

INFORMATIQUE

## UNE AFFICHE POUR LE CVC INFORMATIQUE COLLÈGE



**Image vectorielle** : c'est une image numérique composée d'objets géométriques individuels, des primitives géométriques (segments de droite, arcs de cercle, courbes de Bézier, polygones, etc.), définis chacun par différents attributs (forme, position, couleur, remplissage, visibilité, etc.) et auxquels on peut appliquer différentes transformations (homothéties, similitude, rotations, inclinaison, effet miroir, symétrie, translation...).

Il existe de nombreux formats de fichiers graphiques vectoriels. On peut citer Postscript, PDF, Illustrator, CGM, SVG, EPS. Le Scalable Vector Graphics ou SVG, est un format de données ASCII conçu pour décrire des ensembles de graphiques vectoriels et basé sur XML. Ce format est spécifié par le World Wide Web Consortium.

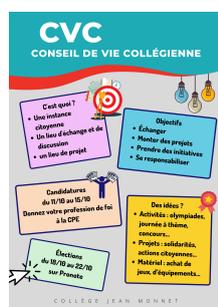
Inkscape est un logiciel de dessin vectoriel libre multiplateforme. Il gère des fichiers conformes aux standards XML, SVG et CSS du W3C. Le logiciel est intégré à la liste des logiciels libres préconisés par l'État français dans le cadre de la modernisation globale de ses systèmes d'information. Il a des fonctionnalités similaires aux logiciels propriétaires CorelDRAW et Adobe Illustrator.

**Image matricielle** : c'est une image constituée d'un pavage carré dont chaque élément, appelé point ou pixel (Bitmap), est coloré selon un code enregistré dans un tableau à deux dimensions. Les formats d'images matricielles sont le PNG, JPEG, BMP, TIFF, GIF. Gimp est un logiciel libre permettant de traiter les images matricielles, il est similaire à Adobe Photoshop.

**Objectifs** : Le 29 novembre 2016 a été mis en place dans tous les collèges le CVC, Conseil de la Vie Collégienne. Le conseil de la vie collégienne donne la parole aux représentants des élèves afin d'impulser une nouvelle dynamique dans les établissements scolaires, de nouveaux projets, un meilleur fonctionnement d'établissement et du mieux-vivre pour les élèves. Ce sont des lieux d'expression de la démocratie scolaire.

Vous êtes un membre élu du CVC et vous souhaitez créer une affiche pour inciter vos camarades à faire des propositions pour améliorer la vie au collège. Pour réaliser cet objectif, vous allez utiliser le logiciel libre de dessin vectoriel Inkscape.

Voici quelques exemples d'affiches qui pourraient vous servir de modèle :

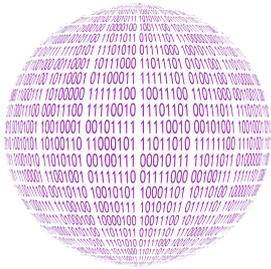


En vous rendant sur la page <https://arnaud.ac3j.fr/CVC> vous trouverez :

- Les six affiches;
- Un modèle au format SVG pour commencer;
- Le formulaire pour poster votre travail.

Quelques pistes pour atteindre le niveau de compétence attendu :

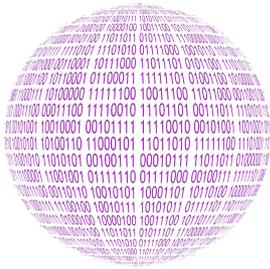
- **Le fond**
  - Déterminer le texte que vous souhaitez mettre en avant, soyez précis;
  - Le contenu doit être réaliste et correspondre aux possibilités offertes au CVC.
- **La forme**
  - Partez du modèle fourni, il contient plusieurs calques pour chaque partie essentielle de l'affiche;
  - Inutile de chercher une affiche toute prête, l'objectif est d'apprendre à utiliser Inkscape et de produire un résultat personnel;
  - Soyez **créatif**, ne négligez aucun détails (alignements, couleurs, polices de caractère...);
  - N'ajoutez ni objets ni logos ni dessins extérieurs à Inkscape, vous devez les concevoir vous-même;
  - Vous pouvez utiliser des photos trouvées sur le Web, mais ce n'est pas l'objectif principal!



# UNE AFFICHE POUR LE CVC — Correction



INFORMATIQUE



INFORMATIQUE



## QUE CONTIENT UN FICHIER ?

### INFORMATIQUE COLLÈGE



## Cryptologie

La **cryptologie** est la science du secret, elle englobe la cryptographie et la cryptanalyse. C'est un domaine aux frontières de la technologie et des mathématiques.

La **cryptographie** est une discipline qui s'attache à protéger des messages en s'aidant souvent de secrets ou de clés. La cryptographie crée des codes secrets qui ne sont lisibles que par les possesseurs d'un secret ou d'une clé. Elle est utilisée depuis l'Antiquité. L'un des plus connus est le code de César qui consiste à décaler toutes les lettres d'un même écart, la clé. Le code de Vigenère est un code de César amélioré où les lettres ne sont pas décalées de la même manière suivant la clé.

La **cryptanalyse** vise à étudier les codes secrets pour déterminer leurs failles et les rendre intelligible. Ce que fit Alan Turing pendant la seconde guerre mondiale pour déchiffrer avec la « bombe » le code de la machine Enigma des nazis.

## Stéganographie

La **stéganographie** est un domaine où l'on cherche à dissimuler discrètement de l'information dans un média de couverture, typiquement un signal de type texte, son, image, vidéo... Elle se distingue de la cryptographie qui cherche à rendre un contenu inintelligible. Lorsqu'un acteur extérieur regarde un contenu chiffré, il peut deviner la nature sensible de l'information qui lui est cachée. L'intérêt de la stéganographie réside précisément dans la possibilité de communiquer en échangeant des contenus d'apparence anodines de façon à ne pas éveiller de soupçons. Pour prendre une métaphore, la stéganographie consisterait à enterrer son argent dans son jardin là où la cryptographie consisterait à l'enfermer dans un coffre-fort — cela dit, rien n'empêche de combiner les deux techniques, de même que l'on peut enterrer un coffre dans son jardin.

Voici un poème écrit par la Résistance française durant la seconde guerre mondiale. De manière surprenante, il semble faire l'éloge des Nazis!

*Aimons et admirons le chancelier Hitler!  
L'Éternelle Angleterre est indigne de vivre.  
Maudissons, écrasons le peuple d'outremer  
Le nazi sur la terre sera seul à survivre.  
Soyons donc le soutien du führer allemand  
De ces navigateurs la race soit maudite.  
À eux seuls appartient ce juste châtement  
La palme du vainqueur répond au vrai mérite.*

En y regardant de plus près, serez-vous capable de lire le véritable message inclus de manière stéganographique dans ce message. Vous pouvez par exemple ne lire que les premiers mots de chaque vers, du haut vers le bas, puis reprendre de même avec la deuxième partie de chaque vers.

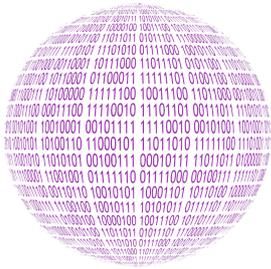
## Fichiers

Un **fichier** un ensemble de données numériques réunies sous un même nom, enregistrées sur un support de stockage permanent comme un disque dur ou une clé usb. Sur le support, les informations sont écrites sous forme de bits, des 0 et des 1, regroupés par groupe de huit, des octets. Ces octets peuvent être regroupés par deux pour former un nombre hexadécimal. Ces informations peuvent aussi être interprétées comme un caractère en utilisant le code ASCII.

Un **format de fichier** est une organisation du fichier qui permet au système d'exploitation de connaître la nature des informations et par conséquent le logiciel qui est capable de le lire. Un fichier contient des **métadonnées** qui peuvent indiquer le nom de l'auteur, la dernière modification, la date, la longueur du fichier... Windows utilise l'extension du nom de fichier pour associer un fichier à un logiciel. D'autres systèmes d'exploitation ignore cette extension.



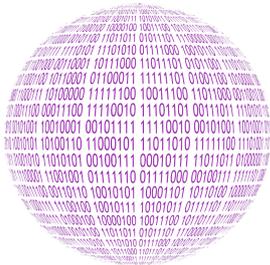
Il est important de ne pas ouvrir un fichier dont on ne connaît pas l'origine. Une image peut en effet cacher un programme malveillant qui pourrait endommager ou prendre le contrôle du terminal (ordinateur, tablette, téléphone...) qui vient de l'utiliser!



# QUE CONTIENT UN FICHIER ? — Correction



INFORMATIQUE



## INFORMATIQUE

La **numérisation** est la conversion des informations d'un support (texte, image, audio, vidéo) ou d'un signal électrique en données numériques que des dispositifs informatiques ou d'électronique numérique pourront traiter.

Les **données numériques** se définissent comme une suite de caractères et de nombres qui représentent des informations.

Pour des raisons électriques, l'information numérisée est stockée ou transmise sous la forme de 0 et de 1. La quantité minimale d'information s'appelle le **bit**. Il correspond à un 0 ou un 1. Pour organiser l'information, les bits sont stockés par groupe de 8, on appelle cela un **octet**.

## Les nombres entiers et le binaire

### Un message secret énigmatique

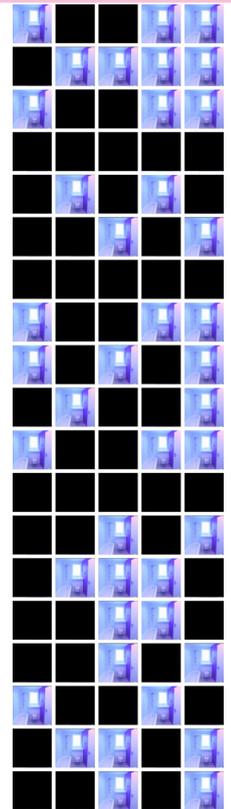
Arthur est pris au piège dans la salle de bain en pleine nuit. La porte vient de claquer et la serrure est bloquée. Il est seul dans la maison. Il ne sert à rien de crier, la maison est un peu isolée. Sa voisine le plus proche est Marie, une informaticienne. Elle est trop loin pour l'entendre, même si elle travaille souvent la nuit. Il n'y a qu'une ouverture dans la salle de bain, une fenêtre pour aérer avec des barreaux métalliques.

Il a alors une brillante idée : il peut utiliser la lumière de la salle de bain pour envoyer un message à Marie. Il pense avoir trouvé un code simple qu'elle sera capable de décoder.

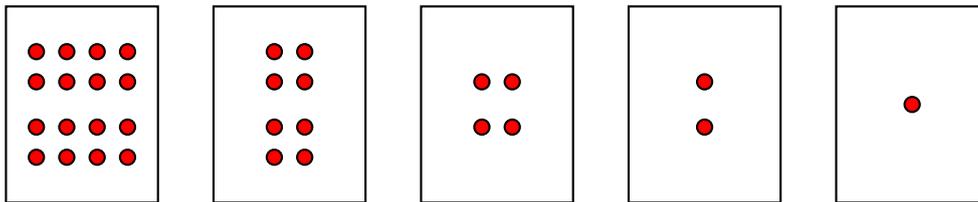
Pouvez-vous lire son message ?



|    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|
| A  | B  | C  | D  | E  | F  | G  | H  | I  | J  | K  | L  | M  |
| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 |
| N  | O  | P  | Q  | R  | S  | T  | U  | V  | W  | X  | Y  | Z  |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |



### La numération binaire



Écrire les nombres entiers de 1 à 30 sous la forme d'une somme de ces étiquettes.

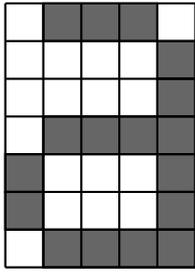
Compléter :

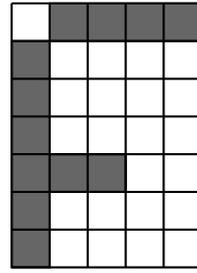
| Décimal | Binaire |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0       |         | 7       |         | 14      |         | 21      |         | 28      |         |
| 1       |         | 8       |         | 15      |         | 22      |         | 29      |         |
| 2       |         | 9       |         | 16      |         | 23      |         | 30      |         |
| 3       |         | 10      |         | 17      |         | 24      |         | 31      |         |
| 4       |         | 11      |         | 18      |         | 25      |         | 32      |         |
| 5       |         | 12      |         | 19      |         | 26      |         |         |         |
| 6       |         | 13      |         | 20      |         | 27      |         |         |         |

Décoder le message de Tom.

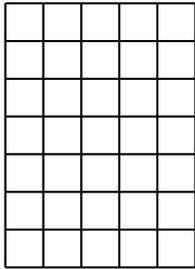
# Numériser une image

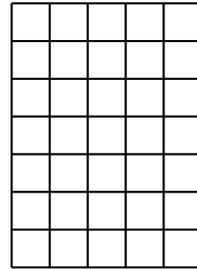
Compléter

|   |         |           |
|---|---------|-----------|
|  | Binaire | Compressé |
|   | _____   | _____     |
|   | _____   | _____     |
|   | _____   | _____     |
|   | _____   | _____     |
|   | _____   | _____     |
|   | _____   | _____     |
|   | _____   | _____     |

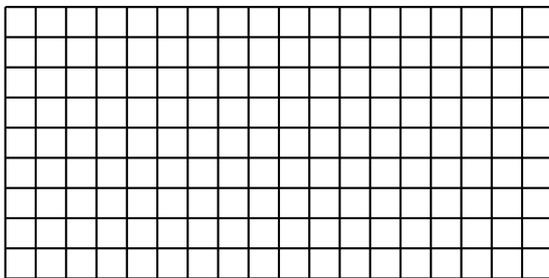
|  |         |           |
|--|---------|-----------|
|  | Binaire | Compressé |
|  | _____   | _____     |
|  | _____   | _____     |
|  | _____   | _____     |
|  | _____   | _____     |
|  | _____   | _____     |
|  | _____   | _____     |
|  | _____   | _____     |

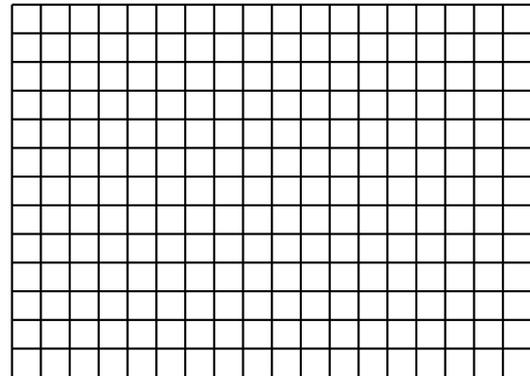
Recommencer en utilisant l'initiale de votre nom ou de votre prénom.

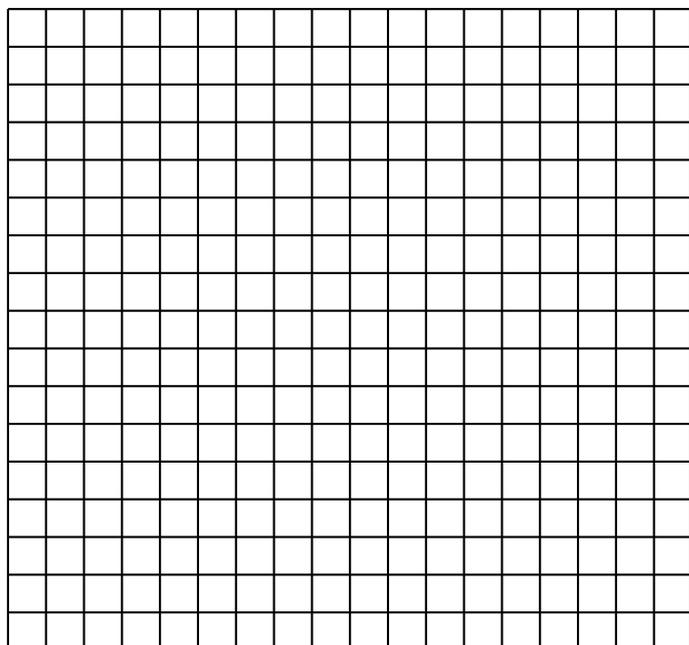
|   |         |           |
|---|---------|-----------|
|  | Binaire | Compressé |
|   | _____   | _____     |
|   | _____   | _____     |
|   | _____   | _____     |
|   | _____   | _____     |
|   | _____   | _____     |
|   | _____   | _____     |
|   | _____   | _____     |

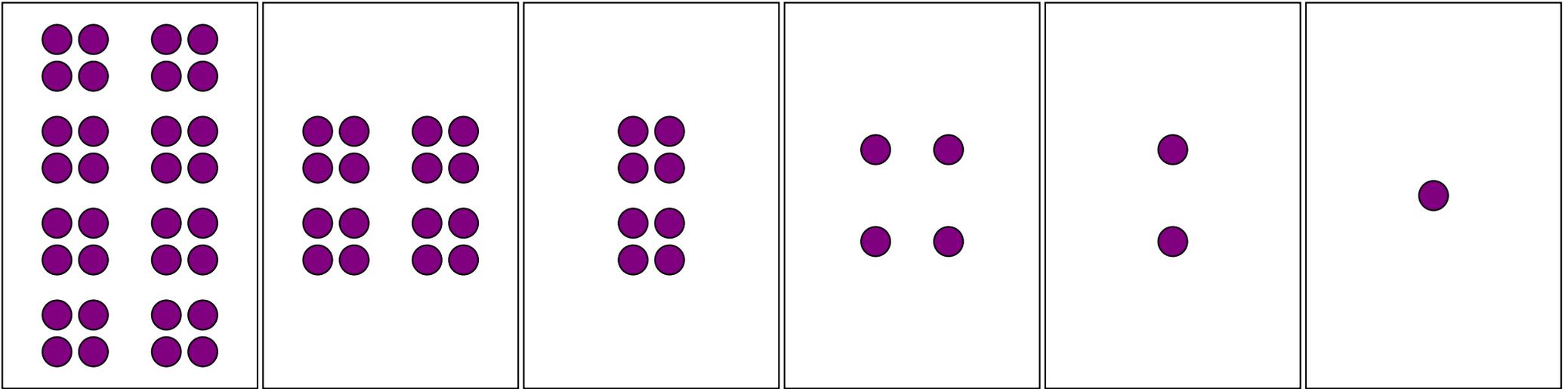
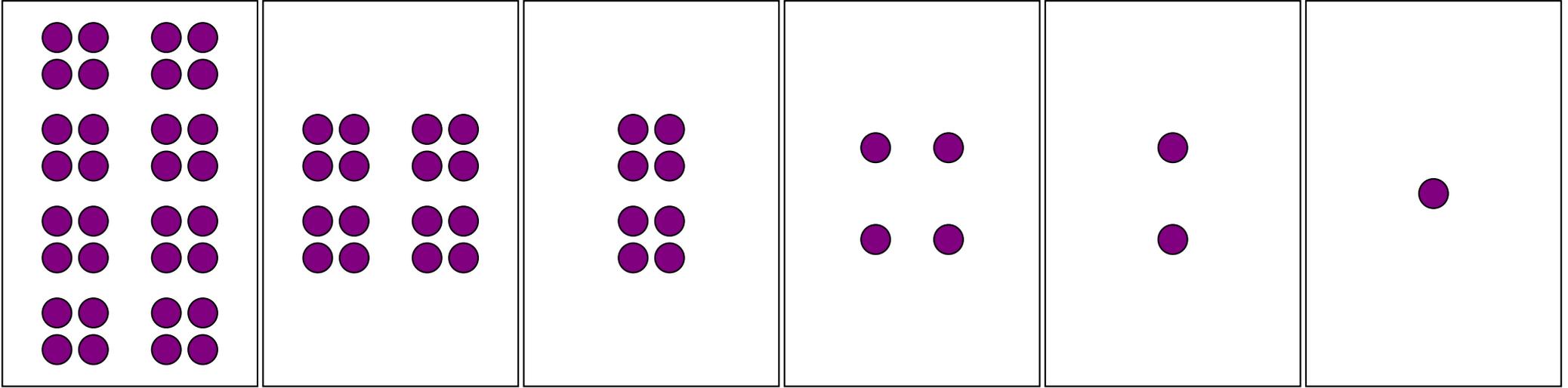
|  |         |           |
|--|---------|-----------|
|  | Binaire | Compressé |
|  | _____   | _____     |
|  | _____   | _____     |
|  | _____   | _____     |
|  | _____   | _____     |
|  | _____   | _____     |
|  | _____   | _____     |
|  | _____   | _____     |

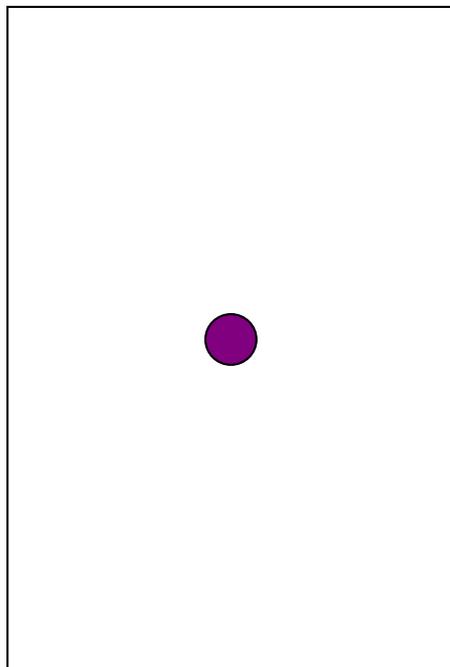
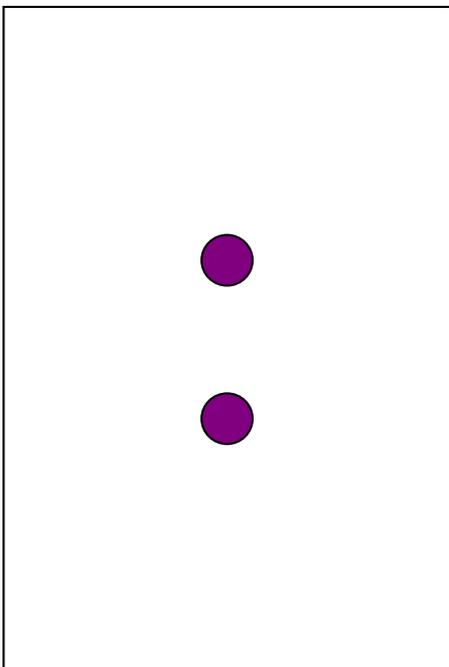
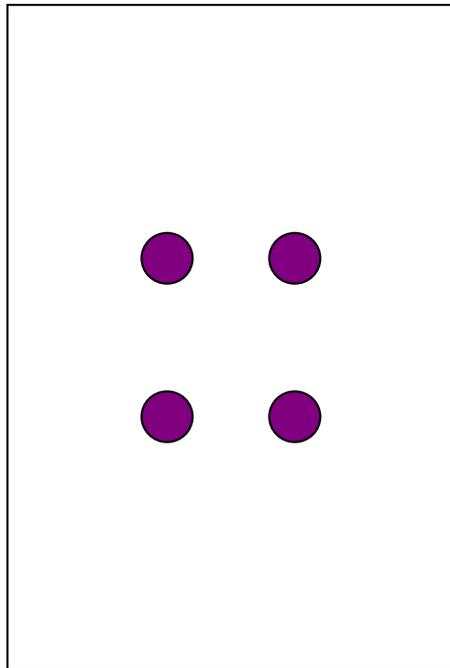
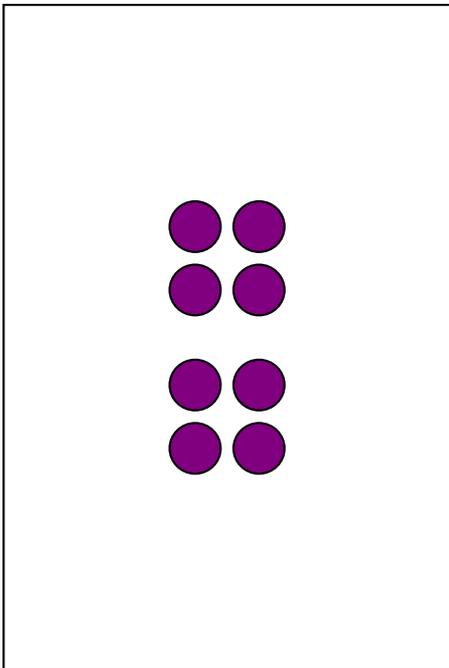
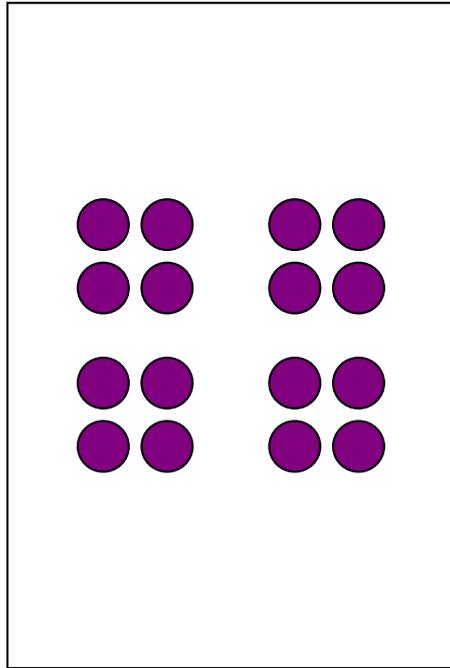
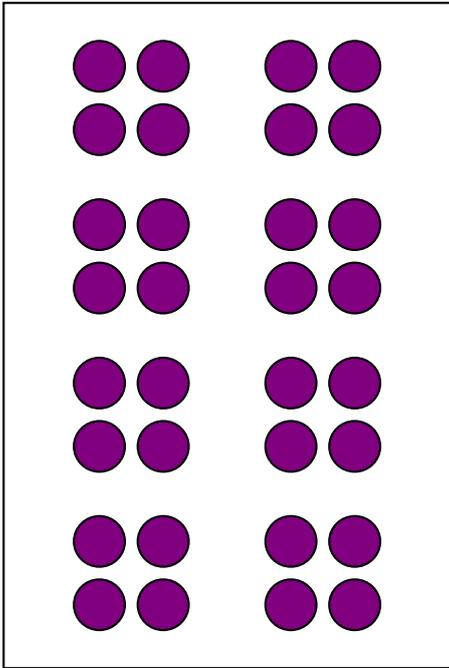
Décoder les trois images suivantes :

|   |            |
|---|------------|
|  | 4, 11      |
|   | 4, 9, 2, 1 |
|   | 4, 9, 2, 1 |
|   | 4, 11      |
|   | 4, 9       |
|   | 4, 9       |
|   | 5, 7       |
|   | 0, 7       |
|   | 1, 15      |

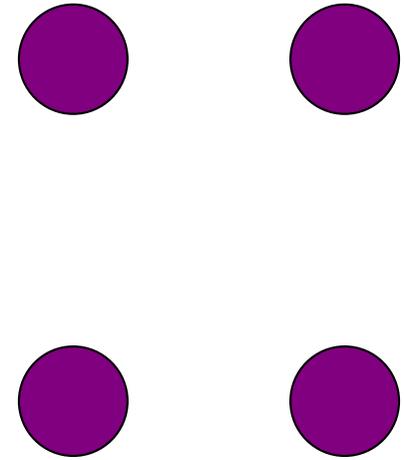
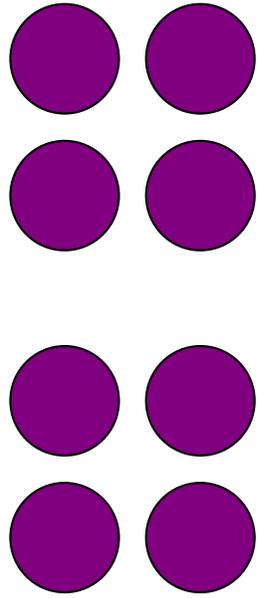
|  |                        |
|--|------------------------|
|  | 6, 5, 2, 3             |
|  | 4, 2, 5, 2, 3, 1       |
|  | 3, 1, 9, 1, 2, 1       |
|  | 3, 1, 9, 1, 1, 1       |
|  | 2, 1, 11, 1            |
|  | 2, 1, 10, 2            |
|  | 2, 1, 9, 1, 1, 1       |
|  | 2, 1, 8, 1, 2, 1       |
|  | 2, 1, 7, 1, 3, 1       |
|  | 1, 1, 1, 1, 4, 2, 3, 1 |
|  | 0, 1, 2, 1, 2, 2, 5, 1 |
|  | 0, 1, 3, 2, 5, 2       |
|  | 1, 3, 2, 5             |

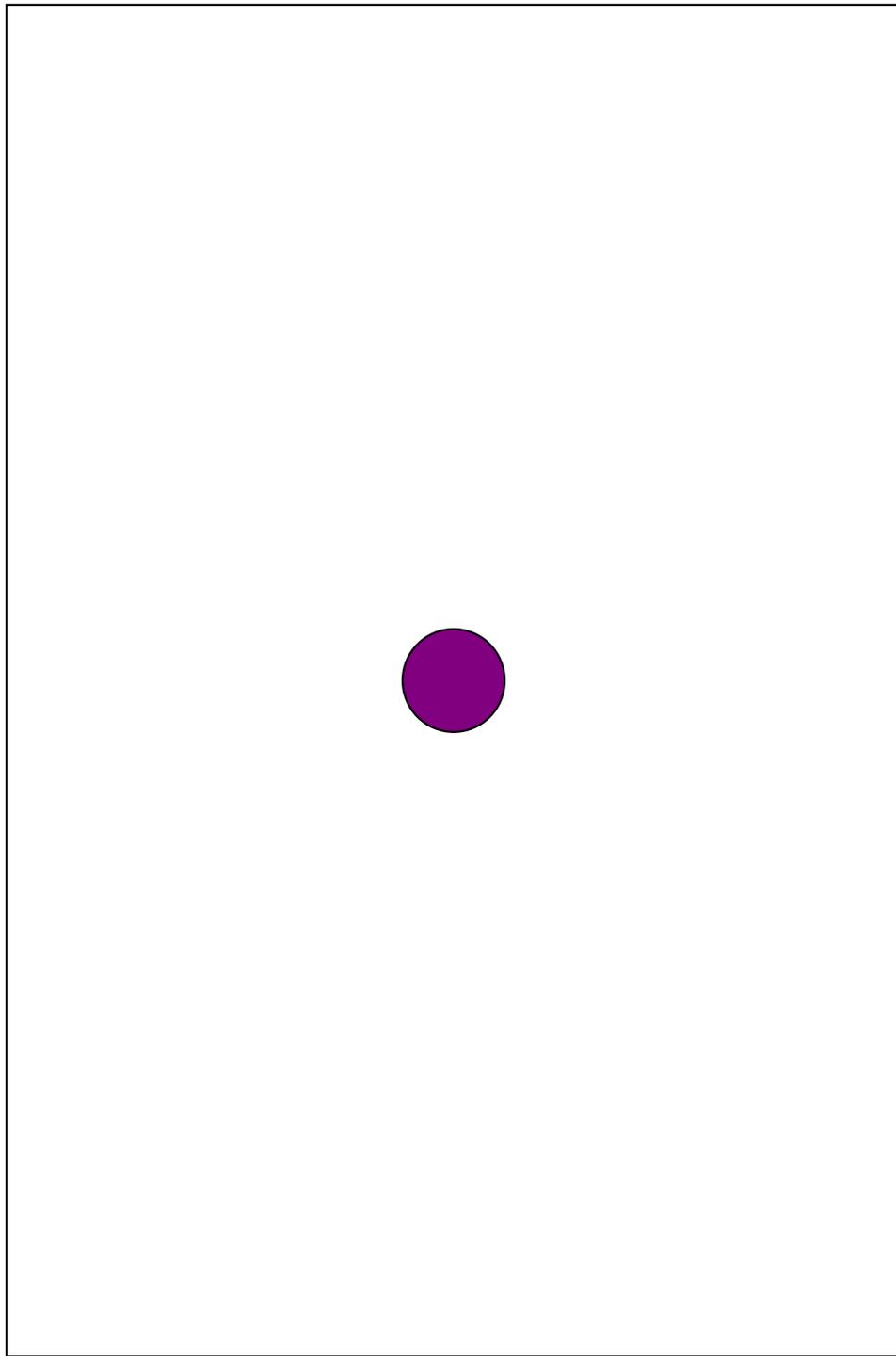
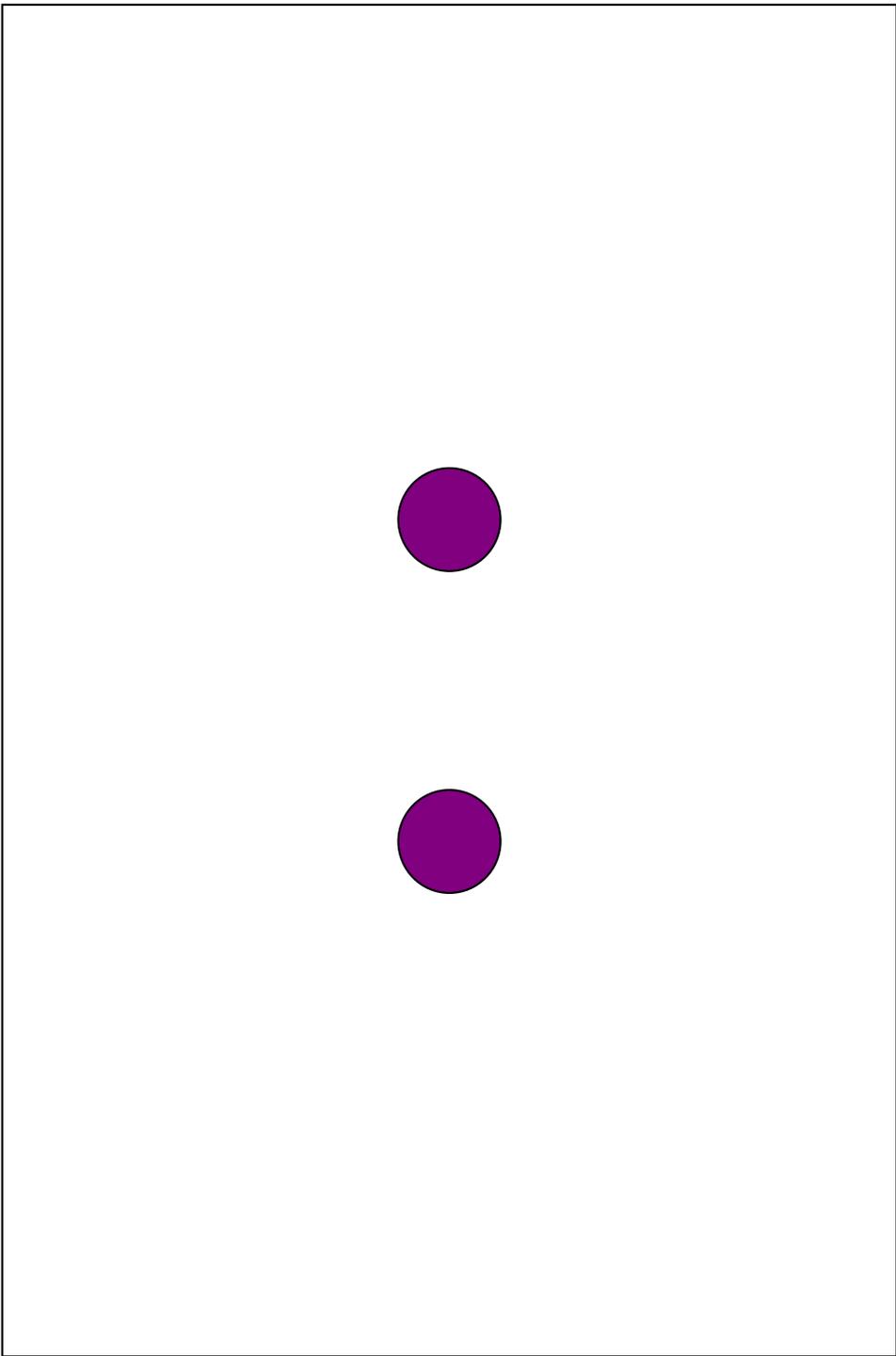
|  |                        |
|--|------------------------|
|  | 6, 2, 2, 2             |
|  | 5, 1, 2, 2, 2, 1       |
|  | 6, 6                   |
|  | 4, 2, 6, 2             |
|  | 3, 1, 10, 1            |
|  | 2, 1, 12, 1            |
|  | 2, 1, 3, 1, 4, 1, 3, 1 |
|  | 1, 2, 12, 2            |
|  | 0, 1, 16, 1            |
|  | 0, 1, 6, 1, 2, 1, 6, 1 |
|  | 0, 1, 7, 2, 7, 1       |
|  | 1, 1, 14, 1            |
|  | 2, 1, 12, 1            |
|  | 2, 1, 5, 2, 5, 1       |
|  | 3, 1, 10, 1            |
|  | 4, 2, 6, 2             |
|  | 6, 6                   |

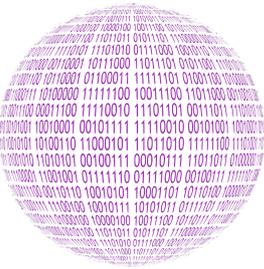












## INFORMATIQUE

### Les nombres entiers et le binaire

#### Un message énigmatique

Arthur allume et éteint successivement et de manière rythmée la salle de bain. On peut imaginer, par exemple, qu'il allume et éteint toutes les dix secondes et qu'il attend plus longtemps quand il change de caractère.

On peut coder 1 la salle de bain allumée et 0 quand elle est éteinte.

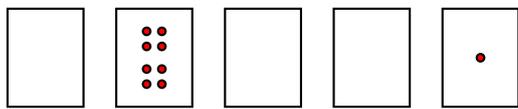
Le code peut se représenter ainsi :

**10011 01111 10011 00000 01010 00101 00000 10011 10101 01001 10011 00000 00101 01110 00110 00101 10010 01101 00101**

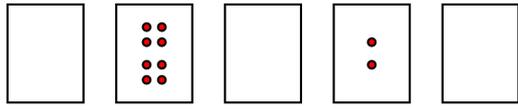
Reste à déterminer le sens de chacun des blocs de 5 bits.

#### La numération binaire

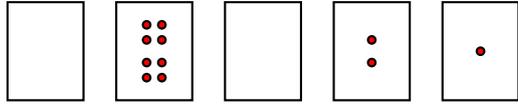
|                          |   |   |   |  |         |
|--------------------------|---|---|---|--|---------|
| <input type="checkbox"/> | <input type="checkbox"/>  | <input type="checkbox"/>                                      | <input type="checkbox"/>                                    | <input type="checkbox"/>                               | 0       |
| <input type="checkbox"/> | <input type="checkbox"/>  | <input type="checkbox"/>                                      | <input type="checkbox"/>                                    | <input style="text-align: center;" type="checkbox"/> • | 1       |
| <input type="checkbox"/> | <input type="checkbox"/>  | <input type="checkbox"/>                                      | <input style="text-align: center;" type="checkbox"/> •<br>• | <input type="checkbox"/>                               | 2       |
| <input type="checkbox"/> | <input type="checkbox"/>  | <input type="checkbox"/>                                      | <input style="text-align: center;" type="checkbox"/> •<br>• | <input style="text-align: center;" type="checkbox"/> • | 3 2+1   |
| <input type="checkbox"/> | <input type="checkbox"/>  | <input style="text-align: center;" type="checkbox"/> ••<br>•• | <input type="checkbox"/>                                    | <input type="checkbox"/>                               | 4       |
| <input type="checkbox"/> | <input type="checkbox"/>  | <input style="text-align: center;" type="checkbox"/> ••<br>•• | <input type="checkbox"/>                                    | <input style="text-align: center;" type="checkbox"/> • | 5 4+1   |
| <input type="checkbox"/> | <input type="checkbox"/>  | <input style="text-align: center;" type="checkbox"/> ••<br>•• | <input style="text-align: center;" type="checkbox"/> •<br>• | <input type="checkbox"/>                               | 6 4+2   |
| <input type="checkbox"/> | <input type="checkbox"/>  | <input style="text-align: center;" type="checkbox"/> ••<br>•• | <input style="text-align: center;" type="checkbox"/> •<br>• | <input style="text-align: center;" type="checkbox"/> • | 7 4+2+1 |
| <input type="checkbox"/> | <input style="text-align: center;" type="checkbox"/> ••<br>••<br>•• | <input type="checkbox"/>                                      | <input type="checkbox"/>                                    | <input type="checkbox"/>                               | 8       |



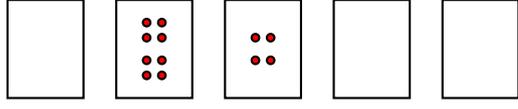
9  $8+1$



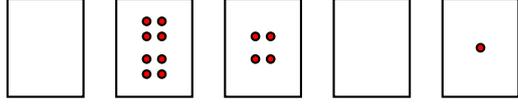
10  $8+2$



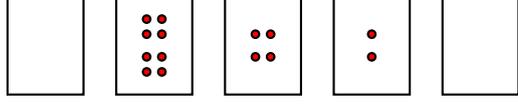
11  $8+2+1$



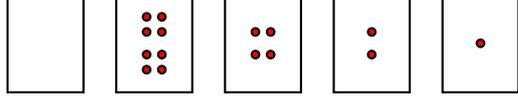
12  $8+4$



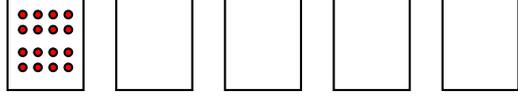
13  $8+4+1$



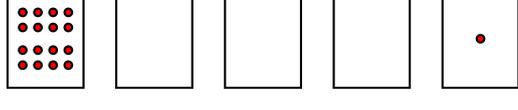
14  $8+4+2$



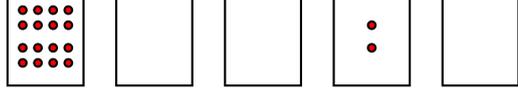
15  $8+4+2+1$



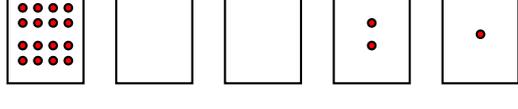
16



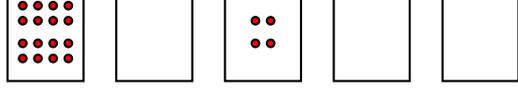
17  $16+1$



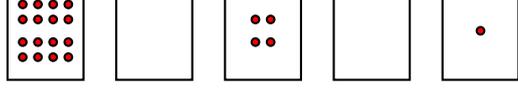
18  $16+2$



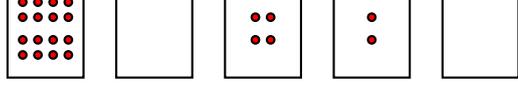
19  $16+2+1$



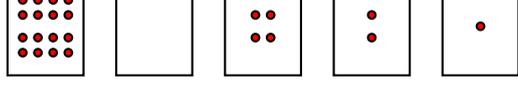
20  $16+4$



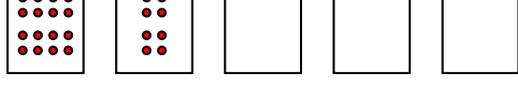
21  $16+4+1$



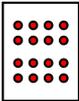
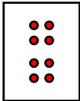
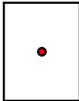
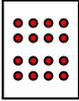
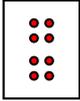
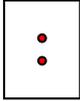
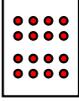
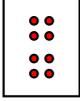
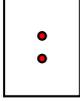
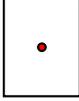
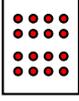
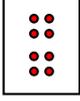
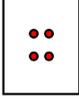
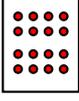
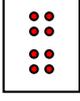
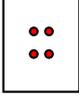
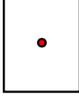
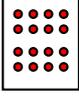
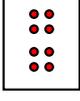
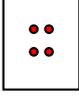
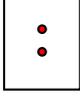
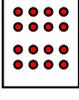
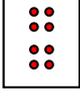
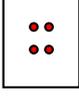
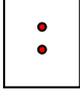
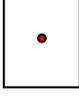
22  $16+4+2$



23  $16+4+2+1$



24  $16+8$

|   |   |   |   |   |    |            |
|---|---|---|---|---|----|------------|
|   |   |   |   |   | 25 | 16+8+1     |
|  |  |  |  |  | 26 | 16+8+2     |
|  |  |  |  |  | 27 | 16+8+2+1   |
|  |  |  |  |  | 28 | 16+8+4     |
|  |  |  |  |  | 29 | 16+8+4+1   |
|  |  |  |  |  | 30 | 16+8+4+2   |
|  |  |  |  |  | 31 | 16+8+4+2+1 |

Compléter :

| Décimal | Binaire |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0       | 0       | 7       | 111     | 14      | 1110    | 21      | 10101   | 28      | 11100   |
| 1       | 1       | 8       | 1000    | 15      | 1111    | 22      | 10110   | 29      | 11101   |
| 2       | 10      | 9       | 1001    | 16      | 10000   | 23      | 10111   | 30      | 11110   |
| 3       | 11      | 10      | 1010    | 17      | 10001   | 24      | 11000   | 31      | 11111   |
| 4       | 100     | 11      | 1011    | 18      | 10010   | 25      | 11001   | 32      | 100000  |
| 5       | 101     | 12      | 1100    | 19      | 10011   | 26      | 11010   |         |         |
| 6       | 110     | 13      | 1101    | 20      | 10100   | 27      | 11011   |         |         |

On peut maintenant décoder le message d'Arthur :

|         |       |       |       |       |       |       |       |       |       |       |       |       |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Binaire | 10011 | 01111 | 10011 | 00000 | 01010 | 00101 | 00000 | 10011 | 10101 | 01001 | 10011 | 00000 |
| Décimal | 19    | 15    | 19    | 0     | 10    | 5     | 0     | 19    | 21    | 9     | 19    | 0     |
| Lettre  | S     | O     | S     |       | J     | E     |       | S     | U     | I     | S     |       |

|         |       |       |       |       |       |       |       |
|---------|-------|-------|-------|-------|-------|-------|-------|
| Binaire | 00101 | 01110 | 00110 | 00101 | 10010 | 01101 | 00101 |
| Décimal | 5     | 14    | 6     | 5     | 18    | 13    | 5     |
| Lettre  | E     | N     | F     | E     | R     | M     | E     |

Le message : **SOS JE SUIS ENFERME**

# Numériser une image

Compléter

|  | Binaire | Compressé  |  | Binaire | Compressé |
|--|---------|------------|--|---------|-----------|
|  | 01110   | 1, 3, 1    |  | 01111   | 1, 4      |
|  | 00001   | 4, 1       |  | 10000   | 0, 1, 4   |
|  | 00001   | 4, 1       |  | 10000   | 0, 1, 4   |
|  | 01111   | 1, 4       |  | 11100   | 0, 3, 2   |
|  | 10001   | 0, 1, 3, 1 |  | 10000   | 0, 1, 4   |
|  | 10001   | 0, 1, 3, 1 |  | 10000   | 0, 1, 4   |
|  | 01111   | 0, 4       |  | 10000   | 0, 1, 4   |

Recommencer en utilisant l'initiale de votre nom ou de votre prénom.

|  | Binaire | Compressé |  | Binaire | Compressé |
|--|---------|-----------|--|---------|-----------|
|  | 10000   | 0, 1, 4   |  | 01111   | 1, 4      |
|  | 10000   | 0, 1, 4   |  | 10000   | 0, 1, 4   |
|  | 10000   | 0, 1, 4   |  | 10000   | 0, 1, 4   |
|  | 10000   | 0, 1, 4   |  | 01111   | 0, 4      |
|  | 10000   | 0, 1, 4   |  | 00001   | 4, 1      |
|  | 10000   | 0, 1, 4   |  | 00001   | 4, 1      |
|  | 01111   | 0, 4      |  | 11110   | 0, 4, 1   |

Décoder les trois images suivantes :

|  |            |
|--|------------|
|  | 4, 11      |
|  | 4, 9, 2, 1 |
|  | 4, 9, 2, 1 |
|  | 4, 11      |
|  | 4, 9       |
|  | 4, 9       |
|  | 5, 7       |
|  | 0, 7       |
|  | 1, 15      |

|  |                        |
|--|------------------------|
|  | 6, 5, 2, 3             |
|  | 4, 2, 5, 2, 3, 1       |
|  | 3, 1, 9, 1, 2, 1       |
|  | 3, 1, 9, 1, 1, 1       |
|  | 2, 1, 11, 1            |
|  | 2, 1, 10, 2            |
|  | 2, 1, 9, 1, 1, 1       |
|  | 2, 1, 8, 1, 2, 1       |
|  | 2, 1, 7, 1, 3, 1       |
|  | 1, 1, 1, 1, 4, 2, 3, 1 |
|  | 0, 1, 2, 1, 2, 2, 5, 1 |
|  | 0, 1, 3, 2, 5, 2       |
|  | 1, 3, 2, 5             |

|  |                        |
|--|------------------------|
|  | 6, 2, 2, 2             |
|  | 5, 1, 2, 2, 2, 1       |
|  | 6, 6                   |
|  | 4, 2, 6, 2             |
|  | 3, 1, 10, 1            |
|  | 2, 1, 12, 1            |
|  | 2, 1, 3, 1, 4, 1, 3, 1 |
|  | 1, 2, 12, 2            |
|  | 0, 1, 16, 1            |
|  | 0, 1, 6, 1, 2, 1, 6, 1 |
|  | 0, 1, 7, 2, 7, 1       |
|  | 1, 1, 14, 1            |
|  | 2, 1, 12, 1            |
|  | 2, 1, 5, 2, 5, 1       |
|  | 3, 1, 10, 1            |
|  | 4, 2, 6, 2             |
|  | 6, 6                   |

# Un tour de magie...

8 9 10 11 12 13 14 15

24 25 26 27 28 29 30 31

40 41 42 43 44 45 46 47

56 57 58 59 60 61 62 63

1 3 5 7 9 11 13 15

17 19 21 23 25 27 29 31

33 35 37 39 41 43 45 47

49 51 53 55 57 59 61 63

32 33 34 35 36 37 38 39

40 41 42 43 44 45 46 47

48 49 50 51 52 53 54 55

56 57 58 59 60 61 62 63

4 5 6 7 12 13 14 15

20 21 22 23 28 29 30 31

36 37 38 39 44 45 46 47

52 53 54 55 60 61 62 63

2 3 6 7 10 11 14 15

18 19 22 23 26 27 30 31

34 35 38 39 42 43 46 47

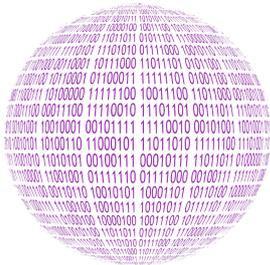
50 51 54 55 58 59 62 63

16 17 18 19 20 21 22 23

24 25 26 27 28 29 30 31

48 49 50 51 52 53 54 55

56 57 58 59 60 61 62 63



## INFORMATIQUE

Le Juste Prix est un jeu télévisé diffusé entre 1987 et 2015. La dernière épreuve de cette émission permettait au candidat de gagner une vitrine constituée de lots prestigieux. Pour cela, il fallait deviner le prix de la vitrine. En 2015, le prix était compris entre 10 000 € et 50 000 €. Le candidat avait alors 30 s pour proposer des prix situés dans cet intervalle. L'animateur du jeu ne pouvait répondre qu'en disant « c'est plus » ou « c'est moins ».

Vous trouverez un exemple de ce jeu en suivant le lien ci-contre.



### La dichotomie

Une des méthodes efficaces pour gagner à ce jeu consiste à utiliser la méthode de **dichotomie**.

Voici cet algorithme pour trouver un nombre entier mystérieux  $x$  compris entre deux nombres entiers  $a$  et  $b$  :

**Étape n° 1** Déterminer le centre de l'intervalle entre  $a$  et  $b$ , on le nomme  $c$

**Étape n° 2** On compare  $x$  et le nombre  $c$ , il y a trois possibilités :

- Si  $x > c$ , on décide que  $a = c$  puis on reprend l'Étape n° 1
- Si  $x < c$ , on décide que  $b = c$  puis on reprend l'Étape n° 1
- Si  $x = c$ , c'est gagné, on stoppe l'algorithme

1. Appliquer cet algorithme pour trouver le nombre  $x = 78$  compris entre les deux nombres entiers 1 et 100.

Compléter le tableau suivant :

|                | $a$ | $b$ | $c$ | C'est plus ou c'est moins? |
|----------------|-----|-----|-----|----------------------------|
| Itération n° 1 | 1   | 100 |     |                            |
| Itération n° 2 |     |     |     |                            |
| Itération n° 3 |     |     |     |                            |
| Itération n° 4 |     |     |     |                            |
| Itération n° 5 |     |     |     |                            |
| Itération n° 6 |     |     |     |                            |
| Itération n° 7 |     |     |     |                            |

2. Comment avez-vous fait pour calculer le nombre  $c$  à chaque étape?

3. Recommencer cette expérience avec un nombre  $x = 699$  compris entre 100 et 1000

Compléter le tableau suivant :

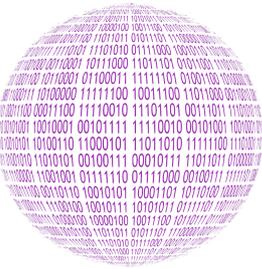
|                 | $a$ | $b$  | $c$ | C'est plus ou c'est moins? |
|-----------------|-----|------|-----|----------------------------|
| Itération n° 1  | 100 | 1000 |     |                            |
| Itération n° 2  |     |      |     |                            |
| Itération n° 3  |     |      |     |                            |
| Itération n° 4  |     |      |     |                            |
| Itération n° 5  |     |      |     |                            |
| Itération n° 6  |     |      |     |                            |
| Itération n° 7  |     |      |     |                            |
| Itération n° 8  |     |      |     |                            |
| Itération n° 9  |     |      |     |                            |
| Itération n° 10 |     |      |     |                            |

4. Combien d'itérations faut-il pour trouver un nombre compris entre 1 et 100? Entre 1 et 1000? Entre 1 et 10 000?

5. Testez une partie de Juste Prix, en partant de l'intervalle 10 000 € et 50 000 €, avec votre voisin.

6. Programmez dans Codablock l'algorithme du Juste Prix. Le programme demande au départ la valeur minimale puis maximale de l'intervalle. L'ordinateur vous propose des nombres entiers et vous devez lui répondre par « C'est plus » ou « C'est moins ». Faire apparaître le nombre d'étapes.

Code Capytale : **2661-1878385**



## INFORMATIQUE

1. Appliquer cet algorithme pour trouver le nombre  $x = 78$  compris entre les deux nombres entiers 1 et 100.  
Compléter le tableau suivant :

|                | <i>a</i> | <i>b</i> | <i>c</i> | C'est plus ou c'est moins ? |
|----------------|----------|----------|----------|-----------------------------|
| Itération n° 1 | 1        | 100      | 50       | +                           |
| Itération n° 2 | 50       | 100      | 75       | +                           |
| Itération n° 3 | 75       | 100      | 88       | -                           |
| Itération n° 4 | 75       | 88       | 82       | -                           |
| Itération n° 5 | 75       | 82       | 79       | -                           |
| Itération n° 6 | 75       | 79       | 77       | +                           |
| Itération n° 7 | 77       | 79       | 78       |                             |

3. Appliquer cet algorithme pour trouver le nombre  $x = 599$  compris entre les deux nombres entiers 100 et 1000.  
Compléter le tableau suivant :

|                 | <i>a</i> | <i>b</i> | <i>c</i> | C'est plus ou c'est moins ? |
|-----------------|----------|----------|----------|-----------------------------|
| Itération n° 1  | 100      | 1000     | 550      | +                           |
| Itération n° 2  | 550      | 1000     | 775      | -                           |
| Itération n° 3  | 500      | 775      | 613      | -                           |
| Itération n° 4  | 500      | 613      | 557      | +                           |
| Itération n° 5  | 557      | 613      | 586      | +                           |
| Itération n° 6  | 586      | 613      | 600      | -                           |
| Itération n° 7  | 586      | 600      | 593      | +                           |
| Itération n° 8  | 593      | 600      | 597      | +                           |
| Itération n° 9  | 597      | 600      | 598      | +                           |
| Itération n° 10 | 598      | 600      | 599      |                             |



SCRATCH



## LE PORTAIL SÉCURISÉ — ÉPISODE I

### INFORMATIQUE COLLÈGE



**OBJECTIFS :** Nous souhaitons mettre en place un portail sécurisé pour une résidence. Il s'agit de programmer la barrière du portail ainsi que la télécommande des utilisateurs pour que l'entrée soit protégée par un mot de passe.

Dans un premier temps, cet épisode, nous allons modéliser ce portail dans Scratch et mettre en place virtuellement le système. Par la suite nous transférerons nos méthodes vers des cartes Microbit.

### PREMIÈRE PARTIE — Le mot de passe

👉 Réaliser dans Scratch un programme qui demande un mot de passe à l'utilisateur et qui vérifie qu'il s'agit bien de celui attendu.

### DEUXIÈME PARTIE — Les trois essais

👉 Modifier le programme précédent de telle manière que l'utilisateur puisse faire au maximum trois essais. Indiquer à chaque fois le numéro de l'essai. En cas d'échec trois fois de suite, faire un message à l'utilisateur.

### TROISIÈME PARTIE — Le portail

Pour l'instant, le portail de ma résidence n'est pas protégé. Pour l'ouvrir il suffit d'appuyer sur le bouton vert. Une fois appuyé sur le bouton vert, le portail se referme 10 s plus tard.



Nous l'avons modélisé dans Scratch. Le fichier dont vous disposez ne contient que les sprites nécessaires au programme. Chacun propose plusieurs costumes pour obtenir la simulation.

On souhaite sécuriser le portail à l'aide d'un code simple : le portail ne s'ouvre que si l'utilisateur a appuyé sept fois de suite sur le bouton vert.



👉 Programmer le code nécessaire à la réalisation de cette fonction.

### QUATRIÈME PARTIE — Sécurisation du portail avec deux boutons

Le code précédent n'est pas trop sécurisé. Il suffit d'appuyer au hasard sur le bouton vert pour ouvrir le portail.

Pour améliorer la situation on a ajouté un deuxième bouton, un bouton rouge. Ce bouton permet de valider ce qui est saisi avec le bouton vert, ce qui évite les tentatives au hasard.

En cas d'erreur de code, le portail est bloqué pendant 10 s sans qu'il soit possible de saisir un nouveau code.

👉 Programmez cette nouvelle situation.

### CINQUIÈME PARTIE — Sécurisation du portail avec un code

Finalement, nous allons nous équiper d'un clavier numérique à dix chiffres. Nous pourrions ainsi imposer un code à quatre chiffres pour chaque résident.

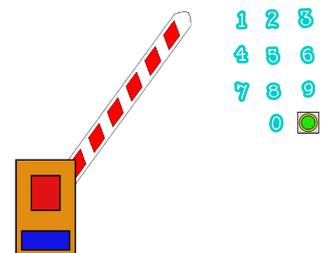
👉 Combien de codes différents sont-ils possibles ?

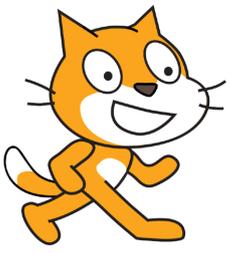
En consultant la bibliothèque de sprites disponibles dans Scratch, vous trouverez les dix chiffres. Placez les sur l'écran de telle manière que l'utilisateur puisse cliquer sur les chiffres pour saisir son code.

👉 Créer le code permettant au portail de s'ouvrir quand un utilisateur saisi le bon code à quatre chiffres.

**Bonus :** il y a dix locataires dans cet immeuble ayant chacun un code personnel à quatre chiffres.

👉 Programmez le portail pour qu'il puisse s'ouvrir lorsque l'un de ces dix codes est saisi.





SCRATCH



LE PORTAIL SÉCURISÉ — ÉPISODE I — Correction



NON



COMMENT FAIRE...

Les structures de contrôle conditionnelles sont communes à la plupart des langages de programmation. Elles permettent d'exprimer des conditions du type : « Si ... Alors »— « Si ... Alors ... Sinon »— « Tant que ... Alors ».

Scratch propose quatre blocs permettant de rendre compte d'une condition dans l'exécution d'un programme.



Dans Scratch, l'expression de la condition se représente sous la forme d'un bloc hexagonal nommé **capteur** ou **opérateur**. Dans les langages classiques, cette condition s'appelle une **expression booléenne**. Une expression booléenne ne peut prendre que deux valeurs : VRAI ou FAUX.

*L'adjectif booléen fait référence au mathématicien britannique Georges Boole (1815-1864), qui a étudié les règles de calcul sur les ensembles constitués de deux valeurs (0 ou 1).*

La plupart des langages de programmation supportent les opérations booléennes et l'algèbre de Boole telles que la **conjonction** (ET), la **disjonction** (OU), l'**équivalence** (=), la **non-équivalence** ( $\neq$ ) et la **négation** (NON).

Voici les blocs élémentaires permettant de construire des expressions booléennes dans Scratch :

Les opérateurs :



Les capteurs :



Chacune de ces expressions booléennes ne renvoie que deux valeurs possibles : VRAI ou FAUX.

On peut combiner les expressions booléennes en utilisant les opérateurs spéciaux suivants :

Les opérateurs logiques :



Voici les tables de calculs de ces opérateurs :

| ET   | VRAI | FAUX |
|------|------|------|
| VRAI | VRAI | FAUX |
| FAUX | FAUX | FAUX |

| OU   | VRAI | FAUX |
|------|------|------|
| VRAI | VRAI | VRAI |
| FAUX | VRAI | FAUX |

| NON  |      |
|------|------|
| VRAI | FAUX |
| FAUX | VRAI |

Exemples : indiquer dans chaque cas la valeur de la variable **A** affichée à la fin de chacun des programmes.

```

quand est cliqué
  Mettre A à 7
  Répéter 5 fois
    Si A < 10 alors
      Ajouter 6 à A
    sinon
      Ajouter -3 à A
  Dire A pendant 2 secondes
  
```

```

quand est cliqué
  Mettre A à 75
  Mettre B à 23
  Répéter jusqu'à ce que A < B
    Mettre A à A - B
    Mettre B à B - 4
  Dire A pendant 2 secondes
  
```

```

quand est cliqué
  Mettre A à 2
  Mettre B à 3
  Répéter jusqu'à ce que Non (A < B)
    Mettre A à A + B
    Mettre B à B * A
  Dire A pendant 2 secondes
  
```



COMMENT FAIRE...





## ALGORITHMIQUE — Le code secret

### PREMIÈRE PARTIE — Le mot de passe

Voici un programme réalisé avec Scratch. Il demande à l'utilisateur un mot de passe et vérifie s'il s'agit bien de celui attendu.

- Se rendre à l'URL : <https://scratch.mit.edu/users/scratch3/> avec le navigateur;
- construire le programme débuté ci-dessous en complétant les blocs manquants;
- changer le mot de passe et choisir « Mathématiques ».

```

quand [drapeau] est cliqué
mettre [Mot de passe] à [Mirzakhani]
demander [Quel est le mot de passe?] et attendre
si [ ] = [ ]
  dire [Vous pouvez rentrer!] pendant [2] secondes
sinon
  dire [Ce n'est pas le bon mot de passe!] pendant [2] seconde

```

### DEUXIÈME PARTIE — Le mot de passe — Épisode 2

Modifier le programme précédent de telle manière que l'utilisateur puisse faire au maximum trois essais. Indiquer à chaque fois le numéro de l'essai. En cas d'échec trois fois de suite, faire un message à l'utilisateur.

Voici quelques blocs qui pourraient vous être utiles :

```

[Essai] ajouter [1] à [Essai]
si [ ] = [ ]

```

### TROISIÈME PARTIE — Le portail

Le portail de ma résidence n'est pas protégé. Pour l'ouvrir il suffit d'appuyer sur le bouton vert. Une fois appuyé sur le bouton vert, le portail se referme 10 s plus tard.

Nous l'avons modélisé dans Scratch.

- Se rendre sur la page des quatrièmes du blog : <https://arnaud.ac3j.fr> ;
- télécharger et enregistrer le fichier **Portail.sb3**;
- importer ce fichier dans Scratch.
- modifier le programme pour qu'il se ferme au bout de 5 s.



On souhaite maintenant sécuriser le portail à l'aide d'un code simple : le portail ne s'ouvre que si l'utilisateur a appuyé sept fois de suite sur le bouton vert.

Ajouter cette fonctionnalité dans le programme Scratch précédent.

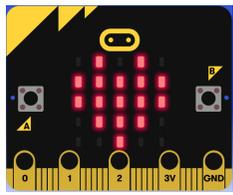
### QUATRIÈME PARTIE — Sécurisation du portail

Le code précédent n'est pas trop sécurisé. Pour améliorer la situation on a ajouté un bouton rouge. Ce bouton permet de valider ce qui est saisi avec le bouton vert, ce qui permet d'éviter les tentatives au hasard.

En cas d'erreur de code, le portail est bloqué pendant 10 s sans qu'il soit possible de saisir un nouveau code.

- Télécharger le fichier **Portail\_securise.sb3** depuis la page du blog;
- importer le fichier dans Scratch;
- modifier le programme pour obtenir le résultat attendu.

*À la fin de la séance, votre travail enregistré doit être envoyé en passant par le formulaire disponible sur le blog!*



## ALGORITHMIQUE — Le code secret

### PREMIÈRE PARTIE — Ouvrir une porte

Voici un début de programme réalisé avec Microbit. Il permet d'afficher le message « Porte ouverte » quand on appuie sur le **bouton A**.

Reproduire ce programme en vous connectant sur le site de Microbit :

- Lancer le navigateur;
- rendez-vous à l'URL : <https://makecode.microbit.org>;
- créer un nouveau projet;
- chercher dans le menu les blocs demandés.



Compléter ce script pour que :

- Lorsque l'on appuie sur le **bouton B** le Microbit affiche « Porte fermée »;
- avant le message « Porte ouvert », faire apparaître un carré pendant 3 s;
- avant le message « Porte fermée », faire apparaître un carré avec une croix à l'intérieur pendant 3 s.

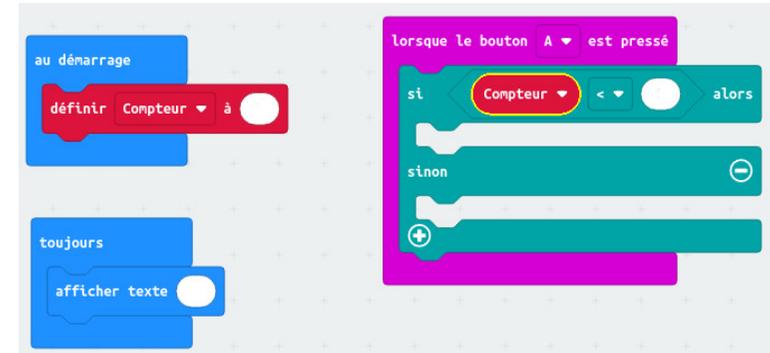
### DEUXIÈME PARTIE — L'affichage numérique

On souhaite utiliser le Microbit comme clavier numérique pour saisir un code d'entrée. Comme il n'y a que deux boutons nous avons imaginé ceci :

- La matrice affiche les chiffres de 0 à 9;
- quand on appuie sur le **bouton B** on passe au chiffre suivant;
- quand on appuie sur le **bouton A** on passe au chiffre précédent;
- le chiffre qui suit le 9 est le 0;

- le chiffre qui précède le 0 est le 9;
- au départ le chiffre 0 est affiché.

Voici le début du programme. Le saisir dans Microbit puis le modifier pour répondre aux six contraintes.



### TROISIÈME PARTIE — Le code secret

On reprend le fonctionnement de l'affichage numérique de la deuxième partie.

On souhaite maintenant ajouter une validation du code à une chiffre saisi. Voici les demandes :

- Le **bouton A** garde le même rôle;
- le **bouton B** sert à valider le chiffre qui apparaît sur la matrice;
- si le chiffre validé est le code secret un carré est affiché sur l'écran;
- si le chiffre validé n'est pas le bon code, un carré avec une croix est affiché;
- le code secret est 6

### QUATRIÈME PARTIE — Sécurisation du code

Le code secret précédent est un nombre compris entre 0 et 9. C'est beaucoup trop facile à trouver.

On souhaite maintenant que le code d'entrée soit un nombre à deux chiffres. Voici les contraintes :

- Le **bouton A** et le bouton **le bouton B** gardent le même rôle;
- le premier chiffre validé correspond au chiffre des dizaines du code;
- le second chiffre validé correspond au chiffre des unités;
- l'affichage est le même que précédemment;
- le code secret est 69

---

## Remarques et intentions pédagogiques

---

### <sup>1</sup> ACTIVITÉ — UNE AFFICHE POUR LE CVC

Les intentions

### <sup>2</sup> ACTIVITÉ — QUE CONTIENT UN FICHIER ?

En cours de rédaction

# INFORMATIONS LÉGALES

- **Auteur** : Fabrice ARNAUD
- **Web** : pi.ac3j.fr
- **Mail** : contact@ac3j.fr
- **Dernière modification** : 23 juin 2024 à 16:20

Ce document a été écrit pour L<sup>A</sup>T<sub>E</sub>X avec l'éditeur VIM - Vi Improved Vim 9.1.  
Il a été compilé sous Linux Ubuntu Noble Numbat 24.04 avec la distribution TeX Live 2023.20240207-101 et LuaHBTeX 1.17.0

Pour compiler ce document, un fichier comprenant la plupart des macros est nécessaires. Ce fichier, Entete.tex, est encore trop mal rédigé pour qu'il puisse être mis en ligne. Il est en cours de réécriture et permettra ensuite le partage des sources dans de bonnes conditions.  
Le fichier source a été réalisé sous Linux Ubuntu avec l'éditeur Vim. Il utilise une balise spécifique à Vim pour permettre une organisation du fichier sous forme de replis. Cette balise %{{{ ... %}}} est un commentaire pour L<sup>A</sup>T<sub>E</sub>X, elle n'est pas nécessaire à sa compilation. Vous pouvez l'utiliser avec Vim en lui précisant que ce code définit un repli. Je vous laisse consulter la documentation officielle de Vim à ce sujet.

## LICENCE CC BY-NC-SA 4.0



### Attribution Pas d'Utilisation Commerciale Partage dans les Mêmes Conditions 4.0 International

Ce document est placé sous licence CC-BY-NC-SA 4.0 qui impose certaines conditions de ré-utilisation.

#### Vous êtes autorisé à :

- Partager** — copier, distribuer et communiquer le matériel par tous moyens et sous tous formats
- Adapter** — remixer, transformer et créer à partir du matériel

L'Offrant ne peut retirer les autorisations concédées par la licence tant que vous appliquez les termes de cette licence.

#### Selon les conditions suivantes :

- Attribution** — Vous devez créditer l'Œuvre, intégrer un lien vers la licence et indiquer si des modifications ont été effectuées à l'Œuvre. Vous devez indiquer ces informations par tous les moyens raisonnables, sans toutefois suggérer que l'Offrant vous soutient ou soutient la façon dont vous avez utilisé son œuvre.
- Pas d'Utilisation Commerciale** — Vous n'êtes pas autorisé à faire un usage commercial de cette Œuvre, tout ou partie du matériel la composant.
- Partage dans les Mêmes Conditions** — Dans le cas où vous effectuez un remix, que vous transformez, ou créez à partir du matériel composant l'Œuvre originale, vous devez diffuser l'œuvre modifiée dans les mêmes conditions, c'est à dire avec la même licence avec laquelle l'œuvre originale a été diffusée.
- Pas de restrictions complémentaires** — Vous n'êtes pas autorisé à appliquer des conditions légales ou des mesures techniques qui restreindraient légalement autrui à utiliser l'Œuvre dans les conditions décrites par la licence.

Consulter : <https://creativecommons.org/licenses/by-sa/4.0/deed.fr>

#### Comment créditer cette Œuvre ?

Ce document, **Cours.pdf**, a été créé par **Fabrice ARNAUD (contact@ac3j.fr)** le 23 juin 2024 à 16:20.

Il est disponible en ligne sur **pi.ac3j.fr**, **Le blog de Fabrice ARNAUD**.

Adresse de l'article : <https://pi.ac3j.fr/mathematiques-college>.