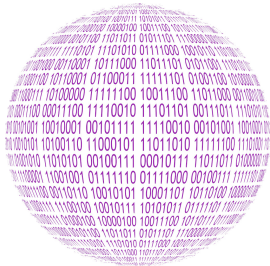


CHAPITRE III



Option informatique en troisième

I — Les outils informatiques



INFORMATIQUE

Un peu de mathématiques

- 1.a. Combien de mot de passe de 8 caractères différents peut-on créer en utilisant seulement des chiffres ?
- 1.b. Combien de mot de passe de 8 caractères différents peut-on créer en utilisant seulement des lettres majuscules ?
- 1.c. Combien de mot de passe de 8 caractères différents peut-on créer en utilisant seulement des chiffres, des lettres majuscules et des lettres minuscules ?
- 1.d. Combien de mot de passe de 8 caractères différents peut-on créer en utilisant seulement des chiffres, des lettres majuscules, des lettres minuscules et 32 caractères spéciaux ?

La méthode par force brute

Une attaque par force brute (bruteforce attack) consiste à tester, l'une après l'autre, chaque combinaison possible d'un mot de passe ou d'une clé pour un identifiant donné afin se connecter au service ciblé.

Il s'agit d'une méthode ancienne et répandue chez les pirates. Le temps nécessaire à celle-ci dépend du nombre de possibilités, de la vitesse que met l'attaquant pour tester chaque combinaison et des défenses qui lui sont opposées.

Ce type d'attaque étant relativement simple, un organisme peut disposer de systèmes permettant de se protéger de ce type de comportement. La première ligne du système de défense est le blocage de comptes après un nombre limité d'échecs d'authentification pour un même identifiant. Cette méthode consiste à déterminer le mot de passe ciblé en testant toutes les possibilités.

```
import time

liste_chiffre = ["0","1","2","3"]
liste_lettre = ["A","B","C","D"]
liste = liste_chiffre + liste_lettre

print("Quel mot de passe voulez-vous tester ? ")
mot_de_passe = input()

stop = False

debut = time.perf_counter()

for caractere1 in liste:
    for caractere2 in liste:
        for caractere3 in liste:
            if caractere1+caractere2+caractere3 == mot_de_passe:
                fin = time.perf_counter()
                stop = True
                break
        if stop:
            break
    if stop:
        break

print("J'ai trouvé ton mot de passe")
print("Il s'agit de "+caractere1+caractere2+caractere3+" en {fin-debut: .5f} secondes" )
```

Modifier le code ci-dessus en tenant compte de tous les chiffres et de l'alphabet complet en majuscule et en minuscule.

Modifier également pour tester un code de 5 caractères.

La méthode du dictionnaire

L'attaque par dictionnaire est une méthode utilisée en cryptanalyse pour trouver un mot de passe ou une clé. Elle consiste à tester une série de mots de passe potentiels, les uns à la suite des autres, en espérant que le mot de passe utilisé pour le chiffrement soit contenu dans le dictionnaire. Si ce n'est pas le cas, l'attaque échouera.

Cette méthode repose sur le fait que de nombreuses personnes utilisent des mots de passe courants (par exemple : un prénom, une couleur ou le nom d'un animal). C'est pour cette raison qu'il est toujours conseillé de ne pas utiliser de mot de passe comprenant un mot ou un nom.

L'attaque par dictionnaire est une méthode souvent utilisée en complément de l'attaque par force brute qui consiste à tester, de manière exhaustive, les différentes possibilités de mots de passe. Cette dernière est particulièrement efficace pour des mots de passe n'excédant pas 5 ou 6 caractères.

Tester le programme suivant :

```
import time

print("Choisir un mot de passe : ")
mot_de_passe = input()
fin = 0

debut = time.perf_counter()

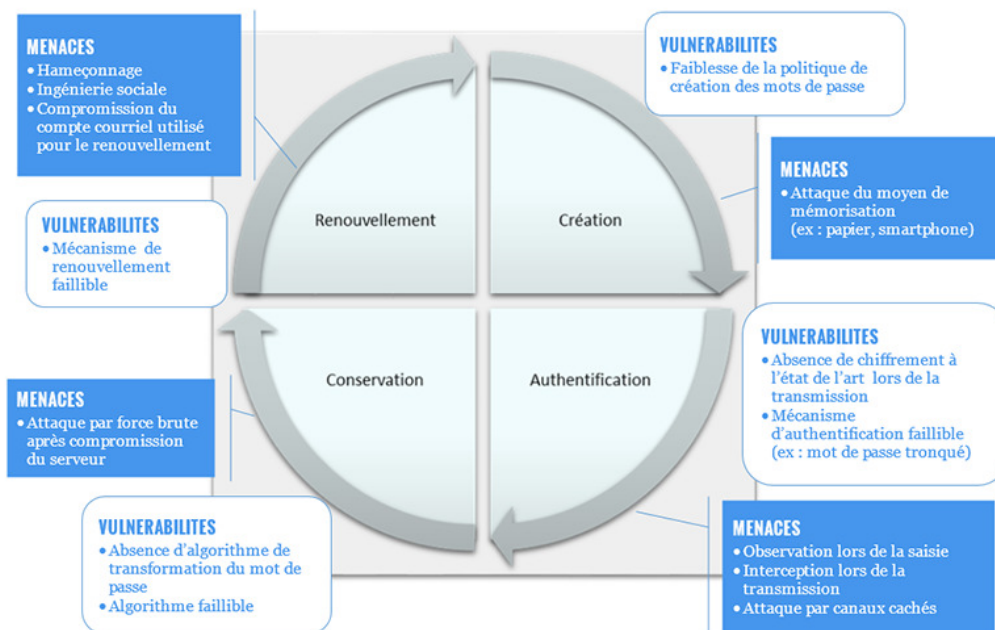
fichier = open("rockyou.txt","r",encoding="latin1")

for mot in fichier:
    if mot_de_passe == mot[:-1]:
        fin = time.perf_counter()
        break

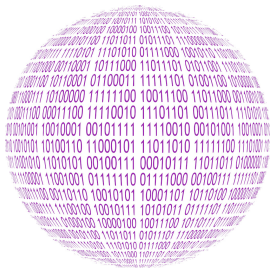
if fin == 0:
    print("Je n'ai pas trouvé ton mot de passe")
else:
    print("J'ai trouvé ton mot de passe")
    print("Il s'agit de "+mot[:-1]+" en : {fin-debut: .5f} secondes")
```

Votre mot de passe est-il sûr ?

Rendez-vous sur le site <https://haveibeenpwned.com/Passwords> pour tester votre mot de passe.







INFORMATIQUE

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.

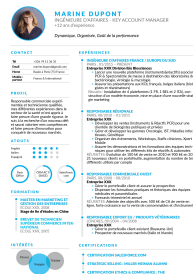
Inkscape est un logiciel de dessin vectoriel libre multiplateforme. Il gère des fichiers conformes aux standards XML, SVG et CSS du W3C. Le logiciel est intégré à la liste des logiciels libres préconisés par l'État français dans le cadre de la modernisation globale de ses systèmes d'information. Il a 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 : Vous souhaitez postuler sur une offre d'emploi. On vous demande un CV (« Curriculum vitæ », du latin « currere », courir et « vitæ » vie, la course de la vie ou déroulement de la vie). En naviguant sur internet, vous avez sélectionné six modèles de CV dont la présentation vous correspond. Il existe en ligne, de nombreux services payants ou presque gratuit, qui vous permettent de préparer un CV ayant une présentation graphique récente. Comme vous avez des compétences particulières en informatique, vous faites le choix de rédiger ce CV directement sur votre ordinateur ce qui vous permettra d'obtenir la charte graphique idéale sans avoir à payer un service supplémentaire.

Pour réaliser cet objectif, vous allez utiliser le logiciel libre de dessin vectoriel Inkscape.

Voici quelques exemples de CV qui pourraient vous servir de modèle :

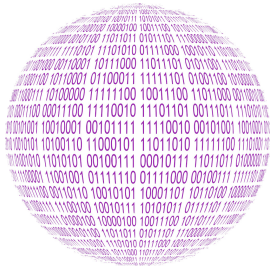


En vous rendant sur la page <https://arnaud.ac3j.fr/CV> vous trouverez :

- Les six CV ;
- 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 un emploi réel sur lequel vous souhaitez postuler, pas de limite pour votre imagination !
 - Inventez vous un CV qui correspond à cet emploi, formation, expériences professionnelles... Toutes les informations pourraient être vraies, si vous n'étiez pas un collégien...
- **La forme**
 - Partez du modèle fourni, il contient plusieurs calques pour chaque partie essentielle du CV ;
 - Inutile de chercher un CV tout prêt, 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 une photo personnelle ou une photo trouvée sur le Web. Il ne faut pas y passer trop de temps !

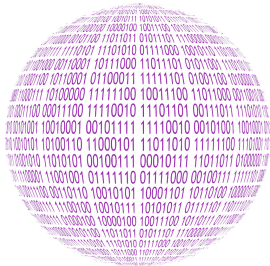


INFORMATIQUE



CURRICULUM VITÆ — Correction





INFORMATIQUE

Le Jeu de la vie est inventé en 1970 par John Conway, professeur de mathématiques à l'université de Cambridge, au Royaume-Uni.

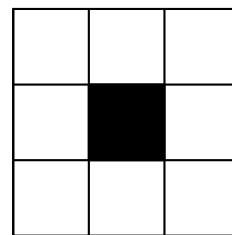
Conway s'intéresse alors à un problème proposé par le mathématicien John Leech dans le domaine de la théorie des groupes et qui avait trait à l'empilement dense de sphères à 24 dimensions. Il découvre quelques propriétés remarquables et publie les résultats de son étude en 1968. Conway est également intéressé par un problème présenté vers les années 1940 par un mathématicien renommé : John von Neumann.

Gardner écrit dans ses colonnes que « le Jeu de la vie rendit Conway rapidement célèbre et il ouvrit aussi un nouveau champ de recherche mathématique, celui des automates cellulaires. En effet, les analogies du jeu de la vie avec le développement, le déclin et les altérations d'une colonie de micro-organismes, le rapprochent des jeux de simulation qui miment les processus de la vie réelle. »

Les règles du jeu

Le Jeu de la vie est un « jeu à zéro joueur », puisqu'il ne nécessite aucune intervention du joueur lors de son déroulement. Il s'agit d'un automate cellulaire, un modèle où chaque état conduit mécaniquement à l'état suivant à partir de règles préétablies.

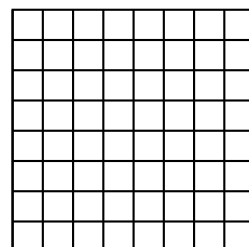
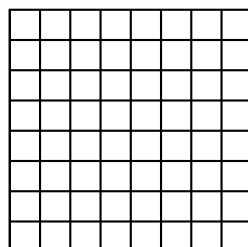
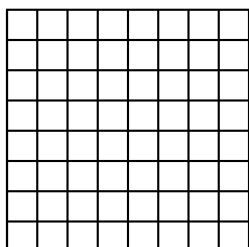
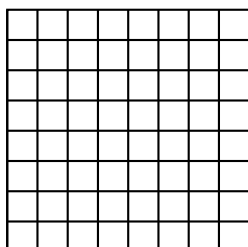
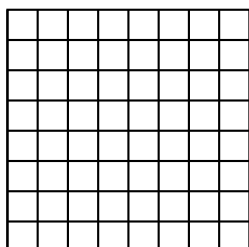
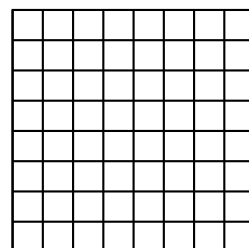
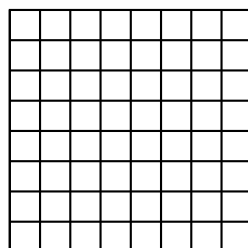
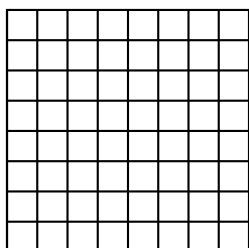
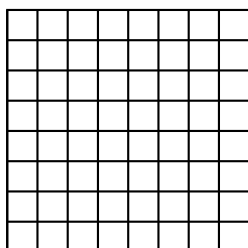
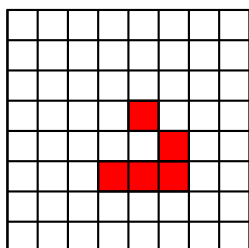
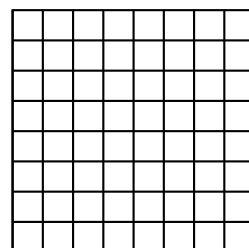
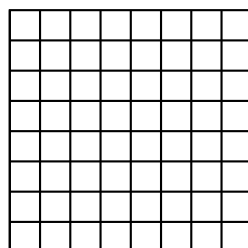
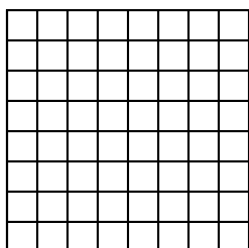
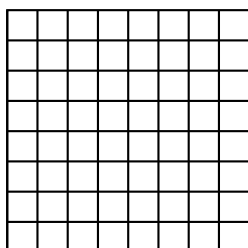
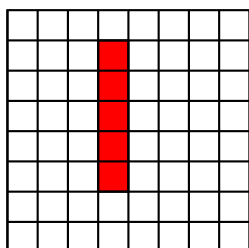
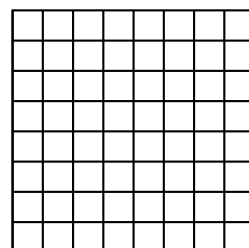
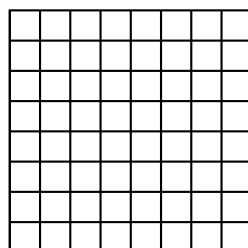
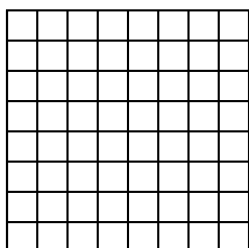
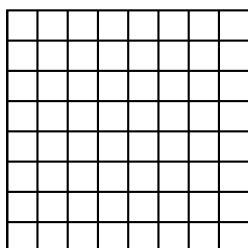
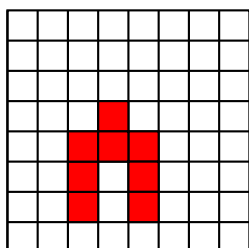
Le jeu se déroule sur une grille à deux dimensions, théoriquement infinie, dont les cases — appelées « cellules », par analogie avec les cellules vivantes — peuvent prendre deux états distincts : « vivante » ou « morte ».



Une cellule possède huit voisines, qui sont les cellules adjacentes horizontalement, verticalement et diagonalement.

- une cellule morte possédant exactement trois cellules voisines vivantes devient vivante (elle naît) ;
- une cellule vivante possédant deux ou trois cellules voisines vivantes le reste, sinon elle meurt.

À la main



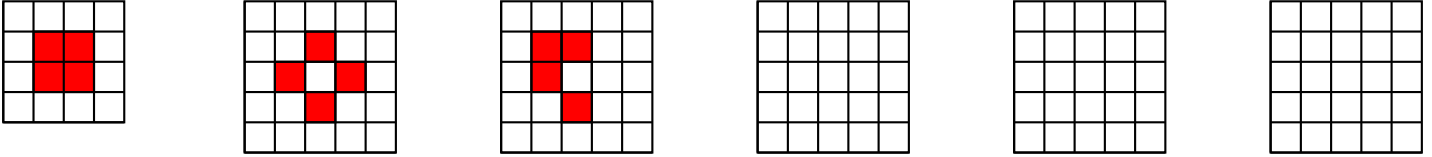
Il existe de nombreuses applications qui simulent le jeu de la vie de Conway.
 Nous utiliserons le site <https://playgameoflife.com/> pour tester ce jeu.

Reprendre les quatre modèles que vous avez testé à la main avec cette application en ligne.

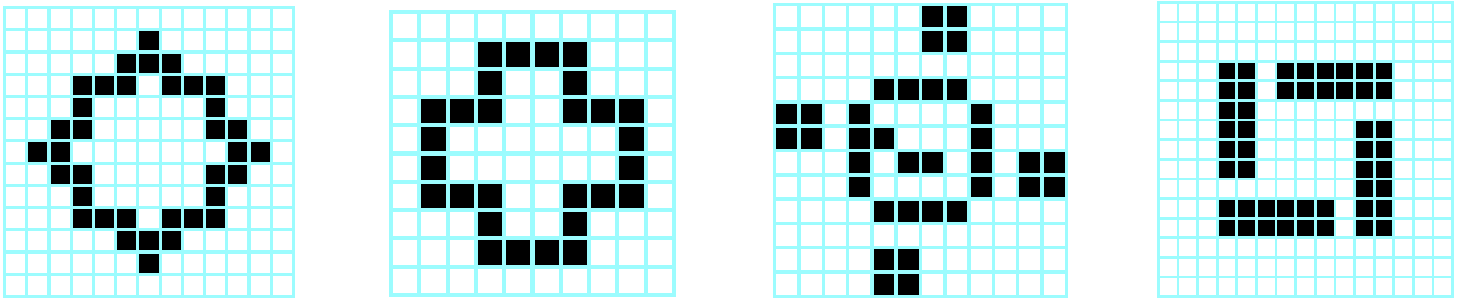
Structures remarquables

Depuis la création de ce jeu par John Conway, de nombreuses structures remarquables ont été découvertes.
 Voici quelques exemples. Testez chacun d'entre eux sur le site en ligne.

LES STRUCTURES STABLES

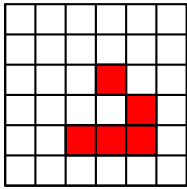


LES OSCILATEURS



LES VAISSEAUX

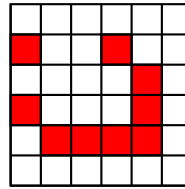
Dans un automate cellulaire, un vaisseau, ou navire, est un objet qui réapparaît au bout d'un certain nombre de générations dans une position différente.



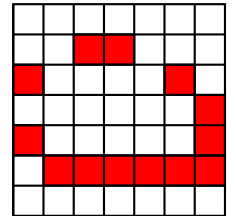
Le glider

Le planeur est une structure du Jeu de la vie, plus exactement le plus petit vaisseau qui existe dans cet automate cellulaire.

Ce symbole porte aussi le nom de glider. Il s'agit du « symbole des hackers ». Dans ce contexte, « hacker » ne désigne pas les crackers mais la culture hacker autour de BSD, MIT, GNU, Linux, Perl... ainsi que la communauté autour du logiciel libre et de l'open source. En utilisant ce symbole, vous exprimez votre accord avec les objectifs des hackers, leurs valeurs, ainsi que leur mode de vie.



Light Weight Spaceship



Heavy Weight Spaceship

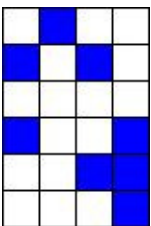
LES PUFFEURS, LES CANONS ET AUTRES CURIOSITÉS

Un puffeur est un objet du jeu de la vie se déplaçant, mais, contrairement au vaisseau, le puffeur laisse un plus ou moins grand nombre de débris (cela varie entre un seul type de débris et un large amas de débris variés)

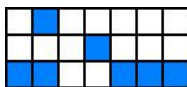
Un canon est un motif fini dont la partie principale se répète périodiquement, comme un oscillateur, et qui émet des vaisseaux à intervalles réguliers.

Un mathusalem est un motif qui met un certain moment avant de se stabiliser en une constellation de débris plus ou moins importante.

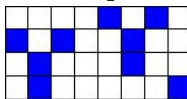
Un puffeur diagonal



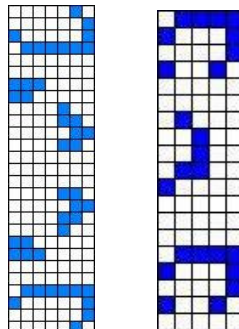
Le gland



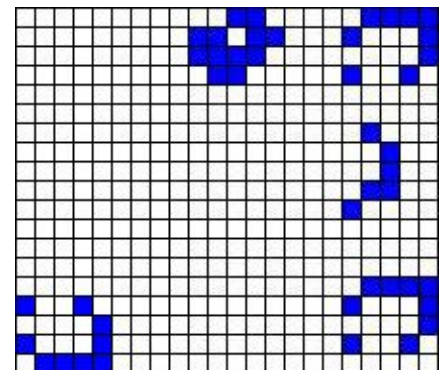
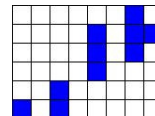
Les lapins



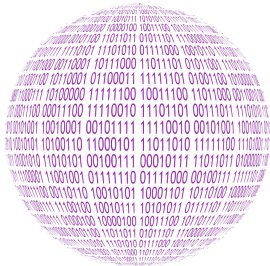
Les deux premiers puffers



Le plus petit puffeur



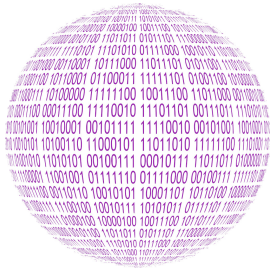
Un rake : un canon à vaisseaux



LE JEU DE LA VIE — Correction



INFORMATIQUE



INFORMATIQUE

Logo est une famille de langages de programmation, né à la fin des années 1960 de la rencontre entre le courant cognitiviste en intelligence artificielle et des théories sur l'apprentissage issues de travaux de Jean Piaget et de ses conceptions en matière d'éducation.

L'appellation, inspirée du grec Logos « parole, discours, intelligence », recouvre, donc, deux concepts étroitement liés quoique distincts : un mode d'apprentissage inspiré des travaux de Jean Piaget sur le développement cognitif de l'enfant et un type d'environnement informatique.

La bibliothèque Turtle de Python est une implémentation de ce langage. Il permet de dessiner des formes complexes en utilisant un programme qui répète des actions élémentaires. En Python, la tortue graphique est semblable à un petit robot avec un crayon qui peut dessiner sur le plan.

Ce langage est réputé pour son usage dans le domaine de l'éducation, il a été créé pour des professeurs pour son usage en classe. Il est dorénavant utile pour les développeurs qui ont besoin de produire des graphiques simplement.

Quelques éléments du langage

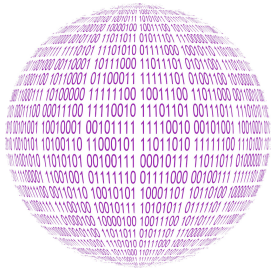
<code>from turtle import *</code>	Charger la bibliothèque turtle
<code>Charlotte = Turtle()</code>	Créer un objet turtle : Charlotte est une tortue
<code>Charlotte.forward(100)</code>	Avancer de 100 pixels
<code>Charlotte.left(90)</code>	Tourner de 90° vers la gauche
<code>Charlotte.right(45)</code>	Tourner de 45° vers la droite
<code>Charlotte.penup()</code>	Lever le crayon, pour stopper le tracé
<code>Charlotte.pendown()</code>	Baisser le crayon, pour commencer le tracé
<code>Charlotte.pensize(2)</code>	Choisir une taille de mine de crayon
<code>Charlotte.pencolor("magenta")</code>	Choisir une couleur de crayon Les couleurs sont celles habituelles en informatique : <i>blue, green, yellow, red, magenta, violet, gray, white, black, snow, cyan...</i>
<code>Charlotte.pencolor(0.78,0.56,0.14)</code>	La couleur en pourcentage de RVB (Rouge, Vert, Bleu)
<code>Charlotte.home()</code>	Charlotte va à la coordonnée (0;0), si le stylo est baissé, elle trace un trait
<code>Charlotte.goto(100,200)</code>	Charlotte va à la coordonnée (100;200)
<code>Charlotte.teleport(-100,-300)</code>	Téléporter la tortue, pas de trait tracé
<code>Charlotte.screen.bgcolor("salmon")</code>	Mettre le fond de l'écran à la couleur saumon
<code>Charlotte.screen.clearscreen()</code>	Effacer l'écran
<code>Charlotte.speed(10)</code>	Régler la vitesse de la tortue
<code>Charlotte.circle(100)</code>	Tracer un cercle de rayon 100 pixels
<code>Charlotte.begin_fill()</code>	Pour commencer un coloriage
<code>Charlotte.fillcolor("orange")</code>	Choix de la couleur de coloriage
<code>Charlotte.circle(200)</code>	Dessiner un cercle pour le colorier
<code>Charlotte.end_fill()</code>	Fin du coloriage
<code>Charlotte.position()</code>	Les coordonnées de la tortue
<code>Charlotte.xcor()</code> et <code>Charlotte.ycor()</code>	L'abscisse ou l'ordonnée de la tortue

`reponse = Charlotte.screen.textinput("Test","Question?")`

Poser une question dans une fenêtre de dialogue

Les couleurs en RVB

snow	indianred	navajowhite1	lightskyblue2	green3	wheat4	lightpink1
ghostwhite	saddlebrown	navajowhite2	lightskyblue3	green4	tan1	lightpink2
whitesmoke	sienna	navajowhite3	lightskyblue4	chartreuse1	tan2	lightpink3
gainsboro	peru	navajowhite4	slategray1	chartreuse2	tan3	lightpink4
floralwhite	burlywood	lemonchiffon1	slategray2	chartreuse3	tan4	palevioletred1
oldlace	beige	lemonchiffon2	slategray3	chartreuse4	chocolate1	palevioletred2
linen	wheat	lemonchiffon3	slategray4	olivedrab1	chocolate2	palevioletred3
antiquewhite	sandybrown	lemonchiffon4	lightsteelblue1	olivedrab2	chocolate3	palevioletred4
papayawhip	tan	cornsilk1	lightsteelblue2	olivedrab3	chocolate4	maroon1
blanchedalmond	chocolate	cornsilk2	lightsteelblue3	olivedrab4	firebrick1	maroon2
bisque	firebrick	cornsilk3	lightsteelblue4	darkolivegreen1	firebrick2	maroon3
peachpuff	brown	cornsilk4	lightblue1	darkolivegreen2	firebrick3	maroon4
navajowhite	darksalmon	ivory1	lightblue2	darkolivegreen3	firebrick4	violetred1
moccasin	salmon	ivory2	lightblue3	darkolivegreen4	brown1	violetred2
cornsilk	lightsalmon	ivory3	lightblue4	khaki1	brown2	violetred3
ivory	orange	ivory4	lightcyan1	khaki2	brown3	violetred4
lemonchiffon	darkorange	honeydew1	lightcyan2	khaki3	brown4	magenta1
seashell	coral	honeydew2	lightcyan3	khaki4	salmon1	magenta2
honeydew	lightcoral	honeydew3	lightcyan4	lightgoldenrod1	salmon2	magenta3
mintcream	tomato	honeydew4	paleturquoise1	lightgoldenrod2	salmon3	magenta4
azure	orangered	lavenderblush1	paleturquoise2	lightgoldenrod3	salmon4	lightsalmon1
aliceblue	red	lavenderblush2	paleturquoise3	lightgoldenrod4	lightsalmon2	lightsalmon3
lavender	hotpink	lavenderblush3	paleturquoise4	lightyellow1	lightsalmon3	lightsalmon4
lavenderblush	deeppink	lavenderblush4	cadetblue1	lightyellow2	orange1	orange2
mistyrose	pink	mistyrose1	cadetblue2	lightyellow3	orange3	orange4
white	lightpink	mistyrose2	cadetblue3	lightyellow4	darkorange1	darkorange2
black	palevioletred	mistyrose3	cadetblue4	yellow1	darkorange3	darkorange4
darkslate gray	maroon	mistyrose4	turquoise1	yellow2	coral1	coral2
dimgray	mediumvioletred	azure1	turquoise2	yellow3	coral3	coral4
slategray	violetred	azure2	turquoise3	yellow4	omato1	tomato2
lightslategray	magenta	azure3	turquoise4	gold1	tomato3	tomato4
gray	violet	azure4	cyan1	gold2	rosybrown1	rosybrown2
lightgray	plum	slateblue1	cyan2	gold3	rosybrown3	rosybrown4
midnightblue	orchid	slateblue2	cyan3	gold4	indianred1	indianred2
seagreen	mediumorchid	slateblue3	cyan4	goldenrod1	indianred3	indianred4
mediumseagreen	darkorchid	slateblue4	darkslategray1	goldenrod2	sienna1	sienna2
lightseagreen	darkviolet	royalblue1	darkslategray2	goldenrod3	sienna3	sienna4
palegreen	blueviolet	royalblue2	darkslategray3	goldenrod4	burlywood1	burlywood2
springgreen	purple	royalblue3	darkslategray4	darkgoldenrod1	burlywood3	burlywood4
lawngreen	mediumpurple	royalblue4	aquamarine1	darkgoldenrod2	wheat1	wheat2
green	thistle	blue1	aquamarine2	darkgoldenrod3	wheat3	
chartreuse	snow1	blue2	aquamarine3	darkgoldenrod4		
mediumspring-	snow2	blue3	aquamarine4	rosybrown1		
green	snow3	blue4	darkseagreen1	rosybrown2		
greenyellow	snow4	dodgerblue1	darkseagreen2	rosybrown3		
limegreen	seashell1	dodgerblue2	darkseagreen3	rosybrown4		
yellowgreen	seashell2	dodgerblue3	darkseagreen4	indianred1		
forestgreen	seashell3	dodgerblue4	seagreen1	indianred2		
olivedrab	seashell4	steelblue1	seagreen2	indianred3		
darkkhaki	antiquewhite1	steelblue2	seagreen3	indianred4		
khaki	antiquewhite2	steelblue3	seagreen4	sienna1		
palegoldenrod	antiquewhite3	steelblue4	palegreen1	sienna2		
lightgoldenrodyel-	antiquewhite4	deepskyblue1	palegreen2	sienna3		
low	bisque1	deepskyblue2	palegreen3	sienna4		
lightyellow	bisque2	deepskyblue3	palegreen4	burlywood1		
yellow	bisque3	deepskyblue4	springgreen1	burlywood2		
gold	bisque4	skyblue1	springgreen2	burlywood3		
lightgoldenrod	peachpuff1	skyblue2	springgreen3	burlywood4		
goldenrod	peachpuff2	skyblue3	springgreen4	wheat1		
darkgoldenrod	peachpuff3	skyblue4	green1	wheat2		
rosybrown	peachpuff4	lightskyblue1	green2	wheat3		



PYTHON ET LA TORTUE — Correction



INFORMATIQUE

3. On démontre que tout nombre entier peut s'écrire de manière unique sous la forme d'une somme de puissances de 2. Pour écrire un nombre **décimal** en **binaire** on utilise la propriété précédente. On code par le chiffre 1 la présence d'une puissance de 2 et par 0 son absence. Par exemple en écrivant le nombre 45 sous la forme $32 + 8 + 4 + 1$ on peut le compléter le tableau suivant :

Puissances de 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Valeur décimale	128	64	32	16	8	4	2	1
Décomposition décimale de 45			32		8	4		1
Écriture binaire de 45	0	0	1	0	1	1	0	1
Décomposition décimale de 1								
Écriture binaire de 1								
Décomposition décimale de 2								
Écriture binaire de 2								
Décomposition décimale de 3								
Écriture binaire de 3								
Décomposition décimale de 4								
Écriture binaire de 4								
Décomposition décimale de 5								
Écriture binaire de 5								
Décomposition décimale de 17								
Écriture binaire de 17								
Décomposition décimale de 63								
Écriture binaire de 63								
Décomposition décimale de 200								
Écriture binaire de 200								
Décomposition décimale de								
Écriture binaire de	1	1	1	0	1	0	0	1

4. En vous inspirant du tableau ci-dessus, compter de 1 à 20 en binaire.

Pour simplifier la compréhension et le stockage des bits on les regroupe par paquet de 8. On appelle cela un octet. Par exemple 0010110 est un octet puisqu'il est constitué de 8 bit. On ajoute des « zéros inutiles » à gauche pour obtenir 8 bits.

5. Quel est le plus grand nombre entier que l'on peut coder avec un octet ?

6. Numériser l'information « VIVE LA TECHNOLOGIE! » en regroupant les bits en octet. Vous utiliserez pour cela le codage ASCII obtenu à la question 1.. Combien d'octets sont nécessaires à cette numérisation ?

7. Décoder le message suivant présenté sous forme de bits regroupés en octet.

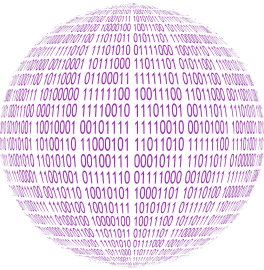
01001111 01001110 00100000 01000001 01000100 01001111 01010010 01000101 00100000 01001101 01000101 01001100 01000001 01001110
 01000111 01000101 01010010 00100000 01010100 01000101 01000011 01001000 01001110 01001111 00100000 01000101 01010100 00100000
 01001101 01000001 01010100 01001000 01010011 00100000 00100001

Voici les unités habituelles utilisées en informatique :

- o** *octet*
- ko** *kilo octet* 1000 o
- Mo** *Méga octet* 1 000 000 o
- Go** *Giga octet* 1 000 000 000 o
- To** *Téra octet* 1 000 000 000 000 o
- Po** *Péta octet* 1 000 000 000 000 000 o

Une clé USB de 64 Go peut stocker :

- 64 000 000 000 de caractères soit 512 000 000 000 de bits;
- 25 films en HD;
- 16 000 photos d'excellente qualité;
- 10 000 livres de 400 pages.



INFORMATIQUE

« VIVE LA TECHNOLOGIE ! » correspond aux codes ASCII suivants :

Caractère	V	I	V	E	L	A	T	E	C	H	N	O	L	O	G	I	E	!			
Codage en ASCII	86	73	86	69	32	76	65	32	84	69	67	72	78	79	76	79	71	73	69	32	33

2.

Puissance de 2	2 ⁰	2 ¹	2 ²	2 ³	2 ⁴	2 ⁵	2 ⁶	2 ⁷	2 ⁸	2 ⁹	2 ¹⁰	2 ¹¹	2 ¹²
Écriture décimale	1	2	4	8	16	32	64	128	256	512	1024	2048	4096

3.

Puissances de 2	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
Valeur décimale	128	64	32	16	8	4	2	1
Décomposition décimale de 45			32		8	4		1
Écriture binaire de 45	0	0	1	0	1	1	0	1
Décomposition décimale de 1								1
Écriture binaire de 1	0	0	0	0	0	0	0	1
Décomposition décimale de 2							2	
Écriture binaire de 2	0	0	0	0	0	0	1	0
Décomposition décimale de 3							2	1
Écriture binaire de 3	0	0	0	0	0	0	1	1
Décomposition décimale de 4						4		
Écriture binaire de 4	0	0	0	0	0	1	0	0
Décomposition décimale de 5						4		1
Écriture binaire de 5	0	0	0	0	0	1	0	1
Décomposition décimale de 17				16				1
Écriture binaire de 17	0	0	0	1	0	0	0	1
Décomposition décimale de 63			32	16	8	4	2	1
Écriture binaire de 63	0	0	1	1	1	1	1	1
Décomposition décimale de 200	128	64			8			
Écriture binaire de 200	1	1	0	0	1	0	0	0
Décomposition décimale de 233	128	64	32		8			1
Écriture binaire de 233	1	1	1	0	1	0	0	1

4.

Décimal	Binaire	Décimal	Binaire	Décimal	Binaire	Décimal	Binaire	Décimal	Binaire	Décimal	Binaire
0	0	10	1010	20	10100	30	11110	40	101000	50	110010
1	1	11	1011	21	10101	31	11111	41	101001	51	110011
2	10	12	1100	22	10110	32	100000	42	101010	52	110100
3	11	13	1101	23	10111	33	100001	43	101011	53	110101
4	100	14	1110	24	11000	34	100010	44	101100	54	110110
5	101	15	1111	25	11001	35	100011	45	101101	55	110111

5. Il s'agit de 11111111 c'est-à-dire $2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$

6.

Caractère	V	I	V	E		L	A		T	E	C
Codage en ASCII	86	73	86	69	32	76	65	32	84	69	67
Binaire	01010110	01001001	01010110	01000101	00100000	01001100	01000001	00100000	01010100	01000101	01000011
Caractère	H	N	O	L	O	G	I	E		!	
Codage en ASCII	72	78	79	76	79	71	73	69	32	33	
Binaire r	01001000	01001110	01001111	01001100	01001111	01000111	01001001	01000101	00100000	00100001	

Soit :

01010110 01001001 01010110 01000101 00100000 01001100 01000001 01010100 01000101 01000011 01001000 01001110 01001111 01001100 01001111 01000111 01001001 01000101 00100000 00100001

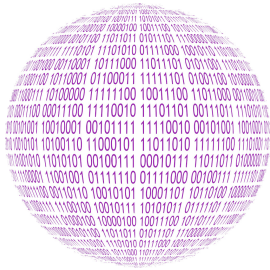
8.

01001111 01001110 00100000 01000001 01000100 01001111 01010010 01000101 00100000 01001101 01000101 01001100 01000001 01001110 01000111 01000101 01010010 00100000 01010100 01000101 01000011 01001000 01001110 01001111 00100000 01000101 01010100 00100000 01001101 01000001 01010100 01001000 01010011 00100000 00100001

Binaire regroupé en octet	01001111	01001110	00100000	01000001	01000100	01001111	01010010	01000101
Codage en ASCII	79	78	32	65	68	79	82	69
Caractère	O	N		A	D	O	R	E
Binaire regroupé en octet	00100000	01001101	01000101	01001100	01000001	01001110	01000111	01000101
Codage en ASCII	32	77	69	76	65	78	71	69
Caractère		M	E	L	A	N	G	E
Binaire regroupé en octet	01010010	00100000	01010100	01000101	01000011	01001000	01001110	01001111
Codage en ASCII	82	32	84	69	67	72	78	79
Caractère	R		T	E	C	H	N	O
Binaire regroupé en octet	00100000	01000101	01010100	00100000	01001101	01000001	01010100	01001000
Codage en ASCII	32	69	84	32	77	65	84	72
Caractère		E	T		M	A	T	H
Binaire regroupé en octet	01010011	00100000	00100001					
Codage en ASCII	83	32	33					
Caractère	S		!					

On obtient donc :

ON ADORE MELANGER TECHNO ET MATHS!



INFORMATIQUE

Nous savons qu'un nombre entier peut s'écrire en binaire, c'est à dire en utilisant seulement les chiffres 0 et 1.
Par exemple, comme $2024 = 1024 + 512 + 256 + 128 + 64 + 32 + 8$, l'écriture binaire de 2024 est **11111101000**.

Objectif : rédiger un script dans Codablock qui prend un nombre entier au départ et fourni en sortie son écriture binaire.

Un algorithme : les divisions euclidiennes successives

Voici un algorithme qui permet d'obtenir l'écriture binaire d'un nombre entier positif :

N un nombre entier positif dont on veut l'écriture binaire.
TABLE une liste vide.

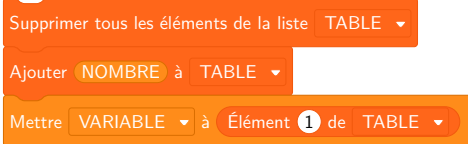
Tant que $N > 0$

Étape n° 1 — Diviser N par 2, Q est le quotient et R le reste
 Étape n° 2 — Stocker R en dernière position du tableau
 Étape n° 3 — $N = Q$

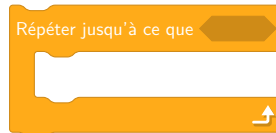
Afficher TABLE dans l'ordre croissant des indices

Tester cet algorithme sur les nombres 2024 et 12 345

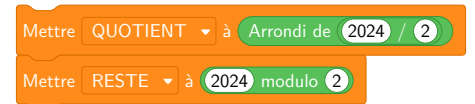
Les outils : une liste, une boucle et la division euclidienne



Quelques blocks pour gérer les listes.



Répéter le contenu de la boucle jusqu'à ce que la condition soit réalisée.

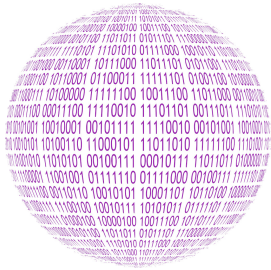


Les opérateurs **Arrondi** et **Modulo**

Exercices : Pour chacun des programmes, indiquer la valeur contenu dans la variable **NOMBRE** à la fin du script.



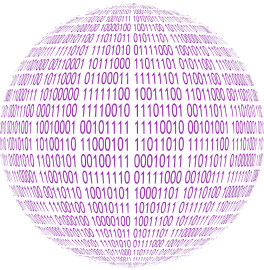
Objectif : rédiger dans Codablock le programme attendu. Voici le code Capytale : **a2a3-1944610**



DU DÉCIMAL AU BINAIRE — Correction



INFORMATIQUE



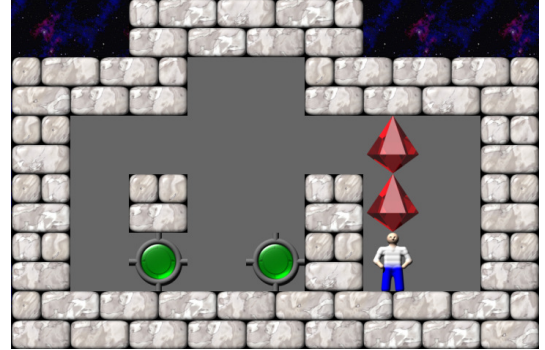
LE SOKOBAN



INFORMATIQUE

倉庫番

sokoban-online.de



Sokoban, ce qui signifie « garde d'entrepôt », est un jeu vidéo de réflexion inventé au Japon.

Le jeu original a été écrit par Hiroyuki Imabayashi et comportait 50 niveaux. Il remporte en 1980 un concours de jeu vidéo pour ordinateur. Plus tard Hiroyuki Imabayashi est devenu président de la compagnie japonaise Thinking Rabbit qui détient aujourd'hui les droits sur le jeu depuis 1982.

Dans Sokoban, le joueur doit ranger des caisses sur des cases cibles. Il peut se déplacer dans les quatre directions, et pousser (mais pas tirer) une seule caisse à la fois. Une fois toutes les caisses rangées (c'est parfois un vrai casse-tête), le niveau est réussi et le joueur passe au niveau suivant, plus difficile en général. L'idéal est de réussir avec le moins de coups possibles (déplacements et poussées).

1. Allez, on joue !

Vous devez lancer une version de Sokoban, JSoko et résoudre les cinq premiers niveaux en moins de 10 minutes!

2. « Hacker » un niveau :

Vous allez sauvegarder le premier niveau du jeu dans un fichier.

En lisant ce fichier sauvegardé, déterminer le code qui permet de modéliser un niveau.

Vous pouvez compléter le tableau suivant :

Symbole dans le fichier	Sprite obtenu dans le jeu
#	
\$	
*	
.	
@	
+	

3. Créer un nouveau niveau

En utilisant un éditeur de texte, par exemple le Notepad de Windows, vous devez créer un nouveau niveau.

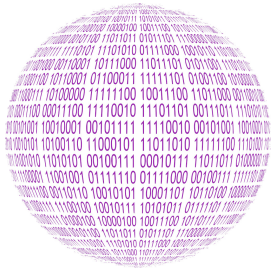
Il doit être viable et avoir une solution.

Le jeu de Sokoban peut être étudié du point de vue de la théorie de la complexité. Il a été démontré que la résolution des niveaux de Sokoban est un problème de complexité NP-difficile. Cela signifie que le temps moyen de résolution de ce problème augmente considérablement quand on ajoute des sprites supplémentaires. Il est facile de vérifier qu'une solution est la bonne solution. Il est très difficile, même pour un ordinateur très puissant, de trouver une solution.

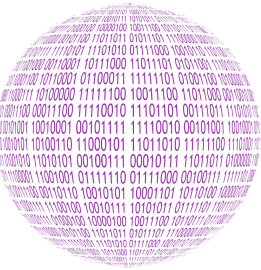
Le jeu est également intéressant pour les chercheurs en intelligence artificielle, car la résolution de niveaux pose des problèmes difficiles, pour lesquels il n'existe pas à ce jour d'algorithmes de résolution rapide.

La difficulté du Sokoban provient de son facteur de branchement, comparable à celui des échecs, bien que très inférieur à celui du jeu de go, mais aussi de la très grande profondeur de son arbre de recherche.

Les problèmes de Sokoban peuvent être résolus automatiquement à l'aide d'un algorithme de recherche à agent unique. C'est la méthode utilisée par Rolling Stone, un solveur développé par le groupe GAMES de l'Université d'Alberta, au Canada.



INFORMATIQUE



INFORMATIQUE

Les jeux de Nim se jouent avec un ou plusieurs tas d'objets (des allumettes par exemple), chaque joueur modifiant un ou plusieurs tas selon les règles adoptées. Il en existe de nombreuses versions : jeu de Marienbad, jeu de Grundy, jeu de Wythoff, jeu de Fort Boyard...

Tous ces jeux sont ce que l'on appelle des jeux à information parfaite, cela signifie que :

- il oppose deux joueurs;
- les joueurs à tour de rôle;
- tous les éléments sont connus;
- le hasard n'intervient pas pendant le déroulement du jeu.

Dans cette activité, on utilise la version de Fort Boyard dont voici les règles :

- le jeu démarre avec vingt objets;
- à chaque tour, un joueur peut choisir un, deux ou trois objets;
- celui qui prend le dernier objet a perdu.



RECHERCHE D'UNE STRATÉGIE DE MANIÈRE EMPIRIQUE

1. Avec le matériel distribué, jouez plusieurs parties avec votre binôme et exprimez une conjecture sur la manière de gagner à cette version du jeu de Nim.
2. Démontrez la conjecture précédente en tenant compte de celui qui commence la partie.

RECHERCHE D'UNE STRATÉGIE À L'AIDES DES NOMBRES DE GRUNDY

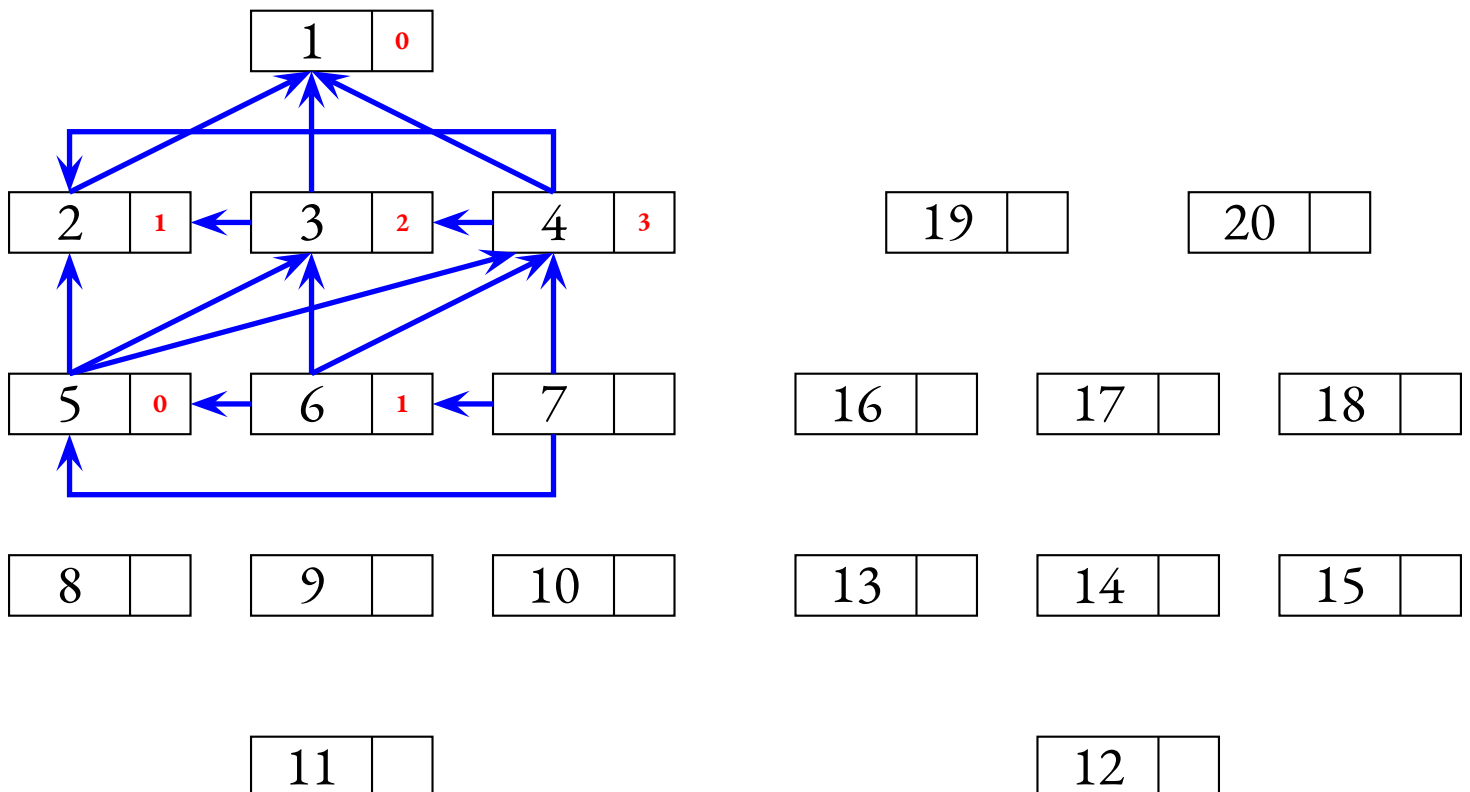
On peut représenter toutes les possibilités du jeu de Nim sous forme d'un arbre.

Chaque cas contient le nombre d'objets restants. Les flèches indiquent qu'il est possible de passer d'un nombre objets à un autre en suivant la règle.

Ainsi la case 5 est reliée aux cases 2, 3 et 4 puisque qu'on passe de 5 à ces cases en retirant 1, 2 ou 3 objets.

De même la case 6 est reliée aux cases 3, 4 et 5.

3. Compléter les flèches de l'arbre ci-dessous :



Les nombres entiers en petit (et en rouge) dans chaque case sont les nombres de Grundy.

Le nombre zéro est associé à la position gagnante finale.

On remonte ensuite les cases dans l'ordre croissant. Le nombre de Grundy de la case suivante est le plus petit nombre entier positif ou nul n'apparaissant pas dans la liste des nombres de Grundy des positions qui suivent immédiatement la position donnée, c'est à dire les cases reliées par les flèches.

Exemples :

Comme la case 1 est reliée à la case 0 et que le nombre de Grundy de la case 0 vaut 0. Le nombre de Grundy de la case 1 est le plus petit entier supérieur à 0, c'est à dire 1.

Comme la case 4 est reliée aux cases 1, 2 et 3 dont les nombres de Grundy sont 1, 2 et 3, le nombre de Grundy de la case 4 est 0, le plus petit nombre entier n'apparaissant pas dans cette liste.

4. Calculer les nombres de Grundy pour toutes les cases restantes.

5. Observer les cases dans lesquelles se trouvent un nombre de Grundy égal à 0. Qu'en concluez-vous?

UNE INTELLIGENCE ARTIFICIELLE AVEC UN ALGORITHME DÉTERMINISTE

6. En utilisant l'interface préparée par votre enseignant dans CodaBlock, implémentez l'algorithme qui permet au joueur qui commence de gagner une partie.

UNE INTELLIGENCE ARTIFICIELLE AVEC DES PASTILLES COLORÉES

Nous allons montrer comment simuler la manière dont une intelligence artificielle apprend par la méthode de **l'apprentissage par renforcement**.

Nous plaçons devant chaque allumette, un verre en carton. Au début de l'apprentissage, on place dans chaque verre des pois chiches : deux jaunes, deux verts et deux rouges.

Voici comment se déroule une partie :

- Un joueur humain peut commencer la partie;
- Quand c'est au tour de notre intelligence artificielle, il suffit de considérer le gobelet situé en face de l'allumette;
- On choisit au hasard un pois-chiche dans le gobelet;
 - S'il est jaune, on prend une allumette;
 - S'il est vert, on prend deux allumettes;
 - S'il est rouge, on prend trois allumettes.
- On laisse devant le gobelet le pois-chiche choisi.
- Le joueur humain joue à son tour et on recommence jusqu'à établir un vainqueur.



On passe ensuite à la phase d'**apprentissage** :

- En cas de **victoire** de l'intelligence artificielle;
 - Il faut **renforcer** son comportement;
 - On observe chacun des pois-chiches devant les gobelets;
 - On replace ces pois-chiches dans les gobelets correspondants en ajoutant un second pois-chiche de la même couleur.
- En cas de **défaite** de l'intelligence artificielle;
 - Il faut **sanctionner** son comportement;
 - On retire les pois-chiches choisis sans rien ajouter dans les gobelets;
 - Si un gobelet est totalement vide, on revient à la situation initiale en plaçant à nouveau deux pois-chiches de chaque couleur.

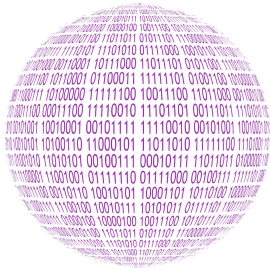
En procédant de cette manière, de nombreuses fois successives, les gobelets vont peut à peut se remplir avec des proportions différentes pour chacun d'entre eux des trois types de pois-chiche.

On constate que de cette manière, après un long entraînement, les proportions de pois-chiche permettent d'améliorer la réussite à ce jeu.

Voici, en pourcentage, la répartition obtenue après 1 000 000 itérations réalisées avec un script en Python :

	J	V	R		J	V	R		J	V	R		J	V	R
n° 1	33,33%	33,33%	33,33%	n° 6	91,30%	00,06%	00,04%	n° 11	00,00%	100,00%	00,00%	n° 16	00,00%	100,00%	00,00%
n° 2	35,39%	29,41%	35,39%	n° 7	00,00%	100,00%	00,00%	n° 12	00,00%	00,00%	100,00%	n° 17	64,12%	19,97%	15,91%
n° 3	00,00%	29,19%	70,81%	n° 8	00,00%	00,00%	100,00%	n° 13	100,00%	00,00%	00,00%	n° 18	00,00%	00,00%	100,00%
n° 4	00,00%	00,00%	100,00%	n° 9	25,00%	75,00%	00,00%	n° 14	04,37%	95,63%	00,00%	n° 19	88,43%	11,57%	00,00%
n° 5	66,67%	33,33%	00,00%	n° 10	100,00%	00,00%	00,00%	n° 15	79,34%	20,66%	00,00%	n° 20	06,78%	28,02%	65,20%

Qu'en pensez-vous?



INFORMATIQUE

prenons une partie de jeu de Nim à partir de la dernière situation : celle où le perdant prend le dernier objet.

Un joueur qui a gagné était juste avant dans l'un des cas suivants :

- il reste 4 objets et il en prend 3;
- il reste 3 objets et il en prend 2;
- il reste 2 objets et il en prend 1.

À l'étape précédente, le joueur perdant était dans un des cas suivants :

- il reste 7 objets et il en prend 3;
- il reste 6 objets et il prend 2 ou 3 objets;
- il reste 5 objets et il prend 1, 2 ou 3 objets;
- il reste 4 objets et il prend 1 ou 2 objets;
- il reste 3 objets et il prend 1 objets;

On constate que l'intérêt du gagnant est que l'autre joueur se trouve face à 5 objets puisque dans ce cas, quelque-soit le choix suivant, il va forcément gagner.

On admet donc que la stratégie du gagnant consiste à conduire son adversaire à avoir 5 objets restants.

À l'étape précédente, le joueur gagnant était dans un des cas suivants :

- il reste 8 objets et il en prend 3;
- il reste 7 objets et il en prend 2;
- il reste 6 objets et il en prend 1.

On revient d'un coup en arrière avec le joueur perdant, il était dans une des situations suivantes :

- il reste 9 objets et il en prend 1, 2 ou 3;
- il reste 8 objets et il en prend 1 ou 2;
- il reste 7 objets et il en prend 1.

Encore une fois, il est avantageux pour le gagnant que le perdant se retrouve avec 9 objets puisque dans ce cas, quelque-soit le choix suivant, la situation est favorable.

On constate que s'il reste 9 puis 5 objets, c'est une stratégie gagnante pour le joueur suivant.

Or $9 = 4 \times 2 + 1$ et $5 = 4 \times 1 + 1$.

On peut faire la conjecture que mettre l'adversaire dans les « positions » : 5; 9; 13 et 17 sont gagnantes!

2. Pour gagner, il faut placer l'adversaire face à 17, 13, 9 puis 5 objets.

Quand on commence la partie, **on gagne à tous les coups**, il suffit de prendre 3 objets pour placer l'adversaire en position 17.

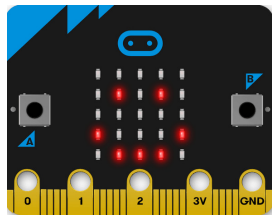
Que l'adversaire retire 1, 2 ou 3 objets, il est toujours possible de le ramener à 13 ($13 = 17 - 1 - 3 = 17 - 2 - 2 = 17 - 3 - 1$).

De même il faut ensuite le mener sur la position 9 puis 5.

Il s'agit des nombres entiers dont le reste dans la division par 4 est égal à 1.

Quand l'adversaire commence la partie, il perd dès qu'il nous permet de le placer sur une des positions précédentes. Il gagne en nous plaçant lui-même sur ces positions.

Le joueur qui commence gagne toujours la partie. Le deuxième joueur ne gagne que si le premier joueur commet une erreur en le plaçant ailleurs qu'en 17, 13, 9 ou 5.



micro:bit



COMMENT TRANSMETTRE UN CODE SECRET ?



OBJECTIFS : Nous souhaitons utiliser la carte **micro:bit** pour communiquer entre nous dans la salle de classe. De manière pratique, nous souhaitons communiquer un code secret en utilisant cette carte. Le **micro : bit** possède un module radio qui permet d'envoyer des informations. Cette séquence va montrer comment une information circule dans un réseau et la nécessité de mettre en œuvre un protocole de communication sécurisé.

PREMIÈRE PARTIE — Envoyer et recevoir une information avec un **micro:bit**

Un premier **micro:bit** est en train d'émettre un message en mode radio dans cette salle de classe. Une seconde carte attend un message de votre part.

👉 Programmer la carte que nous vous avons confiée pour qu'elle affiche le message émit quand on appuie sur le bouton A et qu'elle envoie vos deux prénoms séparés par un tiret quand on appuie sur le bouton B.

DEUXIÈME PARTIE — Broadcast et Unicast sont dans un bateau...

Il est possible de former un groupe pour que les messages envoyés soient diffusés qu'à nos destinataires plutôt qu'à tout le monde. En vous mettant d'accord avec un autre groupe de la classe, vous pouvez décider manuellement d'un numéro de groupe (compris entre 1 et 80) dans lequel vous communiquerez.

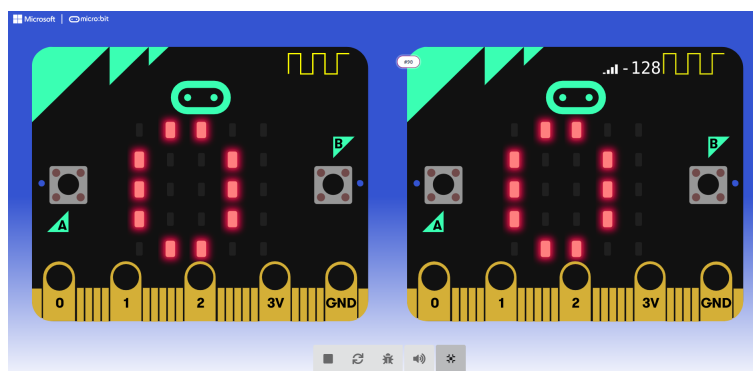
👉 Programmer vos cartes pour qu'elles communiquent entre elles. Quand vous appuyez sur A vous recevez le message envoyé par l'autre groupe. Quand vous appuyez sur B vous envoyez votre message.

TROISIÈME PARTIE — Un premier protocole de communication

Un **micro:bit** diffuse un code secret à quatre chiffre à chacun d'entre vous. Nous avons numéroté les binômes présents en classe. Le message contenant tous les codes secret est diffusé par la carte.

Chaque code secret est précédé d'une **entête** constituée ainsi : **#XX#** où XX désigne le numéro de groupe à deux chiffres. Les codes sont diffusés les uns à la suite des autres en continu.

👉 Programmer votre carte pour qu'elles reçoivent ce message et qu'elle affiche à l'écran uniquement le code secret de votre groupe.

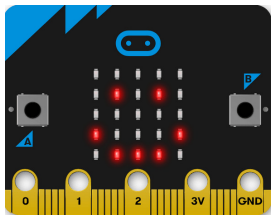


TROISIÈME PARTIE — Un protocole de communication complexe

On souhaite maintenant établir une connexion **Unicast** entre deux cartes **micro:bit** . Il faut pour cela que les cartes commencent par se mettre d'accord sur un numéro de groupe de communication. Pour assurer la confidentialité, il faut que le même groupe ne soit utilisé que par un seul binôme.

👉 Combien de numéro de groupes différents faut-il dans notre classe? Comment, en connaissant les numéros des deux binômes qui veulent communiquer, déterminer un numéro unique de groupe privé?

👉 Mettez en place ce protocole dans votre carte **micro:bit** . Votre carte doit vous permettre de choisir un numéro de binôme avec qui vous voulez communiquer. Vous envoyez ensuite en **Broadcast** un message signalant votre demande sous la forme **#XXYY#** où XX désigne votre **identifiant** et YY celui du binôme avec qui vous voulez parler. Calculez ensuite le numéro de groupe dans lequel vous allez communiquer et envoyez votre message!



micro:bit



COMMENT TRANSMETTRE UN CODE SECRET? — Correction



NON





COMMENT FAIRE...

La carte **BBC micro : bit** possède une antenne Bluetooth à basse énergie (BLE : Bluetooth Low Energy) qui lui permet de transmettre et recevoir des messages. Comparé au Bluetooth, cette technologie permet un débit du même ordre de grandeur 1 Mbit/s pour une consommation électrique dix fois plus faible.



Par défaut, la carte micro:bit diffuse les messages en **broadcast**, les informations sont diffusées à tous les micro:bit actifs sans restriction. On peut simuler une communication **unicast**, c'est à dire une communication entre deux **micro : bit** identifiés seulement, en utilisant la notion de **groupe**.

Voici les six blocs de la bibliothèque Radio disponibles dans l'application **makecode** que nous allons utiliser :



Définition du groupe de communication



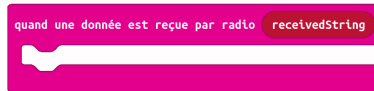
Envoyer une variable



Envoyer une chaîne de caractères



Action quand on reçoit un nombre



Action quand on reçoit une chaîne de caractères



Action quand on reçoit une variable

Les cinq blocs qui suivent servent moins souvent :



Communiquer en utilisant le numéro de série



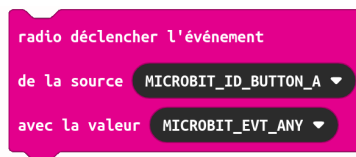
Modifier la force du signal



Lire des variables Radio



Modifie la fréquence d'émission



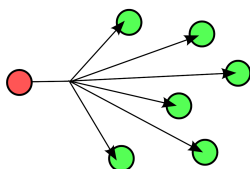
Permet d'envoyer des événements en mode Radio pour les déclencher sur le récepteur.

Quand vous utilisez un de ces blocs, le simulateur inclu dans Makecode vous présente un second **micro:bit**. Cela vous permet de vérifier l'information qui est envoyée.

Broadcast (Télédiffusion) :

Technique de transmission du signal, d'une source vers un grand nombre de clients sans discrimination du destinataire.

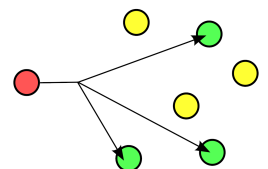
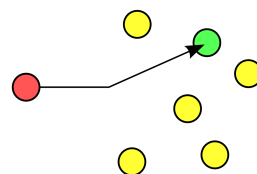
Quand on ne précise pas de numéro de groupe dans Makecode, le **micro:bit** communique en **Broadcast**.



Unicast (Liaison point à point) :

Technique de transmission du signal, d'une source unique vers un destinataire unique, chacun étant bien identifié.

La définition d'un numéro de groupe permet de limiter le nombre de destinataires (**Unicast** : un seul; **Multicast** : plusieurs).





COMMENT FAIRE...



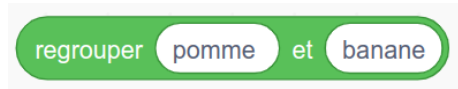


COMMENT FAIRE...

Dans les langages informatiques, les chaînes de caractères sont un des types de données utilisable dans un programme. Une chaîne de caractère correspond à un mot ou une phrase, c'est à dire une suite ordonnées de caractères.

Il est souvent nécessaire de traiter une chaîne de caractères pour **chercher** une information, **extraire** une partie ou un caractère, **concaténer** (regrouper) deux chaînes, **remplacer** certains caractères...

Voici les principales fonctions utilisables sur une chaîne de caractères dans Scratch :



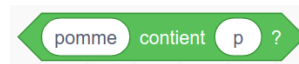
Concatène les deux chaînes de caractères



Récupère un caractère en fonction de sa position dans la chaîne



Indique le nombre de caractères dans la chaîne



Variable booléenne indiquant la présence d'un caractère dans une chaîne

Voici les principales fonctions utilisables sur une chaîne de caractères dans *micro:bit* Makecode :



Indique le nombre de caractères de la chaîne de caractères



Concatène deux chaînes



Convertit une chaîne de caractère en nombre



Variable booléenne indiquant la présence d'un caractère dans une chaîne



Créer un tableau contenant des parties de la chaîne de caractère en tenant compte du séparateur



Renvoie la position d'une sous-chaîne à l'intérieur d'une chaîne de caractères



Variable booléenne indiquant si la chaîne est vide



Récupère une partie de chaîne à partir d'un caractère et d'une taille donnée



Compare deux chaînes de caractères dans l'ordre alphabétique et renvoi 1, 0 ou -1 en fonction du classement



Renvoie le caractère placé à une certaine position



Transforme un nombre en chaîne de caractère



Action quand on reçoit une variable

La plupart des langages de programmation sont **typés**, cela signifie qu'il exige de distinguer les différents types de données : nombres entiers, nombres décimaux, chaînes de caractères, variables booléennes... À chaque type de données correspond des opérations : les opérations avec les nombres, concaténer, extraire avec les chaînes de caractères, ET, OU avec les booléens...

Ainsi dans *micro:bit*, la chaîne de caractère "2022" n'est pas égale au nombre 2022. Il faut utiliser des fonctions de conversions pour passer d'un nombre à une chaîne de caractère et réciproquement.

Quel message est affiché à la fin de ce programme?

```

toujours
définir Texte à "anticonstitutionnellement"
définir Premier résultat à sous-chaîne de Texte de 9 longue de 4
définir Deuxième résultat à concaténation Premier résultat Premier résultat
définir Troisième résultat à "oh"
répéter 4 fois
faire
définir Troisième résultat à concaténation Texte Troisième résultat
afficher texte Troisième résultat

```



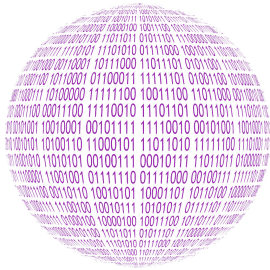
COMMENT FAIRE...



LES CHÂÎNES DE CARACTÈRES — Correction



NON



INFORMATIQUE

Deux défis avec Scratch

Utiliser le langage de programmation par blocs Scratch pour répondre à chacun des défis suivants :

Défi n° 1 (MS) : Calculer $1 + 2 + 3 + 4 + 5 + \dots + 1000$

Bonus (TB) : Créer un bloc **Somme 1000** qui correspond à ce défi.

Défi n° 2 (MS) : Calculer $1^2 + 2^2 + 3^2 + 4^2 + 5^2 + \dots + 1000^2$

Bonus (TB) : Créer un bloc **Carré de la somme 1000** qui correspond à ce défi.

On recommence en Python

Voici un programme en Python que vous pouvez saisir dans la console de Capytale :

```
somme = 0

for i in range(1,100):
    somme = i
    print(somme)
print("La somme finale est égale à : ")

print(somme)
```

Que fait ce programme ?

Défi n° 3 : Calculer $1 + 2 + 3 + 4 + 5 + \dots + 1000000$

Accélérer la vitesse de calcul en éliminant les affichages inutiles. Comparer la vitesse d'exécution avec celle de Blockly.

Voici comment on définit une fonction dans Python :

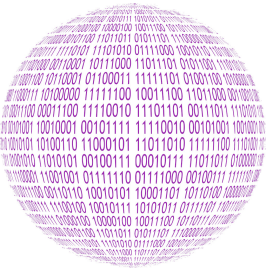
```
def Ma_super_fonction(nombre):
    somme = nombre*2
    return somme

print(Ma_super_fonction(1000))
```

Défi n° 4 : Calculer $1 + 2 + 3 + 4 + 5 + \dots + 1000000000$ en utilisant la commande **Somme(1 000 000 000)**

Défi n° 5 : Calculer $1^2 + 2^2 + 3^2 + 4^2 + 5^2 + \dots + 1000000000^2$ en utilisant la commande **Somme_carres(1 000 000 000)**.

Défi pour les experts : Calculer $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{1000000}$ en utilisant la commande **Somme_harmonique(1 000 000)**.



COMPTER AVEC SCRATCH ET PYTHON — Correction



INFORMATIQUE



INFORMATIQUE

Codablock

Défi (MS) : Écrire un programme dans Codablock qui demande à l'utilisateur la mesure des trois côtés d'un triangle. Le programme doit indiquer si le triangle est rectangle ou pas.

Défi (TB) : Le programme doit au préalable contrôler la saisie de l'utilisateur : les longueurs sont-elles des nombres positifs ? Le triangle existe-t-il ?

Python

Défi (TB) : Écrire un programme en Python qui demande à l'utilisateur la mesure des trois côtés d'un triangle. Le programme doit indiquer si le triangle est rectangle ou pas.

Indice : Voici un programme Python à saisir dans le terminal de Capytale. Que fait ce programme ?

```
print("Saisir deux nombres quelconques")

print("Premier nombre : ")

premier = int(input())

print("Deuxième nombre : ")

deuxieme = int(input())

if premier > deuxieme:
    print("Le premier nombre est plus grand que le deuxième")
    calcul = premier**2-deuxieme**2
    print("La différence des deux carrés vaut : ",calcul)
else:
    print("Le deuxième nombre est plus grand que le premier")
    calcul = deuxieme**2-premier**2
    print("La différence des deux carrés vaut : ",calcul)
```

Quelques explications :

print("Bonjour") : affiche la chaîne de caractères "Bonjour"

print(nombre) : affiche le nombre **nombre**

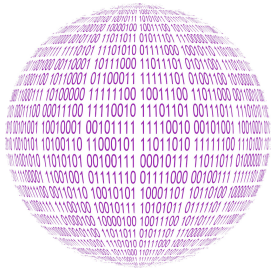
mot = input() : stocke dans la variable mot la chaîne de caractère saisie par l'utilisateur

nombre = int(input()) : stocke dans la variable nombre le nombre saisi par l'utilisateur

if condition:.... else : si... sinon

Il faut un **:** à la fin de la condition, puis une tabulation pour marquer le bloc. Il faut aussi un **:** après le else et des tabulations.

nombre2 :** calcule le carré d'un nombre



PYTHAGORE AVEC CODABLOCK ET PYTHON — Correction



INFORMATIQUE

ÉVALUATION DES COMPÉTENCES EN INFORMATIQUE

ALGORITHMIQUE ET PROGRAMMATION

Variables

Créer une variable

Remarques et intentions pédagogiques

¹ ACTIVITÉ — SÉCURITÉ DES MOTS DE PASSE

Mes intentions sont claires

² ACTIVITÉ — CURRICULUM VITÆ

Les intentions

³ ACTIVITÉ — LE JEU DE LA VIE

Mes intentions sont claires

⁴ ACTIVITÉ — PYTHON ET LA TORTUE

Mes intentions sont claires

⁵ ACTIVITÉ — INTELLIGENCE ARTIFICIELLE ET JEU DE NIM

Mes intentions sont

12

19

21

26

27

31

3 3

3 7

38

41

43

47

48

49

51

53

54

57

61

65

68

69

70

72

75

78

80

81

83

85

8 7

8 8

8 9

9 0

93

95

96

97

9 8

9 9

INFORMATIONS LÉGALES

- **Auteur** : Fabrice ARNAUD
- **Web** : pi.ac3j.fr
- **Mail** : contact@ac3j.fr
- **Dernière modification** : 30 avril 2026 à 12:49

Ce document a été écrit pour \LaTeX avec l'éditeur VIM - Vi Improved Vim 9.1.967
Il a été compilé sous Linux Ubuntu Questing Quokka (Le Quokka en quête) 25.10 avec la distribution TeX Live 2024.20250309 et LuaTeX 1.18.0

Le fichier source a été réalisé sous Linux Ubuntu avec l'éditeur Vim.

J'aimerais beaucoup rendre disponibles mes sources en \TeX . Dans un monde idéal, je le ferai immédiatement. J'ai plusieurs fois constaté que des pilliers du Net me volent mes fichiers pdf, retirent cette dernière page de licence, pour les mettre en ligne et parfois même les rendre payants. N'ayant pas les moyens de mettre un cabinet d'avocats sur cette contravention à la licence CC BY-NC-SA 4.0, je fais le choix de ne pas rendre mes sources disponibles. La plupart des pdf proposés sur ce blog ne contiennent aucun filigrane, je ne les signe pas. Cela permet aux collègues, aux parents, aux élèves, de disposer d'un document anonyme dont chacun peut disposer en respectant la licence qui est particulièrement souple pour les utilisateurs non commerciaux. Je me suis contenté d'ajouter mes références sur cette dernière page. Seules les corrections d'examens contiennent un filigrane vertical. J'ai en effet constaté que certains sites peu scrupuleux, vendaient mes corrections alors qu'elles sont disponibles librement et gratuitement sur mon site. Cette solution est insatisfaisante, je n'ai pas trouvé mieux!

Les QR codes présents sur certains documents pointent vers le fichier pdf lui-même et sa correction. Ce lien ne pointe ni vers une page de mon blog ni vers une quelconque publicité. Vous pouvez le laisser si vous souhaitez que vos élèves accèdent au document en ligne avec sa correction.

Si vous êtes un enseignant et que vous diffusez ce document dans le cadre strict de votre établissement scolaire, inutile de vous poser des questions sur la licence ci-dessous! Dans la mesure où vous limitez cette diffusion à votre classe ou un environnement numérique de travail privé, n'hésitez pas à vous servir!

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-nc-sa/4.0/deed.fr>

Comment créditer cette œuvre ?

Ce document, **Cours.pdf**, a été créé par **Fabrice ARNAUD (contact@ac3j.fr)** le 30 avril 2026 à 12:49.

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

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

INFORMATIONS LÉGALES

- **Auteur** : Fabrice ARNAUD
- **Web** : pi.ac3j.fr
- **Mail** : contact@ac3j.fr
- **Dernière modification** : 30 avril 2026 à 12:49

Ce document a été écrit pour \LaTeX avec l'éditeur VIM - Vi Improved Vim 9.1.967
Il a été compilé sous Linux Ubuntu Questing Quokka (Le Quokka en quête) 25.10 avec la distribution TeX Live 2024.20250309 et LuaTeX 1.18.0

Le fichier source a été réalisé sous Linux Ubuntu avec l'éditeur Vim.

J'aimerais beaucoup rendre disponibles mes sources en \TeX . Dans un monde idéal, je le ferai immédiatement. J'ai plusieurs fois constaté que des pilliers du Net me volent mes fichiers pdf, retirent cette dernière page de licence, pour les mettre en ligne et parfois même les rendre payants. N'ayant pas les moyens de mettre un cabinet d'avocats sur cette contravention à la licence CC BY-NC-SA 4.0, je fais le choix de ne pas rendre mes sources disponibles. La plupart des pdf proposés sur ce blog ne contiennent aucun filigrane, je ne les signe pas. Cela permet aux collègues, aux parents, aux élèves, de disposer d'un document anonyme dont chacun peut disposer en respectant la licence qui est particulièrement souple pour les utilisateurs non commerciaux. Je me suis contenté d'ajouter mes références sur cette dernière page. Seules les corrections d'examens contiennent un filigrane vertical. J'ai en effet constaté que certains sites peu scrupuleux, vendaient mes corrections alors qu'elles sont disponibles librement et gratuitement sur mon site. Cette solution est insatisfaisante, je n'ai pas trouvé mieux!

Les QR codes présents sur certains documents pointent vers le fichier pdf lui-même et sa correction. Ce lien ne pointe ni vers une page de mon blog ni vers une quelconque publicité. Vous pouvez le laisser si vous souhaitez que vos élèves accèdent au document en ligne avec sa correction.

Si vous êtes un enseignant et que vous diffusez ce document dans le cadre strict de votre établissement scolaire, inutile de vous poser des questions sur la licence ci-dessous! Dans la mesure où vous limitez cette diffusion à votre classe ou un environnement numérique de travail privé, n'hésitez pas à vous servir!

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-nc-sa/4.0/deed.fr>

Comment créditer cette œuvre ?

Ce document, **Cours.pdf**, a été créé par **Fabrice ARNAUD (contact@ac3j.fr)** le 30 avril 2026 à 12:49.

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

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