d'instructions simples. Une fenêtre correspondra donc à une espace mémoire qui, elle, peut avoir la valeur 0 (lumière éteinte) et 1 (lumière allumée). Les ordinateurs modernes permettent l'exécution d'instructions composites qui gèrent parallèlement plusieurs espaces mémoire. Dans notre tâche, ces instructions-là correspondraient aux interrupteurs des fenêtres d'un étage ou d'une colonne. En général, les ordinateurs actuels manipulent simultanément plusieurs millions d'espaces mémoire et ceci, à l'intérieur du processeur seulement... la mémoire vive et le disque dur sont encore plus puissants car ils sont capables de manipuler plusieurs milliards et même milliards d'espaces mémoire !

C'est pourquoi il est important que les instructions composites soient clairement définies à savoir que l'on détermine le moment où elles seront exécutées (*pre-condition*) et ce qui suivra à leur exécution (*post-condition*). Dans notre exemple, les conditions suivantes sont valables pour les lumières d'un étage : si une seule lumière est éteinte (*pre-condition*), toutes les autres lumières de cet étage sont allumées au moment où l'interrupteur est actionné (*post-condition*). Sinon (dans le cas où toutes les lumières sont allumées, *pre-condition*), toutes les lumières de l'étage seront éteintes (*post-condition*).

## Sites web et mots clés

langage assembleur (programmation informatique), séquences, opérations en codage binaire

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





## 14. Substitutions

M. Müller est tombé brusquement malade. Dans l'entreprise où il travaille, M. Maier doit le remplacer et accomplir toutes les tâches dont M. Müller était responsable. Heureusement, M. Müller se rétablit plus vite que prévu et retourne au travail deux semaines plus tard. Comme M. Maier a très bien travaillé, les deux collègues décident qu'à partir de maintenant, M. Maier continuera à accomplir les tâches de M. Müller et que M. Müller accomplira les tâches de M. Maier. Par conséquent, la documentation du projet en cours doit être changée comme suit : le nom de M. Müller doit être substitué au nom de M. Maier et vice versa. Dans la documentation, il est possible de substituer chaque texte à un autre.

*Laquelle des démarches suivantes est valable si l'on suppose que le texte ne comporte aucun symbole «#» ?*

- A) Je remplace d'abord tous les «Müller» par «Maier» et puis tous les «Maier» par «Müller».
- B) Je remplace d'abord tous les «Maier» par «Müller» et puis tous les «Müller» par «Maier».
- C) Je remplace d'abord tous les «Müller» par le symbole «#», ensuite le symbole «#» par «Maier» et finalement les «Maier» par «Müller».
- D) Je remplace d'abord tous les «Müller» par le symbole «#», ensuite tous les «Maier» par «Müller» et finalement les «#» par «Maier».



## Solution

La réponse correcte est D) Je remplace d'abord tous les «Müller» par le symbole «#», ensuite tous les «Maier» par «Müller» et finalement les «#» par «Maier».

- A) Dans ce cas-là, il ne restera plus que le nom «Müller» et tous les «Maier» auront été supprimés car suite à la première substitution, le texte ne comportera que le nom «Maier» qui, lui, sera remplacé par «Müller».
- B) Dans ce cas-là, il ne restera que le nom «Maier» et tous les «Müller» auront été supprimés car suite à la première substitution, le texte ne comportera que le nom «Müller» qui, lui, sera remplacé par «Maier».
- C) Dans ce cas-là, il ne restera que le nom «Müller» et tous les «Maier» auront été supprimés car après avoir remplacé tous les «Müller» par le symbole «#», celui-ci est immédiatement remplacé par «Maier» et finalement on remplace l'ensemble des «Maier» par «Müller».
- D) C'est la seule procédure qui aboutira au résultat voulu : les «Müller» seront temporairement remplacés par le symbole «#» et restent ainsi enregistrés tandis que les «Maier» seront tous remplacés par «Müller».

## C'est de l'informatique !

Bien que la procédure de substitution soit très simple, elle est très importante en informatique. Grâce à de telles substitutions, les informaticiens peuvent effectuer des opérations complexes. En informatique théorique, par exemple, les grammaires formelles (à la base des langages de programmation) sont définies comme une liste de règles de remplacement.

Notre exercice insiste sur le fait que l'on a souvent besoin d'un élément temporaire pour pouvoir effectuer l'échange de deux valeurs : ce concept est à la base de l'échange de variables (swap).

## Sites web et mots clés

traitement de texte, suivre des séquences d'instructions, échange de variables (swap)

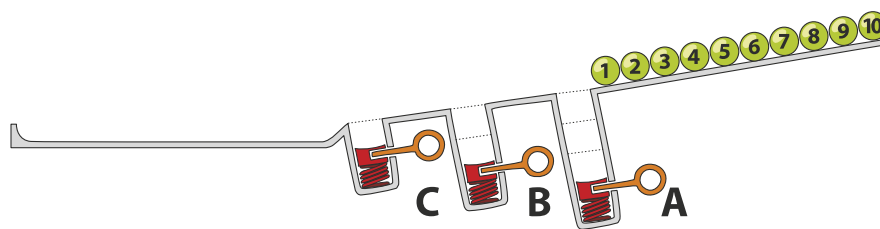
— [https://fr.wikipedia.org/wiki/Grammaire\\_formelle](https://fr.wikipedia.org/wiki/Grammaire_formelle)



## 15. Jeu de billes

Sur une rampe, il y a 10 billes numérotées. Le long de la rampe, il y a trois trous A, B et C : le trou A peut contenir trois billes au maximum, le trou B deux billes et le trou C une seule bille au maximum. Quand les billes roulent sur la rampe, elles tombent successivement dans les trous jusqu'à ce qu'elles les remplissent (les billes 1, 2 et 3 tombent dans le trou A, les billes 4 et 5 tombent dans le trou B et la bille 6 tombe dans le trou C). Les autres billes passent par-dessus et continuent leur chemin jusqu'à la fin de la rampe.

Quand toutes les billes ont parcouru la rampe, les ressorts, placés dans les trous A à C, éjectent les billes qu'ils contenaient : d'abord, les trois billes du trou A, ensuite, celles du trou B et finalement, celle du trou C. Les billes sont ainsi poussées sur la rampe. On attend que toutes les autres billes aient passé avant qu'un ressort ne soit relâché.



*Dans quel ordre les billes de la séquence 1 à 10 seront-elles alignées à la fin ?*

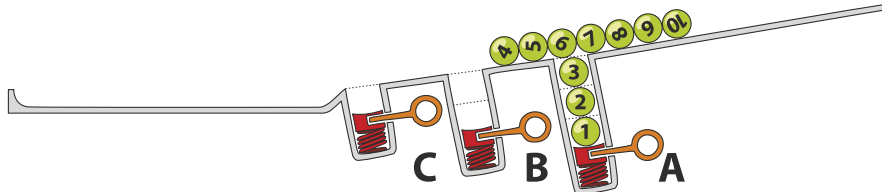




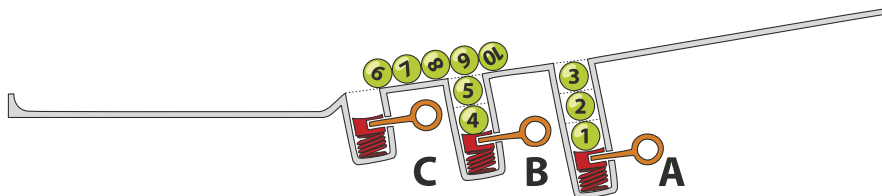
## Solution

La réponse correcte est D) 7 8 9 10 3 2 1 5 4 6.

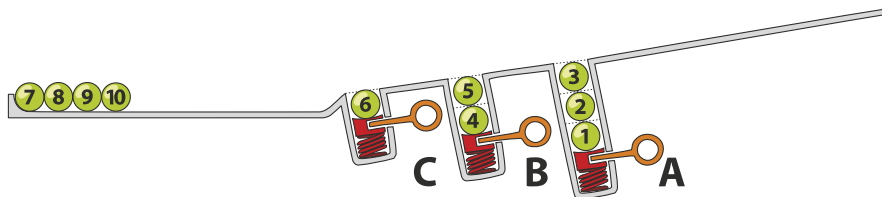
Les billes 1, 2 et 3 tombent dans le trou A, les billes 4 à 10 passent le trou A (qui, lui, contient les billes 1 à 3).



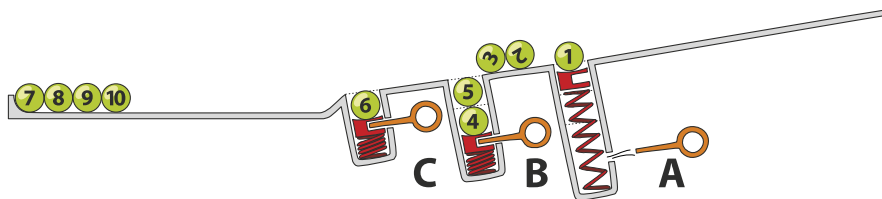
Ensuite, les billes 4 et 5 tombent dans le trou B et les billes 6 à 10 passent le trou B (qui, lui, contient les billes 4 à 5).



Finalement, la bille 6 tombe dans le trou C et les billes 7 à 10 passent le trou C (qui, lui, contient la bille 6). Ces-dernières atteignent la fin de la rampe dans l'ordre de 7 à 10.

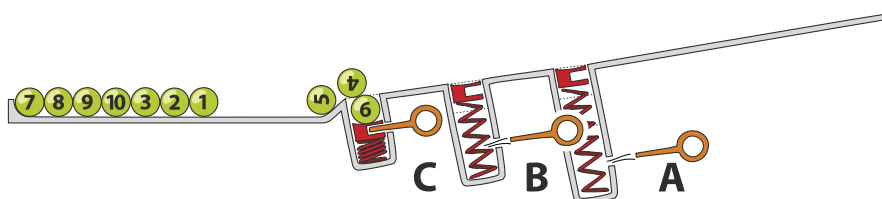


Quand les billes 7 à 10 ont passé les trous, le ressort du trou A libère les billes 3, 2, 1. Celles-ci se retrouvent donc de nouveau sur la rampe et roulent en direction de la fin de la rampe.



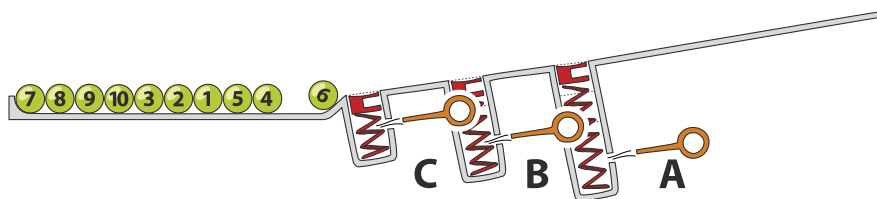
L'ordre de la séquence est donc 7, 8, 9, 10, 3, 2, 1.

Ensuite, le ressort du trou B libère les billes 5 et 4. Celles-ci roulent également en direction de la fin de la rampe. Finalement, le ressort du trou C libère la bille 6 qui s'ajoute aux billes à la fin de la rampe.





Toutes les billes arrivent donc à la fin de la rampe dans l'ordre suivant : 7, 8, 9, 10, 3, 2, 1, 5, 4, 6.



## C'est de l'informatique !

Les trous de la rampe rappellent la structure de la pile (en anglais, *stack*). En informatique, une pile permet d'enregistrer des données pour les réutiliser ensuite selon le principe de *Last-In First-Out (LIFO)* (en français «dernier arrivé, premier sorti») : ainsi, la dernière bille qui est tombée dans le trou sera libérée la première. Aussi simple que ce principe puisse paraître, il est très utile dans bon nombre de situations. On peut par exemple vérifier si les parenthèses d'une expression arithmétique sont équilibrées (dans l'expression  $((1 + 2) \cdot 3)$ , les parenthèses sont équilibrées tandis que dans l'expression  $((4 + 5) \cdot (6 - 7))$ , elles ne le sont pas), en procédant comme suit : chaque parenthèse ouvrante est ajoutée à la pile (à l'aide de l'opération appelée *push*) et quand une parenthèse fermante apparaît, la parenthèse ouvrante est enlevée de la pile (à l'aide de l'opération appelée *pop*). Si on ne trouve plus de parenthèse ouvrante à enlever de la pile bien que l'on en ait encore besoin, ou si à la fin de l'expression il en reste encore dans la pile, nous pouvons être sûrs qu'il y a une erreur. Contrairement à cela, si à la fin de l'expression la pile ne contient plus de parenthèse, cela signifie que les parenthèses sont équilibrées et que donc l'expression est correcte.

## Sites web et mots clés

Stack (en français pile), LIFO

- [https://fr.wikipedia.org/wiki/Last\\_in,\\_first\\_out](https://fr.wikipedia.org/wiki/Last_in,_first_out)
- [https://fr.wikipedia.org/wiki/Pile\\_\(informatique\)](https://fr.wikipedia.org/wiki/Pile_(informatique))



## A. Auteurs des exercices

 Andrea Adamoli  
 Wilfried Baumann  
 Daphne Blokhuis  
 Eugenio Bravo  
 Andrej Brodnik  
 Carmen Bruni  
 Amaury A. Castro Jr.  
 Anton Chukhnov  
 Raluca Constantinescu  
 Valentina Dagienė  
 Christian Datzko  
 Susanne Datzko  
 Janez Demšar  
 Olivier Ens  
 Hanspeter Erni

 Gerald Futschek  
 Martin Guggisberg  
 Urs Hauser  
 Hans-Werner Hein  
 Juraj Hromkovič  
 Ungeol Jung  
 Ivana Kosírová  
 Regula Lacher  
 Greg Lee  
 Milan Lukić  
 Dario Malchiodi  
 Hiroki Manabe  
 Mattia Monga  
 Zsuzsa Pluhár  
 Wolfgang Pohl

 J.P. Pretti  
 Daniel Rakijašić  
 Chris Roffey  
 Frances Rosamond  
 Kirsten Schlüter  
 Eljakim Schrijvers  
 Maiko Shimabuku  
 Taras Shpot  
 Martin Stangl  
 Gergely Tassy  
 Ahto Truu  
 Corina Elena Vint  
 Michael Weigend  
 Hongjin Yeh  
 Momo Yokoyama



## B. Sponsoring : Concours 2017

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

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

**d digitec.ch**

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



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



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



Standortförderung beim Amt für Wirtschaft und Arbeit Kanton Zürich



i-factory (Musée des transports, Lucerne)



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



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



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



**PH LUZERN**  
**PÄDAGOGISCHE**  
**HOCHSCHULE**

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

**ABZ**

AUSBILDUNGS- UND BERATUNGSZENTRUM  
FÜR INFORMATIKUNTERRICHT

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

**n|w**

Fachhochschule  
Nordwestschweiz

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

**Z**

hdk

Zürcher Hochschule der Künste  
Game Design

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



ZUBLER & PARTNER AG  
Informatik

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

**senarclens**  
**leu+partner**  
strategische kommunikation

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





## C. Offres ultérieures

010100110101011001001001  
010000010010110101010011  
010100110100100101000101  
001011010101001101010011  
010010010100100100100001

**SS!E**

[www.svia-ssie-ssii.ch](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.