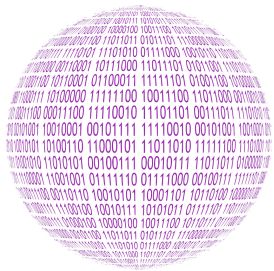
Chapitre III

Option informatique en troisième

I — Les outils informatiques



Sécurité des mots de passe

Informatique Collège

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+f" 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()

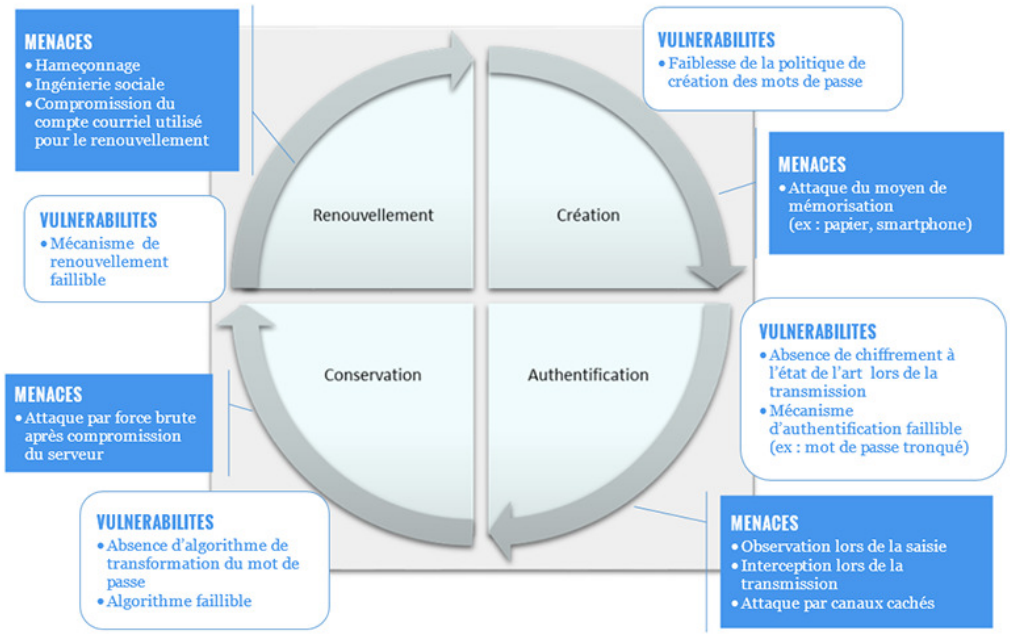
fichier = open("rockyou.txt","r",encoding="latin1")

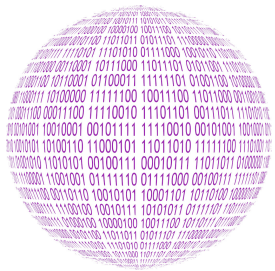
for mot in fichier:
      if mot_de_passe == mot[:-1]:
            fin = time.perf_counter()
            break

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]+f" en : {fin-debut: .5f} secondes")
```

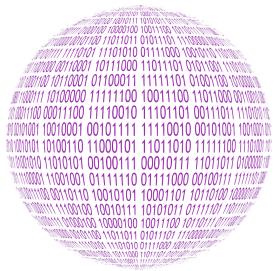
Votre mot de passe est-il sûr ?

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





Sécurité des mots de passe — Correction



Curriculum vitæ

Informatique Collège

✦✦✦

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.

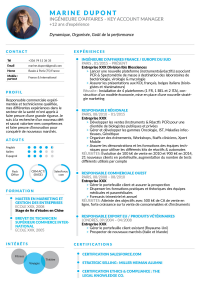
Objectifs : Vous souhaitez postuler sur une offre d'emploi. On vous demande un CV (« Curriculum vitæ », du latin « currere », courrir et « vitæ »vie, la course de la vie ou déroulement de la vie). En naviguant sur internet, vous avez sélectionné six modèles de CV dont la présentation vous correspond. 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 faîtes 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 :











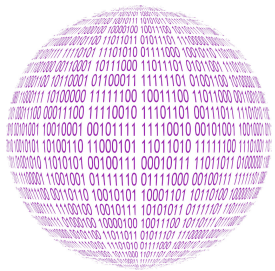


En vous rendant sur la page **https://arnaud.ac3j.fr/CV** vous trouverez :

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

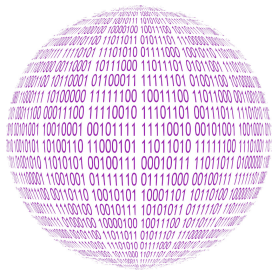
Quelques pistes pour atteindre le niveau de compétence attendu :

- **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



Le jeu de la vie

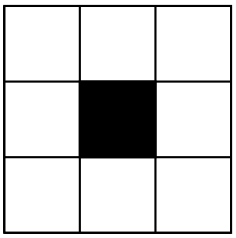
Informatique Collège

✧✧✧

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

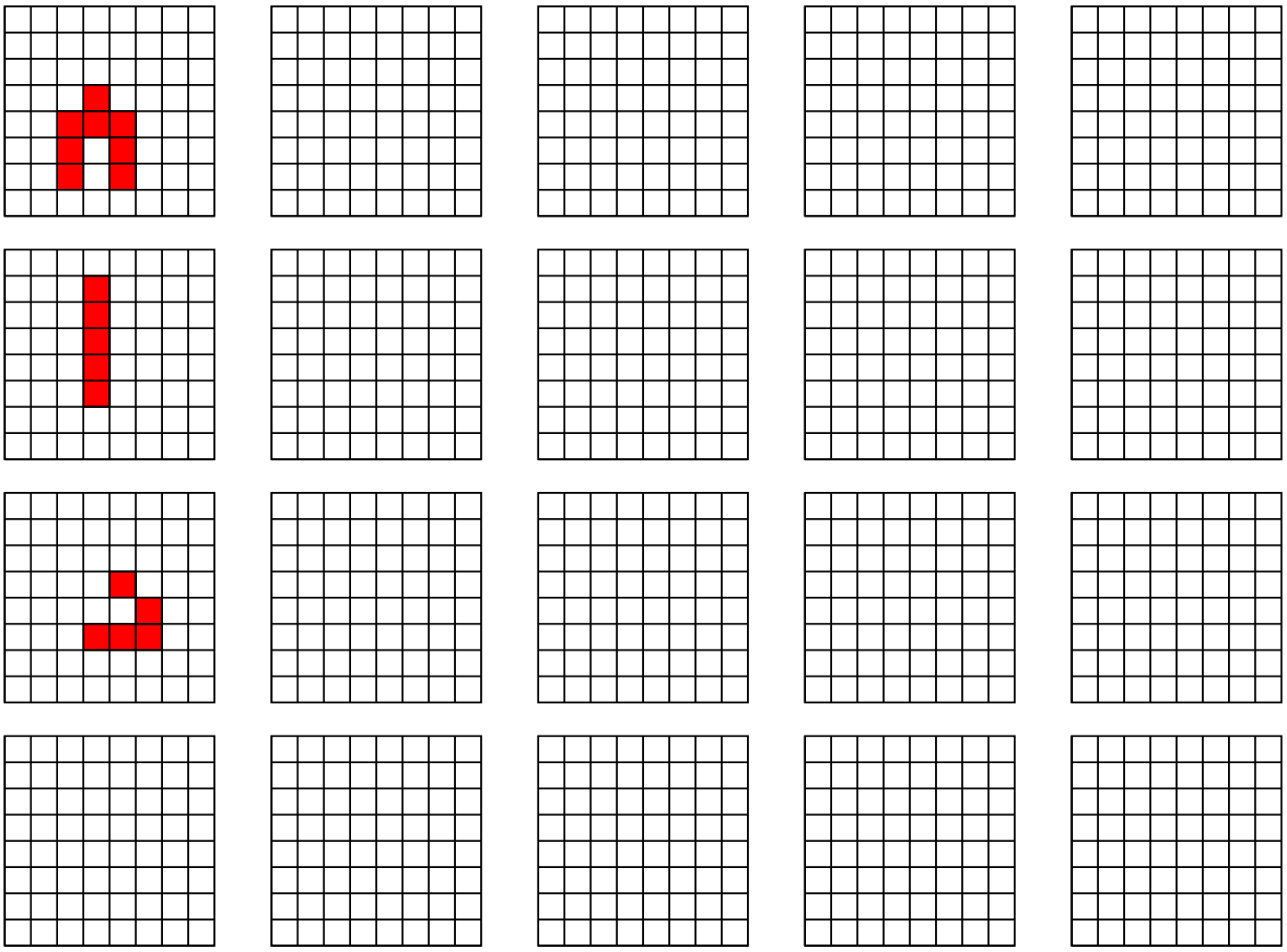


Le Jeu de la vie est un « jeu à zéro joueur », puisqu'il ne nécessite aucune intervention du joueur lors de son déroulement. Il s'agit d'un automate cellulaire, un modèle où chaque état conduit mécaniquement à l'état suivant à partir de règles préétablies.
Le jeu se déroule sur une grille à deux dimensions,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



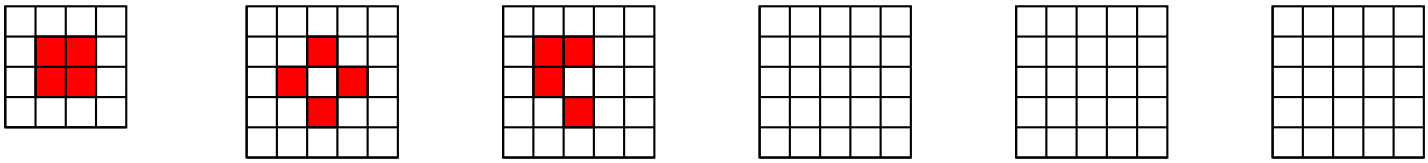
Il existe de nombreuses applications qui simulent le jeu de la vie de Conway.
Nous utiliserons le site https://playgameoflife.com/ pour tester ce jeu.

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

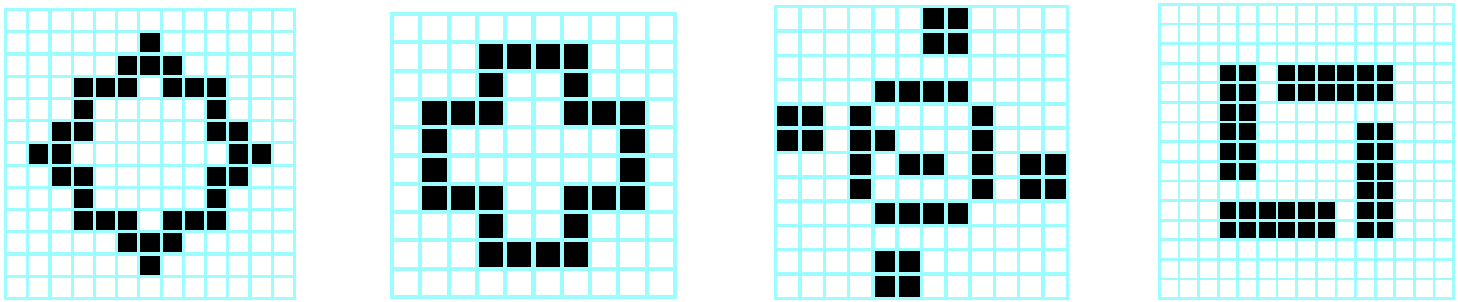
Structures remarquables

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

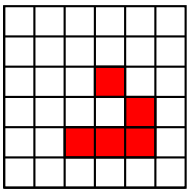


Les oscilateurs



Les vaisseaux

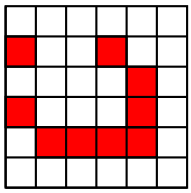
Dans un automate cellulaire, un vaisseau, ou navire, est un objet qui réapparait au bout d'un certain nombre de générations dans une position différente.



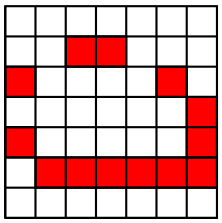
Le glider

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

Ce planeur 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 valeur, ainsi que leur mode de vie.



Light Weight Spaceship

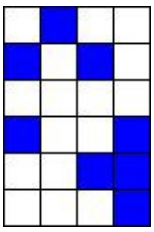


Heavy Weight Spaceship

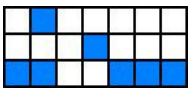
Les puffeurs, les canons et autres curiosités

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.

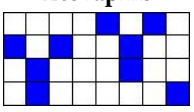
Un puffer diagonal



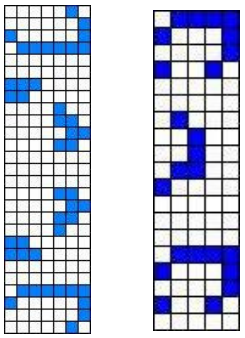
Le gland



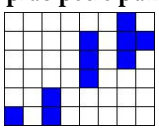
Les lapins

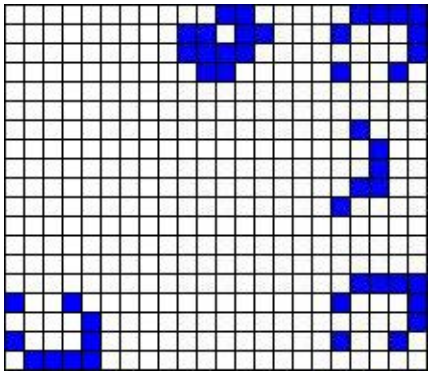


Les deux premiers puffers

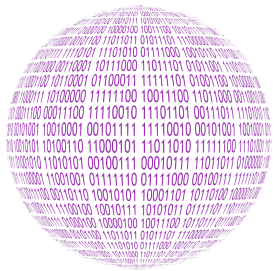


Le plus petit puffer

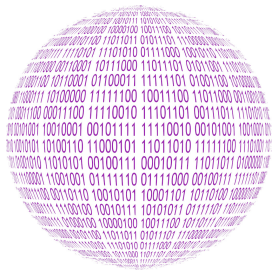




Un rake : un canon à vaisseaux



Le jeu de la vie — Correction



Python et la tortue

Informatique Collège

✧✧✧

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 semblabke à un petit robot avec un crayon qui peut dessiner sur le plan.
Ce langage est réputée pour son usage dans le domaine de l'éducation, il a été crée 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

from turle import *	Charger la bibliothèque turtle
Charlotte = Turtle()	Créer un objet turtle : Charlotte est une tortue
Charlotte.forward(100)	Avancer de 100 pixels
Charlotte.left(90)	Tourner de 90° vers la gauche
Charlotte.right(45)	Tourner de 45° vers la droite
Charlotte.penup()	Lever le crayon, pour stopper le tracé
Charlotte.pendown()	Baisser le crayon, pour commencer le tracé
Charlotte.pensize(2)	Choisir une taille de mine de crayon
Charlotte.pencolor("magenta")	Choisir une couleur de crayon Les couleurs sont celles habituelles en informatique : *blue, green, yellow, red, magenta, violet, gray, white, black, snow, cyan...*
Charlotte.pencolor(0.78,0.56,0.14)	La couleur en pourcentage de RVB (Rouge, Vert, Bleu)
Charlotte.home()	Charlotte va à la coordonnée (0;0), si le stylo est baissé, elle trace un trait
Charlotte.goto(100,200)	Charlotte va à la coordonnée (100;200)
Charlotte.teleport(-100,-300)	Téléporter la tortue, pas de trait tracé
Charlotte.screen.bgcolor("salmon")	Mettre le fond de l'écran à la couleur saumon
Charlotte.screen.clearscreen()	Effacer l'écran
Charlotte.speed(10)	Régler la vitesse de la tortue
Charlotte.circle(100)	Tracer un cercle de rayon 100 pixels
Charlotte.begin_fill()	Pour commencer un coloriage
Charlotte.fillcolor("orange")	Choix de la couleur de coloriage
Charlotte.circle(200)	Dessiner un cercle pour le colorier
Charlotte.end_fill()	Fin du coloriage
Charlotte.position()	Les coordonnées de la tortue
Charlotte.xcor() et **Charlotte.ycor()**	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
ghostwhite
whitesmoke
gainsboro
floralwhite
oldlace
linen
antiquewhite
papayawhip
blanchedalmond
bisque
peachpuff
navajowhite
moccasin
cornsilk
ivory
lemonchiffon
seashell
honeydew
mintcream
azure
aliceblue
lavender
lavenderblush
mistyrose
white
black
darkslate gray
dimgray
slategray
lightslategray
gray
lightgray
midnightblue
seagreen
mediumseagreen
lightseagreen
palegreen
springgreen
lawngreen
green
chartreuse
mediumspring-
green
greenyellow
limegreen
yellowgreen
forestgreen
olivedrab
darkkhaki
khaki
palegoldenrod
lightgoldenrodyel-
low
lightyellow
yellow
gold
lightgoldenrod
goldenrod
darkgoldenrod
rosybrown

indianred
saddlebrown
sienna
peru
burlywood
beige
wheat
sandybrown
tan
chocolate
firebrick
brown
darksalmon
salmon
lightsalmon
orange
darkorange
coral
lightcoral
tomato
orangered
red
hotpink
deeppink
pink
lightpink
palevioletred
maroon
mediumvioletred
violetred
magenta
violet
plum
orchid
mediumorchid
darkorchid
darkviolet
blueviolet
purple
mediumpurple
thistle
snow1
snow2
snow3
snow4
seashell1
seashell2
seashell3
seashell4
antiquewhite1
antiquewhite2
antiquewhite3
antiquewhite4
bisque1
bisque2
bisque3
bisque4
peachpuff1
peachpuff2
peachpuff3
peachpuff4

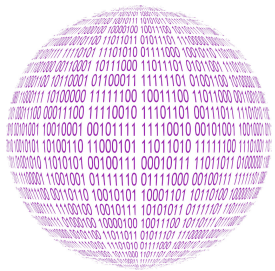
navajowhite1
navajowhite2
navajowhite3
navajowhite4
lemonchiffon1
lemonchiffon2
lemonchiffon3
lemonchiffon4
cornsilk1
cornsilk2
cornsilk3
cornsilk4
ivory1
ivory2
ivory3
ivory4
honeydew1
honeydew2
honeydew3
honeydew4
lavenderblush1
lavenderblush2
lavenderblush3
lavenderblush4
mistyrose1
mistyrose2
mistyrose3
mistyrose4
azure1
azure2
azure3
azure4
slateblue1
slateblue2
slateblue3
slateblue4
royalblue1
royalblue2
royalblue3
royalblue4
blue1
blue2
blue3
blue4
dodgerblue1
dodgerblue2
dodgerblue3
dodgerblue4
steelblue1
steelblue2
steelblue3
steelblue4
deepskyblue1
deepskyblue2
deepskyblue3
deepskyblue4
skyblue1
skyblue2
skyblue3
skyblue4
lightskyblue1

lightskyblue2
lightskyblue3
lightskyblue4
slategray1
slategray2
slategray3
slategray4
lightsteelblue1
lightsteelblue2
lightsteelblue3
lightsteelblue4
lightblue1
lightblue2
lightblue3
lightblue4
lightcyan1
lightcyan2
lightcyan3
lightcyan4
paleturquoise1
paleturquoise2
paleturquoise3
paleturquoise4
cadetblue1
cadetblue2
cadetblue3
cadetblue4
turquoise1
turquoise2
turquoise3
turquoise4
cyan1
cyan2
cyan3
cyan4
darkslategray1
darkslategray2
darkslategray3
darkslategray4
aquamarine1
aquamarine2
aquamarine3
aquamarine4
darkseagreen1
darkseagreen2
darkseagreen3
darkseagreen4
seagreen1
seagreen2
seagreen3
seagreen4
palegreen1
palegreen2
palegreen3
palegreen4
springgreen1
springgreen2
springgreen3
springgreen4
green1
green2

green3
green4
chartreuse1
chartreuse2
chartreuse3
chartreuse4
olivedrab1
olivedrab2
olivedrab3
olivedrab4
darkolivegreen1
darkolivegreen2
darkolivegreen3
darkolivegreen4
khaki1
khaki2
khaki3
khaki4
lightgoldenrod1
lightgoldenrod2
lightgoldenrod3
lightgoldenrod4
lightyellow1
lightyellow2
lightyellow3
lightyellow4
yellow1
yellow2
yellow3
yellow4
gold1
gold2
gold3
gold4
goldenrod1
goldenrod2
goldenrod3
goldenrod4
darkgoldenrod1
darkgoldenrod2
darkgoldenrod3
darkgoldenrod4
rosybrown1
rosybrown2
rosybrown3
rosybrown4
indianred1
indianred2
indianred3
indianred4
sienna1
sienna2
sienna3
sienna4
burlywood1
burlywood2
burlywood3
burlywood4
wheat1
wheat2
wheat3

wheat4
tan1
tan2
tan3
tan4
chocolate1
chocolate2
chocolate3
chocolate4
firebrick1
firebrick2
firebrick3
firebrick4
brown1
brown2
brown3
brown4
salmon1
salmon2
salmon3
salmon4
lightsalmon1
lightsalmon2
lightsalmon3
lightsalmon4
orange1
orange2
orange3
orange4
darkorange1
darkorange2
darkorange3
darkorange4
coral1
coral2
coral3
coral4
omato1
tomato2
tomato3
tomato4
orangered1
orangered2
orangered3
orangered4
red1
red2
red3
red4
deeppink1
deeppink2
deeppink3
deeppink4
hotpink1
hotpink2
hotpink3
hotpink4
pink1
pink2
pink3
pink4

lightpink1
lightpink2
lightpink3
lightpink4
palevioletred1
palevioletred2
palevioletred3
palevioletred4
maroon1
maroon2
maroon3
maroon4
violetred1
violetred2
violetred3
violetred4
magenta1
magenta2
magenta3
magenta4
orchid1
orchid2
orchid3
orchid4
plum1
plum2
plum3
plum4
mediumorchid1
mediumorchid2
mediumorchid3
mediumorchid4
darkorchid1
darkorchid2
darkorchid3
darkorchid4
purple1
purple2
purple3
purple4
mediumpurple1
mediumpurple2
mediumpurple3
mediumpurple4
thistle1
thistle2
thistle3
thistle4
darkgrey
darkgray
dark blue
darkcyan
darkmagenta
dark red
lightgreen



Informatique

Python et la tortue — Correction

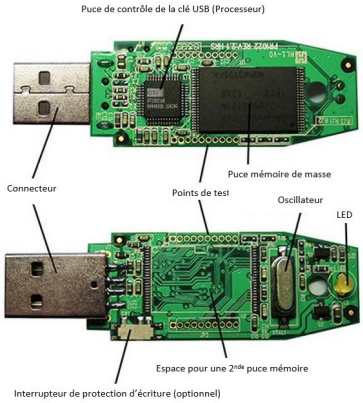
II — Informatique fondamentale

Numérisation de l'information

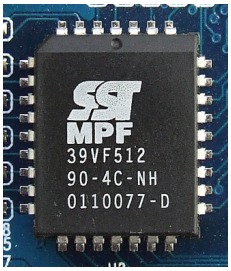
Informatique Collège

Informatique

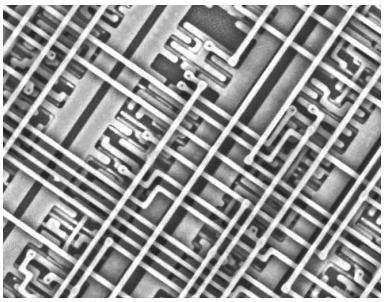
De la clé USB aux transistors



Clé USB



EEPROM



Transistor

Les clés USB, comme les cartes mémoires que nous utilisons tous les jours (disque dur SSD, cartes mémoire), contiennent une EEPROM (Electrically Erasable Programmable Read Only Memory). C'est un type de mémoire morte (ROM : Read Only Memory), contrairement au mémoire vive (RAM : Random Access Memory), qui peut être facilement modifiée ou effacée par l'utilisateur. Une mémoire morte est une mémoire utilisée pour enregistrer des informations qui ne doivent pas être perdues lorsque l'appareil qui les contient n'est plus alimenté en électricité.
Une EEPROM est constituée de semi-conducteurs, les transistors, qui sont des interrupteurs électriques microscopiques. Il faut environ un transistor pour un bit de donnée, soit plusieurs milliards sur une puce EEPROM. Un des enjeux actuels pour les grands constructeurs de semi-conducteurs (Samsung, Intel, AMD, TSMC, Broadcom, STMicroelectronics, Nvidia..) est la taille des transistors. Les plus récents font quelques nanomètres.

Numériser un texte : le code ASCII

Pour que les caractères typographiques (lettres de l'alphabet, ponctuations, majuscules, minuscules...) puissent être traités par les premiers ordinateurs, dès 1960 le codage ASCII (American Standard Code for Information Interechange) apparaît pour standardiser les usages. Ce codage sur 8 bits est une table de 255 caractères.
Voici un bref extrait de la table ASCII pour les caractères habituels :

Caractère :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G
Code ASCII	48	49	50	51	52	53	54	55	56	57	65	66	67	68	69	70	71
Caractère :	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Code ASCII	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88
Caractère :	Y	Z	a	b	c	d	e	f	,	!	.	;	?	é	è	ê	SPACE
Code ASCII	89	90	97	98	99	100	101	102	44	33	46	59	63	232	233	234	032

1. Alan vient de saisir au clavier la phrase « VIVE LA TECHNOLOGIE! ». Écrire les codes ASCII qui correspondent à cette phrase.

Dans une EEPROM, chaque transistor est comme un interrupteur : il peut être allumé ou éteint. Ces deux états codent les chiffres 0 et 1. Numériser une information, une image, une vidéo, un texte, consiste à la transformer en une succession de 0 et de 1. Sous cette forme, l'information peut être stockée sur une mémoire ou envoyée à travers une réseau.

2. Compléter le tableau suivant :

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										

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 ?

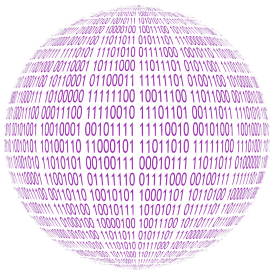
7. Décoder le message suivant présenté sous forme de bits regroupés en octet.
01001111 01001110 00100000 01000001 01000100 01001111 01010010 01000101 00100000 01001101 01000101 01001100 01000001 01001110 01000111 01000101 01010010 00100000 01010100 01000101 01000011 01001000 01001110 01001111 00100000 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.



Numérisation de l'information — Correction

Informatique

1. La phrase « 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
6	110	16	10000	26	11010	36	100100	46	101110	56	111000
7	111	17	10001	27	11011	37	100101	47	101111	57	111001
8	1000	18	10010	28	11100	38	100110	48	110000	58	111010
9	1001	19	10011	29	11101	39	100111	49	110001	59	111011

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

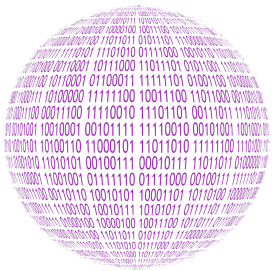
Binaire regroupé en octet	01010010	00100000	01010100	01000101	01000011	01001000	01001110	01001111
Codage en ASCII	82	32	84	69	67	72	78	79
Caractère	R		T	E	C	H	N	O

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!



Du décimal au binaire

Informatique Collège

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 **111111101000**.

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 nº 1 — Diviser *N* par 2, *Q* est le quotient et *R* le reste

Étape nº 2 — Stocker *R* en dernière position du tableau

Étape nº 3 — $N = Q$

Afficher *TABLE* dans l'ordre croissant des indices

Tester cet algorithme sur les nombres 2024 et 12345

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



Quelques blocks pour gérer les listes.

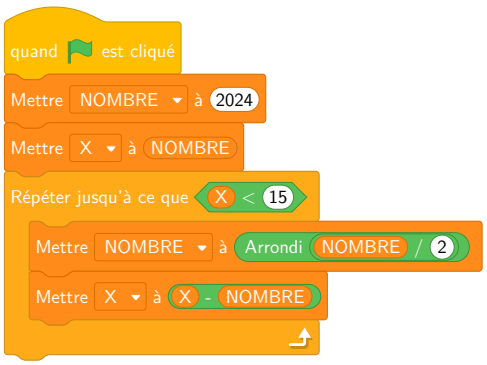


Répéter le contenu de la boucle jusqu'à ce que la condition soit réalisée.



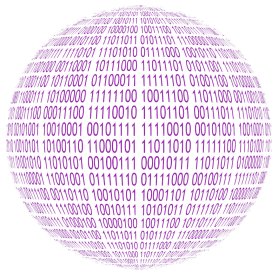
Les opérateurs Arrondi et Modulo

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

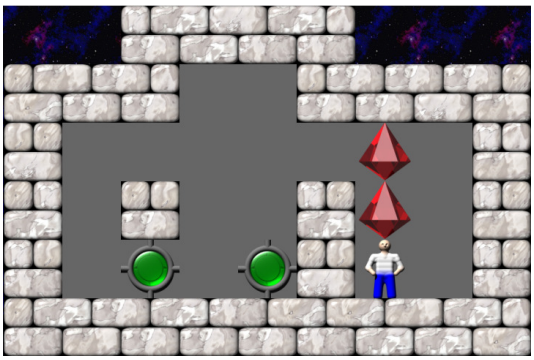
Le sokoban

Informatique

倉庫番

JSOKO

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 :

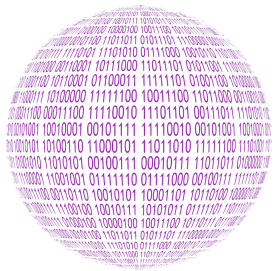
Vous allez sauvegarder le premier niveau du jeu dans un fichier.
En lisant ce fichier sauvegardé, déterminer le code qui permet de modéliser un niveau.
Vous pouvez compléter le tableau suivant :

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.



Le sokoban — Correction

Intelligence Artificielle et jeu de Nim

✧✧✧

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

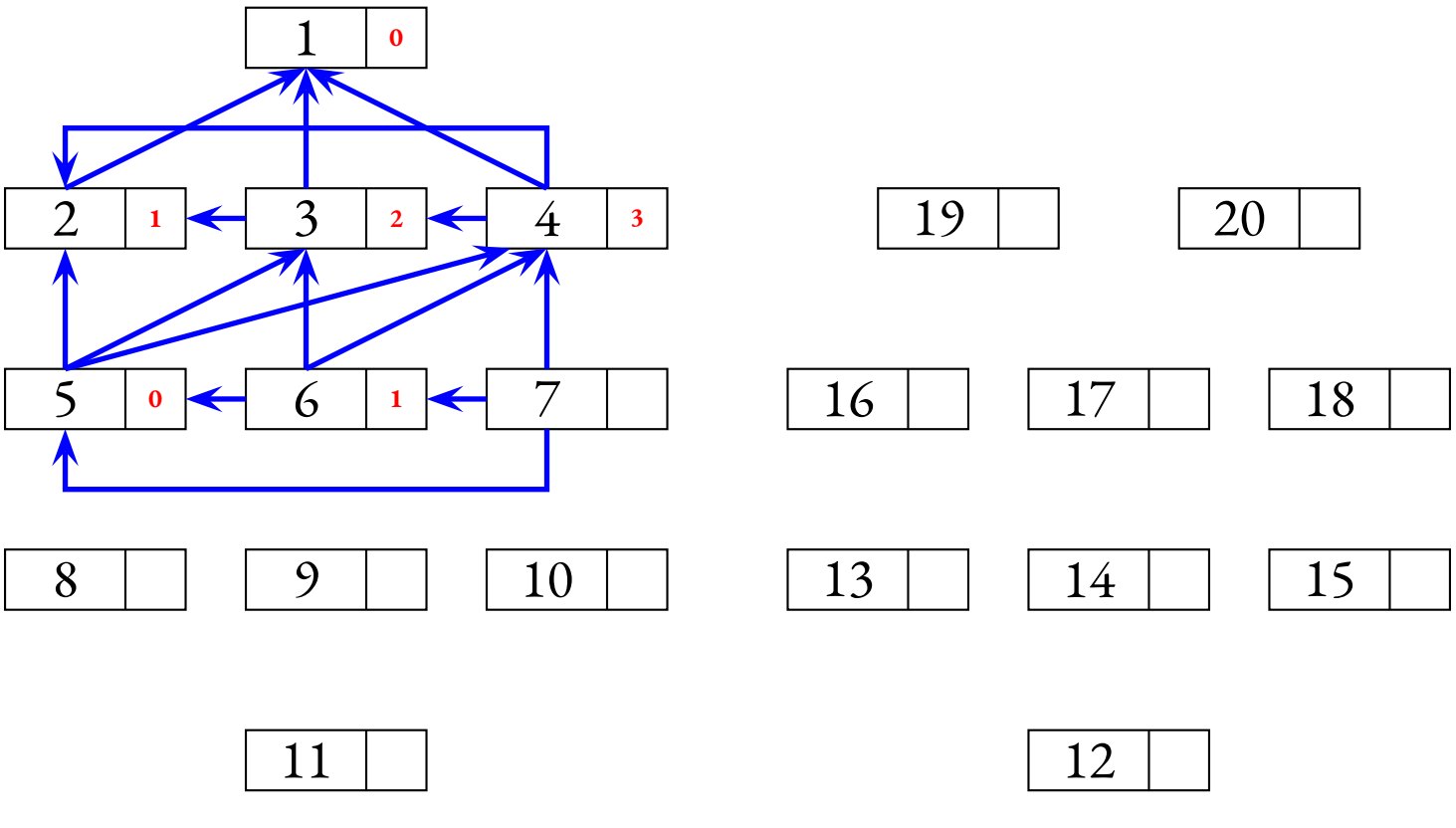
1. Avec le matériel distribué, jouez plusieurs parties avec votre binôme et exprimez une conjecture sur la manière de gagner à cette version du jeu de Nim.

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 prépararée 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** :

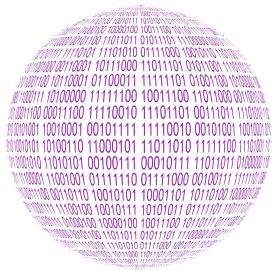
- En cas de **victoire** de l'intelligence artificielle ;
  - Il faut **renforcer** son comportement ;
  - On observe chacun des pois-chiches devant les gobelets ;
  - On replace ces pois-chiches dans les gobelets correspondants en ajoutant un second pois-chiche de la même couleur.
- En cas de **défaite** 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
nº 1	33,33%	33,33%	33,33%	**nº 6**	91,30%	00,06%	00,04%	**nº 11**	00,00%	100,00%	00,00%	**nº 16**	00,00%	100,00%	00,00%
nº 2	35,39%	29,41%	35,39%	**nº 7**	00,00%	100,00%	00,00%	**nº 12**	00,00%	00,00%	100,00%	**nº 17**	64,12%	19,97%	15,91%
nº 3	00,00%	29,19%	70,81%	**nº 8**	00,00%	00,00%	100,00%	**nº 13**	100,00%	00,00%	00,00%	**nº 18**	00,00%	00,00%	100,00%
nº 4	00,00%	00,00%	100,00%	**nº 9**	25,00%	75,00%	00,00%	**nº 14**	04,37%	95,63%	00,00%	**nº 19**	88,43%	11,57%	00,00%
nº 5	66,67%	33,33%	00,00%	**nº 10**	100,00%	00,00%	00,00%	**nº 15**	79,34%	20,66%	00,00%	**nº 20**	06,78%	28,02%	65,20%

Qu'en pensez-vous ?



Intelligence Artificielle et jeu de Nim — Correction

Informatique

1. Reprenons 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.

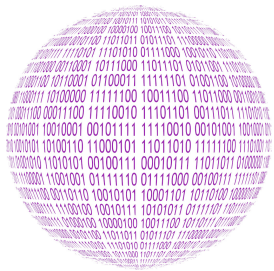
Encore une fois, il est avantageux pour le gagnant que le perdant se retrouve avec 9 objets puisque dans ce cas, quelque-soit le choix suivant, le situation est favorable.
On constate que s'il reste 9 puis 5 objets, c'est une stratégie gagnante pour le joueur suivant.
Or $9 = 4 \times 2 + 1$ et $5 = 4 \times 1 + 1$.

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.



Intelligence Artificielle et jeu de Nim

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, jouer plusieurs parties avec un binôme et exprimer une hypothèse sur la manière de gagner à cette version du jeu de Nim.

2. Justifier votre hypothèse en tenant compte de celui qui commence la partie.

Une intelligence artificielle avec un algorithme déterministe

Un ordinateur joue contre un humain au jeu de Nim avec 20 bâtons, l'ordinateur commence la partie.

3. Écrire un algorithme en langage naturel permettant à l'ordinateur d'utiliser l'algorithme gagnant.

Une intelligence artificielle avec des pois chiches colorés

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 chiches choisis ;
- Le joueur humain joue à son tour et on recommence jusqu'à établir un vainqueur.

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

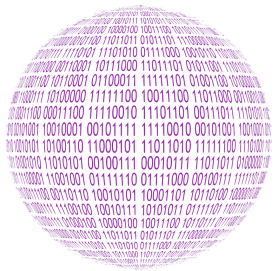
- En cas de **victoire** de l'intelligence artificielle ;
  - Il faut **renforcer** son comportement ;
  - On observe chacun des pois chiches devant les gobelets ;
  - On replace ces pois chiches dans les gobelets correspondants en ajoutant un second pois chiche de la même couleur.
- En cas de **défaite** 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 chiches.
On constate que de cette manière, après un long entraînement, les proportions de pois chiches 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
nº 1	98,34%	1,20%	0,46%	**nº 6**	78,76%	10,01%	11,23%	**nº 11**	7,30%	85,00%	7,70%	**nº 16**	6,98%	11,11%	81,91%
nº 2	81,76%	7,89%	10,35%	**nº 7**	7,45%	82,09%	10,46%	**nº 12**	11,63%	8,43%	79,94%	**nº 17**	31,56%	34,56%	33,88%
nº 3	6,78%	79,03%	14,19%	**nº 8**	7,87%	3,78%	88,35%	**nº 13**	36,56%	32,01%	31,43%	**nº 18**	84,53%	9,47%	6,00%
nº 4	3,56%	9,67%	86,77%	**nº 9**	34,56%	31,32%	34,12%	**nº 14**	79,09%	9,56%	11,35%	**nº 19**	8,67%	81,09%	10,24%
nº 5	33,37%	33,32%	33,31%	**nº 10**	82,56%	11,03%	6,41%	**nº 15**	11,10%	76,78%	12,12%	**nº 20**	08,9%	10,00%	81,10%

Qu'en pensez-vous ?



Intelligence Artificielle et jeu de Nim — Correction

Informatique

1. Reprenons 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, le 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 $5 = 4 \times 1 + 1$.

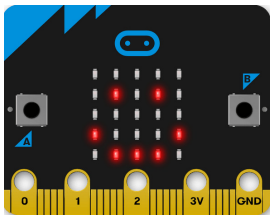
On peut faire la conjecture que mettre l'adversaire dans les « positions » : 5 ; 9 ; 13 et 17 sont gagnantes !

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.

III — Microbit



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...

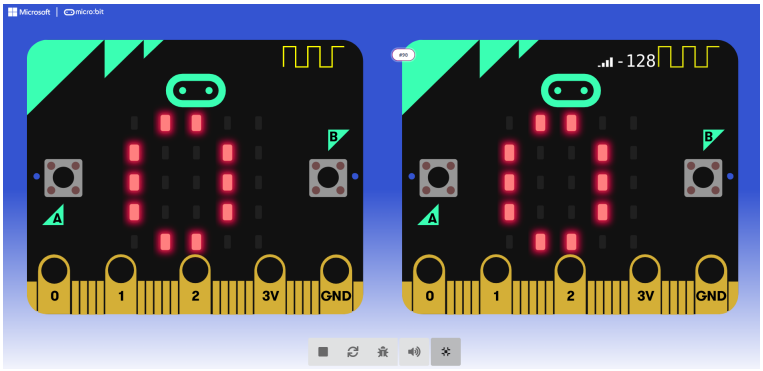
Il est possible de former un groupe pour que les messages envoyés soient diffusés qu'à nos destinataires plutôt qu'à tout le monde.
En vous mettant d'accord avec un autre groupe de la classe, vous pouvez décider manuellement d'un numéro de groupe (compris entre 1 et 80) dans lequel vous communiquerez.

☞ 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.

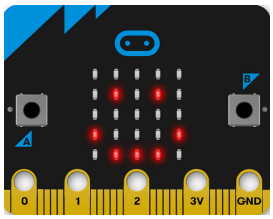


Troisième partie — Un protocole de communication complexe

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

Microbit — La communication radio

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 :

Radio



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



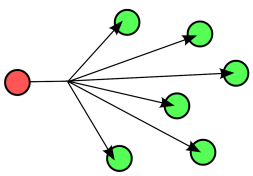
Modifie la fréquence d'émission



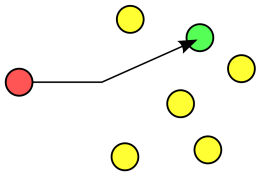
Permet d'envoyer des événements en mode Radio pour les déclencher sur le récepteur.

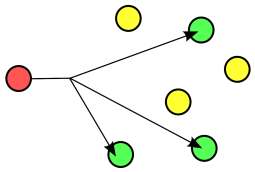
Quand vous utilisez un de ces blocs, le simulateur inclu dans Makecode vous présente un second ***micro:bit*** . Cela vous permet de vérifier l'information qui est envoyée.

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).

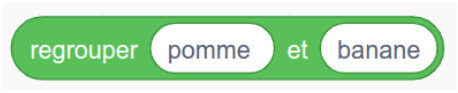




Les chaînes de caractères

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 :



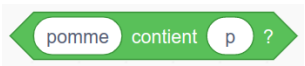
Concatène les deux chaînes de caractères



Récupère un caractère en fonction de sa position dans la chaîne



Indique le nombre de caractères dans la chaîne



Variable booléene 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 booléene 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 booléene indiquant si la chaîne est vide



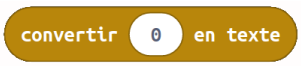
Récupère une partie de chaîne à partir d'un caractère et d'une taille donnée



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

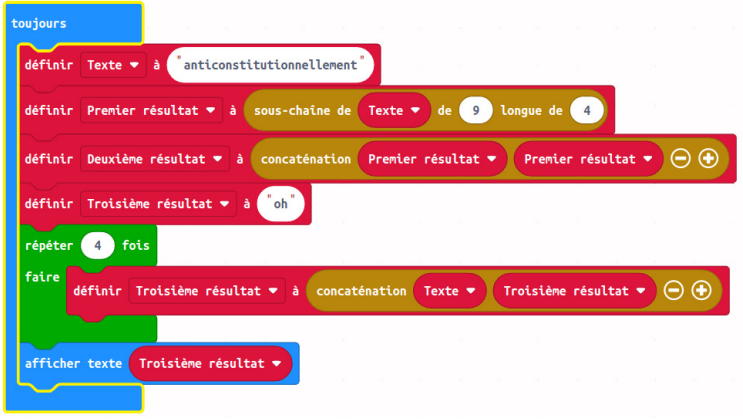


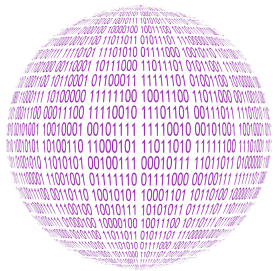
Action quand on reçoit une variable

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 booléenes... 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 ?



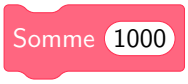


Informatique

Deux défis avec Scratch

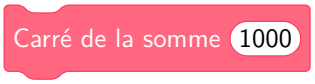
Utiliser le langage de programmation par blocs Scratch pour répondre à chacun des défis suivants :

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

Bonus (TB) : Créer un bloc 

qui correspond à ce défi.

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

Bonus (TB) : Créer un bloc 

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 $1+2+3+4+5+\ldots+1\,000\,000$

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

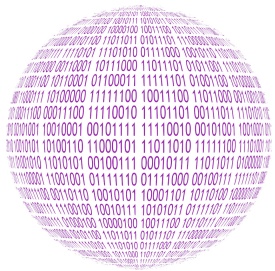
Voici comment on définit une fonction dans Python :

```
def Ma_super_fonction(nombre):
     somme = nombre*2
     return somme
print(Ma_super_fonction(1000))
```

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

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

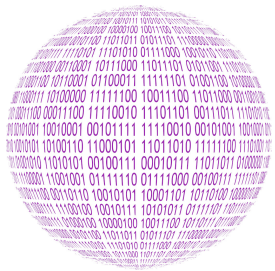
Défi pour les experts : Calculer $1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\ldots+\frac{1}{1\,000\,000}$ en utilisant la commande **Somme_harmonique(**$1\,000\,000$**)**.











Pythagore avec Codablock et Python

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**

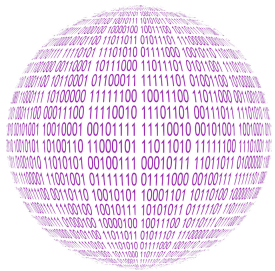
mot = input() : stocke dans la variable mot la chaîne de caractère saisie par l'utilisateur

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

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

[6] **Activité — Intelligence Artificielle et jeu de Nim**

Mes intentions sont

12219

21 26

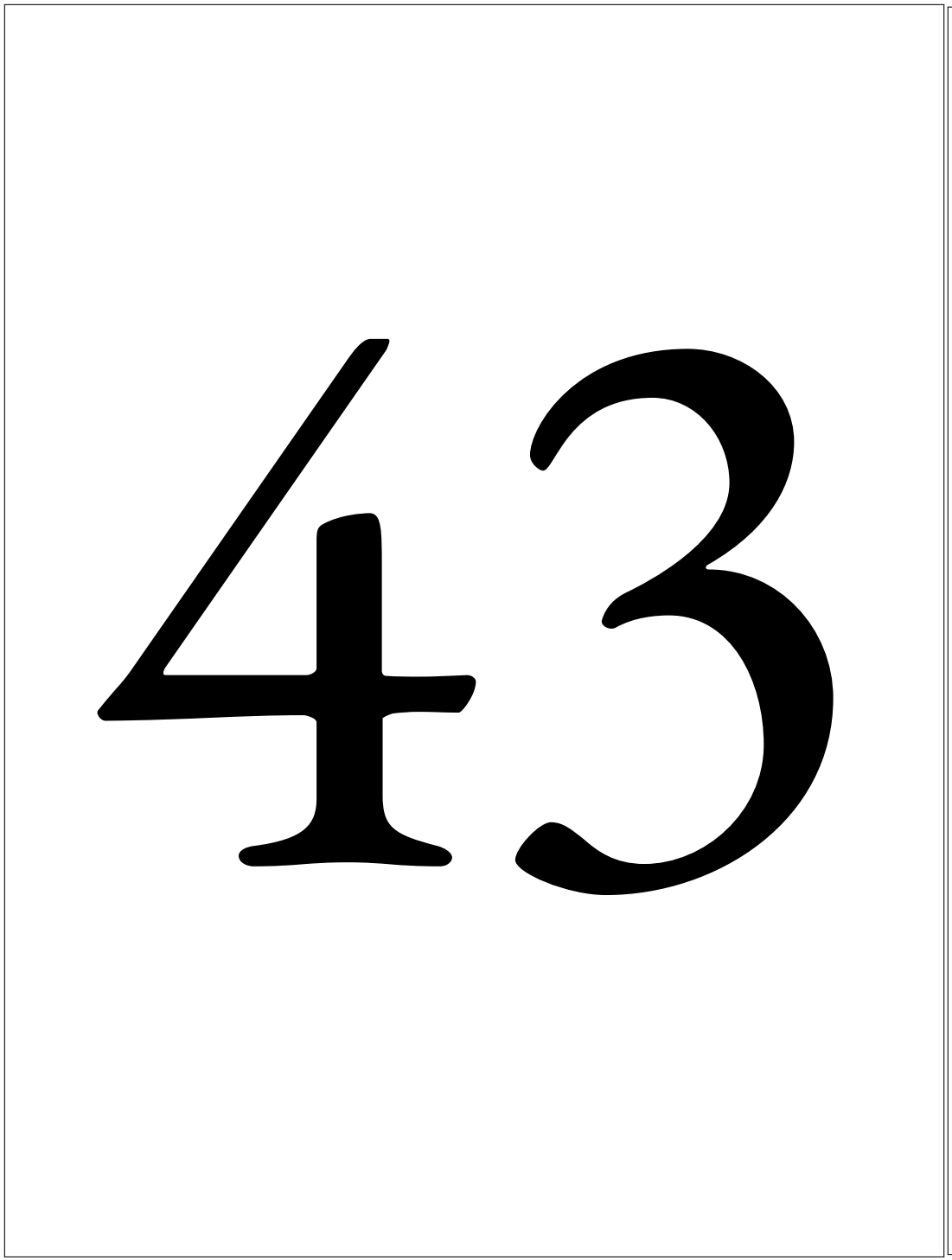
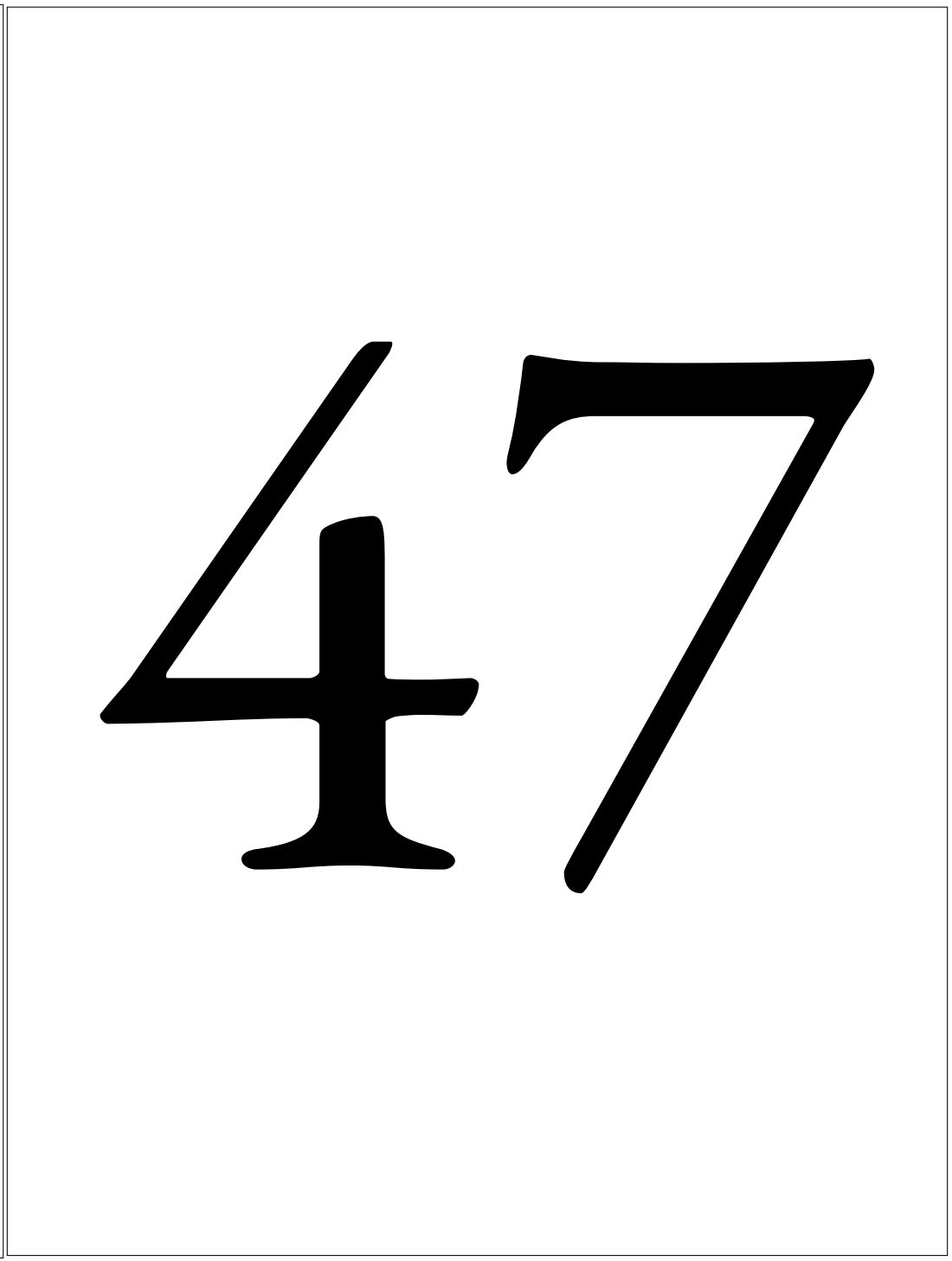
27

31

3337

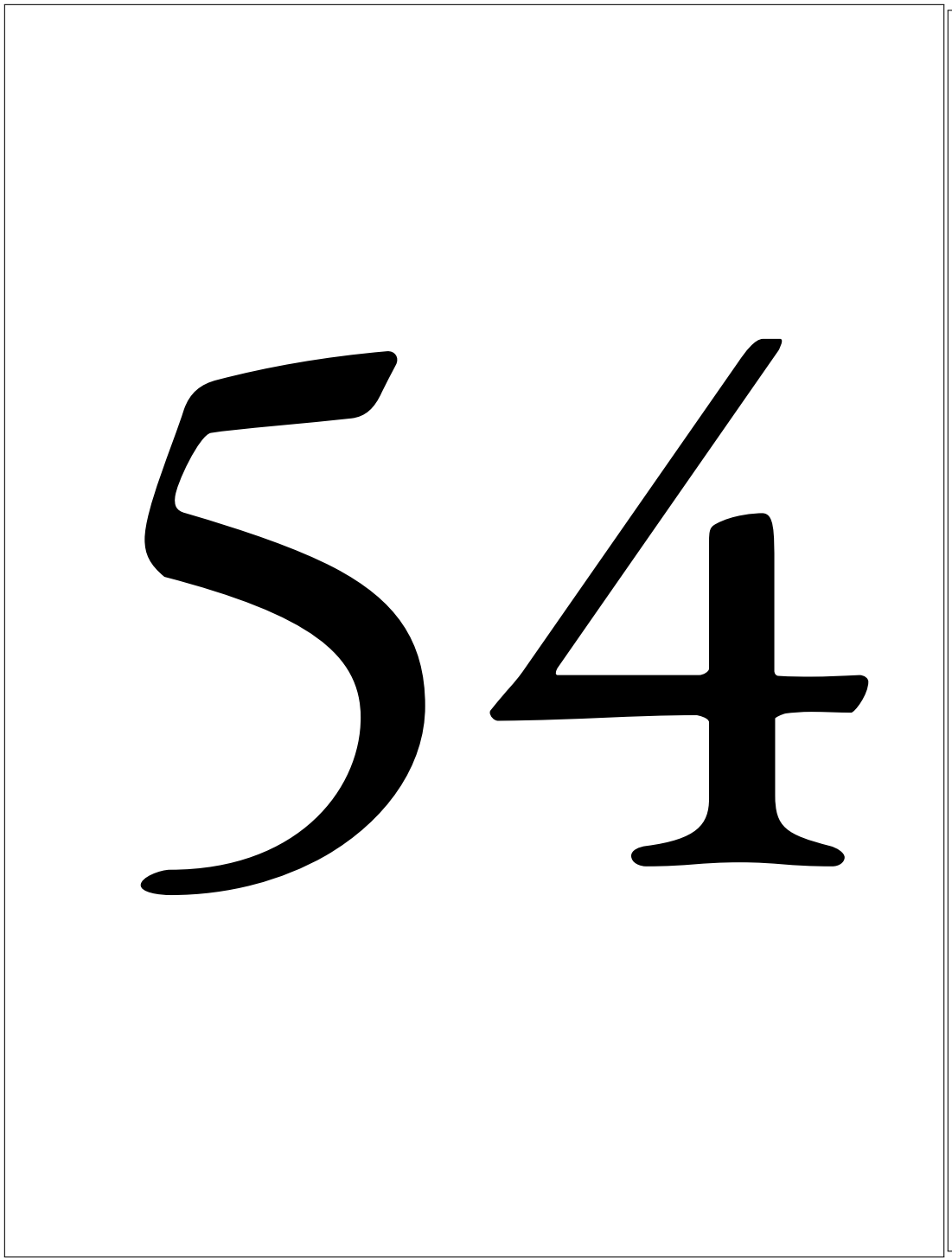
38

41


43

47

513

53


5457

61

65

68

69

70

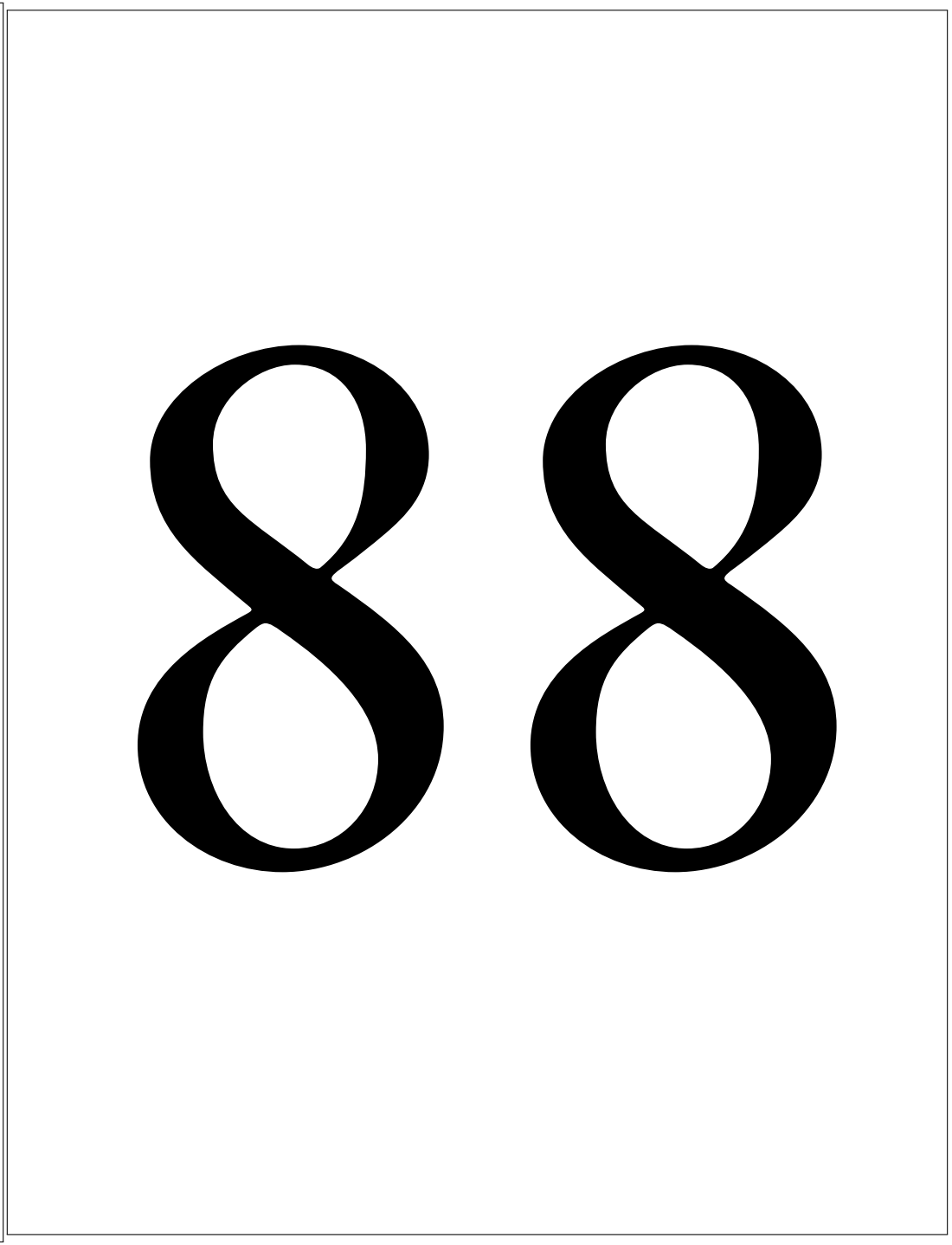
72

7578

80

81

8385


87

88

89

90

9395

96

97

98

99

Informations légales

- **Auteur :** Fabrice ARNAUD
- **Web :** pi.ac3j.fr
- **Mail :** contact@ac3j.fr

- **Dernière modification :** 1er novembre 2026 à 11:08

Ce document a été écrit pour LaTeXavec l'éditeur VIM - Vi Improved Vim 9.1.2141
Il a été compilé sous Linux Ubuntu Resolute Raccoon (Le Raton Laveur résolu) 26.04 avec la distribution TeX Live 2025.20260124 et LuaTex 1.22.0

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 pilleurs 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 même 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'Oeuvre 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 crée par **Fabrice ARNAUD (contact@ac3j.fr)** le 1er novembre 2026 à 11:08.
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 :** 1er novembre 2026 à 11:08

Ce document a été écrit pour LaTeXavec l'éditeur VIM - Vi Improved Vim 9.1.2141
Il a été compilé sous Linux Ubuntu Resolute Raccoon (Le Raton Laveur résolu) 26.04 avec la distribution TeX Live 2025.20260124 et LuaTex 1.22.0

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 pilleurs 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 même 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'Oeuvre 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 crée par **Fabrice ARNAUD (contact@ac3j.fr)** le 1er novembre 2026 à 11:08.
Il est disponible en ligne sur **pi.ac3j.fr**, **Le blog de Fabrice ARNAUD**.
Adresse de l'article : https://pi.ac3j.fr/mathematiques-college