

## Table des matières

### Dates de la formation

2

### Mode d'emploi

2

### Progression

3

### Le monde Informatique

4

### Représentation de l'information

5

- (110) Codage binaire . . . . . 5
- (111) Écriture d'un entier positif dans une base  $b \geq 2$  . . . . . 6
- (112) Représentation d'un entier relatif . . . . . 9
- (113) Représentation d'un nombre décimal . . . . . 13
- (114) Représentation d'un texte . . . . . 14
- (115) Logique booléenne . . . . . 19
- (120) Types construits : les p-uplets . . . . . 24
- (121) Types construits : les p-uplets nommés . . . . . 26
- (122) Types construits : les tableaux . . . . . 28
- (123) Types construits : les dictionnaires . . . . . 31
- (130) Le vocabulaire des données . . . . . 34
- (131) Manipuler des données avec un tableau . . . . . 36
- (132) Manipuler des données avec python . . . . . 38
- (133) Table(s) de données . . . . . 39
- (134) Requêtes . . . . . 40

### algorithmique

42

- (210) Module turtle pour présenter python . . . . . 42
- (220) Parcours séquentiel d'un tableau . . . . . 44
- (225) Parcours dichotomique . . . . . 47
- (230) Tri : Réflexions préalables . . . . . 51
- (231) Algorithme de tri par sélection . . . . . 53
- (232) Algorithme de tri par insertion . . . . . 54

### Langage et programmation

55

- (310) Algorithme versus programme . . . . . 55
- (320) les fonctions informatiques - effets de bord . . . . . 56
- (330) Récursivité . . . . . 58
- (340) Langage de description - le fond. Exemple de l'html . . . . . 60
- (341) Langage de description - la forme. Exemple du CSS . . . . . 63
- (350) POO : vocabulaire de la programmation orienté objet . . . . . 64
- (351) POO : Comprendre la programmation orienté objet . . . . . 65
- (352) POO : Manipuler la programmation orienté objet . . . . . 68
- (353) POO : héritage . . . . . 70

### Machine

71

- (410) Éléments d'un pc . . . . . 71
- (415) Le langage assembleur . . . . . 72
- (420) Modèle TCP/IP . . . . . 75
- (421) Comprendre les algorithmes de routage . . . . . 76
- (422) routage élastique . . . . . 78
- (430) Filius : simulation d'un réseau . . . . . 83

### Projet : retouche d'images

86

- (510) Image : premiers codages . . . . . 86
- (520) Image : prise en main de PILLOW . . . . . 88
- (530) Image : premières retouches d'images . . . . . 89
- (540) Image : tramage . . . . . 91
- (550) Image : Filtre de convolution . . . . . 93
- (560) Image : Stéganographie . . . . . 94
- (570) Image : traitement parallèle . . . . . 95
- (580) Image : POO et images . . . . . 96

### Correction des exercices

97

# Mode d'emploi

## Auteurs

Maxime Fourny

Patrice Péröt

## Mode d'emploi du fichier pdf

1. Ce PDF est cliquable :
  - En cliquant en haut au centre : "Retour : table des matières", vous serez ramenés à la table des matières.
  - La table des matières est cliquable : un clic sur un document vous y amènera.
  - Si vous cliquez sur le numéro d'un exercice, vous serez envoyés vers sa correction (Elles ne sont pas toutes faites). Un clic sur le numéro de la correction d'un exercice vous ramènera vers l'exercice
2. Ce PDF contient des fichiers attachés :
 

une icône de la forme représente un tel fichier.  
 Un clic droit puis enregistrer sous permet de le récupérer.  
 Attention : les fichiers python sont au format .txt car certains lecteurs de pdf refusent le .py.  
 Il conviendra donc d'en faire un renommage via un IDE python.
3. Si vous détectez une erreur/coquille, n'hésitez pas à le signaler à :

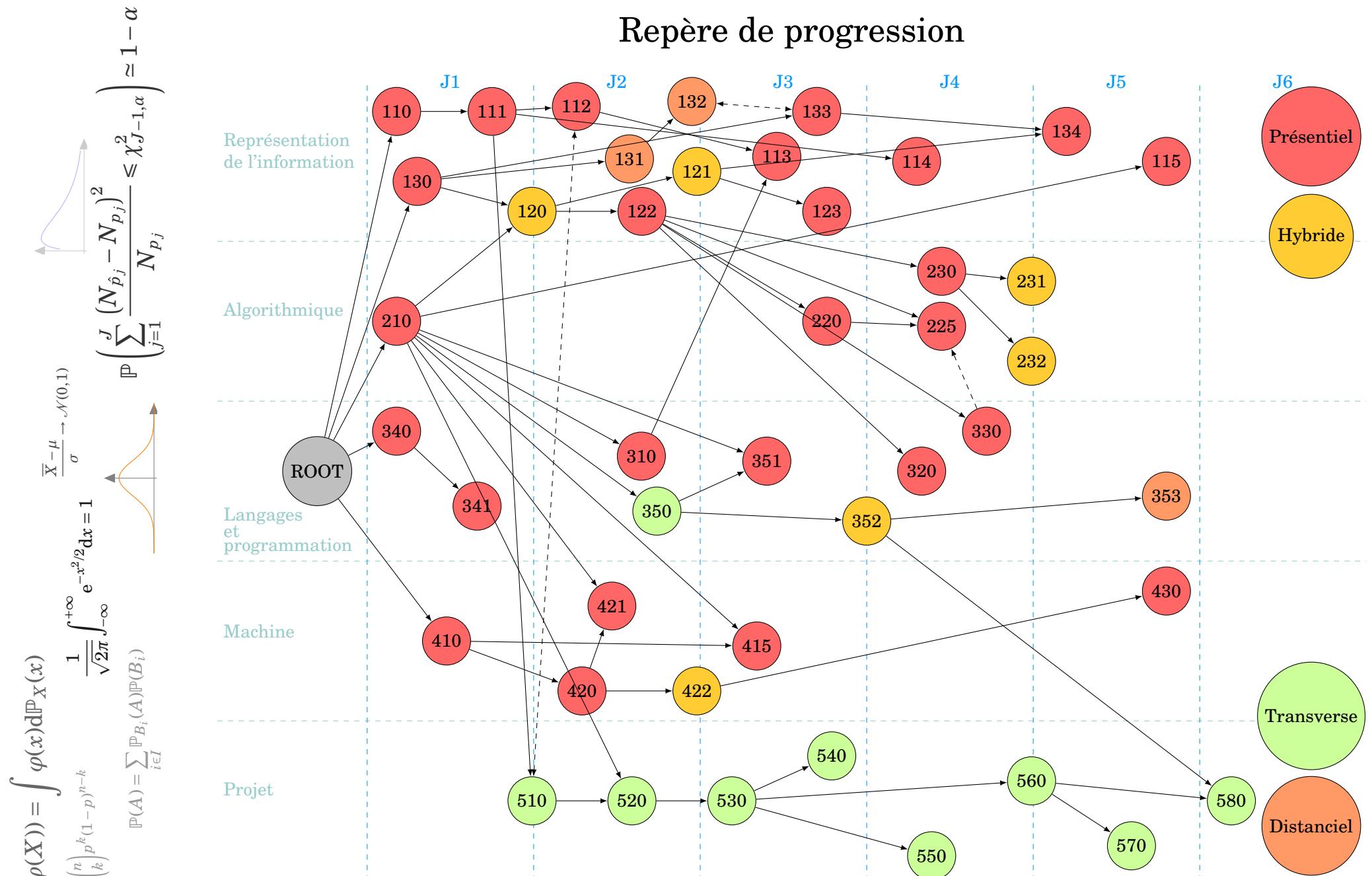
[maxime-simon.fourny@ac-besancon.fr](mailto:maxime-simon.fourny@ac-besancon.fr)

## Dates de la formation

1. Vendredi 4 septembre
2. Vendredi 18 septembre

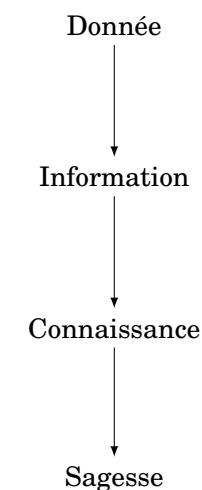
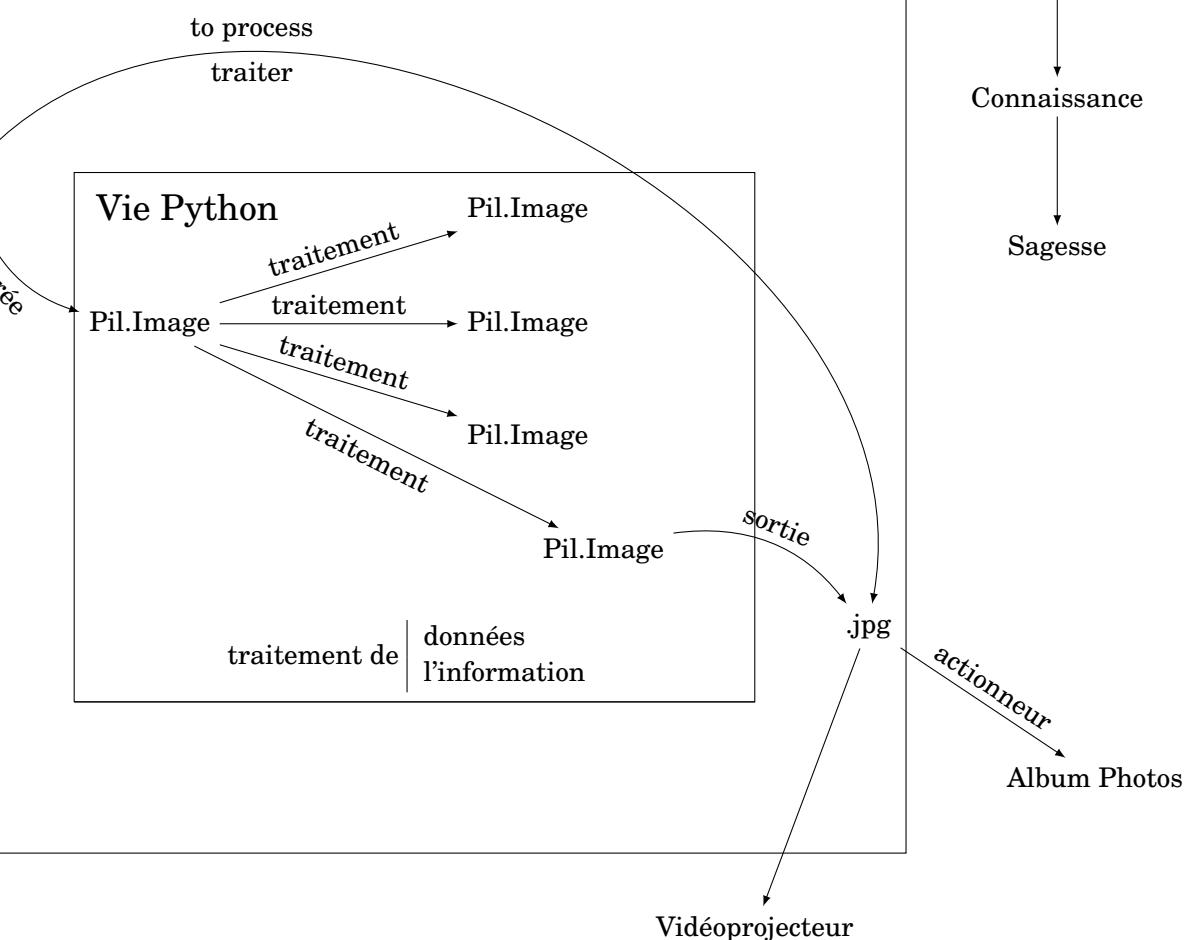
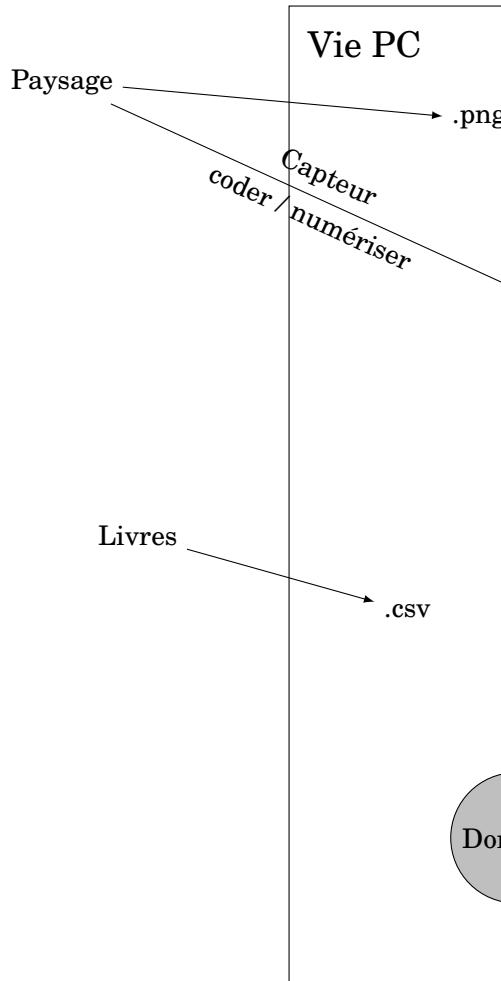
3. Vendredi 25 septembre
4. Vendredi 2 octobre
5. Vendredi 9 octobre
6. Vendredi 16 octobre

# Repère de progression



# Le monde informatique

Monde réel



# Codage binaire

Toute machine informatique manipule une représentation des données dont l'unité minimale est le bit - binary digit (chiffre binaire)- valant soit 0 soit 1, ce qui permet d'**unifier logique et calcul**.

Actuellement La mémoire des ordinateurs est implémenté technologiquement par une multitude de minuscules circuits électroniques qui ne peuvent être, chacun, que dans deux états. On a décidé (convention internationale) d'appeler ces deux états 0 et 1. L'état d'un circuit, composé de plusieurs de ces circuits mémoire un bit, se décrit donc par une suite finie de 0 et de 1 que l'on appelle mot.

## Définition 1 (Mot binaire).

Un mot binaire est un mot dont les seules lettres sont soit 0 soit 1.

## Représentation de l'information en binaire

### 1 Mot de 8 bit

Comment appelle-t-on un mot de 8 bits ? Qu'est ce qu'un byte ?

### 2 Circuit mémoire de 4 et 10 bits

- On imagine un ordinateur dont la mémoire est constituée de quatre circuits mémoire un bit.

Quel est le nombre d'états possibles de la mémoire de cet ordinateur ?

- Même question pour un ordinateur dont la mémoire est constituée de dix circuits mémoire un bit.

### 3 Et avec un peu plus de bits ?

Même question pour un ordinateur dont la mémoire est constituée de 34 milliards de tels circuits.

### 4 Du bit vers le booléen

Trouvez trois informations de la vie courante qui peuvent être exprimées par un mot d'un bit.

## Définition 2 (Coder).

L'action qui consiste à représenter un objet en un mot binaire s'appelle un codage. Un codage peut être standardisé internationalement ou être localisé.

### 5 Codage de l'arc en ciel

On veut représenter chacune des sept couleurs de l'arc-en-ciel par un mot binaire, les sept mots devant être distincts et de même longueur.

- Quelle est doit être la longueur minimale de ces mots ?
- Proposer un codage des couleurs de l'arc en ciel.
- Combien de couleurs contient un arc en ciel ?

### 6 Codage d'une panne

On considère une « box internet » qui est munie d'une diode électroluminescente éteinte ou allumée selon un motif de 0 et de 1 qui change à chaque demi-seconde. Lorsque la box est éteinte, la diode aussi, le motif est 000000000... Lorsque la box est allumée et fonctionne, la diode est allumée en continu, le motif vaut donc 111111111...

Lorsque la box est en panne, le fournisseur d'accès souhaite que la diode clignote selon différents motifs durant 6 secondes, en fonction du type de panne : pas de réseau, réseau saturé, facture non payée, etc.

On parle ainsi de clignotement rapide pour le motif 010101010101.

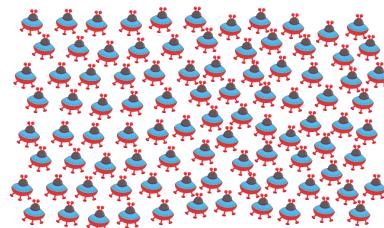
- Proposer deux motifs qui pourraient correspondre aux descriptions « clignotement lent » et « clignotement très lent » ?
- Comment peut-on décrire les motifs suivants, répétés indéfiniment :
  - 000001000000 et 010101000000 ?
- Combien y a t-il de motifs possibles avec une série de dix booléens ?
- Comment faut-il procéder pour qu'un motif donné ne s'affiche que toutes les dix secondes ?
- Dans une situation où il y aurait deux types de panne en même temps, comment pourrait-on procéder pour afficher les motifs correspondant aux deux messages d'erreurs différents ?

# Représentation d'un nombre entier positif

## Du dénombrement vers la base

### 7 Nombre de vaisseaux

Déterminer le nombre de vaisseaux ci-dessous :



### 8 chiffres

Qu'est ce qu'un chiffre ?

## Un peu d'histoire

Depuis le Moyen Âge, on écrit les nombres entiers naturels en *notation décimale à position*. Cela signifie que, pour écrire le nombre entier naturel  $n$ , on commence par imaginer  $n$  objets, que l'on groupe par paquets de dix, puis on groupe ces paquets de dix objets en paquets de dix paquets, etc. À la fin, il reste entre zéro et neuf objets isolés, entre zéro et neuf paquets isolés de dix objets, entre zéro et neuf paquets isolés de cent, etc. Et on écrit cet entier naturel en écrivant de droite à gauche, le nombre d'objets isolés, le nombre de paquets de dix, le nombre de paquets de cent, le nombre de paquets de mille, etc. Chacun de ces nombres étant compris entre zéro et neuf, seuls dix chiffres sont nécessaires : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9. Par exemple, l'écriture 2359 exprime un entier naturel formé de 9 unités, 5 dizaines, 3 centaines et 2 milliers.

Pourquoi des paquets de 10 ? On aurait pu tout aussi bien décider de faire des paquets de deux, de cinq, de douze, de vingt, de soixante, etc.

On écrirait alors les nombres entiers naturels en *notation à position en base deux, cinq, douze, vingt ou soixante*.

Ce choix est lié à la physiologie humaine : il correspond à notre nombre de doigts. D'ailleurs doigt et digit (chiffre en anglais) ont la même racine étymologique.

La notation décimale à position s'appelle donc aussi la *notation à position en base dix*.

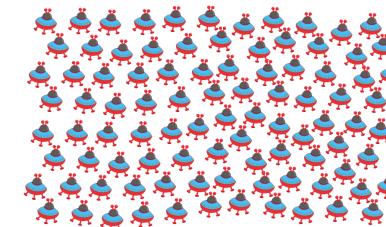
**9** Un horloger excentrique a eu l'idée de fabriquer une montre sur laquelle l'heure est indiquée par 10 diodes électroluminescentes appelées 1 h, 2 h, 4 h, 8 h, 1 mn, 2 mn, 4 mn, 8 mn, 16 mn et 32 mn. Pour connaître l'heure, il suffit d'ajouter la valeur de toutes les diodes allumées.

1. Quelle heure est-il quand sont allumées les diodes 1 h, 2 h, 4 h, 1 mn, 2 mn, 8 mn, 16 mn et 32 mn ?
2. Quelles sont les diodes allumées à 5 h 55 ?
3. Est-il possible de représenter toutes les heures ?
4. Toutes les configurations sont-elles la représentation d'une heure ?

## La base cinq

### 10 Nombre de vaisseaux

1. Déterminer le nombre de vaisseaux ci-dessous en *notation à position en base cinq* :



2. De combien de chiffres différents a-t-on besoin en base 5 ?

**11** Vider une trousse et exprimer le nombre de stylos qu'elle contenait en *notation à position en base cinq*

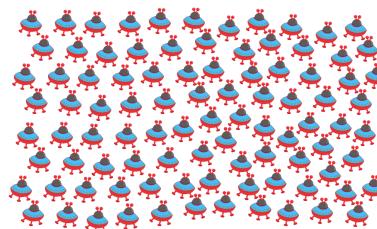
**12** Trouver la représentation en base cinq de 47, de 944, de 289.

**13** Trouver la représentation en base dix du nombre 401302 (en base cinq), de  $2341_5$ , de  $444_5$ .

### La base deux

#### 14 Nombre de vaisseaux

- Déterminer le nombre de vaisseaux ci-dessous en *notation à position en base deux* :



- De combien de chiffres différents à-t-on besoin en base 2 ?

**15** Vider votre trousse et exprimer le nombre de stylos qu'elle contenait en *notation à position en base deux*

**16** Trouver la représentation en base deux du nombre 13, de 1000.

**17** Donner la représentation en base deux des nombres 1, 3, 7, 15, 31 et 63.

**18** Trouver la représentation en base dix du nombre 11111111, puis de 10010110

**19** C'est en 11110010000 qu'a été démontré le théorème fondamental de l'informatique. Exprimer ce nombre en base dix.

**20** Combien de nombres entiers positifs différents peut-on représenter avec un mot de  $n$  bits ?

#### Définition 3 (Codage binaire d'un nombre entier positif).

Cette représentation d'un nombre entier positif par le mot binaire obtenu comme précédemment constitue le codage le plus souvent utilisé pour représenter un nombre entier positif en informatique.

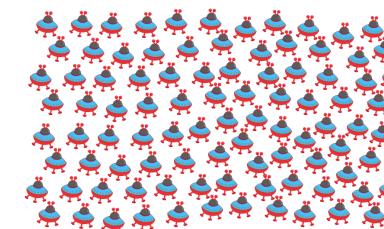
**21** Pour multiplier par dix un entier naturel exprimé en base dix, il suffit d'ajouter un 0 à sa droite, par exemple,  $12 \times 10 = 120$ . Quelle est l'opération équivalente pour les entiers naturels exprimés en base deux ? Exprimer en base deux les nombres 3, 6 et 12 pour illustrer cette remarque.

### En base 16

**22** Quel est le problème de la base 16 par rapport à précédemment ?

#### 23 Nombre de vaisseaux

Déterminer le nombre de vaisseaux ci-dessous en *notation à position en base 16* :



**24** Trouver la représentation en base seize du nombre 965, de 6725 et de 18379.

**25** Trouver la représentation en base dix du nombre 4E2C (en base 16), ABCD (en base 16), 281EF (en base 16).

**26** *Intérêt ?*

Quel est l'intérêt de la base 16 en informatique ?

**27** Chercher sur le web ce qu'est le système de numération Shadok. Est-ce un système de numération à position ? Si oui, en quelle base et avec quels chiffres ?

# Représentation d'un nombre entier relatif

## Rappel : notion de base

### 28 L'algorithme est-ce si compliqué ?

Pour jouer au jeu : choisis un nombre entier positif que je dois retrouver. Tu ne peux répondre à mes questions que par oui ou par non.

1. Combien faut-il de questions au minimum pour retrouver avec certitude n'importe quel nombre entre 0 et 100 ?
2. Combien faut-il de questions au minimum pour retrouver avec certitude n'importe quel nombre entre 0 et 255 ?
3. Comment s'appelle cet algorithme ?

### 29 Conversion

- Convertir en base 10 le nombre  $10110101_{(2)}$
- Convertir en base 2 le nombre  $243_{(10)}$
- Convertir en base 10 le nombre  $2F3_{(16)}$
- Convertir en base 2 le nombre  $3EC2_{(16)}$

### 30 Opérations

1. Additionner  $10110101_{(2)}$  et  $01011100_{(2)}$ . Vérifier que le calcul correspond en base 10.  
Quel problème cela peut-il soulever ?
2. Multiplier  $10110101_{(2)}$  par  $01011100_{(2)}$ .
3. Diviser  $10110100_{(2)}$  par  $00000010_{(2)}$
4. Soustraire  $01011100_2$  à  $10110101_2$ .

## Nombres entiers signés

**Définition 4 (Nombres entiers signés).** Les nombres entiers signés des informaticiens sont les entiers relatifs pour les mathématiciens.

### Premier codage des nombres entiers signés :

Au début de la programmation informatique, la méthode utilisée pour représenter en binaire les nombres entiers signés a été d'utiliser une représentation de type «signe, magnitude», le nombre de bits utilisés étant fixé.

Ainsi le premier bit (le bit de poids fort) valait 0 si le nombre était positif et 1 si le nombre était négatif.

Les bits restants représentent la valeur absolue du nombre entier signé en utilisant la représentation d'un entier positif en base 2 comme vu précédemment.

### 31 Donner la représentation en mot binaire selon cette norme de codage des nombres 123, -123, 46, -99 et 0.

### 32 Sur 8 bits

Avec 8 bits quels nombres entiers signés peut-on numériser selon ce codage ?

### 33 Sur 4 bits

1. Avec 4 bits quels nombres entiers signés est-on capable de numériser selon ce codage ?
2. Donner toutes les correspondances.

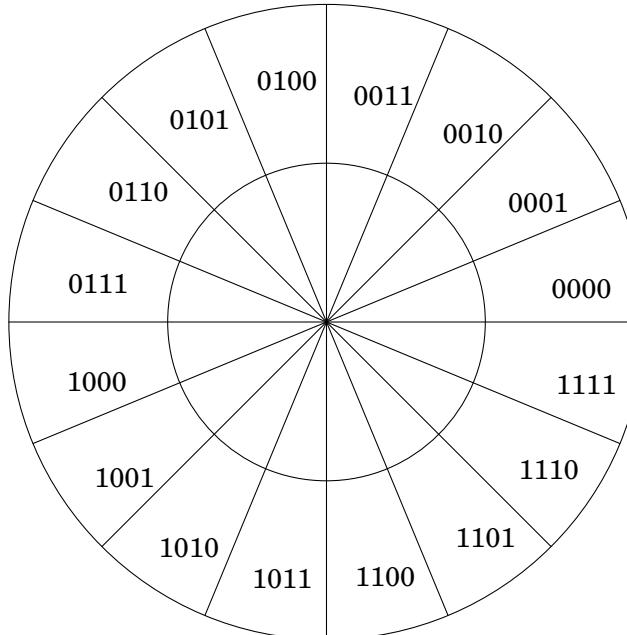
### 34 Calculer $(+2) + (-2)$

### 35 Nombre de bits fixé

Quel problème est généré par le fait que le nombre de bits pour représenter un nombre entier est fixé.

**36** Dans le cercle ci dessous, des nombres entiers signés ont été positionnés. Ils sont représentés par le mot binaire selon le codage évoqué jusqu'à présent.

1. Compléter le cercle suivant avec la correspondance en base 10.



2. Quels sont les défauts de cette représentation ?

### Complément à un

#### Définition 5 (Complément à un).

Le complément à un d'un mot binaire est le mot binaire obtenu en inversant tous les bits du mot de départ.

#### 37 Complément à un

Donner le complément à un des mots 1010 1100 et 0101 0001.

#### Deuxième codage des entiers signés :

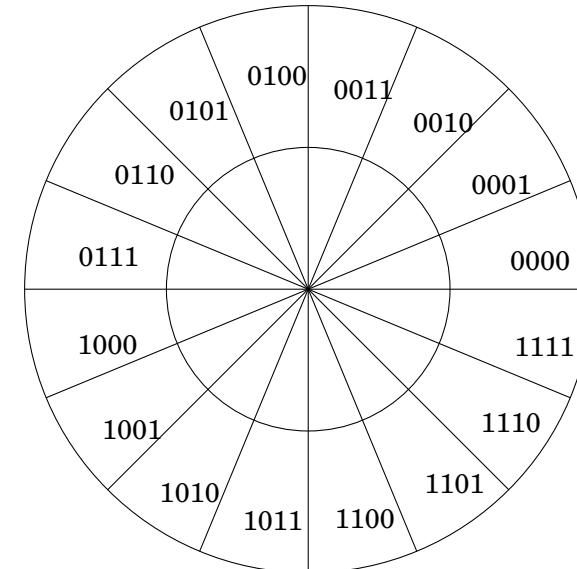
Maintenant on code l'opposé d'un nombre positif par le complément à un du nombre positif. Le codage d'un nombre positif, lui ne change pas.

#### 38 Calcul de complément à un

1. Donner la représentation de -68 sur 8 bits.
2. Donner la représentation de -45 sur 8 bits.
3. Quel est le nombre correspondant à  $10101100_2$  ?
4. Quel effet à ce codage sur le bit de poids fort.

**39** Dans le cercle ci dessous la représentation des nombres est notée en binaire sur 4 bits selon la méthode du complément à un.

1. Compléter le cercle suivant avec la correspondance en base 10.



2. Quel défaut subsiste-t-il ?

## Complément à deux

**Définition 6 (Complément à deux).** Le complément à deux d'un mot binaire est la valeur numérique obtenue en base 2 en ajoutant 1 à son complément à un (le mot étant alors assimilé à un nombre. Numériser vous avez dit ?).

### 40 Complément à deux

Donner le complément à deux des mots 1010 1100 et 0101 0001.

#### Troisième codage des entiers signés :

L'opposé d'un nombre positif est codé par le complément à deux du nombre positif correspondant.

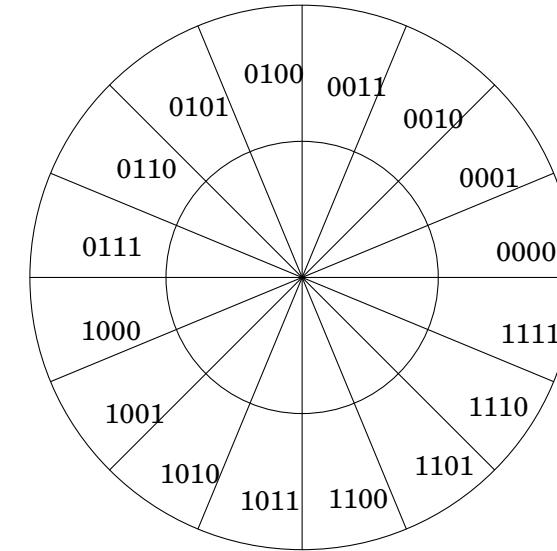
Le codage d'un nombre positif, lui ne change pas.

### 41 Calcul de complément à deux

1. Donner la représentation de -68 sur 8 bits.
2. Donner la représentation de -45 sur 8 bits.
3. Quel est le nombre correspondant à  $10101100_2$  ?
4. Quel effet à ce codage sur le bit de poids fort.

**42** Dans le cercle ci dessous la représentation des nombres est notée en binaire sur 4 bits selon la méthode du complément à deux.

1. Compléter le cercle suivant avec la correspondance en base 10.
2. Quel défaut subsiste-t-il ?
3. Quel est le réel intérêt d'un codage des entiers relatifs si compliqué ?



### 43 Taille d'un entier : implémentation langage

Souvent les langages de programmation implémentent différents types d'entiers signés : les courts (2 octets), les longs (4 octets), les long long (8 octets)...

1. Déterminer dans chacun des cas, après avoir préciser le nombre de bits alloués à la valeur proprement dite du nombre, la palette de nombres que l'on peut représenter.
2. Déterminer la palette des nombres que l'on peut représenter avec  $n$  bits.

### 44 Implémentation en python

Au vu de ce qui a été vu précédemment, essayer de déterminer le nombre d'octets alloués au codage des entiers signés en python.

## Pixel Art

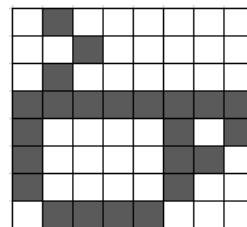
### 45 Pixel art

Barnabé et Cerise qui s'ennuient en NSI décident de mettre en place un chiffrement pour s'envoyer en toute discréte des petites images en noir et blanc, carrées, de 8 pixels de côté.

En plein cours l'enseignant intercepte le message suivant :

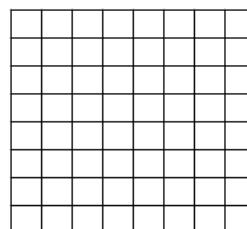
$$(-61, -127, 0, 36, -127, -61, -127, 36)$$

1. A quelle image correspond ce message chiffré ?
2. En utilisant le même chiffre, chiffrer l'image suivante :



3. Déchiffrer la suite suivante :

$$(-1, -33, 31, 30, -63, -63, -35, -103)$$



#### 46 Négatif et pixel art

1. Donner l'encodage des négatifs des images de l'exercice précédent.
2. Proposer un protocole rapide permettant de générer l'encodage des négatifs.

#### Complément à un et deux - un peu de théorie

#### 47 Complément à un

Quelle est la représentation binaire du nombre 57 ? Et celle du nombre 198 ?

Soit  $m$  un mot de 8 bits,  $n$  l'entier naturel représenté en binaire par le mot  $m$ ,  $m'$  le mot obtenu en remplaçant dans  $m$  chaque 0 par un 1 et chaque 1 par un 0 et  $n'$  l'entier naturel représenté en binaire par le mot  $m'$ . Exprimer  $n$  et  $n'$  comme une somme de puissances de 2, montrer que  $n + n' = 255$ . Montrer que la représentation binaire du nombre  $255 - n$  est obtenue en remplaçant dans celle de  $n$  chaque 0 par un 1 et chaque 1 par un 0.

#### 48 complément à deux

1. Quelle est la représentation binaire en complément à deux de 42 ? de -43 ? de -1 ?
2. Soit  $m$  un mot de 8 bits,  $n$  l'entier relatif représenté par le mot  $m$  par la méthode du complément à deux,  $m'$  le mot obtenu en remplaçant dans  $m$  chaque 0 par un 1 et chaque 1 par un 0 et  $n'$  l'entier relatif représenté par le mot  $m'$  par la méthode du complément à deux.  
Démontrer que  $n + n' = -1$ .
3. Retrouver à l'aide de cette formule la méthode pratique pour obtenir la représentation binaire d'un entier relatif  $n$  par la méthode du complément à deux lorsque  $n$  est négatif.

# Décimale flottante

## 49 Des puissances de deux

1. Donner la valeur exacte de  $2^i$  pour  $i$  allant de  $-1$  à  $-8$ .
2. Écrire les nombres suivants comme somme de puissances de 2 :

- |            |             |
|------------|-------------|
| (a) 0,75   | (d) 14,8125 |
| (b) 0,375  | (e) 49,625  |
| (c) 0,4375 | (f) 0,1     |

3. Peut-on écrire n'importe quel nombre décimal comme somme d'une puissance de 2 ?
4. Déterminer tous les nombres décimaux que l'on peut écrire comme somme de puissances de 2 de  $2^{-1}$  à  $2^{-4}$ . (On pourra s'aider d'un arbre des possibles)
5. Parmi les nombres décimaux écrits trouvés dans la question précédente, déterminer le plus proche de :

- |         |         |                   |
|---------|---------|-------------------|
| (a) 0,1 | (b) 0,4 | (c) $\frac{1}{3}$ |
|---------|---------|-------------------|

## 50 Nombres à virgule en base 2

1. Donner l'écriture décimale des nombres :
 

(a) 0 <sub>2</sub>	(c) 0,01 <sub>2</sub>
(b) 0,1 <sub>2</sub>	(d) 0,101 <sub>2</sub>
2. Donner la représentation binaire de :
 

(a) 0,5	(d) 0,0625
(b) 0,75	(e) 0,421875
(c) 0,125	(f) 0,1

## 51 En binaire

1. Déterminer la représentation en base 2 des nombres suivants :

- |           |              |
|-----------|--------------|
| (a) 1,5   | (c) 39,3125  |
| (b) 7,625 | (d) 100,6875 |

2. Déterminer une représentation approchée binaire sur 4 bits après la virgule des nombres suivants :

- |          |           |
|----------|-----------|
| (a) 28,7 | (b) 50,44 |
|----------|-----------|

## 52 Écriture flottante normalisée

Déterminer l'écriture flottante normalisée de :

- -0,3125
- 1,5
- 7,625
- 39,3125

## 53 un algorithme pour convertir

Utiliser la méthode algorithmique vue en cours pour déterminer l'écriture binaire à virgule des nombres suivants :

- |            |        |                  |
|------------|--------|------------------|
| 1. 15,4375 | 2. 0,7 | 3. $\frac{1}{3}$ |
|------------|--------|------------------|

## 54 le même en Python

Écrire une fonction Python qui prend en argument un nombre décimal et un entier  $k$  et qui retourne l'écriture binaire à virgule de ce nombre tronqué à  $k$  bits après la virgule.

# Représentation d'un texte en machine - Encodage

**Fichiers attachés :** extension\_ascii\_latin1.pdf

**Fichiers attachés :** table\_ascii.pdf

## Code versus chiffre

### 55 déchiffrer/décoder/coder/chiffrer/décrypter

Deux élèves trouvant le cours peu intéressant ont décidé de mettre en place un système pour communiquer par écrit. Comme ils ne souhaitaient pas pouvoir être relus par d'autres élèves, ils ont réfléchi à un chiffrage des caractères.

1. Leur enseignant a intercepté le message suivant :

2 - 15 - 14 - 10 - 15 - 21 - 18

Déchiffrer ce message.

2. Proposer une table de codage pouvant être celle des deux protagonistes.
3. Le professeur, facétieux, leur rend un devoir dont l'appréciation est :

1 - 21 - 27 - 2 - 15 - 20 - 12 - 15 - 16 - 28

À l'aide de la table de codage, déchiffrer ce message.

4. Proposer une définition de :

- Coder
- Chiffrer
- Décoder
- Déchiffrer
- Décrypter
- Crypter ?

5. Énoncer tous les caractères qu'un américain pourrait utiliser.

## Encodage ASCII

### 56 Table ASCII

L'American Standard Code for Information Interchange (Code américain normalisé pour l'échange d'information), plus connu sous l'acronyme ASCII, est une norme informatique de codage de caractères apparue dans les années 1960. C'est la norme de codage de caractères la plus influente à ce jour. ASCII définit 128 codes à 7 bits, comprenant 95 caractères imprimables : les chiffres arabes de 0 à 9, les lettres minuscules et capitales de A à Z, et des symboles mathématiques et de ponctuation.

1. Expliquer pourquoi L'American Standard Code for Information Interchange s'est limité à 7 bits ?
2. Classer les différents caractères de la table ASCII en catégories
3. Pourquoi 31 caractères de contrôle ?
4. Pourquoi ne pas avoir mis les majuscules à la suite des minuscules ?

### 57 utilisation de la table ASCII

1. Déterminer l'encodage ASCII de la chaîne de caractères suivante, en hexadécimal et en binaire :

"Hello World !"

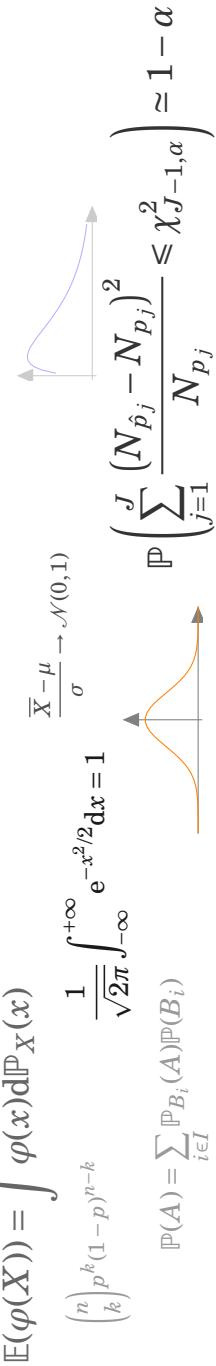
2. Déterminer la chaîne de caractères dont l'encodage ASCII en hexadécimal est le suivant, et vous découvrirez le créateur du langage Python :

47 75 69 64 6F 20 56 61 6E 20 52 6F 73 73 75 6D

### 58 Huitième bit : bit de parité

1. L'ASCII définit 128 caractères numérotés de 0 à 127 et codés en binaire de 0000000 à 1111111. Sept bits suffisent donc.

Toutefois, les ordinateurs travaillant presque tous sur un multiple de huit bits (byte = un octet) depuis les années 1970, chaque caractère d'un texte en ASCII est souvent stocké dans un octet dont le 8e bit est 0.



Écrire, selon cette convention le mot "Bonjour" en binaire.

**Pour la culture :** Aujourd'hui encore, certains systèmes de messagerie électronique et de SMS fonctionnent avec des bytes ou multiplets composés de seulement sept bits (contrairement à un octet qui est un byte ou multiplet standardisé à huit bits).

2. Une autre utilisation possible est d'utiliser le 8ème bit comme bit de parité : le nombre de 1 dans l'écriture binaire sur un octet doit être pair.

Écrire, selon cette deuxième convention, le mot "Bonjour" en binaire.

- (a) Quel est l'intérêt de ce bit de parité ?
- (b) Quel en est la limite ?

## Extension de la table ASCII

### 59 Extension à 8 bits

En 1986, le progrès des technologies a permis d'intégrer à la table de codage le huitième bit des octets. Cette année-là a vu la création de ce qui deviendra la table ISO 8859-1, dont le nom complet est ISO/CEI 8859-1.

Souvent dénommée "Latin-1" ou "Europe occidentale", elle étend la table ASCII avec 128 caractères supplémentaires.

Elle contient 191 caractères de l'alphabet latin, chacun d'entre eux étant codé sur un octet (soit 8 bits).

1. Combien de caractères peut-on coder si l'on décide d'utiliser le 8ème bit ?
2. Quel est le code du caractère "é" ?
3. Quel est le code du caractère "1" ? Après l'avoir écrit en binaire, comparer avec le codage du nombre 1 sur un octet.
4. Quel est le code de la chaîne de caractère "-42". Après l'avoir écrit en binaire, Comparer avec le codage du nombre -42 sur un octet.
5. Est-il possible de coder tous les caractères du monde en étendant la table ASCII sur un octet ?

6. Expliquer pourquoi il n'est pas judicieux d'étendre la table ASCII pour coder tous les caractères du monde sur plusieurs octets.

### 60 norme ISO-8859-1

1. Ouvrir NotePad++ puis créer un nouveau fichier. Cliquer sur "Encoding" puis "Convert to ANSI" (il s'agit de la norme ISO-8859-1).
2. Dans le fichier texte, taper "Comment ça va ?" puis enregistrer.
  - (a) Quelle est la taille du fichier ainsi créé ?
  - (b) Ouvrir ce fichier avec EditHexa. Quel est l'encodage du "ç" ?
3. Ouvrir de nouveau le fichier avec NotePad++. Ajouter à la ligne "Très bien, et toi ?" puis enregistrer.
  - (a) Quelle est maintenant la taille du fichier ?
  - (b) Ouvrir ce fichier avec EditHexa. Quel est l'encodage du retour à la ligne ?

### 61 ord et chr

1. Déterminer ce que font les fonctions python ord et chr ?
2. Écrire une fonction en python, qui prendra une lettre minuscule et la convertira en majuscule.
3. Énoncer une pré-condition à cette fonction.
4. Énoncer une post-condition à cette fonction. Est-il nécessaire de l'implémenter.
5. Implémenter la pré-condition de cette fonction.
6. Écrire une fonction python qui prendra en entrée une chaîne de caractères et la convertira en minuscule.
7. Écrire une fonction python qui prendra en entrée une chaîne de caractères et la convertira en majuscule.

## Norme Unicode et encodage UTF-8

### Pourquoi une nouvelle norme :

Pour pallier le fait que chaque pays, chaque langue fasse sa propre extension de la table ASCII...

Par ailleurs, tous les alphabets autres que l'alphabet latin ne sont pas gérés par l'ASCII. L'Asie existe !

Chaque langue se voit dotée d'une extension de la norme ASCII pour coder son alphabet.

Une seule norme pour toute la planète (tour de Babel)

### Unicode :

Unicode est un standard informatique qui permet des échanges de textes dans différentes langues, à un niveau mondial. Il est développé par le Consortium Unicode, qui vise au codage de textes écrits en donnant à tout caractère de n'importe quel système d'écriture un nom et un identifiant numérique, et ce de manière unifiée, quels que soient la plate-forme informatique ou le logiciel utilisés.

La dernière version, Unicode 12.0, a été publiée en mars 2019.

### Limite de l'octet :

Pour s'affranchir des contraintes rigides des normes précédentes (une suite de bits, une représentation), Unicode sépare dorénavant d'une part la définition du jeu de caractères (liste des caractères par leur nom) et leur index (point de code) de celle du codage.

### La représentation en mémoire n'est plus égale au point de code.

Ainsi, on ne peut donc pas parler de la taille d'un caractère Unicode, car elle dépend du codage choisi, et celui-ci peut donc varier à volonté. En pratique, UTF-8 est très utilisé dans les pays occidentaux.

Là où l'ASCII utilise 7 bits et ISO/CEI 8859-1 8 bits (comme la plupart des pages de codes nationales), Unicode, qui rassemble les caractères de chaque page de code, avait besoin d'utiliser plus que les 8 bits d'un octet. La limite fut dans un premier temps fixée à 16 bits pour les premières versions d'Unicode, et à 32 bits pour les premières versions de la norme ISO/CEI 10646.

### UTF-8 :

UTF-8 (abréviation de l'anglais Universal Character Set Transformation Format1 - 8 bits) est un codage de caractères informatiques conçu pour coder l'ensemble des caractères du « répertoire universel de caractères codés », initialement développé par l'ISO dans la norme internationale ISO/CEI 10646, aujourd'hui totalement compatible avec le standard Unicode, en restant compatible avec la norme ASCII limitée à l'anglais de base, mais très largement répandue depuis des décennies.

#### 62 point de code versus représentation en mémoire

1. A quel caractère correspond le point de code 8364 ? commenter *ord* et *chr* au regard de l'encodage utilisé.
2. Combien de bits sont nécessaires pour l'écrire ? Combien d'octets faudrait-il ?
3. Trouver la représentation en mémoire de ce caractère selon la norme d'encodage UTF8, UTF16, UTF32.
4. Ce symbole apparaît dans un fichier texte encodé selon la norme UTF-8.  
Quel(s) symbole(s) verrait-on apparaître si on lit ce fichier texte selon la norme d'encodage ISO-8859-1 ?

#### 63 Deuxième illustration de l'incompatibilité unicode-ascci

1. Quel est l'encodage de é selon la norme d'encodage ISO-8859-1 ?
2. Quel est le point de code "é" selon l'unicode. Déterminer son encodage selon la norme d'encodage utf8.
3. Pourquoi est-il judicieux de ne pas respecter le codage binaire d'un caractère comme étant la valeur en binaire de son point de code ?
4. Expliquer pourquoi l'UTF8 prolonge parfaitement la table ASCII mais ne peut prolonger la table ISO/CEI 8859-1.

#### 64 Encodage UTF-8

1. Encoder les caractères suivants selon la norme UTF-8 :
  - (a) A (point de code 65)
  - (b) è (point de code 232)
  - (c) ★ (point de code 8902)
2. Déterminer le point de code des caractères dont l'encodage selon la norme UTF-8 est la suivante :
  - (a) 10110100
  - (b) 11001101 10110110
  - (c) 11100101 10000110 10100101

### 65 Une fonction Python

Écrire une fonction en Python `utf8(n)` qui prend en argument un entier  $n$  et qui renvoie l'encodage selon la norme UTF-8 du caractère dont le point de code est  $n$ .

On pourra librement utiliser la fonction `bin` dont la documentation indique :

```
1 bin ( number ) -> string
2     Return the binary representation of an integer .
```

### 66 Petite question

Si c'est normé, pourquoi a-t-on toujours plus de 20 ans après la mise en place de l'unicode des problèmes d'encodage ?

### Petit ? projet de cryptographie

Fichiers attachés :

- |                |                |
|----------------|----------------|
| — fichier1.txt | — fichier5.txt |
| — fichier2.txt | — fichier6.txt |
| — fichier3.txt | — fichier7.txt |
| — fichier4.txt | — fichier8.txt |

### 67 Mini-projet – autour des caractères

Par groupe de 2, vous devez réaliser le mini-projet suivant. Pour vous aider, vous avez à votre disposition les fichiers textes suivants :

- `fichier1.txt`
- `fichier2.txt`
- `fichier3.txt`

Faire un fichier `.py` par exercice et garder tracer des fichiers texte qui ont été créés.

Chaque fonction Python créée devra être documentée (y compris pré et post-conditions), la documentation devra être accessible à l'aide de l'instruction `help(NomFonction)`.

#### 1. Avant de commencer :

Pour lire un fichier texte caractère par caractère, on utilise les instructions suivantes :

```
1 with open ( " fichier1 . txt " , " r " ,
2             encoding = " utf8 " ) as* fichier :
3     car = fichier . read (1) # lecture du premier
4                         # caractère
5     while car != " " :
6         # instructions à réaliser
7         car = fichier . read (1) # lecture du
8                         # caractère suivant
```

Compter le nombre de "a" dans le fichier texte `fichier1.txt`.

#### 2. Formater un texte :

Écrire une fonction `formater(NomFichier)` qui prend en argument une chaîne de caractères correspondant au nom d'un fichier texte, et qui crée un autre fichier texte, dont le nom est le même auquel on a rajouté « \_formate » (par exemple, le fichier `texte1.txt` donnera `texte1_formate.txt`), qui formate le texte contenu dans le fichier texte de la façon suivante :

- mise en majuscule de toutes les lettres minuscules ;
- remplacement des lettres accentuées par des lettres non accentuées ;

- remplacement des ç, œ, æ par C, OE et AE.

**Remarque :** Pour écrire dans un fichier, il faut l'ouvrir en mode "w" au lieu du mode "r" et utiliser la fonction write.

### 3. Analyse fréquentielle :

Écrire une fonction analyse(NomFichier) qui prend en argument une chaîne de caractères correspondant au nom d'un fichier texte formaté comme dans l'exercice 1, et qui renvoie un dictionnaire dont les clés sont les lettres majuscules et les valeurs sont les fréquences d'apparition de ces lettres dans le fichier texte.

### 4. Le chiffre César :

Le chiffre César est une méthode de chiffrement d'un texte par décalage de chaque lettre d'un même nombre que l'on appelle la clé du chiffre César.

Écrire une fonction chiffrer(NomFichier, cle) qui prend en argument une chaîne de caractères correspondant au nom d'un fichier texte et un entier cle, et qui crée un autre fichier texte, dont le nom est le même auquel on a rajouté « \_chiffre ». Ainsi, le fichier

texte1\_formate.txt

donnera

texte1\_formate\_chiffre.txt

qui chiffre le texte avec la clé donnée.

### 5. Déchiffrer :

(a) Écrire une fonction CleCesar(NomFichier) qui prend en argument une chaîne de caractères correspondant au nom d'un fichier texte qui a été chiffré avec le chiffre César, et qui détermine la clé qui a été utilisée à l'aide d'une analyse fréquentielle.

(b) Déchiffrer les fichiers textes suivants :

- fichier4.txt
- fichier5.txt
- fichier6.txt

- fichier7.txt

### 6. Pour aller plus loin : Décrypter :

Le texte du fichier fichier8.txt a été chiffré par substitution monoalphabétique, c'est-à-dire que chaque lettre est remplacée par une autre mais on ne connaît pas la correspondance utilisée.

Décrypter le texte contenu dans fichier8.txt.

# Logique et calcul booléen

## Booléen

### 68 Propositions

Que vous inspirent les propositions suivantes :

- A :  $2^{10} = 1024$
- B :  $5 < 4$
- C : 3 est un nombre impair
- D : Il pleut dehors
- E : Je rase tous ceux qui ne se rasent pas
- F : Si je suis français alors je suis européen.
- G : Si je suis européen alors je suis français.
- H : La phrase que je viens de dire est fausse.

### 69 Table de vérité

1. Compléter la propriété suivante :

**Propriété 7 (Table de vérité).** Une proposition a un sens dans la théorie où l'on se place.

A toute proposition, on peut associer une valeur de vérité (ou booléen) qui ne prend que deux états :

- Soit .....
- Soit .....

2. Quelle convention les informaticiens ont-ils choisi pour représenter les deux états.

### 70 Logique non ?

Une réunion de cosmonautes du monde entier a lieu à Paris. Les cosmonautes américains portent tous une chemise rouge.

1. Ecrire l'affirmation précédente sous la forme SI ... ALORS ...
2. Ecrire l'affirmation précédente sous la forme d'une implication.
3. A l'aéroport, quelqu'un porte une chemise blanche. Est-il cosmonaute américain ?

4. A côté de la personne précédente, on voit quelqu'un qui porte une chemise rouge.  
Est-il cosmonaute américain ?
5. Le haut-parleur annonce l'arrivée d'un cosmonaute russe. Porte-t-il une chemise rouge ?
6. Dans le hall, on voit un cosmonaute américain qui porte un manteau. Porte-t-il une chemise rouge ?

## Opérateurs booléens

### 71 Une petite belote ?

Dans un jeu de 32 cartes, on en tire une au hasard. On considère les propositions suivantes :

- A : La carte tirée est un as.
  - B : La carte tirée est un coeur.
  - C : La carte tirée est de couleur rouge.
1. À quelle condition la proposition (A et B) est vérifiée ?
  2. À quelle condition la proposition (A ou B) est vérifiée ?
  3. À quelle condition la proposition (non C) est vérifiée ?
  4. À quelle condition la proposition (B et C) est vérifiée ?
  5. À quelle condition la proposition (B ou C) est vérifiée ?

### 72 Premiers calculs booléens

Déterminer quelle est la valeur des booléens suivants :

1.  $4 > 3$
2.  $5 \leq 8$
3.  $6 \leq 6$
4.  $3 > 3$
5.  $(3 \geq 2) \text{ and } (7 < 50/3)$
6.  $(a < 3) \text{ and } (a**2 > 9)$  avec  $a = 2.8$  puis  $a = -3.1$
7.  $((4 \geq 2) \text{ and } (-4 \geq -2)) \text{ or } (-1 < 0)$
8.  $0.1 + 0.2 == 0.3$

**73** *Devinette*

1. Quelle est le contenu de la variable *a* après exécution des instructions suivantes ?

```
a = 2
a == 3
a = a == 3
```

**Expressions booléennes****74** *Année bissextile*

On note : *a* le booléen « l'année est divisible par 4 », *b* le booléen « l'année est divisible par 100 » et *c* le booléen « l'année est divisible par 400 ».

1. Donner la définition d'une année bissextile
2. Écrire une expression booléenne à partir des booléens *a*, *b* et *c* et des opérateurs and, or et not permettant de déterminer si une année est bissextile ou non.

**Table de vérité d'opérateurs booléens****75** *Fonction non*

Soit *A* une proposition.

1. À quelle condition la proposition (*non A*) est vérifiée ?
2. Dresser la table de vérité de l'expression booléenne (*non A*) :

<i>A</i>	<i>non(A)</i>
0	1
1	0

**76** *Fonctions et*

Soit *A* et *B* deux propositions.

Dresser la table de vérité de l'opérateur booléen *et*

<i>A</i>	<i>B</i>	<i>(A et B)</i>
0	0	0
0	1	0
1	0	0
1	1	1

**77** *Fonctions ou*

Dresser la table de vérité de l'opérateur booléen *ou*.

**Table de vérité d'une expression booléenne****78** *Tables de vérité*

Soit *a* et *b* deux propositions. Dresser les tables de vérité des expressions booléennes suivantes :

1. (*a and not(b)*)
2. (*not(a) and b*) or (*a and not(b)*)
3. *not(a or not(b))*

**79** *Avec trois variables booléennes*

Soit *a*, *b* et *c* trois booléens. Dresser les tables de vérité des expressions booléennes suivantes :

1. (*a and not(b)*) or *c*
2. (*not(a) and b*) or (*c and not(b)*)
3. *not(a or not(b)) and (b or not(c))*

**80** *Priorité ?*

Soit *a*, *b* et *c* deux propositions. Dresser la tables de vérité de l'expressions booléennes suivantes :

a ou b et c

**expressions D'une fonction booléenne**

**81 Constante 0**

Soit  $a$  un booléen.

1. Soit  $C$  la fonction booléenne constante égale à 0. Quelle est sa table de vérité :

$a$	$C(a)$
0	
1	

2. Écrire la fonction  $C$  comme une expression booléenne de  $a$  en utilisant juste les trois opérateurs booléens *non*, *et* et *ou*.

**82 Constante 1**

Soit  $a$  un booléen.

1. Soit  $C$  la fonction booléenne constante égale à 1. Quelle est sa table de vérité :

$a$	$C(a)$
0	
1	

2. Écrire la fonction  $C$  comme une expression booléenne de  $a$  en utilisant juste les trois opérateurs booléens *non*, *et* et *ou*.

**83 identité**

Soit  $a$  un booléen.

1. Soit  $I$  la fonction Identité. Quelle est sa table de vérité :

$a$	$I(a)$
0	
1	

2. Écrire la fonction  $I$  comme une expression booléenne de  $a$  en utilisant juste les trois opérateurs booléens *non*, *et* et *ou*.

**84 Simplification**

1. Montrer que  $A \text{ ou } B = \overline{\overline{A} \text{ et } \overline{B}}$

2. Que peut-on en conclure ?

**xor et multiplexeur**

**85** opérateur booléen xor (Yaourt ou fromage *xor* ("ou exclusif") est un opérateur booléen prenant en paramètre deux booléens  $a$  et  $b$ .

1. Dresser la table de vérité  $a \text{ xor } b$
2. Écrire la fonction *xor* comme une expression booléenne de  $a$  et  $b$  utilisant uniquement les trois opérateurs booléens *non*, *et* et *ou*.

**86 Multiplexeur**

Soit  $A, B$  et  $C$  trois propositions.

1. Déterminer la sémantique de la fonction  $\text{mux}(A, B, C)$  dont la table de vérité est ci dessous :

$A$	$B$	$C$	$\text{mux}(A, B, C)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

2. Écrire la fonction *mux* comme une expression booléenne de  $a$ ,  $b$  et  $c$  utilisant uniquement les trois opérateurs booléens *non*, *et* et *ou*.
3. Trouver une façon plus simple d'écrire *mux*

**Plus difficile****87 ?**

On considère la fonction booléenne  $F$  suivante :

$a$	$b$	$F(a,b)$
0	0	1
0	1	0
1	0	0
1	1	1

1. Trouver la sémantique lié à  $F$
2. Écrire la fonction  $F$  comme une expression booléenne de  $a$  et  $b$  utilisant uniquement les trois opérateurs booléens *non*, *et* et *ou*.

**88 ??**

On considère la fonction booléenne  $F$  suivante :

$a$	$b$	$F(a,b)$
0	0	1
0	1	1
1	0	0
1	1	1

1. Trouver la sémantique lié à  $F$
2. Écrire la fonction  $F$  comme une expression booléenne de  $a$  et  $b$  utilisant uniquement les trois opérateurs booléens *non*, *et* et *ou*.

**Application : l'addition binaire**

**89 Algorithme d'addition** Expliquer comment additionner le nombre 728 à 456

**90 Addition en base deux**

1. A quels nombres entiers (en base 10) correspondent les nombres 101 et 111 (en base 2).
2. Faire l'addition des deux nombres 101 et 111.

3. Pourquoi faut il considérer une somme de trois nombres ?
4. Écrire les tables d'additions en base deux.
5. On assimile par la suite chiffre et booléen.  
Donner les tables de vérité du chiffre des unités et de celui des deuzaines.
6. Écrire les fonction booléennes *unite* et *deuzaine* comme une expression booléenne de  $a$ ,  $b$  et  $r$  utilisant uniquement les trois opérateurs booléens *non*, *et* et *ou*.
7. À quel "bug" inhérent à l'addition binaire sur une machine faut-il faire attention ? Déterminer la catastrophe la plus célèbre l'illustrant.

**caractère séquentiel ou évaluation paresseuse****91 Anticipation des résultats**

1. Déterminer ce qui se passerait si l'on rentre les programme suivant sous Python :

(a) Programme 1

```
1 a = 0
2 0 < 1/a and a != 0
```

(c) Programme 3

```
1 a = 0
2 0 < 1/a or a == 0
```

(b) Programme 2

```
1 a = 0
2 a != 0 and 0 < 1/a
```

(d) Programme 4

```
1 a = 0
2 a == 0 or 0 < 1/a
```

2. Que penser du script python suivant :

```
1 n = len(tab)
2 i = 0
3 while i < n and tab[i] != x :
4     i = i + 1
5     ...
6     ...
```

## Programmation python

### 92 Avec Python

Dans cet exercice, on représente un booléen par les entiers 0 et 1.

1. Programmer la fonction NOT(x) en python, où x est un entier égal à 0 ou 1 représentant un booléen, et qui renvoie la valeur de not(x), en utilisant uniquement des opérations sur les entiers.  
Elle devra passer le jeu de test suivant :

```
1 assert NOT (1) == 0
2 assert NOT (0) == 1
```

2. Programmer la fonction AND(x, y) en python, où x et y sont des entiers représentant des booléens, et qui renvoie la valeur de x and y, en utilisant uniquement des opérations sur les entiers.

Elle devra passer le jeu de test suivant :

```
1 assert AND (0 , 1) == 0
```

3. Programmer la fonction OR(x, y) en python, où x et y sont des entiers représentant des booléens, et qui renvoie la valeur de x or y, en utilisant uniquement des opérations sur les entiers.

Elle devra passer le jeu de test suivant :

```
1 assert OR (0 , 1) == 1
```

4. Programmer la fonction XOR(x, y) en python, où x et y sont des entiers représentant des booléens, et qui renvoie la valeur de x xor y.

5. Programmer la fonction MULT(x, y, z) en python, où x, y et z sont des entiers représentant des booléens, et qui renvoie la valeur de mult(x,y,z).

## Pour se faire plaisir

### 93 Un roi cruel

Un roi cruel avait pour habitude de proposer le marché suivant à ses condamnés :

Tout prisonnier doit choisir entre deux cellules. Chaque cellule peut contenir une corde ou une clef (un seul objet par cellule).

Si le prisonnier choisit une cellule avec une corde, il sera pendu. S'il choisit une cellule avec clef, il sera libéré.

Pour rendre plus cruelle la situation, les deux cellules contiennent parfois toutes deux une corde, mais aussi parfois toutes deux une clef et parfois l'une contient une clef et l'autre une corde.

Le roi donne par ailleurs des indications. Voici les indications données à l'un des prisonniers :

- sur la porte de la cellule 1 est inscrit : « Au moins l'une des deux cellules contient une clef ».
- sur la porte de la cellule 2 est inscrit : « L'autre cellule contient une corde ».

Le roi annonce par ailleurs au prisonnier que les affirmations sur les portes sont vraies toutes deux ou fausses toutes deux.

1. On note  $x$  : « la cellule 1 contient une clef » et  $y$  : « la cellule 2 contient une corde ».

Traduire les deux affirmations inscrites sur les deux portes à l'aide de  $x$  et  $y$  et des opérateurs booléens and, or et not.

2. Écrire un programme Python qui affiche la table de vérité des deux affirmations inscrites sur les portes en fonction des valeurs de  $x$  et de  $y$ .

3. Conclure sur le choix que doit faire le prisonnier.

## Les *p*-uplets

## 94 *p-uplet : accès à un élément*

On considère le  $p$ -uplet  $T = ("Barnabé", "Belfort", 17)$ .

1. Que retourne l'instruction `T[1]` ?
  2. Quelle instruction permet d'obtenir la valeur 17 ?
  3. Quel est l'état de la variable `T` après exécution de l'instruction suivante ?

```
1 T[1] = "Yesoul"
```

**95** problèmes de date

On représente une date par le 3-uplet  $(j, m, a)$ , où  $j$  est un entier correspondant au numéro du jour dans le mois,  $m$  est un entier correspondant au numéro du mois dans l'année et  $a$  est un entier correspondant à l'année.

- Écrire et programmer la fonction répondant aux spécifications suivantes :

```
1 def EstBissextile(date) :
2     """ EstBissextile(tuple) -> bool
3         renvoie True si l'année du jour représentée
4         par le 3-uplet date est bissextile, False
5         sinon """
6
```

2. L'algorithme de Mike Keith permet la détermination du jour de la semaine connaissant une date.

  - (a) Faire une recherche sur cet algorithme.
  - (b) Écrire et programmer la fonction qui répond aux spécifications suivantes :

```
1 def jourSemaine(date) :
2     """ jourSemaine(tuple) -> int
3     renvoie un entier correspondant au jour
4     dans la semaine (0 = dimanche,
5     1 = lundi...), du jour représenté par le
6     3-uplet date, en utilisant l algorithme
7     de Mike Keith"""
8
```

- (c) Modifier cette fonction pour qu'elle renvoie le nom du jour de la semaine au lieu d'un entier.
  - (d) Quel jour de la semaine êtes-vous nés ?
  - (e) Combien de vendredi 13 comporte l'année 2021 ?

96 en géométrie

On se place dans un repère orthonormé  $(O, \vec{i}, \vec{j})$ . On représente les coordonnées d'un point dans ce repère par un 2-uplet.

1. Écrire et programmer la fonction `milieu` qui calcule les coordonnées du milieu d'un segment  $[AB]$  :

```
1 def milieu(A, B) :
2     """ milieu(tuple, tuple) -> tuple
3     renvoie les coordonnées du milieu du segment
4     [AB], où A et B sont des 2-uplets (on
5     identifie le point avec ses coordonnées)"""
6
```

2. Écrire et programmer la fonction droite qui détermine une équation cartésienne de la droite ( $AB$ ) :

```
1 def droite(A, B) :
2     """ droite(tuple, tuple) -> tuple
3     renvoie un 3-uplet (a, b, c) correspondant aux
4     coef d une équation cartésienne  $ax + by + c = 0$ 
5     de la droite (AB)"""
6
```

**97 conversion binaire**

On représente l'écriture binaire d'un entier naturel  $n$  par un  $p$ -uplet, composé uniquement de 0 et de 1.

Écrire et programmer la fonction `binToDec(T)` permettant de déterminer la valeur décimale d'un entier dont on connaît l'écriture binaire :

```

1 def binToDec(T) :
2     """ binToDec(tuple) -> int
3         renvoie l entier écrit en base 10 dont l écriture
4         binaire est donnée par le p-uplet T"""
5
6 >>> binToDec((1, 0, 0, 1, 1))
7 19

```

**98 Génération de p-uplets**

1. Générer un p-uplet comprenant 100 zéros.
2. Générer un p-uplet comprenant 100 nombres entiers aléatoires entre 0 et 1 000 000.

**99 Interface versus implémentation**

1. Décrire l'interface d'un p-uplet
2. Critiquer l'implémentation python utilisée dans ce TP

# Les $p$ -uplets nommés

## 100 $p$ -uplet nommé : accès à un élément

On considère le  $p$ -uplet nommé

```
T = ("nom" : "Barnabe", "ville" : "Belfort", "age" : 17)
```

et s la chaîne de caractères définie par s = "age".

Parmi les instructions suivantes, quelles sont celles qui permettent d'obtenir la valeur 17 ?

```
T[2] T["2"] T[17] T["17"] T[age] T["age"] T[s] T["s"]
```

## 101 carte d'identité

Sur le recto d'une carte d'identité apparaissent les éléments suivants :

- Numéro (12 chiffres)
- Nom
- Prénom et parfois deuxième et troisième prénom
- Sexe (M ou F)
- Date de naissance
- Lieu de naissance
- Taille en mètres

1. Proposer une représentation de ces informations sous la forme d'un  $p$ -uplet nommé. Préciser le type des valeurs associées à chaque descripteur.

Écrire le  $p$ -uplet nommé correspondant à sa propre carte d'identité à l'aide de cette représentation.

2. Les quatre premiers chiffres du numéro de la carte d'identité correspondent à l'année et au mois de délivrance de la carte d'identité.

Écrire et programmer une fonction `estValable(carte)` qui prend en argument une carte d'identité représentée par un  $p$ -uplet nommé et qui renvoie `True` si la carte d'identité est valable à la date d'aujourd'hui, `False` si elle est périmée.

```
1 def estValable(carte) :
2     """ estValable(dict) -> bool
3     carte est un p-uplet nomme implemente par un
4     objet de type dict. Renvoie True si la carte
5     est valable, False si elle est perimee"""
6
```

**Rappel** : la carte d'identité d'une personne majeur au moment de la délivrance de la carte est valable 15 ans et celle d'une personne mineur au moment de la délivrance de la carte d'identité est valable 10 ans.

## 102 mini-projet : une bataille un peu spéciale

On considère un jeu de bataille un peu spéciale, se jouant à l'aide d'un jeu classique de 52 cartes :

- le jeu de cartes est distribué équitablement entre les deux joueurs ;
- les joueurs retournent la première carte de leur paquet. Celui qui a la carte la plus forte<sup>1</sup> remporte les deux cartes (mais ne les remet pas dans son paquet) ;
- en cas d'égalité, les deux cartes sont écartées ;
- la manche s'arrête lorsque les deux joueurs sont arrivés à la fin de leur paquet. Le gagnant est celui qui a empoché le plus de cartes.

Votre travail est de réaliser une simulation d'une partie de ce jeu de bataille.

Les cartes empochées par chaque joueur et les cartes écartées devront être consignées dans un fichier texte, qui constituera un rapport de la partie qui s'est déroulée.

### Aide :

On représentera une carte par un  $p$ -uplet nommé et on pourra utiliser un objet de type `list` pour stocker un ensemble de cartes et la méthode `append`. On pourra également utiliser les fonctions et méthodes suivantes : `open`, `print`, `shuffle` de la bibliothèque `random` (lire la

documentation de ces fonctions pour en comprendre le fonctionnement).

### 103 Interface versus implémentation

1. Décrire l'interface d'un p-uplet nommé
2. Critiquer l'implémentation python utilisée dans ce TP

$$\begin{aligned} \mathbb{E}(\varphi(X)) &= \int \varphi(x) d\mathbb{P}_X(x) \\ &= \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i) \\ &= \sum_{i \in I} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1 \\ &\quad \text{Car } \int_{-\infty}^{+\infty} e^{-x^2/2} dx = \int_{-\infty}^{+\infty} \frac{1}{\sigma} \mathcal{N}\left(\frac{\bar{X} - \mu}{\sigma}\right) dx = 1 \end{aligned}$$

- 
1. Pour déterminer la carte la plus forte, on utilise les critères suivants :
    - une carte noire est toujours plus forte qu'une carte rouge ;
    - si les deux cartes sont rouges, la plus forte est celle dont la valeur est la plus petite ;
    - si les deux cartes sont noires, la plus forte est celle dont la valeur est la plus grande.

# Les tableaux

## 104 accès à un élément

On considère le tableau tab défini de la façon suivante : `tab = [2,5,-8,12,7,3]`

- Que vaut `tab[1]` ? `tab[6]` ? Quelle instruction permet d'obtenir le premier élément du tableau ?
- Après la suite d'instructions suivantes, que vaut tab ?

```
1 for i in range(6) :
2     if tab[i]%2 == 0 :
3         tab[i] = -tab[i]
```

- Même question (en repartant du tableau initial), avec la suite d'instructions :

```
1 for i in range(6) :
2     if i%2 == 0 :
3         tab[i] = -tab[i]
```

## 105 somme des éléments

On veut écrire une fonction somme qui prend en argument un tableau d'entiers et qui renvoie la somme des éléments du tableau.

- Écrire le prototype de cette fonction.
- Programmer cette fonction.
- Créer un jeu de tests et tester cette fonction.

## 106 nombres premiers

- Créer un tableau de taille 101 dont toutes les valeurs sont initialisées à 1.
- Écrire une suite d'instructions de façon à ajouter 1 aux éléments du tableau dont l'indice est pair.

- Écrire une suite d'instructions de façon à ajouter 1 aux éléments du tableau dont l'indice est un multiple de 3.
- À l'aide d'une boucle pour, écrire une suite d'instructions qui ajoute 1 aux éléments du tableau dont l'indice est un multiple de k, pour k allant de 4 à 100.
- À quoi correspondent les valeurs contenues dans le tableau ?

## 107 renverser un tableau

On veut écrire une fonction `renverse` qui prend en argument un tableau d'entiers et qui renvoie un nouveau tableau, dont les éléments sont les mêmes mais rangés dans l'ordre inverse.

- Écrire le prototype de cette fonction.
- Programmer cette fonction.
- Créer un jeu de tests et tester cette fonction.

## 108 décalage

On considère la fonction `décalage` qui prend en argument un tableau d'entiers, et renvoie un nouveau tableau obtenu par décalage de tous les éléments d'une place vers la droite (le dernier élément étant remis en première place).

- Écrire le prototype de cette fonction.
- Programmer cette fonction.
- Créer un jeu de tests et tester cette fonction.
- Modifier cette fonction pour qu'elle réponde aux spécifications suivantes :

```
1 def decalage(tab, k) :
2     """ renvoie un nouveau tableau obtenu par
3     decalage de tous les elements de k place vers
4     la droite
5     tab est un tableau d entiers, k est un
6     entier"""
7
```

**109 la suite de Fibonacci**

1. Écrire une fonction qui prend en argument un entier  $n$  et qui retourne le tableau des  $(n + 1)$  premiers termes de la suite de Fibonacci.
2. Écrire une fonction qui prend en argument un entier  $n$  et qui retourne le tableau des quotients de 2 termes consécutifs de la suite de Fibonacci.
3. Représenter graphiquement les 100 premiers quotients. Pour cela :
  - (a) Créer un tableau X contenant les entiers de 1 à 50 et un tableau Y contenant les 50 premiers quotients de la suite de Fibonacci ;
  - (b) Créer le graphique à l'aide de l'instruction :

```

1 import matplotlib.pyplot as plt
2 plt.plot(X,Y)
3 plt.show()

```

- (c) Quelle conjecture peut-on faire ?

**110 écart**

1. Programmer une fonction plusGrandEcart qui prend en argument un tableau d'entiers et qui renvoie le plus grand écart entre deux nombres du tableau.

```

1 >>> plusGrandEcart([7 , 1, -1, 5, -2, 4])
2 9

```

Quelle est la complexité de cette fonction ?

2. Programmer une fonction plusPetitEcart qui prend en argument un tableau d'entiers et qui retourne le plus petit écart entre deux nombres du tableau.

```

1 >>> plusPetitEcart([7 , 1, -1, 5, -2, 4])
2 1

```

Quelle est la complexité de cette fonction ?

**111 conversion de la base b vers la base 10**

Écrire une fonction ConvertVersBase10(tab, b) qui prend en arguments un tableau d'entiers tab et un nombre entier b supérieur ou égal à 2, où tab représente les chiffres d'un entier écrit en base b et qui renvoie cet entier écrit en base 10.

```

1 >>> ConvertVersBase10([1 ,2 ,4] ,5)
2 39

```

**112 somme en base 2**

1. Lorsqu'on ajoute deux entiers positifs dont l'écriture en base 2 possède respectivement  $k_1$  et  $k_2$  bits, sur combien de bits s'écrira la somme de ces deux entiers ?
2. Écrire une fonction Somme(tab1, tab2) qui prend en arguments deux tableau d'entiers représentant les chiffres de deux entiers écrit en base 2 et qui retourne un tableau représentant la somme de ces deux entiers écrit en base 2.

```

1 >>> Somme([1 , 0, 1, 0, 0], [1, 0, 1, 1])
2 [0, 1, 1, 1, 1]
3
4 >>> Somme([1 , 0, 1, 1], [1, 1, 0, 1])
5 [1, 1, 0, 0, 0]

```

**113 longueur d'une ligne brisée**

On se place dans le plan muni d'un repère orthogonal.

On représente un point par un 2-uplets, correspondant à ses coordonnées.

- Programmer une fonction longueur(ligne) qui prend en argument un tableau de 2-uplets correspondant aux coordonnées des points formant une ligne brisée et qui calcule la longueur totale de cette ligne brisée.

```

1 >>> longueur([(1 , 2) , (2, -1) , (4, 1) , (7, 0)])
2 9.15298244508295

```

- Application :** Calculer une approximation de la longueur des deux courbes suivantes :

- parabole représentant la fonction carré sur l'intervalle  $[-1; 2]$
- demi-cercle de rayon 1.

#### 11.4 Interface versus implémentation

- Décrire l'interface d'un tableau
- Critiquer l'implémentation python utilisée dans ce TP

# Les dictionnaires

Fichiers attachés : communes.csv 

## 115 instructions de base

Tester dans la console les instructions suivantes :

```

1 >>> objet = {"nom" : "Barnabe", "age" : 23}
2 >>> objet["nom"]
3 >>> objet["telephone"]
4 >>> objet["telephone"] = "03 99 99 99 99"
5 >>> objet
6 >>> len(objet)
7 >>> type(objet)
8 >>> del objet["telephone"]
9 >>> objet

```

Décrire le rôle de chacune de ces instructions.

## 116 ajout dans un dictionnaire

Écrire et programmer une fonction

```
1 def ajout(dico, cle, val) :
```

qui prend en arguments un dictionnaire dico, une clé cle et une valeur val, et qui ajoute dans le dictionnaire dico la valeur val associée à la clé cle.

## 117 recherche d'une clé, d'une valeur

- Écrire une fonction RechercheValeur(dico, cle) qui prend en arguments un dictionnaire dico et une clé cle, et qui renvoie la valeur associée à la clé cle si cle est une clé du dictionnaire, et qui retourne None sinon.
- Écrire une fonction RechercheCle(dico, val) qui prend en arguments un dictionnaire dico et une valeur val, et qui renvoie la clé associée à la valeur val si val est une valeur du dictionnaire, et qui retourne None sinon.

## 118 suppression d'une clé, d'une valeur

- Écrire une fonction SuppressionCle(dico, cle) qui prend en arguments un dictionnaire dico et une clé cle, et qui supprime la clé cle si cle est une clé du dictionnaire.
- Écrire une fonction SuppressionValeur(dico, val) qui prend en arguments un dictionnaire dico et une valeur val, et qui supprime la clé associée à la valeur val si val est une valeur du dictionnaire.

## 119 carnet d'adresses

On modélise un carnet d'adresses par une liste dont les éléments sont des dictionnaires ayant des clés identiques.

Chaque dictionnaire correspond à une entrée du carnet d'adresses et les clés de chaque dictionnaire sont : "nom", "prenom", "adresse", "ville", "telephone".

**Exemple** : une entrée peut être

```
{"nom" : "Martin", "prenom" : "Pierre", "adresse" : "2 rue du muguet",
"ville" : "Belfort", "telephone" : "03 99 99 99 99"}.
```

- Écrire une fonction ajout(carnet, nom, prenom, adresse, ville, telephone) qui ajoute l'entrée correspondante au carnet d'adresses carnet.

**Exemple** : ajout(carnet, "Martin", "Pierre", "2 rue du muguet", "Belfort", "03 99 99 99 99")

- Écrire une fonction appartient ( carnet, nom) qui renvoie True si nom apparaît dans le carnet d'adresses, False sinon.

- Écrire une fonction RechercheNom(carnet, nom) qui renvoie la liste des dictionnaires dont la valeur associée à clé "nom" est nom.

**Exemple** : RechercheNom(carnet, "Martin")

- Écrire une fonction supprime(carnet, nom, prenom) qui supprime l'entrée du carnet d'adresses correspondante.

**Exemple** : supprime(carnet, "Martin", "Pierre")

### 120 créer un dictionnaire

On dispose de deux listes de même taille. La première est une liste de clés et la deuxième est une liste de valeurs.

- Écrire et programmer une fonction

```
CreerDictionnaire ( listeCles, listeValeurs)
```

qui prend en argument deux telles listes et qui construit et renvoie le dictionnaire dont les clés sont les éléments de listeCles et les valeurs sont les éléments de listeValeurs.

- Tester avec listeCles = ["nom", "age", "telephone"] et listeValeurs = ["Barnabe", 23, "03 99 99 99 99"].

### 121 charger un fichier csv avec Python

On souhaite stoker le fichier communes.csv dans une variable afin de pouvoir travailler dessus.

- Ouvrir un nouveau script Python avec Edupython et l'enregistrer dans le même dossier que le fichier communes.csv.
- Tester le script suivant :

```
1 with open("communes.csv", "r", encoding="utf-8")
2
3     ligne = fichier.readline()
```

Que contient la variable ligne ?

- Dans la console, taper ensuite ligne = ligne.strip(). Quel est le rôle de la fonction strip ?
- Dans la console, taper ensuite  
descripteurs = ligne.split(" ;").  
Quel est le rôle de la fonction split ?
- On va implémenter chaque objet de ce fichier csv dans un dictionnaire et on va regrouper l'ensemble de ces dictionnaires dans une liste.

On aura donc une variable de type liste dont les éléments sont des dictionnaires pour lesquels les clés seront les descripteurs.

Pour parcourir l'ensemble des lignes du fichier communes.csv avec Python, il suffit d'écrire le script suivant (à compléter) :

```
1 with open("communes.csv", "r", encoding="utf-8") as fichier :
2
3     ligne = fichier.readline()
4     descripteurs = # à compléter (1)
5     ligne = fichier.readline()
6     table = []
7
8     while ligne != "":
9         # instructions à faire sur la ligne en
10        # cours (2)
11         ligne.readline()
```

- Compléter l'instruction (1) de façon à ce que la variable descripteurs contienne la liste des descripteurs de cette table.
- Compléter les instructions (2) de façon à ce que la variable table contienne la liste des objets (sous la forme de dictionnaires dont les clés sont les descripteurs). On pourra utiliser la fonction CreerDictionnaire de l'exercice 2.

### 122 traiter des données

Maintenant qu'on a implémenté le fichier communes.csv dans une variable, on va pouvoir utiliser Python pour traiter les données.

Pour chacune des questions suivantes, écrire un script pour répondre aux questions :

- Quel est le nombre de communes en France métropolitaine ?
- Quel est le nombre de communes du territoire de Belfort ?
- Quelle est la commune de France située le plus au Nord ?
- Faire la moyenne des coordonnées géographiques de toutes les communes de France et placer le point obtenu sur  
<https://www.cordonnees-gps.fr/>  
pour visualiser où se situe le « centre » de la France.

5. Faire la même chose pour trouver le « centre » du territoire de Belfort.
6. Quel est le département français qui comporte le plus de communes (on pourra créer un dictionnaire dont la clé est le nom du département et la valeur est le nombre de communes de ce département).

### 123 Interface versus implémentation

1. Décrire l'interface d'un dictionnaire
2. Critiquer l'implémentation python utilisée dans ce TP

# Vocabulaire des données

**Matériel :** Des livres du CDI

## 124 Tweet

1. Choisir un livre au CDI.
2. Résumer en 1 tweet l'objet que vous avez entre les mains afin qu'un de vos camarades puisse l'identifier parmi tous les objets à partir de la lecture des 280 caractères.

## 125 Identifier un objet

Après avoir récupéré le tweet d'un de vos camarades, identifier l'objet décrit par ce tweet dans toute la collection d'objet.

## 126 Analyse des descriptions

1. Quelles données ont systématiquement été décrites dans les tweets et de quels types sont elles ?
2. Quelles données ont quelque fois été décrites dans les tweets et de quels types sont elles ?
3. Tous les nombres sont ils écrits pareils ?
4. Quelle(s) donnée(s) permet(tent) à coup sûr d'identifier le livre ?  
Pour une personne physique, ce genre de données est appelé "données personnelles"

## 127 Et vous ?

Choisir un objet, dont on peut faire la collection. En énoncer une liste de descripteurs.

## 128 Descripteurs humains

1. Énoncer des descripteurs pouvant s'appliquer à un humain.  
Préciser leur type.

2. Énoncer des descripteurs correspondant à une donnée personnelle pour un humain.

## 129 Mise en commun

Écrire son tweet dans le pad indiqué par votre enseignant.

## 130 Travail sur le document unique

A partir du pad élaboré collaborativement, Résumer la collection en un seul fichier avec le logiciel de votre choix. Ce fichier devra :

- être compréhensible par quelqu'un d'extérieur à la classe.
- être pratique à consulter
- permettre le rajout facile d'un nouvel objet à la collection
- supprimer facilement un objet de la collection
- être synthétique.

## 131 Données personnelles

Déterminer la quantité de données possédés par Facebook et/ou GOOGLE sur soi. En quelle unité se mesure cette quantité ?

## 132 Comprendre la structure du fichier .csv

Fichiers attachés :

1. Quel logiciel est utilisé par le système d'exploitation pour ouvrir le fichier budget\_parisv1.csv
2. Forcer l'ouverture de ce fichier par un bloc note (wordpad).
3. Décrire comment sont structurés les fichiers .csv
4. Déterminer le caractère séparateur des fichiers :
  - budget\_parisv2.csv
  - budget\_parisv3.csv
  - budget\_parisv4.csv
  - budget\_parisv5.csv

# Structure de tables

## 133 Déterminer des descripteurs

- projets du budget participatif de la ville de Paris

Fichiers attachés : 

Déterminer les descripteurs utilisés par la ville de Paris pour la budgétisation de ses projets participatifs.

- Choisissez un objet, dont vous faites par exemple la collection.  
Énoncez en des descripteurs possibles.

## 134 Yuka

### Matériel : étiquette de produits alimentaires

- Calculer le nutriscore de chaque étiquette.
- Déterminer comment fonctionne l'application smartphone Yuka
- Quel est la taille de la base de données "OpenFoodFact".
- Donner quelques descripteurs utilisés par "OpenFoodFact" pour décrire un aliment.

# Les fichiers CSV avec le tableur

Fichiers attachés : surface agricole bio.csv : 

## 135 préparation des données

Placer le fichier csv dans le dossier personnel.

Ouvrir le fichier à l'aide du logiciel Calc de libre Office (veiller à sélectionner le point-virgule comme séparateur et jeu de caractères Europe occidentale ISO-8859-1).

1. De combien de lignes et de colonnes est composé ce jeu de données ?
2. Quels sont les descripteurs de ce jeu de données ?
3. Décrire les objets de ce jeu de données.

## 136 Utiliser un tri

Trier le tableau par culture, puis par région et enfin par année.

Quelle est la donnée de la cellule F10 ?

## 137 utiliser un filtre

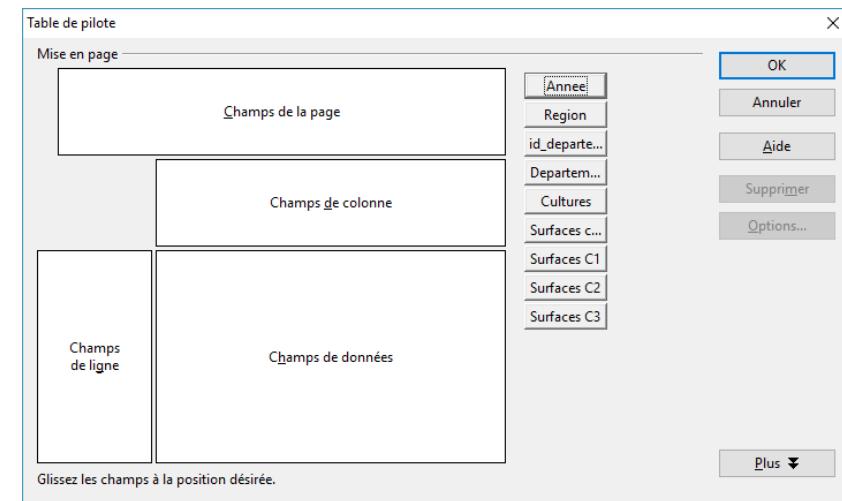
Utiliser un filtre pour ne garder que les cultures de fruits de la région Midi-Pyrénées.

Quelle est la surface dédiée au bio en Ariège en 2010 ?

## 138 utiliser un tri croisé

Effacer les filtres.

Cliquer sur Données / Table dynamique / Créer pour réaliser un tri croisé et obtenir la fenêtre suivante :



Glisser les étiquettes puis utiliser des filtres pour construire le tableau suivant (il s'agit des données filtrées pour l'année 2010) :

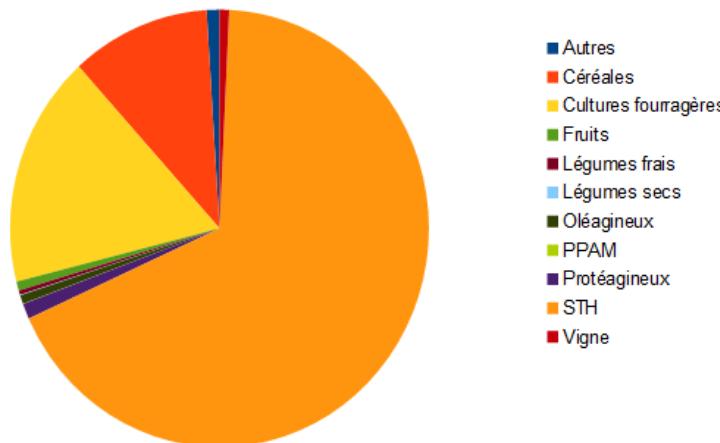
	A	B	C	D	E	F
1	Filtrer					
2						
3	Somme - Surfaces certifiées	Cultures				
4	Region	Céréales	Fruits	Oléagineux	Vigne	Total Résultat
5	ALSACE	1273,6	119,72	105,63	1030,52	2529,47
6	CHAMPAGNE-ARDENNE	1885,47	13,32	171,01	150,55	2220,35
7	FRANCHE-COMTE	2830,92	184,54	202,11	174,66	3392,23
8	LORRAINE	2922,53	203,27	69,98	1,85	3197,63
9	Total Résultat	8912,52	520,85	548,73	1357,58	11339,68

**139 construire des graphiques**

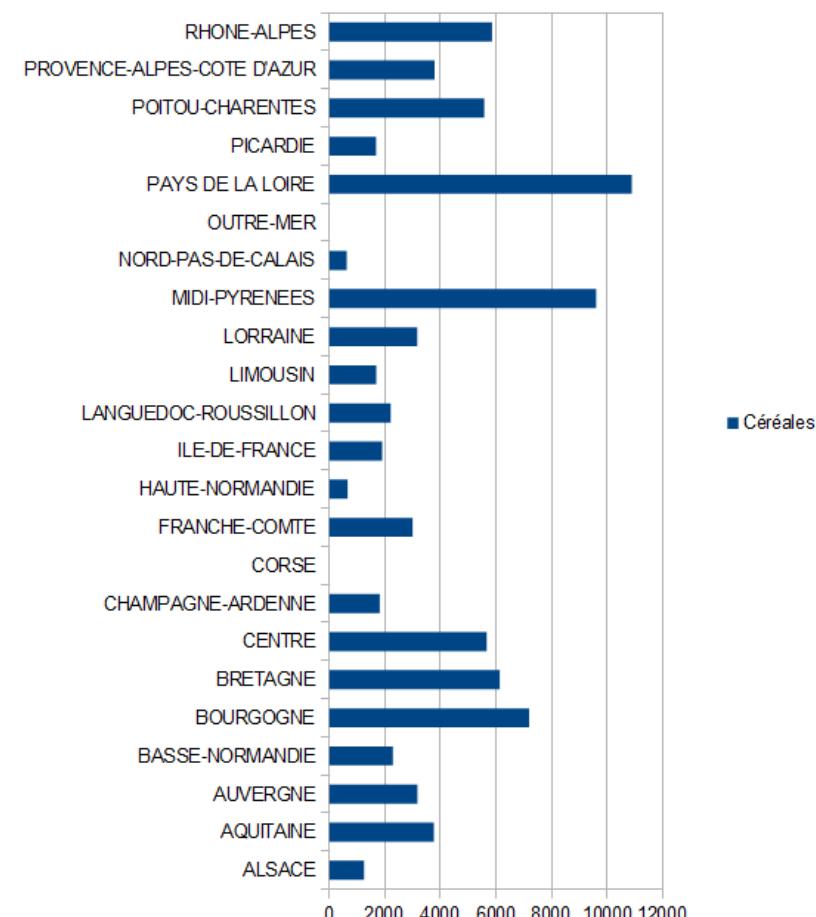
En modifiant les filtres du tri croisé, construire les deux graphiques suivants :

**Graphique 1 :**

Répartition des surfaces certifiées bio en Franche-Comté en 2011

**Graphique 2 :**

Surfaces certifiées bio en céréales, pour l'année 2009



# Les fichiers CSV avec Python

Il est possible de traiter les données d'un jeu de données au format csv avec Python et son module csv.

## Fichiers attachés :

communes.csv : 

cinema.csv : 

villes.csv : 

## 140 chargement du fichier csv dans un tableau

1. Placer le fichier communes.csv dans son dossier personnel.
2. Taper le script Python suivant et l'enregistrer dans le même dossier que le fichier csv.

```
1 import csv
2 with open("communes.csv","r",encoding = "utf-8") as fichCSV :
3     tableCSV = csv.DictReader(fichCSV,delimiter=";")
4     tab_communes = []
5     for objet in tableCSV :
6         tab_communes.append(objet)
7     descripteurs = tableCSV.fieldnames
```

3. Afficher le contenu de la variable descripteurs. Quels sont les descripteurs de ce jeu de données ?
4. Afficher le contenu de tab\_communes[0]. Que contient la variable tab\_communes ?
5. Expliquer chacune de ces lignes de codes.

## 141 traitement des données

1. Après le code précédent, taper les lignes suivantes :

```
1 nb = 0
2 for objet in tab_communes :
3     if objet["numéro_département"] == "25" :
4         nb = nb + 1
```

2. Expliquer ces lignes de codes.
3. Combien y a-t-il de communes dans le Doubs ?
4. Modifier ce script pour déterminer quel est le département de la région Bretagne qui possède le plus de communes.
5. Pour aller plus loin : Écrire un script qui recherche la commune française située le plus au Nord.
6. Pour aller plus loin : Écrire un script qui fait la moyenne des longitudes et des latitudes de toutes les communes. Afficher le point géographique obtenu sur

<https://www.cordonnees-gps.fr/>

pour visualiser où se situe le « centre » de la France.

## 142 croisement de données avec Python

On dispose du fichier cinema.csv dont les descripteurs sont cinema, nombre de salles et ville et du fichier villes.csv dont les descripteurs sont ville, latitude et longitude. Ces deux tables ont été rencontrées dans une autre activité.

1. En se basant sur l'exercice précédent, charger les fichiers cinema.csv et villes.csv dans deux tables, appelés respectivement tab\_cinema et tab\_villes.
2. Écrire un script permettant d'afficher le nom de chacun des cinémas ainsi que les coordonnées géographiques de la ville dans laquelle il se trouve.
3. Écrire un script permettant d'afficher le nom de chaque ville ainsi que le nombre de salles de cinéma dont elle dispose.

# Deux tables ?

**Fichiers attachés :**

communes.csv :

cinema.csv :

villes.csv :

## 143 Intérêt d'avoir deux tables plutôt qu'une seule

### Partie A : Avec une seule table

On considère la table suivante :

nom_eleve	prenom_eleve	classe	nom_lycee	ville	latitude	longitude
Bucher	Léo	2nde1	Condorcet	Belfort	47.633333	6.866667
Hissler	Julien	TG8	Pergaud	Besançon	47.25	6.033333
Muller	Léna	1G4	Haberges	Vesoul	47.633333	6.166667
Jeudy	Eméric	1G3	Pergaud	Besançon	47.25	6.033333
Guillot	Melody	1G7	Condorcet	Belfort	47.633333	6.866667
Dumont	Germain	1G7	Condorcet	Belfort	47.633333	6.866667
Mignot	Cédric	TG1	Haberges	Vesoul	47.633333	6.166667
Yuksel	Hanae	TG8	Pergaud	Besançon	47.25	6.033333
Ratoni	Elisa	2nde4	Pergaud	Besançon	47.25	6.033333

- Quelles sont les données redondantes dans ce tableau ?
- Les anomalies :
  - On veut ajouter un nouvel élève. Combien de valeurs doit-on renseigner ? Quelles anomalies peuvent apparaître ?
  - On se rend compte que les coordonnées géographiques de Belfort ne sont pas correctes. Quel problème se pose ?
  - Les élèves Muller Léna et Mignot Cédric ont quitté le lycée et doivent être supprimés de la base. Quelle information liée à ces deux élèves va-t-on perdre ?
- Proposer deux tables pour séparer les données de façon à éviter les redondances et les problèmes repérés dans la question précédente. élèves, un pour les établissements
- Quel est le pourcentage de place économisée en stockant les informations dans deux tables au lieu d'une seule ?

### Partie B : Avec deux tables

On considère les deux tables suivantes :

cinema	nombre de salles	ville
Pathé Belfort	14	Belfort
Mégarama Beaux-Arts	8	Besançon
Mégarama Ecole-Valentin	13	Besançon
Plazza Victor Hugo	3	Besançon
Kinépolis	14	Mulhouse
Le Palace	8	Mulhouse
Cinéma Bel Air	1	Mulhouse

ville	latitude	longitude
Belfort	47.633333	6.866667
Besançon	47.25	6.033333
Mulhouse	47.75	7.333333

Ces deux tables ont un descripteur en commun et vont nous permettre d'obtenir de nouvelles informations :

- Construire une nouvelle table dont les descripteurs sont le nom du cinéma et les coordonnées géographiques de la ville dans laquelle il se trouve.
- Construire une nouvelle table dont les descripteurs sont le nom de la ville et le nombre total de salles de cinéma dans la ville.

# Autour du monde

## Fichiers attachés :

langue.csv

pays.csv

villes.csv

### 144 Mise en place

Après avoir récupéré les trois fichiers csv, les enregistrer dans un dossier à côté d'un fichier monde.py.  
monde.py sera le fichier de travail de toute la séance.

## Données

La base est fournie sous la forme de trois fichiers csv : villes.csv, pays.csv, langues.csv.

**villes.csv** : Le fichier villes.csv contient 4080 lignes, une ligne décrivant une ville avec dans l'ordre quatre informations :

- Identifiant (numérique) : l'identifiant unique de la ville
- Nom (chaine de caractères) : le nom de la ville
- CodePays (chaine de caractères) : l'identifiant du pays de la ville
- Etat (chaine de caractères) : le nom de l'état, du district ou du département dans lequel se trouve la ville

**pays.csv** : Le fichier pays.csv contient 239 lignes, une ligne décrivant un pays avec dans l'ordre quinze informations :

- Code (chaine de caractères) : le code identifiant de manière unique le pays
- Nom (chaine de caractères) : le nom du pays
- Continent (énuméré) : le nom du continent où se situe le pays. Les continents prennent leur nom dans la liste : Asia, Europe, North America, Africa, Oceania, Antarctica, South America
- Region (chaine de caractères) : le nom de la région dans laquelle se situe le pays
- Aire (réel) : l'aire du pays

- AnneeIndependance (entier) : l'année d'indépendance du pays
- Population (entier) : la taille de la population du pays
- EsperanceVie (réel) : l'espérance de vie dans le pays
- GNP (réel) : estimation de la production de produits et de services
- GNPAncien (réel) : idem, valeur ancienne, peut-être de valeur NULL
- NomLocal (chaine de caractères) : nom local donné au pays
- FormeGouvernement (chaine de caractères) : forme de gouvernement du pays
- ChefEtat (chaine de caractères) : nom du chef de l'état (à l'époque de la constitution de la base)
- Capitale (entier) : identifiant de la capitale. Cet identifiant correspond à une ville dans le fichier villes.csv
- Code2 (chaine de caractères) : autre code pour le pays

**langues.csv** : le fichier langues.csv décrit les langues parlées dans chaque pays. Le fichier comprend 984 lignes, chaque ligne donnant une information sur une langue parlée dans un pays. Chaque ligne contient 4 informations avec dans l'ordre :

- CodePays (chaine de caractères) : le code du pays concerné, qu'on retrouve en champ 1 du fichier pays.csv
- Langue (chaine de caractères) : la langue parlée dans le pays
- Officiel (énuméré valant T ou F) : information précisant si la langue concernée est la langue officielle (T) du pays ou non (F)
- Pourcentage (réel) : pourcentage de la population parlant cette langue

A noter que pour ce dernier fichier, une ligne est identifiée de manière unique par le couple (CodePays, Langue) et que dans un pays, plusieurs langues peuvent être parlées.

## Pour bien commencer

### 145 Programme à comprendre

Déterminer ce que fait le programme suivant déjà rencontré précédemment :

```

1 import csv
2 with open("langues.csv", "r", encoding = "utf-8") as fichier :
3     table = csv.DictReader(fichier, delimiter = " ; ")
4     tab_langues = []
5     for objet in table :
6         tab_langues.append(objet)
7     descripteurs = tableCSV.fieldnames

```

**146 Entrée...**

Écrire une fonction permettant de récupérer l'équivalent de "descripteurs" et "tab\_langues" sans utiliser le module csv mais juste la fonction open de base.

**147 ... Sortie**

Écrire une fonction permettant de créer un fichier CSV à partir de "tab\_langues"

**C'est parti**

**148 requêtes**

Effectuer les requêtes suivantes en python. Les résultats de chaque requête devront se trouver dans un fichier CSV.

1. Donnez toutes les villes dont le nom commence par « pa » (sans tenir compte de la casse)
2. Donnez tous les pays d'Amérique du Sud.
3. Donnez toutes les villes d'Europe dont le nom commence par pa
4. Donnez toutes les villes de plus de 100 000 habitants d'Europe.
5. Donnez le nombre de formes de gouvernements présentes dans la base.
6. Donnez le nombre de pays répertoriés dans la base.
7. Donnez tous pays où l'on parle français.
8. Donnez tous les pays où le français est langue officielle.

9. Donnez toutes les villes d'Afrique de moins de 100 000 habitants, ayant le français pour langue officielle.
10. Quels sont les pays d'Amérique du Sud de plus de 10 000 000 d'habitants ayant un régime républicain ?
11. Quelles sont les villes de plus de 100000 habitants de pays nord-américains où l'on parle espagnol ?
12. Donnez la surface de l'Europe.
13. Donnez la surface de la Polynésie.
14. Combien y-a-t 'il de pays en Océanie de plus 10000 km<sup>2</sup> ?
15. Quelles sont les langues officielles des pays d'Europe de l'est ?
16. Quelle est la population moyenne dans les pays d'Asie ?
17. Quelle est la population moyenne dans les villes des pays d'Asie ?
18. Quelles sont les capitales d'Europe, ordonnées par ordre alphabétique ?
19. Donnez les villes des pays d'Afrique où la capitale a plus de 3 000 000 d'habitants ?
20. Quels sont les pays d'Amérique du Nord ayant accédé à l'indépendance avant 1912, où l'on parle portugais et pour lesquels sont répertoriées dans la base plus de 49 villes ?

**149 Défi**

Refaire toutes les requêtes de l'exercice précédent sans jamais utiliser d'affectations.

# Module turtle pour introduire python

## Niveau : \*

Python est un langage de programmation dont la syntaxe est particulièrement simple, il est très puissant et très utilisé (le créateur de ce langage travaillait chez Google et maintenant chez dropbox. La Nasa l'utilise aussi, pour ne citer qu'eux !).

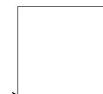
Pour vous familiariser avec le langage Python, essayez ces quelques commandes directement dans l'interpréteur python :

**150** Déterminer ce que fait chacune des instructions suivantes

```
>>> from turtle import *
>>> forward(120)
>>> left(90)
>>> color("red")
>>> forward(80)
>>> reset()
```

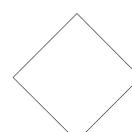
**151 Carré**

Tracer un carré de 100 pixels de côté.



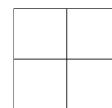
**153 Carré ?**

Tracer la figure ci-dessous.  
Le carré fait 100 pixels de côté.



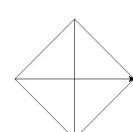
**152 Carrés**

Tracer la figure ci-dessous.



**154 Carré ? (2)**

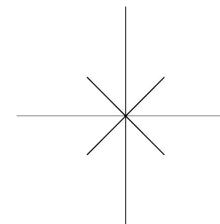
Tracer la figure ci-dessous.  
Le carré fait 100 pixels de côté.



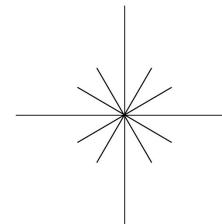
## Niveau : \*\*

**155 Cristal**

Tracer le cristal ci-dessous. Les grands segments mesurent 300 pixels, les petits 150.

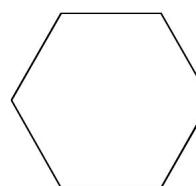


**156 Cristal (2)** Tracer le cristal ci-dessous. Les grands segments mesurent 300 pixels, les petits 150.



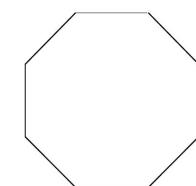
**157 Hexagone**

Tracer l'hexagone ci-dessous. Les 6 côtés mesurent 100 pixels.



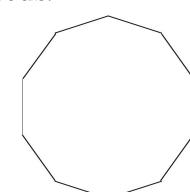
**158 Octogone**

Tracer l'octogone ci-dessous. Le côté mesure 100 pixels.



**159 (Double) décagone**

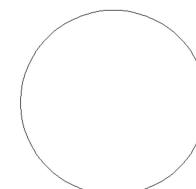
1. Tracer le décagone ci-dessous.



2. Tracer un double décagone

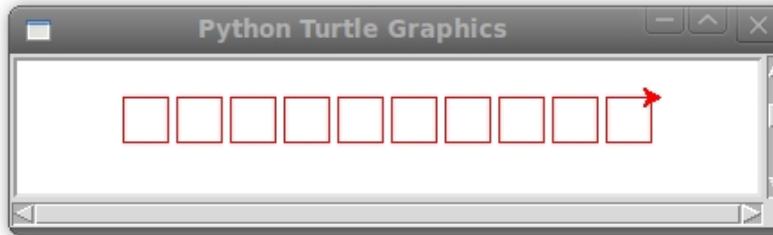
**160 Cercle**

Tracer le cercle ci-dessous.

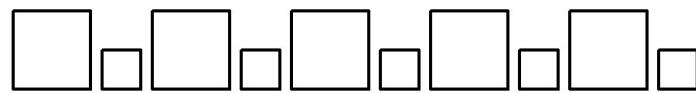


**Niveau : \*\*\***

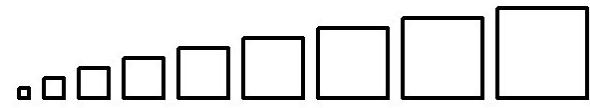
- 161** Réaliser le dessin ci-dessous à l'aide du module turtle :



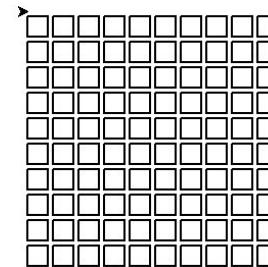
- 162** Réaliser le dessin ci-dessous à l'aide du module turtle :



- 163** Réaliser le dessin ci-dessous à l'aide du module turtle :



- 164** Réaliser le dessin ci-dessous à l'aide du module turtle :

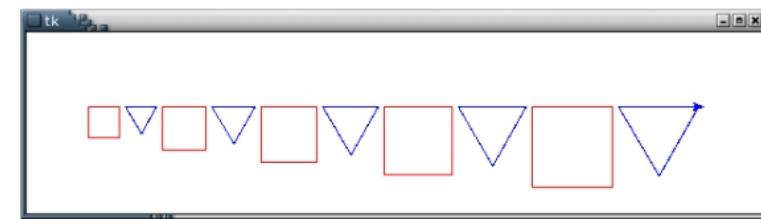


- 165** Modifier le code du programme précédent pour que le carré se trouvant en ligne 4 colonne 8 soit rouge

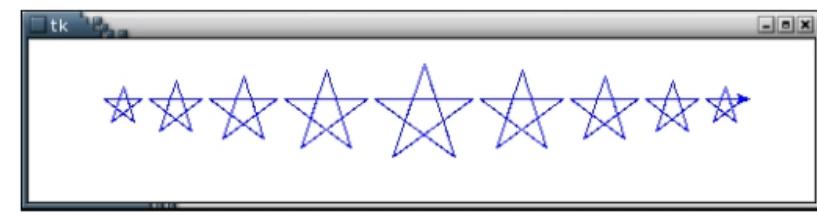
**Niveau : \*\*\*\***

- 166** Ecrire un programme qui dessine un carré de coté 100 "plein"

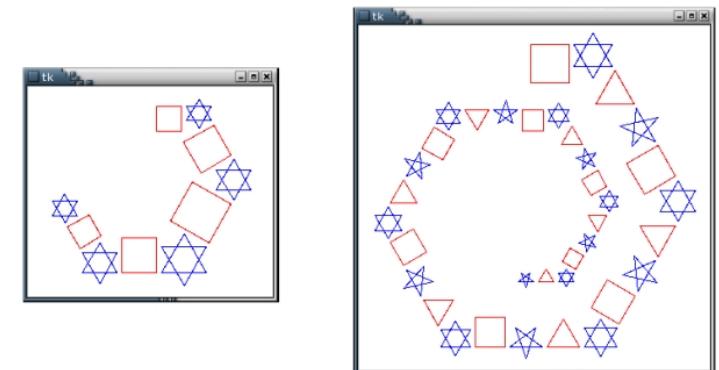
- 167** Réaliser le dessin suivant :



- 168** Dessiner la figure suivante :



- 169** Créer un script qui dessinera une série de ces étoiles :



# Parcours séquentiel d'un tableau

## Compréhensible non ?

### 170 Liste en compréhension

- Exécuter l'algorithme suivant :

```
1 from random import *
2 res = [randrange(0,100) for i in range(10)]
```

- Expliquer la sémantique de cette syntaxe.

## Niveau 1

### 171 Programme de calcul

On considère le programme de calcul suivant :

- Choisir un nombre
- multiplier par 3
- ajouter 6
- mettre au carré
- soustraire le nombre de départ

- Programmer cet algorithme
- Créer un tableau donnant le résultat de cet algorithme pour tous les nombres de 0 à 40.

### 172 Recherche d'occurrences

- Créer un tableau de 10000 nombres entiers aléatoires entre 0 et 100
- Déterminer le nombre d'occurrences du nombre 42 dans votre tableau.

### 173 Complexité

- Dans l'algorithme précédent, déterminer l'opération effectuée le plus souvent. Combien de fois est-elle exécutée ?

- Combien de fois serait-elle exécutée si le tableau avait une taille de 1 million ? 1 milliard ?
- Déterminer le nombre total d'opérations effectuées par l'algorithme en fonction de la taille du tableau.

### 174 Maximum

- Créer un tableau de 1000 nombres entiers entre 0 et 10 milliards.
- Déterminer le maximum du tableau ainsi créé.
- Quel est la complexité de cet algorithme ?

### 175 Moyenne

- Créer un tableau de 10000 nombres entiers aléatoires entre 0 et 100
- Déterminer la moyenne des nombres du tableau.
- Quel est la complexité de cet algorithme ?

## Niveau 2

### 176 appartenance

- Écrire une fonction qui renvoie True si un nombre appartient à un tableau de nombres et False sinon.

Le tableau et le nombre seront passés en paramètres de la fonction.

```
1 #prototype
2 def appartient(nombre,tableau) :
```

- (a) À quelle situation correspond le meilleur des cas qu'on puisse rencontrer ?

Pour un tableau de taille  $n$ , on appelle  $C_1(n)$  le nombre de comparaisons effectuées dans le meilleur des cas. Exprimer  $C_1(n)$ .

- (b) À quelle situation correspond le pire des cas qu'on puisse rencontrer ?

Pour un tableau de taille  $n$ , on appelle  $C_2(n)$  le nombre de comparaisons effectuées dans le pire des cas. Exprimer  $C_2(n)$ .

### 177 Fonction sur spécification : première occurrence

- Écrire une fonction qui renverra le nombre d'occurrences d'un nombre dans un tableau de nombres.

Le tableau et le nombre seront passés en paramètres de la fonction.

```
1 #prototype
2 def nombre_occurrence(nombre,tableau) :
```

- Écrire une fonction permettant de connaître l'indice de position de la première occurrence d'un nombre dans un tableau de nombres. Si le nombre n'est pas présent, la fonction renverra -1. Le tableau et le nombre seront passés en paramètres de la fonction.

```
1 #prototype
2 def occurrence_1(nombre,tableau) :
```

### 178 Fonction sur spécification : occurrences

Écrire une fonction permettant de connaître la ou les indices de position où se trouve(nt) un nombre dans un tableau de nombres. La fonction renverra une liste des indices. Si le nombre n'est pas présent, la liste renvoyée sera vide.

Le tableau et le nombre seront passés en paramètres de la fonction.

```
1 #prototype
2 def position_occurrence(nombre,tableau) :
```

### 179 Nombres supérieurs à un paramètre

On considère une fonction Compte qui prend en argument un tableau d'entiers tab et un entier x et qui renvoie le nombre d'éléments du tableau supérieurs ou égaux à x.

Programmer cette fonction.

### 180 Nombres supérieurs à l'indice

Écrire une fonction qui renverra le nombre d'éléments d'un tableau de nombres qui sont supérieurs à leurs indices dans un tableau de nombres.

Le tableau et le nombre seront passés en paramètres de la fonction.

```
1 #prototype
2 def superieur_indice(nombre,tableau) :
```

### 181 Extremum

- Écrire une fonction qui renverra le maximum d'un tableau de nombres :

```
1 #prototype
2 def maximum(tableau) :
```

- Écrire une fonction qui renverra le minimum d'un tableau de nombres :

```
1 #prototype
2 def minimum(tableau) :
```

- À quoi correspondent le meilleur et le pire des cas qu'on puisse rencontrer ?

- On appelle  $C(n)$  le nombre de comparaison effectuées. Exprimer  $C(n)$  en fonction de  $n$ .

- Modifier ces fonctions pour qu'elles renvoient un 2-uplet contenant l'extremum et sa ou ses positions dans le tableau.

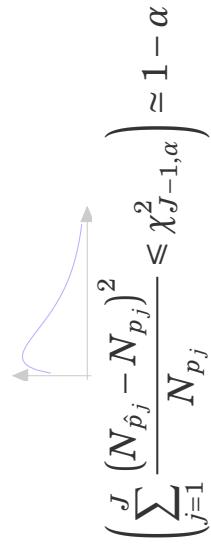
## 182 Position des extrêmes

1. Écrire une fonction permettant de connaître le ou les indices du maximum d'un tableau de nombres. La fonction renverra une liste des indices.

```
1 #prototype
2 def position_maximum(tableau) :
```

2. Écrire une fonction permettant de connaître le ou les indices du minimum d'un tableau de nombres. La fonction renverra une liste des indices. Expliquer le problème algorithmique posé par cet implémentation ?

```
1 #prototype
2 def position_minimum(tableau) :
```

**Fichiers attachés :**

recherches.py



graphique.py

**Principe de la dichotomie****183 Castor informatique**

1. Aller effectuer l'exercice "Attraper le monstre" du castor informatique 2014
2. Aller effectuer l'exercice "Tas de graines" du castor informatique 2017

**Le juste prix****184 Le juste prix**

Écrire une fonction qui prend un nombre entre 0 et 20 000. La fonction devra le retrouver par dichotomie et renvoyer le nombre d'essai.

**Recherche dichotomique dans un tableau**

**Position du problème :** on dispose d'un tableau d'éléments et on veut chercher si un élément donné est dans le tableau.

**185 Recherche séquentielle**

On a déjà rencontré l'algorithme suivant :

```

1 def Recherche(tab, element) :
2     n = len(tab)
3     for i in range(n) :
4         if tab[i] == element :
5             return True
6     return False

```

1. Dresser le tableau de l'état des variables au cours de l'exécution de Recherche(tab, element) avec tab = [1, 4, 2, 7] et element = 4.

i	tab[i]

2. Dresser le tableau de l'état des variables au cours de l'exécution de Recherche(tab, element) avec tab = [1, 4, 2, 7] et element = 3.
3. Soit  $n$  la longueur du tableau. Quel est le nombre de tests d'égalité dans le meilleur des cas ? Dans le pire des cas ? Quel est la complexité de cet algorithme ?

Lorsque le tableau contient des éléments que l'on peut ordonner (typiquement des nombres) et qu'il est trié, il est possible de faire mieux. Il s'agit de l'algorithme de recherche dichotomique.

**Définition 8 (Dichotomie).** Du grec ancien *διχοτομία*, dikhotomia (« division en deux parties »)

**Principe de recherche dichotomique** Le principe est simple : on regarde l'élément placé au milieu du tableau. Si c'est l'élément cherché, on retourne True, sinon, on relance avec la partie gauche ou la partie droite du tableau suivant que l'élément du milieu est plus grand ou plus petit que l'élément cherché. On s'arrête quand on a trouvé l'élément ou que le tableau dans lequel on cherche est réduit à rien, auquel cas on retourne False.

**186 manipulation dichotomique**

Faire fonctionner pas à pas, en précisant à chaque étape l'élément choisi comme pivot et la partie du tableau choisi, cet algorithme avec :

1. tab = [1, 2, 4, 5, 7, 8, 10, 12, 15] et element = 5.
2. tab = [1, 2, 4, 5, 7, 8, 10, 12, 15] et element = 11.

**187 implémentation**

1. Déterminer une stratégie permettant de déterminer la partie du tableau sur laquelle travailler.

2. Déterminer le critère d'arrêt permettant de savoir que l'élément recherché n'est pas dans le tableau.
3. Programmer cet algorithme en Python

### 188 état des variables

1. Dresser le tableau de l'état des variables au cours de l'exécution de Recherche(tab, element) avec tab = [1, 2, 4, 5, 7, 8, 10, 12, 15] et element = 5.

début	fin	milieu	tab[milieu]

2. Dresser le tableau de l'état des variables au cours de l'exécution de Recherche(tab, element) avec tab = [1, 2, 4, 5, 7, 8, 10, 12, 15] et element = 11.

**189** On suppose que le tableau a une taille  $n = 2^k$ . et on se place dans le pire des cas, c'est à dire lorsque l'élément cherché n'est pas dans le tableau.

1. Quelle est la valeur de  $fin - debut$  avant le premier passage dans la boucle ? Exprimer cette valeur en fonction de  $k$ .
2. Quelle est la valeur de  $fin - debut$  après le premier passage dans la boucle ? Exprimer cette valeur en fonction de  $milieu$ .
3. Lors du dernier passage dans la boucle, quelle est la valeur de  $fin - debut$  ?
4. Dans le pire des cas, combien de passages dans la boucle Tant que sont nécessaires ?
5. Déterminer un variant de boucle et en déduire la terminaison de l'algorithme de la recherche dichotomique.

### 190 pour aller plus loin

On cherche à comparer la complexité de ces deux algorithmes dans le pire des cas, c'est-à-dire lorsque l'élément cherché n'est pas dans le tableau.

Soit  $n$  la taille du tableau.

Dans cette partie, tab = [i for i in range(n)] et element = -1.

1. (a) pour  $n = 10000$ , quelle est le nombre de tests d'égalité pour l'algorithme de recherche naïf dans le pire des cas ?
- (b) pour  $n = 10000$ , quelle est le nombre de passages dans la boucle pour l'algorithme de recherche dichotomique dans le pire des cas ?
2. Le fichier recherches.py contient les deux fonctions de recherche vues précédemment. Modifier ces deux fonctions pour qu'ils retournent :
  - (a) pour l'algorithme de recherche naïve le nombre  $f(n)$  de tests d'égalité effectués ;
  - (b) pour l'algorithme de recherche dichotomique le nombre  $g(n)$  de passages dans la boucle Tant que.
3. Le code du fichier graphique.py permet de tracer le nuage de points  $(X, Y)$  où  $X$  et  $Y$  sont deux listes de même taille. On a alors  $X$  en abscisses et  $Y$  en ordonnées.  
Modifier ce bout de code pour représenter d'abord sur deux graphiques différents puis sur un même graphique les fonctions  $f$  et  $g$  pour  $n$  allant de 100 à 10 000 par pas de 100.

### Recherche dichotomique récursive dans un tableau

#### 191 recherche dichotomique

1. Rappeler l'algorithme de recherche dichotomique d'un élément dans un tableau trié, vu en première.
2. Pour implémenter l'algorithme de recherche dichotomique par une fonction récursive, on va utiliser la structure suivante :

```

1 def rechercheDicho(tab, element) :
2     """ rechercheDicho(list, int) -> bool """
3
4     def rechercheDichoRec(tab, element, debut, fin) :
5         ...
6
7     n = len(tab)
8     return rechercheDichoRec(tab, element, 0, n - 1)

```

$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\bar{X} - \mu \xrightarrow{\sigma} \mathcal{N}(0, 1)$$

$$\mathbb{P}\left(\sum_{j=1}^J \frac{(N_{p_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1, \alpha}^2\right) \approx 1 - \alpha$$

C'est la fonction `rechercheDichoRec` qui est une fonction récursive ; elle renvoie True si `element` apparaît dans la partie du tableau dont les indices sont compris entre `début` et `fin` et False sinon.

Le principe est le suivant :

- on regarde l'élément du tableau d'indice `milieu` = (`début` + `fin`) // 2.
- si c'est l'élément cherché, on renvoie True
- s'il est plus petit que l'élément cherché, on effectue un appel récursif sur la partie du tableau dont les indices sont compris entre `milieu` + 1 et `fin`
- s'il est plus petit que l'élément cherché, on effectue un appel récursif sur la partie du tableau dont les indices sont compris entre `début` et `milieu` - 1

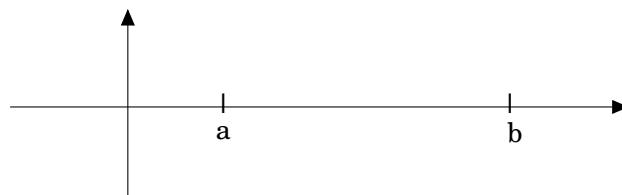
- Il manque un cas de base. Quel est-il ?
- Programmer la fonction `rechercheDichoRec`
- Proposer un jeu de tests et effectuer les tests.

## Recherche de racines

### 192 Et pour une fonction

On considère une fonction  $f$  continue définie sur  $[a, b]$ .

Il a préalablement été démontré que  $f$  ne s'annulait qu'une et une seule fois sur  $[a, b]$  mais où ?



Comment peut-on faire, en utilisant la dichotomie pour affiner un encadrement de la racine ? Quand s'arrête-t-on ?

### 193 A vous le clavier

- Programmer un algorithme permettant de connaître un encadrement à  $10^{-5}$  de la racine de  $f(x) = e^{1/x} - 2 \cos(x)$  sur l'intervalle  $[4; 6]$ .
- En proposer un encadrement à  $10^{-10}$ .
- Modifiez votre algorithme pour savoir combien de tours de boucle il a été amené à faire.

## Défi

### 194 Mélange des genres

Donner un encadrement à  $10^{-14}$  de toutes les racines de  $f(x) = e^{1/x} - 2 \cos(x)$  sur  $[1; 100]$ .

## Trichotomie

### 195 Le trésor et la fausse pièce

Vous êtes un ou une pirate qui vient de récupérer sa part du trésor constitué d'un beau tas de pièces d'or. Hélas, vous avez vent d'une supercherie : l'une des pièces n'est pas en or massif, mais en un alliage plus léger que l'or et est simplement plaquée or, pour qu'on ne puisse pas la distinguer visuellement des autres pièces. Le seul signe distinctif est qu'elle est plus légère que les autres. Vous n'avez pas à votre disposition un atelier de chimiste ni un laboratoire scientifique vous permettant d'étudier très précisément ces pièces, mais simplement une balance à plateaux sans aucun poids, comme celle donnée ci-dessous :

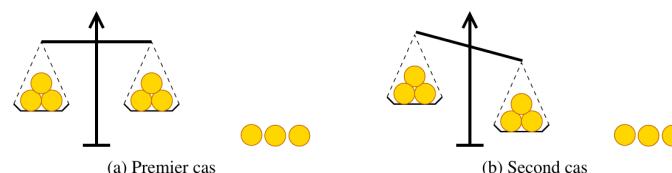


### 196 Trichotomie

1. Exemples pour un petit nombre de pièces dans le trésor.

- Expliquer comment procéder avec 2 pièces pour retrouver la fausse pièce.
- Expliquer comment procéder en une seule pesée quand il y a 3 pièces.
- Expliquer comment conclure en deux pesées quand il y a 4 pièces.

Examinons maintenant comment faire avec 9 pièces. On procède en divisant les 9 pièces en 3 tas, et selon la réponse de la balance on identifie le tas contenant la pièce plus légère, comme indiqué sur la figure ci-dessous.



Dans le cas illustré en figure (a), le tas contenant la pièce la plus légère est le tas qui n'est pas sur un plateau. On identifie, en une pesée supplémentaire, la pièce la plus légère parmi les 3 pièces de ce tas. Dans le cas illustré en figure (b), le tas contenant la pièce la plus légère est le tas qui est sur le plateau de gauche. On identifie, en une pesée supplémentaire, la pièce la plus légère parmi les 3 pièces de ce tas. On peut donc conclure en deux pesées. Examinons maintenant le cas général d'un trésor de  $n$  pièces.

2. Algorithme dans le cas général d'un trésor de  $n$  pièces ( $n$  est une donnée d'entrée de l'algorithme).

- Dans le cas où  $n$  est une puissance de 3 ( $n = 3^k$ ,  $k$  entier naturel) : écrire un algorithme itératif qui permet d'identifier, parmi  $n$  pièces, la pièce fausse qui est plus légère que les autres.
- Expliquer comment votre algorithme fonctionne avec 27 pièces. Aide : on devrait obtenir le résultat en 3 pesées.

(c) Dans le cas d'un nombre quelconque de pièces, adapter l'algorithme précédent pour identifier la fausse pièce.

(d) Afin de comparer avec la recherche dichotomique, remplir le tableau suivant, en indiquant, pour chaque méthode et pour différentes valeurs de  $n$ , le nombre maximal de pesées.

Quelle est, d'après vous, la méthode la plus efficace en nombre de pesées ?

$n$	Dichotomie	Trichotomie
2		
3		
6		
8		
9		
16		
27		

# Préparation au tri

**Position du problème :** On dispose d'un tableau d'éléments pouvant être ordonnés. Les algorithmes de tris servent à ordonner ce tableau par ordre croissant ou décroissant.

**Matériel :** jeu de tarot

## 197 A la main - Variantes 1

Vous disposez d'une variable "tableau" constitué de 1 000 000 de nombres entiers.

1. Chacun de son côté, proposer un algorithme pour trier ce tableau par ordre croissant. Écrire cet algorithme sur papier de façon à le rendre compréhensible par son binôme.
2. Pour aller plus loin : Si votre binôme a compris votre algorithme et pense qu'il renvoie le résultat attendu, programmez votre algorithme en Python : on créera une fonction `tri(tab)` qui prend en argument un tableau de nombres `tab` et qui retourne un tableau trié.

## 198 A la main - variantes 2

Par groupe de deux :

1. Chacun de son côté, proposer un algorithme pour trier par ordre croissant, à la main, les cartes qui auront été découpées au préalable. Écrire cet algorithme sur papier de façon à le rendre compréhensible par son binôme.
2. Échanger son algorithme avec son binôme et tester l'algorithme de son binôme.
3. Pour aller plus loin : Si votre binôme a réussi à trier ses cartes par ordre croissant en utilisant votre algorithme, programmez votre algorithme en Python : on créera une fonction `tri(tab)` qui prend en argument un tableau de nombres `tab` et qui retourne un tableau trié.

## 199 A la main - variantes 3

Par groupe de deux :

1. Chacun de son côté, proposer un algorithme pour trier par ordre croissant, à la main, les atouts du jeu de tarot. Écrire cet algorithme sur papier de façon à le rendre compréhensible par son binôme.
2. Échanger son algorithme avec son binôme et tester l'algorithme de son binôme.
3. Pour aller plus loin : Si votre binôme a réussi à trier ses cartes par ordre croissant en utilisant votre algorithme, programmez votre algorithme en Python : on créera une fonction `tri(tab)` qui prend en argument un tableau de nombres `tab` et qui retourne un tableau trié.

## 200 Tri décroissant

Comment peut on "simplement" trier un tableau de nombre par ordre décroissant ?

$$\mathbb{E}(\Phi(X)) = \int \Phi(x) q \mathbb{P}^X(x)$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}^i(A) \mathbb{P}(B^i)$$

$$\left( \sum_{i \in I} \mathbb{P}^i(J - h) \right)_{h \in \mathbb{R}}$$

$$\frac{\lambda \sum_{i=1}^n e^{-x_i \lambda} q x_i}{\lambda - \mu} \rightarrow \lambda(0, T)$$

$$\mathbb{P}\left(\sum_{i=1}^n W^{b_i} - \sum_{i=1}^n W^{b_i} \leq \lambda - \alpha\right) = 1 - \alpha$$

17	27	20
5	11	31
19	36	8
2	24	7

17	27	20
5	11	31
19	36	8
2	24	7

# Tri par sélection

## Principe du tri par sélection :

Trier par sélection consiste à déterminer le minimum du tableau, puis le minimum parmi les nombres restants, puis le minimum parmi les nombres restants , puis ... jusqu'à épuisement des nombres restants.

**201** *A la main* Dérouler à la main l'algorithme du tri par sélection sur le tableau [9, 8, 2, 7, 1, 6, 10, 2, 4]. On fera apparaître le tableau d'état des variables.

**202** *Programmation du tri par sélection*

Tout est dans le titre.

**203** *Méthode .append*

La méthode .append a-t-elle été utilisée ? Si oui programmer un deuxième algorithme ne l'utilisant pas.

**204** *En place*

Le tri est-il en place ? Si non, programmer un troisième (deuxième, premier ?) algorithme respectant cette complexité en mémoire.

**205** *Complexité*

1. On considère un tableau de taille  $n$  que l'on veut trier à l'aide du tri par sélection.
  - (a) Quel est le pire des cas qui puisse se présenter ? Dans ce cas, déterminer le nombre  $c(n)$  de comparaisons pour trier ce tableau en fonction de  $n$ .
  - (b) Quel est le meilleur cas qui puisse se présenter ? Dans ce cas, déterminer le nombre  $c(n)$  de comparaisons pour trier ce tableau en fonction de  $n$ .

**206** *Correction du tri par sélection*

- 1.
2. Justifier que la propriété « le tableau  $\text{tab}[0 : i+1]$  est trié » est un invariant de boucle pour l'algorithme du tri par sélection.
3. En conclure que l'algorithme du tri par insertion est correct.

**207** *Terminaison du tri par sélection*

1. Déterminer un variant de boucle pour prouver la terminaison du tri par insertion.
2. En déduire la preuve de l'algorithme du tri par sélection.

**208** *Récuratif*

Programmer le tri par sélection de manière récursive.

## Étayage

**209** *Étayage pour la programmation en place de l'algorithme*

Aller voir dans la correction de cet exercice une version papier de l'algorithme pour le programmer en python.

# Tri par insertion

## Principe du tri par insertion :

La plupart des personnes utilisent ce tri pour trier un jeu de cartes à jouer.

Trier par insertion consiste à aller prendre le premier élément puis l'insérer au bon endroit. Puis le premier élément des nombres restants et l'insérer au bon endroit, puis le premier élément des nombres restants et l'insérer au bon endroit...

### 210 A la main

Dérouler à la main l'algorithme du tri par insertion sur le tableau [9, 8, 2, 7, 1, 6, 10, 2, 4]. On fera apparaître le tableau d'état des variables.

### 211 Programmation du tri par insertion

Tout est dans le titre.

### 212 Méthode .append

La méthode .append a-t-elle été utilisée ? Si oui programmer un deuxième algorithme ne l'utilisant pas.

### 213 En place

Le tri est-il en place ? Si non, programmer un troisième (deuxième, premier ?) algorithme respectant cette complexité en mémoire.

**Aide :** Dans l'algorithme, on parcourt le tableau à trier du début à la fin. Au moment où on considère le  $i$ -ème élément, les éléments qui le précédent sont déjà triés. Pour faire l'analogie avec l'exemple du jeu de cartes, lorsqu'on est à la  $i$ -ème étape du parcours, le  $i$ -ème élément est la carte saisie, les éléments précédents sont la main triée et les éléments suivants correspondent aux cartes encore en désordre sur la table.

L'objectif d'une étape est d'insérer le  $i$ -ème élément à sa place parmi ceux qui précédent. Il faut pour cela trouver où l'élément doit être

inséré en le comparant aux autres, puis décaler les éléments afin de pouvoir effectuer l'insertion.

En pratique, ces deux actions sont fréquemment effectuées en une passe, qui consiste à faire « remonter » l'élément au fur et à mesure jusqu'à rencontrer un élément plus petit.

### 214 Complexité

1. On considère un tableau de taille  $n$  que l'on veut trier à l'aide du tri par sélection.
  - (a) Quel est le pire des cas qui puisse se présenter ? Dans ce cas, déterminer le nombre  $c(n)$  de comparaisons pour trier ce tableau en fonction de  $n$ .
  - (b) Quel est le meilleur cas qui puisse se présenter ? Dans ce cas, déterminer le nombre  $c(n)$  de comparaisons pour trier ce tableau en fonction de  $n$ .

### 215 Correction du tri par insertion

1. Justifier que la propriété « le tableau  $\text{tab}[0 : i+1]$  est trié » est un invariant de boucle pour l'algorithme du tri par insertion.
2. En conclure que l'algorithme du tri par insertion est correct.

### 216 Terminaison du tri par insertion

1. Montrer que l'algorithme du tri par insertion termine.
2. En déduire la preuve de l'algorithme du tri par insertion.

### 217 Récursif

Programmer le tri par sélection de manière récursive.

### Étayage

### 218 Étayage pour la programmation en place de l'algorithme

Aller voir dans la correction de cet exercice une version papier de l'algorithme pour le programmer en python

# Algorithme versus programme

## Bizarre ? Vous avez dit bizarre ?

### 219 Étonnant non ?

- Effectuer les deux calculs suivants :

$$3.11 + 2.08$$

$$0.1 + 0.2$$

- Effectuer ces calculs avec python

### 220 Étonnant non ? - le retour

On considère un triangle rectangle ABC de côté  $a = AB$ ,  $b = AC$  et  $c = BC$ .

- Énoncer une condition permettant de savoir si le triangle est rectangle en A.
- Programmer une fonction en python dont le prototype est :

```
1 def est_rectangle(a,b,c) :
2     """(nombre,nombre,nombre) -> boolean"""
```

Elle renverra Vrai si le triangle ABC est rectangle en A, Faux sinon.

- Tester cette fonction sur le triangle ABC avec :

$$AB = \sqrt{3}$$

$$AC = \sqrt{2}$$

$$BC = \sqrt{5}$$

- Quelle vigilance doit on avoir en programmation ? Énoncer un moyen de contourner cette difficulté ? Est ce pour autant une garantie ?

## Tracé de courbes

### 221 module matplotlib.pyplot

- Que fait le programme suivant :

```
1 from matplotlib.pyplot import *
2 x = 1
3 y = 2
4 plot(x,y,"o")
5 show()
```

### 222 Tracé de courbes

Tracer la courbe représentative de la fonction  $f(x) = 2x^2 - 2x + 3$  sur :

- $[-1\ 000; 1\ 000]$
- $[-10\ 000; 10\ 000]$
- $[-1\ 000\ 000; 1\ 000\ 000]$
- $[-4; 4]$

### 223 Tracé d'une autre courbe

Tracer la courbe représentative de la fonction  $g(x) = \frac{2x-1}{x+3}$  sur  $[-4; 4]$

## Défi : Faire planter la machine

### 224

- Calculer, pour différentes valeurs de  $n$  :

$$\sum_{k=1}^n \frac{1}{k} \quad \text{et} \quad \sum_{k=n}^1 \frac{1}{k}$$

- Calculer pour  $x = 0.5$ ,  $x = 0.25$  et  $x = 0.1$

$$\sum_{k=1}^{1000} x$$

- Programmer la suite de Muller (on admettra qu'elle converge vers 6).

$$\begin{cases} u(0) = 2 \\ u(1) = -4 \\ u_{n+1} = 111 - \frac{1130}{u_{n-1}} + \frac{3000}{u_{n-1}u_{n-2}} \end{cases}$$

## Les fonctions

## 225 portée des variables

*Cet exercice est à faire sans machine.*

Pour chacun des trois scripts suivants, préciser ce qu'affiche l'instruction `f()`, ou si cette instruction lève une exception :

```
1     a = 2
2
3 def f() :
4     a = 3
5
6     def g() :
7         a = 4
8
9     print(a)
10
11
12 f()
```

```
a = 2

def f():
    a = 3

def g():
    print(a)

g()

f()
```

```
a = 2  
  
def f(): :
```

```
4
5     def g() :
6         print(a)
7
8     g()
9
10    f()
```

226 effet de bord ?

Cet exercice est à faire sans machine.

1. On considère la fonction suivante :

```
1 def f(x) :  
2     x = x + 1  
3     return x
```

Quelles sont les valeurs affichées lors de la séquence d'instructions suivante ?

```
1 x = 2
2 print(f(x))
3 print(x)
```

2. On considère la fonction suivante

```
1 def f(x) :  
2     x = x + [4]  
3     return x
```

Quelles sont les valeurs affichées lors de la séquence d'instructions suivante ?

```
1 x = [1, 2, 3]
2 print(f(x))
3 print(x)
```

3. On considère la fonction suivante :

```
1 def f(x) :
2     x.append(4)
3     return x
```

Quelles sont les valeurs affichées lors de la séquence d'instructions suivante ?

```
1 x = [1, 2, 3]
2 print(f(x))
3 print(x)
```

$$\begin{aligned} \mathbb{E}(\varphi(X)) &= \int \varphi(x) d\mathbb{P}_X(x) \\ &= \binom{n}{k} p^k (1-p)^{n-k} \\ \mathbb{P}(A) &= \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i) \\ &\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1 \\ \frac{\bar{X} - \mu}{\sigma} &\rightarrow \mathcal{N}(0,1) \end{aligned}$$

# Récurseurité

## Fonctions récursives

### 227 des mises en abyme

1. On considère la fonction suivante :

```

1 def f(n, p) :
2     """ f(int, int) -> int """
3     if n == 0 :
4         return p
5     else :
6         return f(n - 1, p + 1)

```

Quelles valeurs sont renvoyées par les appels suivants ?

- |               |                |
|---------------|----------------|
| (a) $f(0, 4)$ | (c) $f(3, -1)$ |
| (b) $f(1, 4)$ | (d) $f(-1, 3)$ |

2. On considère la fonction suivante :

```

1 def g(n) :
2     """ g(int) -> int """
3     if n == 0 :
4         return 1
5     else :
6         return 2 * g(n)

```

Quelles valeurs sont renvoyées par les appels suivants ?

- |            |            |
|------------|------------|
| (a) $g(0)$ | (b) $g(4)$ |
|------------|------------|

3. On considère la fonction suivante :

```

1 def h(n, p) :
2     """ h(int, int) -> int """
3     if n%2 == 0 :
4         return h(n//2, p+1)
5     else :
6         return h(n + 1, p-1)

```

Quelles valeurs sont renvoyées par les appels suivants ?

- |               |               |
|---------------|---------------|
| (a) $h(4, 3)$ | (b) $h(7, 1)$ |
|---------------|---------------|
4. (a) Quels sont les problèmes rencontrés dans les fonctions précédentes ?  
 (b) Citer deux éléments nécessaires pour s'assurer de la terminaison de ces fonctions.  
 (c) Proposer une modification de ces fonctions pour que les problèmes rencontrés soient corrigés.

### 228 exposant

Soient  $x$  et  $n$  deux entiers, avec  $x > 0$ .

On veut écrire une fonction puissance, qui prend en argument  $x$  et  $n$  et qui calcule  $x^n$ .

**Contrainte** : on ne s'autorise qu'une seule opération : la multiplication.

- Proposer une fonction puissanceRec( $x, n$ ), récursive, qui répond à la spécification précédente.
- Proposer une version non récursive puissanceIter( $x, n$ ) qui répond à la spécification précédente.

### 229 somme et racine digitales

- La somme digitale d'un entier naturel  $n$  est la somme de ses chiffres.

On veut écrire la fonction sommeDigitale( $n$ ), récursive, qui prend en argument un entier naturel  $n$  et qui renvoie la somme de ses chiffres.

- (a) Quel est le cas de base ?  
 (b) On suppose qu'on n'est pas dans un cas de base. Comment se rapprocher du cas de base ?  
 (c) Programmer la fonction sommeDigitale(n).
2. La racine digitale d'un entier  $n$  est calculée de la manière suivante :
- on calcule la somme digitale de  $n$  ;
  - si cette somme digitale a plusieurs chiffres, il faut calculer de nouveau la somme digitale de cette somme ;
  - on réitère jusqu'à ce qu'on obtienne un nombre à un chiffre. La racine digitale de  $n$  est ce dernier nombre.

Programmer la fonction

racineDigitale(n),

réursive, qui calcule la racine digitale de  $n$ . On pourra utiliser la fonction sommeDigitale.

### 230 autour des tableaux

Pour chacune des fonctions suivantes, rappeler la version itérative et proposer une version récursive :

1. la fonction effectif(tab, element) qui compte le nombre de fois où apparaît element dans tab.
2. la fonction maximum(tab) qui détermine le maximum du tableau.
3. la fonction recherche(tab, element) qui renvoie True si element est dans le tableau, False sinon.

### 231 la suite de Fibonacci

La suite de Fibonacci s'obtient de la façon suivante : les deux premiers termes de la suite sont 1 et 1, puis chaque terme est obtenu en faisant la somme des deux termes précédents.

1. Calculer à la main les 10 premiers termes de cette suite.
2. Écrire une fonction FiboRec(n), récursive, qui calcule le n-ième terme de la suite de Fibonacci.

3. Démarrer tous les appels intermédiaires, lorsqu'on appelle FiboRec(5). Que constate-t-on ?
4. Combien d'appels récursifs seraient nécessaires pour calculer le 100<sup>ème</sup> terme de la suite de Fibonacci avec la fonction FiboRec ?
5. Expliquer pourquoi ce n'est pas un bon algorithme.
6. Proposer une version itérative permettant de calculer le  $n$ -ième terme de la suite de Fibonacci. Quelle est sa complexité ?

# Tarte au citron meringuée

Fichiers attachés : recette tarte au citron 

## 232 Pour bien commencer

Créer un dossier "mon\_site" dans votre espace personnel. Tous les fichiers devront être sauvegardés ici.

## 233 Premier site web

- Ouvrir un éditeur de texte de type bloc-note, taper les éléments suivants :

```

1 <!doctype html>
2
3 <html lang="fr">
4
5 </html>
```

- Enregistrer votre fichier en l'appelant "tarte\_citron.html"
- Double-cliquer sur l'icône ainsi créée.
- Quel est le rôle de !doctype ? html ?
- Dans votre navigateur, faire un clic-droit et cliquer sur "afficher le code source"

## Rôle des balises

Le jeu des exercices suivants consistent à déterminer le rôle des balises

## 234 head/meta/title

Dans la balise html taper :

```

<head>
  <meta charset="utf-8" />
  <title> Tarte au citron </title>
</head>
```

On pensera à bien enregistrer le fichier "tarte\_citron.html" systématiquement et à actualiser le navigateur.

Quel est le rôle de head ? meta ? title ?

## 235 body/h1/p

Dans la balise html sous la balise head taper :

```

1 <body>
2   <h1> Tarte au citron meringuée </h1>
3   <p> Voici la célèbre recette de
4     la tarte au citron </p>
5   <p> Bon appétit </p>
6   <h2> préparation </h2>
7   <h2> ingrédients </h2>
8 </body>
```

Quel est le rôle de body ? h1 ? h2 ? p

## 236 Sous les ingrédients taper :

```

1 <ul>
2   <li> 1 pâte sablée</li>
3   <li> 4 citrons de taille moyenne sans
4     traitement après récolte </li>
5   <li> 300 g de sucre </li>
6   <li> 3 oeufs + 3 blanc d'oeufs</li>
7   <li> 1 cuillère à soupe de Maïzena </li>
8   <li> 1 demi cuillère à café de levure
9     chimique</li>
10 </ul>
```

Quel est le rôle de ul ? li ?

$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\left( \frac{n}{k} \right) p^k (1-p)^{n-k}$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\frac{\bar{X} - \mu}{\sigma} \xrightarrow{\mathcal{N}(0,1)}$$

$$\mathbb{P}\left( \sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1, \alpha}^2 \right) \approx 1 - \alpha$$

**237** Sous préparation taper :

<ol>

- <li> Faire cuire à blanc 10 mn, à 180 degrés Celsius la pâte sablé (baisser ensuite le four à 120/150 degrés environ pour la meringue)
- </li>
- <li> Laver les citrons, en zester 2 et mettre les zestes très fins dans une casserole.
- </li>
- <li> Presser les citrons et mettre le jus avec les zestes dans la casserole. Verser le sucre et la Maïzena, remuer, et commencer à faire chauffer à feux doux.
- </li>
- <li> Battre les oeufs dans un récipient séparé, puis les incorporer tout en remuant le jus de citron, le sucre, la Maïzena et les zestes.
- </li>
- <li> Mettre à feu fort et continuer à remuer à l aide d un fouet. Le mélange va alors commencer à s épaisser (attention de toujours remuer lorsque les oeufs sont ajoutés, car la crème de citron pourrait brûler).
- </li>
- <li> Une fois le mélange épaisse, ôter du feu et verser la crème au citron sur le fond de tarte cuit.
- </li>
- <li> Monter les blancs en neige avec une pincée de sel. Quand ils commencent à être fermes, ajouter le sucre puis la levure et mixer jusqu a ce que la neige soit ferme.
- </li>
- <li> Recouvrir la tarte avec les blancs en neige et napper. Cuire à feu doux (120/150 degrés) Pendant 1H.

</li>  
</ol>

Quel est le rôle de ol ?

**238** Rajouter l'astuce

**239** *strong/mark/em/i*

1. Mettre le mot célèbre en contenu d'une balise <strong>.
2. Mettre "cuire à blanc" en contenu d'une balise <em>
3. Dans les ingrédients, mettre "sans traitement après récolte" en contenu d'une balise <i>.
4. Mettre "attention de toujours remuer lorsque les oeufs sont ajoutés, car la crème de citron pourrait brûler" en contenu d'une balise <mark>
5. Quel est le rôle de strong ? mark ? em ? i ?

**240** *img*

Fichiers attachés :

image tarte 

image citron 

1. Enregistrer les deux images dans un dossier images à côté du fichier html.
  2. Taper au début du body :
- ```
1   
```
3. Mettre l'autre image juste avant l'image des citrons
  4. Quel est le rôle de img ?

**241 Pâte sablée****Fichiers attachés :**

recette en txt

recette en pdf

Première image

Deuxième image

Sur le même modèle que la recette de la tarte au citron meringuée, créer une page web de la recette de la pâte sablée.

**là où tout commence : les liens hypertextes (balise a)****242 a**

Tout en bas de votre page web, taper :

- 1 `< a href="https://www.marmiton.org/">Pour plus d'idées.</a>`

Quel est le rôle de a ?

**243 D'une page vers une autre : lien relatif**

Créer un lien hypertexte de la recette de la tarte au citron meringuée vers la pâte sablée lors d'un clic sur l'ingrédient "pâte sablée".

**244 Pour aller plus loin**

Tester :

```

< a href="mailto:maxime.fourny@gmail.com">
    Envoyez-moi un e-mail ! </a>
< a href="images/citron.png"> Télécharger
    le fichier </a>

```

# CSS : La mise en forme

## Fichiers attachés :

- recette tarte au citron 
- recette pâte sablée 
- style1.css 
- style\_base.css 
- livre de création de site web 

## Mise en place d'un site

### 245 Structure

1. Créer un dossier "mon\_site" qui contiendra les sous dossiers "images", "styles", "js" et "html"
2. Enregistrer les deux fichiers html dans le dossier html.
3. Enregistrer les deux fichiers de style dans le dossier style

## Choix du style

### 246 C'est parti pour le style

1. Afficher la page de la recette de la tarte au citron dans un navigateur.
  2. Rajouter juste en dessous de la balise title :
- ```
<link rel="stylesheet" type="text/css"
      href="../styles/style1.css">
```
3. Rafraîchir la page dans le navigateur
  4. Modifier style1 par style\_base. Quelle est l'utilité de cette feuille de style ?

### 247 Et les images

Modifier les pages html pour permettre l'affichage de l'image pour chacune des deux recettes.

### 248 index

Créer la page principale du site que l'on appellera index.html. Elle se trouvera à la racine.

## Fonctionnement fondamentale

dans le fichier `mon_style.css`, pour décrire le style d'une `<balise>`, il suffit de taper :

```
balise
{
    propriete1 : valeur1;
    propriete2 : valeur2;
    ...
}
```

où chaque propriété (couleur, font ...) ne sera modifiée que pour la balise

# Vocabulaire de la programmation orienté objet

## 249 Définitions

1. Donner une définition des mots suivants sous le prisme de la programmation orienté objet :

- objet
- classe
- attributs
- attributs public
- attributs privé
- méthodes
- interface
- implémentation
- encapsulation
- instanciation
- instance
- constructeur
- destructeur
- polymorphisme
- héritage
- getter
- setter
- accesseurs
- mutateurs

2. Faire évoluer ces définitions au cours des séances.

# Comprendre la programmation orienté objet

## Philosophie de la POO

La programmation orientée **objet** (POO) permet de créer des entités (objets) que l'on peut manipuler. Les objets que l'on a créés sont optimisés pour ce que l'on veut faire.

Une **classe** regroupe des **attributs** et des **méthodes** qui définissent et permettent de manipuler un objet. Les attributs et les méthodes sont en général **encapsulés** dans l'objet.

Créer sa propre classe revient à créer un nouveau type.

Un objet est une **instance** d'une classe.

## Point géométrique

L'objectif de cette partie est de créer une classe permettant de numériser un point géométrique : abscisse, ordonnée, distance à l'origine...

### 250 Première classe

Exécuter le programme suivant en Python :

```
1 class Point(object) :
2     "point geometrique"
```

### 251 Premier objet

1. À la suite du programme précédent, rajouter la ligne suivante :

```
1 A=Point()
```

2. Interpréter le résultat obtenu lors de l'appel des instructions suivantes :

```
1 >>> A
2 >>> type(A)
3 >>> help(Point)
4 >>> help(A)
```

3. Déterminer le lien poo entre Point et A.

### 252 Premiers attributs

1. À la suite du programme précédent, rajouter la ligne suivante :

```
1 A.x=2
```

2. Interpréter le résultat obtenu lors de l'appel des instructions suivantes :

```
1 >>> A.x
2 >>> type(A.x)
3 >>> help(A.x)
```

3. Compléter l'objet A pour avoir une numérisation d'un point géométrique dans le plan.

4. Créer de la même manière un deuxième point B.

5. Déterminer le lien poo entre Point, A et A.x.

6. Quelle interface aimeraient avoir pour simplifier la création d'une instance de point ?

7. Quelle difficulté liée à l'implémentation peut-on anticiper ?

### 253 Méthode constructeur

1. Modifer la classe Point avec les instructions suivantes :

```
1 class Point(object) :
2     "point geometrique"
3     def __init__(self,x = 0,y = 0) :
4         self.x = x
5         self.y = y
```

2. Interpréter le résultat obtenu lors de l'appel des instructions suivantes :

```

1 A = Point()
2 B = Point(2,3)
3 A.x
4 A.y
5 Point.__doc__
6 __doc__

```

La fonction `__init__` qui a été encapsulée dans la classe est appelée méthode constructeur. Elle porte bien son nom.

3. Expliquer le sens de `self`.
4. Modifier le programme pour régler la difficulté lié au type de `self.x` et `self.y`.

#### 254 Première méthode ?

1. Écrire une fonction "distance\_origine" qui renvoie la distance d'un Point à l'origine
2. Écrire une fonction "distance" qui renvoie la distance entre deux Points
3. Après avoir définis quels attributs pourraient être intéressants, créer une classe vecteur.
4. Écrire une fonction qui renvoie un vecteur connaissant d'un vecteur si on lui fournit deux points en entrée.

#### 255 Critique de l'exercice précédent

1. Qu'est ce qui différencie une fonction d'une méthode
2. Quelles fonctions créées précédemment pourraient être catégorisées comme des méthodes ?

#### 256 Syntaxe d'une méthode

Un élève a trouvé le script suivant :

```

1 import math as je_suis_un_gros_boulet
2
3 class Point(object) :
4     "point geometrique"
5     def __init__(self,x = 0,y = 0) :
6         self.x = x
7         self.y = y
8     def distance_origine(self) :
9         return sqrt(self.x**2+self.y**2)
10
11 A = Point(2,3)

```

1. Comment faire pour connaître la distance à l'origine du point `A` ?
2. Quel erreur python peut-on anticiper ? Comment la lever ?
- 3.Modifier le programme en rajoutant une méthode permettant de connaître la distance d'une instance à n'importe quel autre point du plan.

#### 257 Quelle est la différence entre un attribut d'instance et un attribut de classe ?

#### 258 getter/setter

1. Quel peut être les limites et les dangers de la syntaxe :
- ```

1 mon_objet.mon_attribut = valeur

```
2. Écrire une méthode permettant d'obtenir les coordonnées d'un Point.
  3. Écrire une méthode permettant d'obtenir les coordonnées d'un Point.
  4. Ces deux dernières méthodes sont-elles indispensables, pratiques, intéressantes ?

### 259 Ce qui est privé doit resté privé

1. Modifier le programme pour que l'instruction python

```
1 A.x = 5
```

renvoie un message d'erreur "interdit".  $A$  étant un Point crée préalablement.

2. Modifier le programme pour que l'instruction python

```
1 abscisse = A.x
```

renvoie un message d'erreur "interdit".  $A$  étant un Point crée préalablement.

3. Modifier le programme pour que l'instruction python

```
1 A.bidule = 5
```

renvoie un message d'erreur "interdit".  $A$  étant un Point crée préalablement.

# Manipuler la POO

## Un grand classique : la classe Time

### 260 *class Time*

Créer une classe Time.

- Une instance devra être instancié avec 3 attributs : heure, minutes, secondes.
- Elle contiendra un getter qui permettra d'afficher l'heure
- Elle contiendra un setter qui permettra de modifier une heure.

## Domino

### 261 *Jeu de domino*

Définir une classe Domino qui permette d'instancier des objets simulant les pièces d'un jeu de dominos. Le constructeur de cette classe initialisera les valeurs des points présents sur les deux faces A et B du domino (valeurs par défaut = 0). Deux autres méthodes seront définies :

- Une méthode affiche\_points() qui affiche les points présents sur les deux faces ;
- Une méthode valeur() qui renvoie la somme des points présents sur les 2 faces.

## Compte bancaire

### 262 *Simulation d'une banque*

Définir une classe compte\_bancaire(), qui permette d'instancier des objets tels que compte1, compte2, etc. Le constructeur de cette classe initialisera deux attributs d'instance nom et solde, avec les valeurs par défaut 'Dupont' et 1000. Trois autres méthodes seront définies :

- depot(somme) permettra d'ajouter une certaine somme au solde ;
- retrait(somme) permettra de retirer une certaine somme du solde ;
- affiche() permettra d'afficher le nom du titulaire et le solde de son compte.

## Combat de chien

### 263 *class Chien*

Créer une classe Chien

- Un chien sera instancié avec pléthore d'attributs : son nom, son aboiement caractéristique, son nombre de point de vie, son niveau de fatigue, puissance d'attaque, le coût d'une attaque...
- Un chien peut en mordre un autre. Cela lui fait baisser son nombre de point de vie de la puissance d'attaque du chien mordueur. Le chien mordueur se voit fatigué du cout de son attaque.
- Un chien peut manger. Cela augmente ses points de vie.
- Un chien peut mâchouiller une chaîne de caractères. La conséquence est que la chaîne de caractère se retrouve mélangée.
- Un Chien peut grogner
- ...

## boiboites

### 264 *Des boîtes, toujours des boîtes*

1. Créer une classe Boite.

Cette classe a pour attributs :

- Longueur
- Largeur
- Hauteur

Ces trois attributs devront êtres dans un ordre décroissant : longueur  $\geq$  largeur  $\geq$  hauteur

Cette classe a pour méthodes :

- Volume, qui comme son nom l'indique donne le volume d'une boite
- rentre\_dans (autre\_boite), qui renvoie vrai si l'objet Boite rentre dans autre\_boite.

2. Créer aléatoirement une liste d'une vingtaine de boîtes (les dimensions seront choisies entre 1 et 50).

3. A l'aide d'un algorithme glouton, donner une suite de boîtes aussi grande que possible qui rentrent les unes dans les autres.

## Écosystème

Dans cette partie, on va modéliser le fonctionnement d'un écosystème, avec de l'herbe, des moutons et des loups. Cette simulation est discrète : tous les « tours de jeu » (tous les ticks d'horloge), les moutons ainsi que les loups se déplacent, mangent ou meurent éventuellement, et l'herbe pousse.

**265**

### classe Monde

Cette classe est la carte sur laquelle vont évoluer les moutons.

Cette carte va représenter une prairie carrée qui sera matriciée sur laquelle va pousser de l'herbe dans chaque case de la carte.

- Cette classe a pour attributs :

- dimension
- carte

C'est une matrice carrée de dimension largeur, hauteur

exemple de David

# Héritage

## Fichiers attachés :

— atome.py 

- 266** Analyser et faire fonctionner le programme atome.py
- 267** Créer une classe Point\_espace à partir de la classe Point.
- 268** Créer une classe Time\_annuel à partir de la classe Time. Elle contiendra comme attribut supplémentaire : le jour, le mois et l'année.
- 269** Définissez une classe Cercle(). Les objets construits à partir de cette classe seront des cercles de tailles variées. En plus de la méthode constructeur (qui utilisera donc un paramètre rayon), vous définirez une méthode surface(), qui devra renvoyer la surface du cercle. Définissez ensuite une classe Cylindre() dérivée de la précédente. Le constructeur de cette nouvelle classe comportera les deux paramètres rayon et hauteur. Vous y ajouterez une méthode volume() qui devra renvoyer le volume du cylindre (rappel : volume d'un cylindre = surface de section × hauteur).
- 270** Complétez l'exercice 269 en lui ajoutant encore une classe Cone(), qui devra dériver cette fois de la classe Cylindre(), et dont le constructeur comportera lui aussi les deux paramètres rayon et hauteur. Cette nouvelle classe possédera sa propre méthode volume(), laquelle devra renvoyer le volume du cône.

# Éléments d'un PC

## 271 Démontage

1. Démonter la tour (attention, vous aurez à la remonter, souvenez vous des emplacements des différentes pièces).
2. Identifier les éléments d'architecture de l'ordinateur : composants de base (unité centrale, mémoires, périphériques) pour pouvoir compléter la fiche.
3. Expliquer le rôle des constituants d'un ordinateur.
4. Faites la même opération avec le PC portable

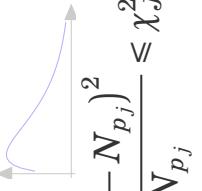
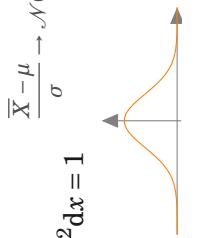
$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\binom{n}{k} p^k (1-p)^{n-k}$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\frac{\bar{X} - \mu}{\sigma} \rightarrow \mathcal{N}(0,1)$$

$$\mathbb{P}\left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1, \alpha}^2\right) \approx 1 - \alpha$$



# Le langage assembleur

## Du langage machine ...

Le langage machine est une succession de combinaisons de bits.

Les instructions machines sont relativement basiques. ce sont des instructions de bas niveau.

Parmi elles :

- les instructions arithmétiques de base (addition, soustraction...);
- les instructions de transfert de données, qui permettent de transférer les données des registres dans la mémoire vive et réciproquement ;
- les instructions de branchement, qui permettent de sauter à une autre partie du code.

Une instruction machine est une chaîne binaire composée de deux parties :

- le champs opération qui indique au processeur le type de traitement à réaliser ;
- le champs opérandes qui indique la nature des données sur lesquelles l'opération doit être réalisée.

Les opérandes peuvent être soit des valeurs immédiates, soit des valeurs rangées dans un des registres, soit des valeurs stockées en mémoire vive.

Chaque famille de processeurs utilise des jeux d'instructions différents.

## ... vers le langage assembleur

En langage assembleur, une instruction machine est représentée par un symbole dit mnémoniques (c'est-à-dire faciles à retenir).

### 272 instructions de base

Expliquer le rôle des instructions suivantes :

1. LDR R0, 78
2. MOV R1, #25
3. STR R2, 131

4. ADD R0, R1, #37

5. SUB R0, R1, R2

6. CMP R0, #18

BEQ laBas

7. HALT

### 273 premières instructions en assembleur

Écrire les instructions en assembleur correspondant aux phrases suivantes :

1. Additionne la valeur stockée dans le registre R0 et la valeur stockée dans le registre R1. Le résultat est stockée dans le registre R0.
2. Place la valeur stockée à l'adresse mémoire 251 dans le registre R3.
3. Place le contenu du registre R0 en mémoire vive à l'adresse 124.
4. La prochaine instruction à réaliser se situe après l'étiquette monLabel.
5. Si la valeur stockée dans le registre R0 est égale à 4, alors la prochaine instruction à exécuter se situe en mémoire vive à l'adresse représentée par l'étiquette parIci.

### 274 un programme complet

On considère le programme suivant :

```

1    LDR R0, 78
2    LDR R1, 131
3    CMP R0, R1
4    BGT siAlors
5    MOV R2, R0
6    STR R2, 83
7    HALT
8    siAlors :
9    MOV R2, R1

```

10 STR R2, 83  
11 HALT

- Détailler les instructions de ce programme. À quoi sert ce programme ?
- On suppose qu'en Python, l'emplacement mémoire 78 correspond à la variable a, l'emplacement mémoire 83 correspond à la variable b et l'emplacement mémoire 131 correspond à la variable c.  
Traduire cet algorithme en Python.

### 275 pour aller plus loin

On considère le programme Python suivant :

```
1 x = 1
2 n = 0
3 while x < 100 :
4     x = 2*x
5     n = n + 1
```

Traduire ce programme en assembleur.

Pour programmer en assembleur, on va utiliser le simulateur développé par Peter L. Higginson, disponible à l'adresse :

<http://www.peterhigginson.co.uk/AQA/>

### 276 gestion de la mémoire

- Tester le programme suivant dans le simulateur et observer le comportement des différentes mémoires (registres et mémoire vive) :

```
1     MOV RO, #12
2     STR RO, 100
3     LDR R1, 100
```

4 MOV R2, #28  
5 MOV R3, R2  
6 STR R3, 101  
7 HALT

- Écrire un programme en assembleur qui exécute les instructions suivantes puis le tester dans le simulateur :
  - range dans le registre R0 la valeur immédiate 55 ;
  - stocke le contenu du registre R0 en mémoire vive à l'adresse 128 ;
  - charge le contenu de l'adresse 128 de la mémoire vive dans le registre R1 ;
  - range dans le registre R5 le contenu du registre R1.

### 277 les opérations

Écrire un programme en assembleur qui exécute les instructions suivantes puis le tester dans le simulateur :

- range dans le registre R0 la valeur immédiate 8 ;
- range dans le registre R1 la somme de la valeur rangée dans le registre R0 et de la valeur immédiate 32 ;
- stocke en mémoire vive à l'adresse 110 le contenu du registre R1 ;
- range dans le registre R2 le double de la valeur rangée dans le registre R1.

### 278 les branchements

- Tester le programme suivant et comprendre comment il fonctionne :

```
1     MOV RO, #12
2     CMP RO, #8
3     BGT siOui
4     STR RO, 100
5     HALT
6     siOui :
```

```

7   SUB R0, R0, #8
8   STR R0, 100
9   HALT

```

2. Écrire et tester un programme correspondant au programme Python suivant :

```

1 x = 7
2 y = 12
3 if x < y :
4     x = x + y
5 elif x == y :
6     y = x + y
7 else :
8     x = x - y

```

### 279 les boucles

1. Écrire et tester un programme correspondant au programme Python suivant :

```

1 s = 0
2 for i in range(12) :
3     s = s + i

```

2. Écrire et tester un programme correspondant au programme Python suivant :

```

1 s = 0
2 while s < 100 :
3     s = s + i

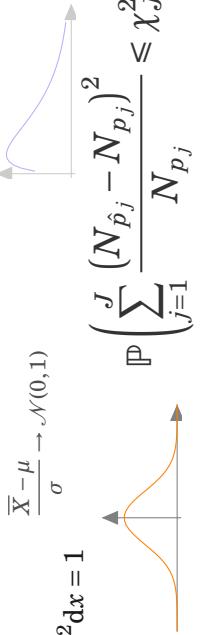
```

3. Écrire et tester un programme qui calcule le produit du contenu du registre R0 par le contenu du registre R1 et qui stocke le résultat en mémoire vive à l'adresse 100.

4. Pour aller plus loin : Écrire et tester un programme qui calcule le quotient et le reste de la division euclidienne du contenu du registre R0 par le contenu du registre R1 et qui stocke le quotient en mémoire vive à l'adresse 100 et le reste en mémoire vive à l'adresse 101.

# Modèle TCP/IP

**280** Expliquer en quoi consiste le modèle TCP/IP

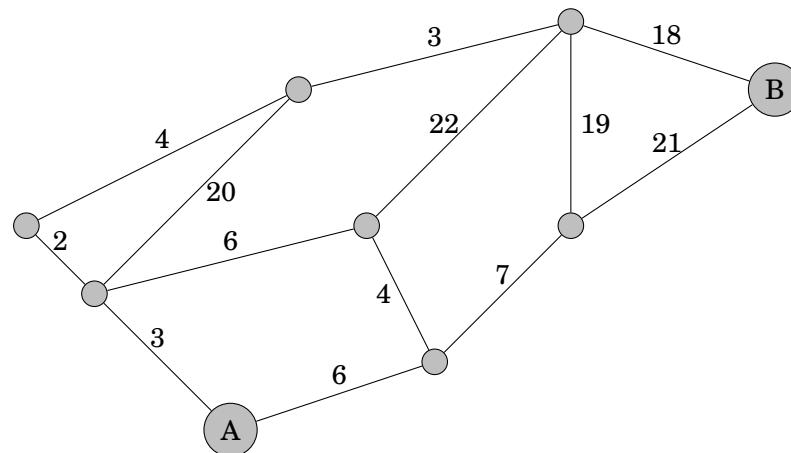
$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$
$$\binom{n}{k} p^k (1-p)^{n-k}$$
$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$
$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$
$$\frac{\bar{X} - \mu}{\sigma} \rightarrow \mathcal{N}(0,1)$$
$$\mathbb{P}\left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1, \alpha}^2\right) \approx 1 - \alpha$$


# Graphe et distance

## GPS

### 281 Copilote

Vous souhaitez vous rendre de la ville A à la ville B. Une carte schématisée se trouve ci-dessous.

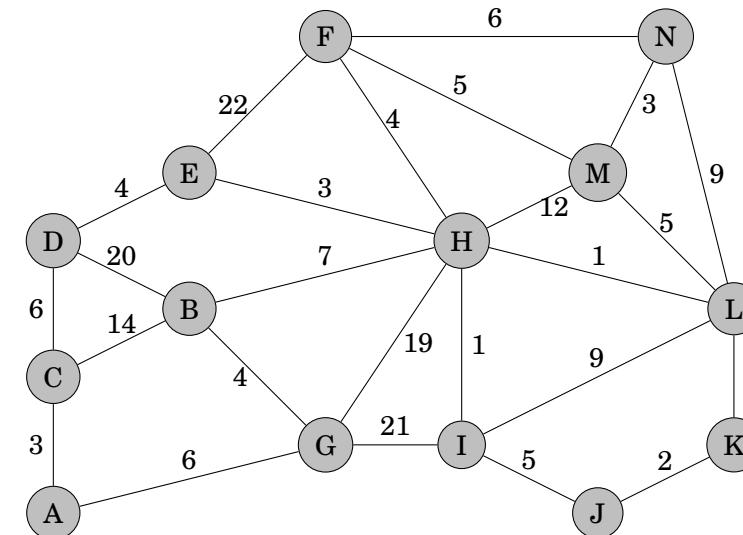


- Déterminer la route à prendre pour y arriver le plus vite possible.
- Pour faire du tourisme et prendre son temps, déterminer la route à prendre sans passer deux fois par la même ville.
- Préciser comment être sûr du résultat.

### 282 Copilote efficace

Déterminer dans la carte ci contre :

- Le chemin le plus court entre la ville A et la ville N.
- Le chemin le plus long entre la ville A et la ville N.



### 283 plus court ?

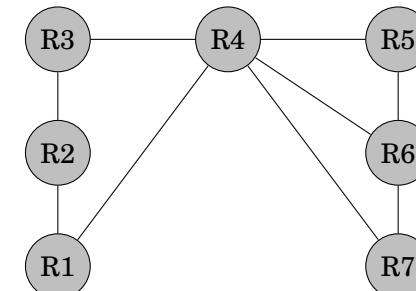
Le chemin le plus court est mesuré selon quel distance ?

## Routage sur internet

### 284 Route la plus rapide

Le routeur numéro 1 (R1) souhaite envoyer un message au routeur numéro 7 (R7).

- Indiquer quel chemin le message doit prendre.



- Expliquer pourquoi ce choix de chemin (qui est effectivement implémenté sur Internet dans le protocole de routage RIP) est critiquable et améliorabile.

**285 meilleur protocole** Le protocole de routage RIP est remplacé/complété par un autre protocole de routage dénommé protocole OSPF qui fonctionne différemment, car il prend en compte la bande passante. Le protocole attribue un coût à chaque liaison afin de privilégier le choix de certaines routes plus rapides. Plus le coût est faible, plus le lien est intéressant.

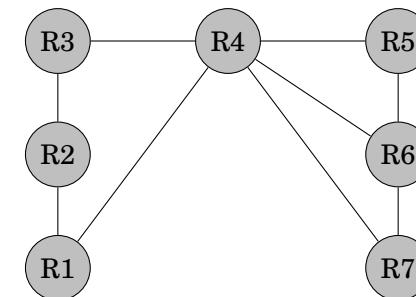
La formule de calcul du coût d'une liaison est donnée par la formule :

$$\text{coût} = \frac{10^8}{\text{bande passante du lien en bps}}$$

Voici le tableau référençant Les coûts des liaisons en fonction du type de liaison entre deux routeurs (Ethernet, fibre optique...) :

| Type de réseau     | Coût par défaut |
|--------------------|-----------------|
| FDDI, FastEthernet | 1               |
| Ethernet 10 Mbps   | 10              |
| E1 (2,048 Mbps)    | ?               |
| T1 (1,544 Mbps)    | 65              |
| 64 Kbps            | ?               |
| 56 Kbps            | 1758            |
| 19.2 Kbps          | 5208            |

1. Compléter les coûts manquants dans le tableau.
2. Déterminer la bande passante du réseau FDDI (fibre optique).
3. Compléter le graphe ci-contre avec le coût de chaque liaison et déterminer alors la route la plus rapide
  - Le réseau entre R1 et R2 est de type Ethernet 10 Mbps.
  - Le réseau entre R2 et R3 est de type E1.
  - Le réseau entre R3 et R4 est de type FDDI.
  - Le réseau entre R1 et R4 est de type 19.2 Kbps.
  - Le réseau entre R4 et R7 est de type 19.2 Kbps.
  - Le réseau entre R4 et R6 est de type E1.
  - Le réseau entre R4 et R5 est de type 64 Kbps.
  - Le réseau entre R5 et R6 est de type T1.
  - Le réseau entre R6 et R7 est de type FastEthernet.

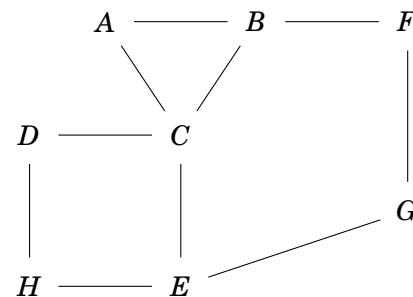


4. Déterminer alors le chemin le plus rapide.

# Routage élastique

## Étude d'un exemple

On considère le réseau suivant formé de huit routeurs  $A, B, C, D, E, F, G$  et  $H$  :



La table de routage de ce réseau est donnée par le tableau suivant :

| Routeur A |      | Routeur B |      | Routeur C |      | Routeur D |      |
|-----------|------|-----------|------|-----------|------|-----------|------|
| Dest.     | Lien | Dest.     | Lien | Dest.     | Lien | Dest.     | Lien |
| B         | B    | A         | A    | A         | A    | A         | C    |
| C         | C    | C         | C    | B         | B    | B         | C    |
| D         | C    | D         | C    | D         | D    | C         | C    |
| E         | C    | E         | C    | E         | E    | E         | H    |
| F         | B    | F         | F    | F         | B    | F         | C    |
| G         | C    | G         | F    | G         | E    | G         | H    |
| H         | C    | H         | C    | H         | E    | H         | H    |

| Routeur E |      | Routeur F |      | Routeur G |      | Routeur H |      |
|-----------|------|-----------|------|-----------|------|-----------|------|
| Dest.     | Lien | Dest.     | Lien | Dest.     | Lien | Dest.     | Lien |
| A         | C    | A         | B    | A         | F    | A         | D    |
| B         | C    | B         | B    | B         | F    | B         | D    |
| C         | H    | C         | G    | C         | E    | C         | D    |
| D         | H    | D         | B    | D         | E    | D         | D    |
| F         | G    | E         | G    | E         | E    | E         | E    |
| G         | G    | G         | G    | F         | F    | F         | E    |
| H         | H    | H         | G    | H         | E    | G         | E    |

Cette table indique, pour chaque routeur, comment envoyer un message à un autre routeur. Par exemple, si le routeur  $A$  veut envoyer un message au routeur à destination du routeur  $F$ , il doit passer par le routeur  $B$ .

286

Répondre aux questions suivantes en utilisant la table de routage ci-dessus :

- Le routeur  $D$  doit envoyer un message au routeur  $C$ . A qui doit-il le transmettre ?
- Le routeur  $E$  doit envoyer un message au routeur  $F$ . A qui doit-il le transmettre ?
- Un message doit aller du routeur  $A$  au routeur  $G$ . Quel parcours suit-il ?
- Un message doit aller du routeur  $F$  au routeur  $C$ . Quel parcours suit-il ?

287 *Travaux de groupes*

Par groupes de huit élèves :

- Attribuer à chacun une des huit lettres des routeurs précédents.
- Transmettre, pour chaque élève, un objet à un autre élève (proche ou éloigné) en suivant la table de routage.

**Un peu de réflexion****288**

En utilisant ce qui a été vu au paragraphe précédent, répondre, en justifiant, aux questions suivantes :

1. Y a-t-il unicité des tables de routage ?
2. Entre différentes tables de routage qui fonctionnent, peut-on considérer que certaines sont meilleures que d'autres ?
3. Proposer, pour le réseau précédent, une table de routage estimer meilleure que celle fournie. Tester cette nouvelle table comme dans l'exercice 287.

**Réseau mouvant...****289** *changement de topologie*

1. Pourquoi la topologie d'un réseau peut-elle changer ?
2. Que se passe-t-il dans ce cas là au niveau des tables de routage ?

**290** Travaux de groupes II

Par groupes de huit élèves :

1. Choisir l'un des quatre réseaux suivants :

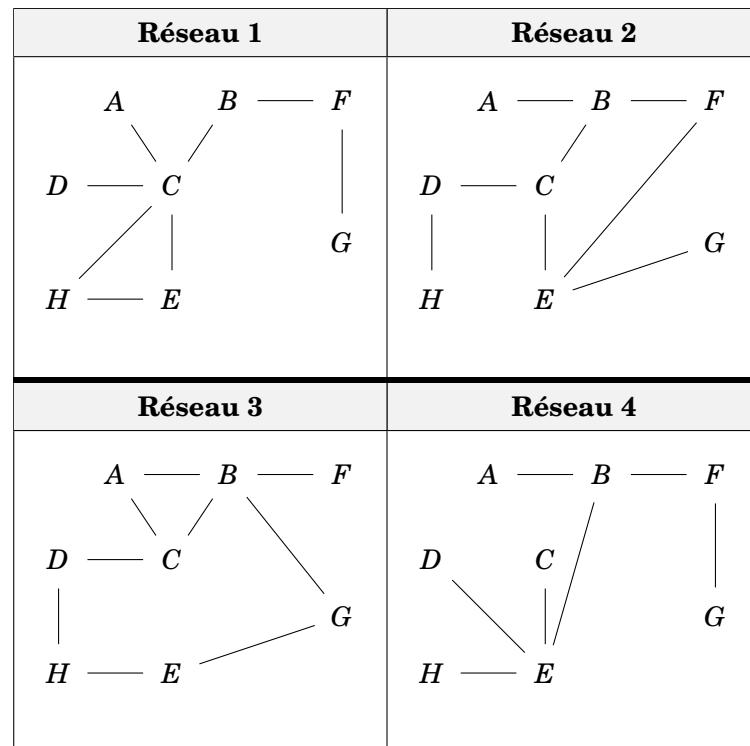
$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\binom{n}{k} p^k (1-p)^{n-k}$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\frac{\bar{X} - \mu}{\sigma} \xrightarrow{\text{distrib. normale}} \mathcal{N}(0,1)$$



2. Chacun connaissant ses voisins, définir le protocole d'envoi d'un message à travers ce réseau est respectant les règles suivantes :
  - On ne peut communiquer qu'avec ses voisins.
  - Toutes les informations que l'on reçoit se font via des messages.
  - Il n'existe pas de personne extérieure pour aller voir ce qui se passe globalement sur le réseau.
  - On a le droit de montrer sa table de routage à ses voisins.

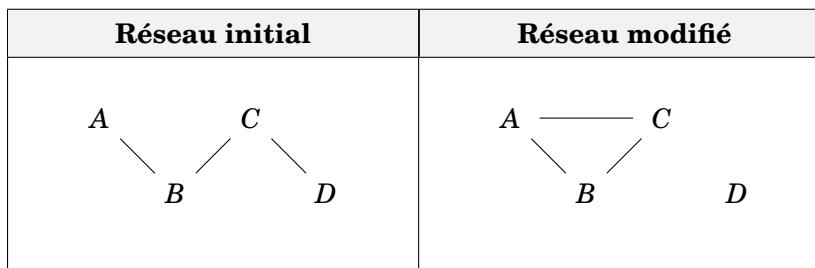
*On pourra éventuellement se poser les questions suivantes :*

  - (a) Qu'est-ce que j'envoie à mes voisins ?
  - (b) Que faire si j'ai perdu la connexion avec un de mes voisins ?
  - (c) Que faire si j'ai un nouveau voisin ?

- (d) Que faire si mon voisin me répond alors que je passais par lui pour atteindre un autre routeur ?  
 (e) Que faire si l'un de mes voisins peut accéder à un routeur alors que moi je n'avais pas de chemin pour y accéder ?  
 (f) Quand est-ce qu'on s'arrête ?

### Réseau mouvant... et aberrant...

L'algorithme vu au paragraphe précédent sur les réseaux mouvants est relativement simple, mais ne fonctionne plus dans le cas suivant :



En effet, puisque la table de routage du réseau initial est

| Routeur A |      | Routeur B |      | Routeur C |      | Routeur D |      |
|-----------|------|-----------|------|-----------|------|-----------|------|
| Dest.     | Lien | Dest.     | Lien | Dest.     | Lien | Dest.     | Lien |
| B         | B    | A         | A    | A         | B    | A         | C    |
| C         | B    | C         | C    | B         | B    | B         | C    |
| D         | B    | D         | C    | D         | D    | C         | C    |

la modification de topologie (suppression de la liaison entre C et D et création de la liaison entre A et C) oblige les routeurs A, C et D à modifier leur table comme suit :

| Routeur A |      | Routeur B |      | Routeur C |      | Routeur D |      |
|-----------|------|-----------|------|-----------|------|-----------|------|
| Dest.     | Lien | Dest.     | Lien | Dest.     | Lien | Dest.     | Lien |
| B         | B    | A         | A    | A         | B    | A         | ...  |
| C         | C    | C         | C    | B         | B    | B         | ...  |
| D         | B    | D         | C    | D         | ...  | C         | ...  |

Le routeur A envoie sa table à ses voisins : le routeur B ne modifie rien et n'envoie rien ; en revanche, le routeur C ayant reçu la table de A, il modifie sa table en conséquence et la renvoie :

| Routeur A |      | Routeur B |      | Routeur C |      | Routeur D |      |
|-----------|------|-----------|------|-----------|------|-----------|------|
| Dest.     | Lien | Dest.     | Lien | Dest.     | Lien | Dest.     | Lien |
| B         | B    | A         | A    | A         | A    | A         | ...  |
| C         | C    | C         | C    | B         | B    | B         | ...  |
| D         | B    | D         | C    | D         | A    | C         | ...  |

Les routeurs A et B n'ont pas modifié leur table et le routeur D ne reçoit rien et n'a personne à qui envoyer sa table. Tout le monde a donc terminé et l'algorithme s'arrête.

### 291 Problème

Quel est le problème de la table de routage finale ?

Pour contourner ce problème, la solution consiste à ajouter les distances entre les routeurs ; par exemple, le routeur A va au routeur B via B à distance 1. Ainsi, lorsqu'on crée un cycle comme ci-dessus, la distance augmente et lorsque celle-ci dépasse le diamètre du graphe (= la distance entre les deux routeurs les plus éloignés dans le graphe), on élimine le chemin concerné de la table (dans la pratique on garde un diamètre petit). Il va donc falloir stocker dans les tables à la fois à qui on doit faire suivre le message et la distance qui sépare le routeur de la destination. Pour le réseau initial, on a la table de routage suivante :

| Routeur A |       | Routeur B |       | Routeur C |       | Routeur D |       |
|-----------|-------|-----------|-------|-----------|-------|-----------|-------|
| Dest.     | Lien  | Dest.     | Lien  | Dest.     | Lien  | Dest.     | Lien  |
| B         | B (1) | A         | A (1) | A         | B (2) | A         | C (3) |
| C         | B (2) | C         | C (1) | B         | B (1) | B         | C (2) |
| D         | B (3) | D         | C (2) | D         | D (1) | C         | C (1) |

**292 Évolution**

Compléter l'évolution des tables de routages avec distances pour le changement de topologie en respectant les informations fournies :

- Les routeurs *C* et *D* retirent les destinations impossibles, le routeur *A* note qu'il a le routeur *C* comme voisin et envoie sa table :

| Routeur A |      | Routeur B |          | Routeur C |          | Routeur D |      |
|-----------|------|-----------|----------|-----------|----------|-----------|------|
| Dest.     | Lien | Dest.     | Lien     | Dest.     | Lien     | Dest.     | Lien |
| <i>B</i>  |      | <i>A</i>  | <i>A</i> |           | <i>A</i> |           |      |
| <i>C</i>  |      | <i>C</i>  |          | <i>B</i>  |          | <i>B</i>  |      |
| <i>D</i>  |      | <i>D</i>  |          | <i>D</i>  |          | <i>C</i>  |      |

- Le routeur *B* pouvait déjà aller en *C* directement, donc le message de *A* ne change rien pour lui ; il n'envoie donc rien. En revanche, le routeur *C* met sa table à jour :

| Routeur A |      | Routeur B |      | Routeur C |      | Routeur D |      |
|-----------|------|-----------|------|-----------|------|-----------|------|
| Dest.     | Lien | Dest.     | Lien | Dest.     | Lien | Dest.     | Lien |
| <i>B</i>  |      | <i>A</i>  |      | <i>A</i>  |      | <i>A</i>  |      |
| <i>C</i>  |      | <i>C</i>  |      | <i>B</i>  |      | <i>B</i>  |      |
| <i>D</i>  |      | <i>D</i>  |      | <i>D</i>  |      | <i>C</i>  |      |

- Comme le routeur *C* se rend compte que la distance obtenue est plus grande que le diamètre du réseau, il enlève le chemin vers *D* via *A* dans sa table et renvoie sa table qui ne contient pas de chemin vers *D*. Le routeur *A* ne modifie rien, mais le routeur *B* peut modifier sa table :

| Routeur A |      | Routeur B |      | Routeur C |      | Routeur D |      |
|-----------|------|-----------|------|-----------|------|-----------|------|
| Dest.     | Lien | Dest.     | Lien | Dest.     | Lien | Dest.     | Lien |
| <i>B</i>  |      | <i>A</i>  |      | <i>A</i>  |      | <i>A</i>  |      |
| <i>C</i>  |      | <i>C</i>  |      | <i>B</i>  |      |           |      |
| <i>D</i>  |      | <i>D</i>  |      | <i>D</i>  |      | <i>C</i>  |      |

- Le routeur *A* n'a pas changé sa table, donc n'envoie rien. En revanche, lorsqu'il reçoit la table du routeur *B*, il peut mettre à jour la sienne :

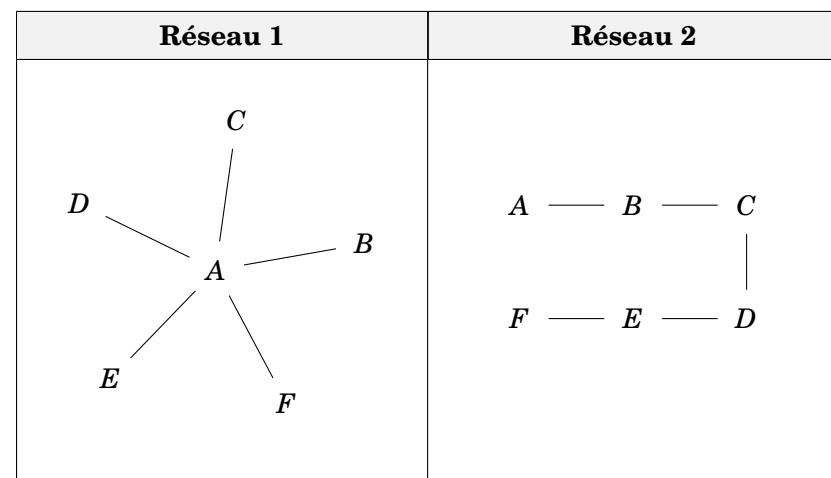
| Routeur A |      | Routeur B |      | Routeur C |      | Routeur D |      |
|-----------|------|-----------|------|-----------|------|-----------|------|
| Dest.     | Lien | Dest.     | Lien | Dest.     | Lien | Dest.     | Lien |
| <i>B</i>  |      | <i>A</i>  |      | <i>A</i>  |      | <i>A</i>  |      |
| <i>C</i>  |      | <i>C</i>  |      | <i>B</i>  |      | <i>B</i>  |      |
| <i>D</i>  |      | <i>D</i>  |      | <i>D</i>  |      | <i>C</i>  |      |

- Cette dernière table va-t-elle bouger après l'envoi de *A* de sa nouvelle table ?

**Bon réseau ?**

**293**

Quels sont les avantages et les inconvénients des deux réseaux suivants ?



**Compléments**

- Une vidéo sur le protocole IP :  
<https://www.youtube.com/watch?v=sLJ-QztAqNE&app=desktop>
- Deux outils permettant de voir une carte quel chemin emprunte les données pour aller de votre ordinateur au serveur cible :  
<https://www.monitis.com/traceroute/>  
<https://visualtraceroute.net/>



— Bilan du protocole IP :

Le protocole IP (= *Internet Protocol*) permet aux ordinateurs reliés à internet de dialoguer entre eux. Plus précisément, il se charge du routage de chaque paquet vers sa destination (c'est-à-dire de l'acheminement des données sur le réseau).

Pour passer d'un réseau à un autre, on utilise des routeurs. Le rôle d'un routeur est de décider quel chemin doit prendre le paquet qu'il reçoit et donc de choisir vers quel autre routeur envoyer le paquet jusqu'à ce que la destination soit atteinte. Pour prendre sa décision, le routeur utilise une table de routage qui sert d'aiguillage : « pour telle destination, envoyer le paquet à mon routeur voisin de droite, pour telle autre destination envoyer le paquet à mon routeur voisin de gauche », et ainsi de suite... En général, le paquet transite par des dizaines de routeurs et donc d'ordinateurs jusqu'à atteindre leur destinataire.

Pour que les paquets puissent parvenir à leur destination, une adresse IP *unique* composée de quatre nombres entre 0 et 255 séparés par des points (par exemple, 204.35.129.3) est attribuée à chaque ordinateur connecté au réseau et comme pour un courrier postal, chaque paquet de données contient l'adresse IP du destinataire.

Toutefois, le protocole IP n'est pas suffisant car il est considéré comme « non fiable », c'est-à-dire qu'il n'offre aucune garantie concernant le contenu des paquets à savoir : la corruption des données, l'ordre d'arrivée des paquets, la perte ou la destruction de paquets et la duplication des paquets. Les garanties concernant ces différents points sont déléguées à d'autres protocoles, comme le protocole TCP, afin de réduire le niveau de complexité des routeurs et ainsi de leur permettre de disposer d'une plus grande rapidité.

# Simulation d'un réseau avec le logiciel Filius

Ce TP utilise le logiciel Filius, qui va nous permettre de simuler un réseau.

Ce logiciel dispose de deux modes :

- le mode conception 
- le mode simulation 

## 294 installation

a faire

## 295 deux ordinateurs

1. En mode conception, placer deux ordinateurs et les relier par un câble ethernet.
2. (a) Double-cliquer sur chacun des deux ordinateurs pour accéder à leur configuration réseau.  
 (b) Modifier l'adresse IP de chacun des ordinateurs en 192.168.1.1 pour l'un et 192.168.1.2 pour l'autre. Cocher également la case **Utiliser l'adresse IP comme nom**.

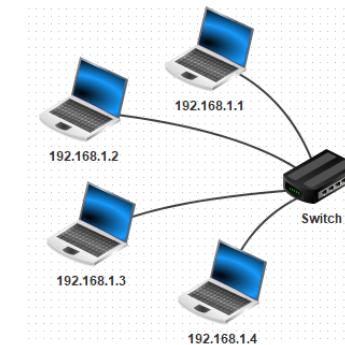


3. Passer en mode simulation puis :
  - (a) Double-cliquer sur la première machine. La fenêtre qui s'ouvre correspond au bureau de cette machine. Ouvrir l'installateur de logiciels et installer la ligne de commande.
  - (b) Ouvrir la ligne de commande par double-clic et saisir l'instruction ipconfig.
  - (c) Noter les informations qui apparaissent et les expliquer.

4. (a) Toujours en ligne de commande, faire un ping vers la deuxième machine (l'instruction est ping 192.168.1.2).  
 (b) Expliquer toutes les informations qui sont apparues. À quoi sert l'instruction ping ?
5. (a) Modifier l'adresse IP de la deuxième machine en lui affectant l'adresse 192.168.2.2.  
 (b) Refaire un ping depuis la première machine vers l'adresse 192.168.2.2. Que constate-t-on ? Expliquer.

## 296 plusieurs ordinateurs

1. En mode conception, ajouter deux autres ordinateurs. Configurer leurs adresses IP.
2. Peut-on relier ces deux machines aux deux premières ? Pourquoi ?
3. Ajouter un switch et relier toutes les machines au switch à l'aide d'un câble ethernet.



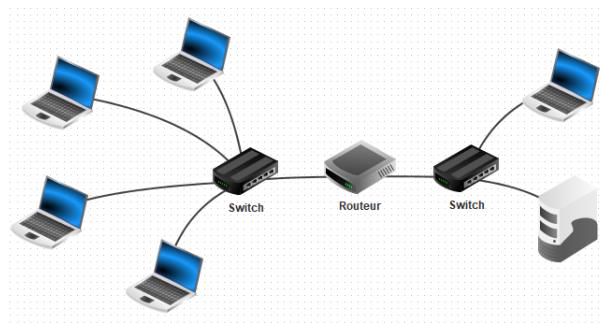
4. En mode simulation, double-cliquer sur le switch pour afficher sa table.
5. Tester toutes les connexions du réseau à l'aide de ping (il faut installer la ligne de commande sur chaque machine).
6. Afficher à nouveau la table du switch. Que constate-t-on ?
7. Qu'est-ce qu'une adresse MAC ?

**297 avant d'aller plus loin**

1. Qu'est-ce qu'une adresse réseau ?
2. Qu'est-ce qu'un masque ?
3. Comment obtient-on l'adresse réseau quand on connaît l'adresse IP d'une machine ?
4. Qu'est-ce que le broadcast ?
5. Combien de machines peut-on mettre sur un réseau dont le masque est 255.255.255.0 ?

**298 relier deux réseaux**

1. Réaliser le réseau suivant, composé de deux sous-réseaux reliés par un routeur (à deux interfaces).



2. Configurer l'adresse IP de chacune des machines pour que le sous-réseau de gauche ait pour adresse 192.168.1.0 et le sous-réseau de droite ait pour adresse réseau 192.168.2.0.
3. Peut-on faire communiquer une machine du sous-réseau de gauche avec une machine du sous-réseau de droite (tester avec un ping) ?
4. Faire un clic droit sur le routeur pour configurer ses deux interfaces (192.168.1.254 pour la première et 192.168.2.254 pour la deuxième).
5. Répondre à nouveau à la question 3. Le problème est-il résolu ?

6. Pour que les machines puissent utiliser le routeur, il faut les configurer en leur ajoutant une adresse de passerelle. Cette adresse est l'adresse IP de l'interface du routeur correspondant au réseau sur lequel se situe la machine.

Configurer **toutes** les machines en leur ajoutant une adresse de passerelle.

7. Vérifier alors que chaque machine peut communiquer avec n'importe quelle autre.
8. Effectuer un traceroute de la machine 192.168.1.4 vers la machine 192.168.2.1 (l'instruction est traceroute 192.168.2.1). Visualiser le chemin parcouru par les paquets de données entre ces deux machines.

**299 serveur Web**

On va installer un serveur Web sur la machine 192.168.2.2.

1. En mode simulation, installer sur la machine 192.168.2.2 :
  - l'explorateur de fichiers
  - le serveur Web
  - l'éditeur de texte
2. Vérifier à l'aide de l'explorateur de fichiers la présence d'un dossier webserver qui contient un fichier nommé index.html et une image. Ouvrir le fichier index.html à l'aide de l'éditeur de texte et imaginer à quoi ressemble la page web créée.
3. Installer sur la machine 192.168.1.1 un navigateur web.
4. Dans le navigateur web de la machine 192.168.1.1, taper dans la barre d'adresse http://192.168.2.2/. Que se passe-t-il ?
5. Cliquer sur l'icône **serveur Web** de la machine 192.168.2.2 puis **démarrer** puis recommencer la question précédente.
6. Retourner sur l'application Serveur web de la machine 192.168.2.2.  
Quelles sont les requêtes HTTP qui ont été envoyées par la machine 192.168.1.1 ? Combien y en a-t-il ? Pourquoi ? Quelles sont les réponses envoyées par le serveur ?

**300 serveur DNS**

Pour obtenir la page index.html sur un navigateur, on aimerait utiliser l'adresse `http://www.NSI.fr` plutôt que l'adresse `http://192.168.2.2/`.

On va donc installer un serveur DNS pour répondre à cela :

1. Sur le routeur, ajouter une troisième interface.
2. Relier une troisième machine sur le routeur. La configurer pour que son adresse réseau soit 192.168.3.0. Faire le nécessaire pour que cette machine puisse joindre toutes les autres et qu'elle soit joignable depuis toutes les autres.
3. En mode simulation, installer sur cette machine un serveur DNS. Le configurer en ajoutant le nom de domaine `www.NSI.fr` lié à l'adresse 192.168.2.2.
4. En mode conception, configurer l'adresse du serveur DNS sur la machine 192.168.1.1.
5. Tester sur le navigateur de la machine 192.168.1.1 l'adresse `www.NSI.fr` et vérifier qu'on obtient bien la page voulue.
6. Observer le chemin parcouru par les paquets de données lorsqu'on demande la page web.

# Premier codage d'une image

## Fichiers attachés :

- cercle.ggb
- cercle.pbm
- cercle.pgm
- image.ppm

## Premier codage

### 301 Cercle

1. Ouvrir cercle.ggb
2. Amusez vous avec.
3. Expliquer le principe de base pour coder ? numériser ? chiffrer ? l'information contenue dans une image.

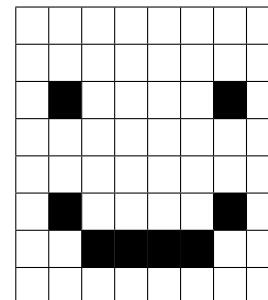
### 302 Première numérisation d'une image

On propose comme protocole de numérisation d'une image  $8 \times 8$  l'encodage suivant :

A chaque case de l'image est associé un bit. Lorsque le bit est à 0, la case est noire, lorsque le bit est à 1, la case est blanche.

Une ligne de l'image fait 8 cases, soit 8 bits, soit 1 octet.

Chaque ligne de l'image peut ainsi être décrite par un mot binaire assimilable à un nombre binaire. La lecture se faisant de la gauche vers la droite et de haut en bas.



1. Donner l'octet permettant d'encoder la ligne 3 en binaire puis en décimal.
2. Donner l'octet permettant d'encoder la ligne 7 en binaire puis en décimal.
3. Donner le mot binaire permettant d'encoder cette image.
4. Dessiner le négatif de l'image.
5. Donner le mot binaire encodant le négatif en binaire puis en décimal.

### 303 Négatif

Au vu de l'exercice précédent déterminer un protocole rapide pour faire le négatif d'une image  $8 \times 8$  selon notre protocole d'encodage.

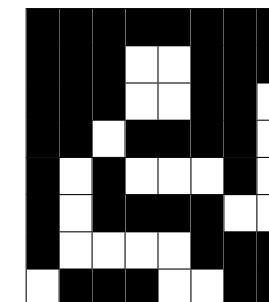
### 304 Numérisation d'une image (suite)

On finalise le protocole vu précédemment en notant

$$n_1.n_2.n_3.n_4.n_5.n_6.n_7.n_8$$

les huit nombres permettant d'encoder notre image.  
Ainsi  $n_i$  correspond à la  $i$ ème ligne.

1. Déterminer l'encodage de l'image suivante :



2. Dessiner l'image correspondant à la numérisation  
0111 1110.1011 1101.1100 0011.1010 0101.  
1100 0011.00000000.0101 1010.0100 0010
3. Donner l'encodage du négatif de cette image.

## Premiers formats d'images

---

### 305 .pbm

1. Ouvrir le fichier cercle.pbm avec un logiciel type gimp ou XNview
2. Ouvrir le fichier cercle.pbm avec un bloc note.
3. Amusez-vous à modifier les valeurs.
4. Expliquer le protocole d'encodage dans le format .pbm.

### 306 .pgm

1. Ouvrir le fichier cercle.pgm avec un logiciel type gimp ou XNview
2. Ouvrir le fichier cercle.pgm avec un bloc note.
3. Amusez-vous à modifier les valeurs.
4. Expliquer le protocole d'encodage dans le format .pgm.

### 307 .ppm

1. Ouvrir le fichier image.ppm avec un logiciel type gimp ou XNview
2. Ouvrir le fichier image.ppm avec un bloc note.
3. Amusez-vous à modifier les valeurs.
4. Expliquer le protocole ppm.

### 308 Limites et améliorations

Expliquer les limites de ces trois protocoles d'encodage et proposer des améliorations.

# Images numériques

## Prendre en main la bibliothèque PIL

### 309 Dans l'interpréteur d'Edupython

- Ouvrir un nouveau fichier python avec **Edupython** que vous appellerez `depart.py`. **Sauvegarder** le dans votre dossier personnel, puis **exécuter** le.
- Dans l'interpréteur d'Edupython taper :
 

```
>>> from PIL import Image
>>> im = Image.new("RGB", (600,300), "grey")
>>> type(im)
>>> im.save("trace.png", "PNG")
```
- A quoi sert la fonction `Image.new` et comment fonctionne-t-elle ? On n'hésitera pas à regarder ce qu'il se passe dans le dossier contenant le fichier `depart.py`.
- A quoi sert la méthode `.save` et comment fonctionne-t-elle ?

### 310 Toujours dans l'interpréteur d'Edupython

- Dans l'interpréteur python taper :
 

```
>>> im.putpixel((10,10), (0,0,0))
>>> im.save("trace.png", "PNG")
```
- A quoi sert la méthode `.putpixel` et comment fonctionne-t-elle ?
- Pourquoi `.save` est-il indispensable ?

## Création d'images

### 311 Premier programme

- Dans le bloc note d'Edupython, écrire un programme qui créera une image toute jaune de taille 800 par 800 pixels. Cette image devra s'appeler `premiere_image.png`
- Modifier le programme pour que l'image contienne un pixel noir, un bleu, un rouge, un vert et un blanc.

3. Modifier votre programme pour que votre image contienne un pixel de couleur "skyblue".

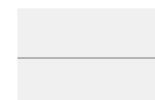
4. Comparer le résultat si on avait décidé d'appeler notre image `premiere_image.jpg`

Expliquer pourquoi les résultats sont différents.

Pour les exercices ci-dessous, on partira d'une image carrée de taille 800 par 800 dont le fond est gris.

### 312 Tracer un segment de droite.

Vous devez obtenir un résultat comme ci-contre :



**313** Tracer un segment de droite. Vous devez obtenir un résultat comme ci-dessous :



**314** Tracer un segment de droite. Vous devez obtenir un résultat comme ci-dessous :



**315** Écrire un programme permettant d'obtenir une "ligne en pointillée" comme ci-dessous :



**316** Écrire un programme permettant d'obtenir une "ligne bicolore" comme ci-dessous :



**317** Dessiner les drapeaux suivants :



# Traitement d'images

## Quelques commandes en plus

### 318 Fichiers attachés :

- fichier retouche.py :
- La Joconde en couleur : jpg , png
- La Joconde en couleur 1024 pixels : jpg
- La Joconde en noir et blanc : jpg
- papillon : jpg
- La Joconde en niveau de gris 10 par 10 : jpg
- canyon Noir et blanc : jpg
- canyon couleurs : jpg

1. Dans votre espace personnel, créer un dossier contenant toutes les images.
2. Ouvrir le fichier retouche.py avec **Edupython** et **enregistrer** le dans le même dossier que celui qui contient les quatre images.
3. Après avoir **exécuter** le fichier retouche.py, déterminer ce que fait la commande .open
4. Précisez le type de img ?

### 319

Dans cet exercice nous allons travailler sur l'image papillon.png qui est en noir et blanc.

*Préalable* : Modifier le programme retouche.py de manière à importer le papillon dans une ImageFile que nous appelerons pap.

**Exécuter** le programme retouche.py.

Grâce à l'interpréteur, déterminer ce que font les commandes suivantes :

1. >>> pap.format
2. >>> pap.size
3. >>> pap.show()
4. >>> pap.getpixel((3,7))

5. >>> pap.putpixel((3,7),128)
6. >>> pap2 = pap.copy()
7. >>> pap.mode

## Négatif

### 320 Négatif d'une image en noir et blanc

1. Effectuer le négatif de l'image papillon.png
2. Effectuer le négatif de l'image Joconde\_10.png.
3. Effectuer le négatif de l'image Joconde\_384.png

### 321 Comprendre les images couleurs

1. importer Joconde\_384.jpg dans une Imagefile que nous appellerons joco (en dehors d'une fonction).
2. Exécuter le fichier.
3. Tester et comprendre les commandes suivantes dans l'interpréteur :
 

```
>>> valeurs = joco.getpixel((3,7))
>>> valeurs
>>> valeurs[0] = 35
>>> valeurs
>>> (R,G,B) = joco.getpixel((3,7))
>>> G
```
4. Comment fonctionne joco.getpixel ?

### 322 Négatif d'une image en couleurs

Effectuer le négatif de l'image Joconde\_384.jpg

## Conversion couleur - noir et blanc

### 323 Transformer une image couleur en noir et blanc

- Traiter Joconde\_384.jpg en remplaçant, pixel par pixel, chacune des valeurs des composantes des couleurs par la moyenne des trois.
- Traiter Joconde\_384.jpg, toujours pixel par pixel, en remplaçant chaque composante des couleurs par la valeur commune :

$$0,21 \times R + 0,71 \times G + 0,07 \times B$$

- Répéter la même opération mais en prenant cette fois ci :

$$0,299 \times R + 0,587 \times G + 0,114 \times B$$

## Miroir

### 324 miroir

- Effectuer le miroir de l'image papillon.png
- Effectuer le miroir de l'image Joconde\_384.png
- Effectuer le miroir de l'image Joconde\_384.jpg

## Images non carrées

### 325 Images non carrées

- Effectuer le négatif de canyon.png et canyon.jpg
- Effectuer le miroir de canyon.png et canyon.jpg

## Autres retouches possibles

### 326 Noir ou blanc

- Écrire une fonction permettant de transformer une image en noir et blanc en noir ou blanc.
- Écrire une fonction permettant de transformer une image couleur en noir ou blanc.

### 327 Filtre

- Programmer les trois fonctions filtres, rouge, vert, bleu.

- Transformer une image couleur en cyan et rouge.

### 328 Bruitage

- Écrire une fonction permettant de brouter (donner comme valeur blanc ou noir)  $n$  pixels d'une image noir et blanc.
- Écrire une fonction permettant de brouter (donner comme valeur blanc ou noir)  $n$  pixels d'une image couleur.

### 329 Algorithme de la médiane

- Écrire une fonction qui remplace chaque pixel d'une image noir et blanc par la médiane des pixels environnants.
- A quoi cela peut il servir ?
- Effectuer la même opération mais sur une image couleur.

### 330 Algorithme de la moyenne

- Écrire une fonction qui remplace chaque pixel d'une image noir et blanc par la moyenne des pixels environnants.
- A quoi cela peut il servir ?
- Effectuer la même opération mais sur une image couleur.

### 331 Utilité ?

- Écrire un programme qui trie les couleurs d'une image noir et blanc par ordre croissant par ligne.
- Adapter votre programme à une image couleur.

## Principe de traitement d'images pixel par pixel

### 332 Structure algorithmique

- Quelle structure algorithmique classique a été massivement utilisée dans cette activité
- Proposer d'autres exemples où cette structure peut être retrouvée

# Tramage

## Fichiers attachés :

- paysage.jpg : 
- Turing.jpg : 

### 333 tramage par diffusion de l'erreur

L'objectif est de transformer une image en niveaux de gris en une image n'utilisant que du noir et du blanc. Typiquement, ce type de transformation est nécessaire lorsqu'on veut par exemple imprimer sur une imprimante noir et blanc une image en niveaux de gris.

1. Proposer une première solution à ce problème et la programmer. L'image obtenue est-elle satisfaisante ?
2. On va utiliser une méthode de tramage par diffusion de l'erreur pour obtenir une image de meilleur qualité :

**Principe :** l'erreur de quantification pour le traitement d'un pixel est partiellement distribuée sur les pixels voisins non encore traités.

#### Algorithme :

- 1) on choisit un seuil.
- 2) on parcourt le tableau de pixels de gauche à droite et de haut en bas :

pour chaque pixel de l'ancienne image :

- si la valeur  $p$  du pixel est inférieure au seuil :
  - on met la valeur du pixel de la nouvelle image à 0. L'erreur commise est  $p$ .
  - on ajoute au pixel à droite de l'ancienne image la moitié de l'erreur commise (sans dépasser 255) ;
  - on ajoute au pixel en bas de l'ancienne image le quart de l'erreur commise (sans dépasser 255) ;
  - on ajoute au pixel en bas de l'ancienne image à droite le quart de l'erreur commise (sans dépasser 255).
- si la valeur  $p$  du pixel est supérieure au seuil :
  - on met la valeur du pixel de la nouvelle image à 255. L'erreur commise est  $255 - p$ .

- on soustrait au pixel à droite de l'ancienne image la moitié de l'erreur commise (en restant positif ou nul) ;
- on soustrait au pixel en bas de l'ancienne image le quart de l'erreur commise (en restant positif ou nul) ;
- on soustrait au pixel en bas de l'ancienne image à droite le quart de l'erreur commise (en restant positif ou nul).

Programmer cet algorithme et le tester avec l'image "Turing.jpg", en utilisant un seuil égal à 127. Le mode "1" sera utilisé pour exporter la matrice vers une image et l'image créée aura le format bmp avec par exemple pour nom "TuringTramage.bmp", pour éviter les approximations de compression liées au format "jpg".

#### 3. Pour aller plus loin :

On peut adapter cet algorithme pour transformer une image en couleurs en une image n'utilisant que 8 couleurs. Il suffit d'appliquer la transformation précédente aux trois composantes RVB de l'image.

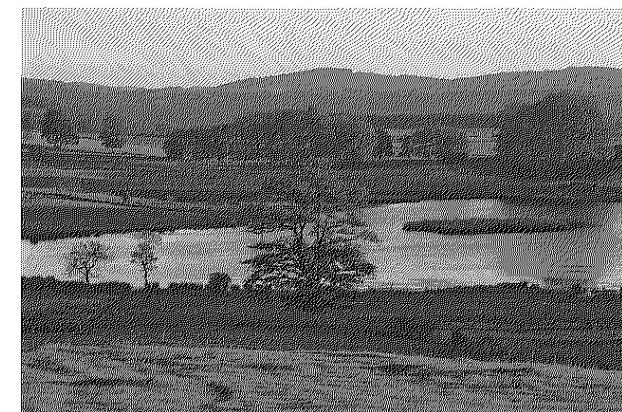
- (a) Pourquoi le principe de l'algorithme précédent appliqué aux trois composantes RVB permet d'obtenir une image n'utilisant que 8 couleurs ?
- (b) Programmer cet algorithme pour transformer une image en couleurs en un image n'utilisant que 8 couleurs et le tester sur l'image "paysage.jpg".



original



niveaux de gris



tramage NB



tramage 8 couleurs

$$\mathbb{E}(\Phi(X)) = \int \Phi(x) q \mathbb{P}^X(x)$$

$$\mathbb{B}(A) = \sum_{i \in I} \mathbb{B}^i(A) \mathbb{B}(B^i)$$

$$\left( \sum_{i \in I} \mathbb{B}^i(A) \mathbb{B}(B^i) \right)_{\text{NB}} = \sum_{i \in I} \mathbb{B}^i(A) \mathbb{B}(B^i) = \mathbb{B}(A)$$

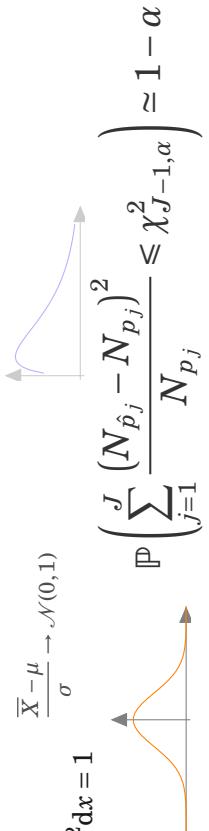
$$\mathbb{B}\left(\sum_{i=1}^n \frac{W^{b_i}}{\sum_{j=1}^n (W^{b_j} - W^{b_i})} \leq \chi_{\sum_{j=1}^n (W^{b_j} - W^{b_i})} \right) = 1 - \alpha$$

# Matrice de convolution (filtre)

Dans la partie précédente, le traitement de chaque pixel était indépendant des pixels voisins. Ici les traitements utiliseront les pixels environnants. Cela donne une puissance de traitement bien supérieur (contour, flou, embossage...)

## Principe d'un filtre type matrice de convolution

Le principe de ce type de traitement, une fois qu'on a laissé l'intitulé de côté, est très simple. Illustrons le sur un exemple d'images en niveau de gris (Je laisserai le soin d'extrapoler aux images RGB). Prenons le cas d'une matrice  $3 \times 3$  de nombres qui contiendra les paramètres du filtre. On va l'appliquer sur un pixel de coordonnées  $(i, j)$ .



|   |   |   |
|---|---|---|
| a | b | c |
| d | e | f |
| g | h | i |

|         | ... | $i - 1$ | i   | $i + 1$ | ... |
|---------|-----|---------|-----|---------|-----|
| ...     | ... | ...     | ... | ...     | ... |
| $j - 1$ | ... | 130     | 104 | 95      | ... |
| $j$     | ... | 34      | 234 | 56      | ... |
| $j + 1$ | ... | 146     | 213 | 21      | ... |
| ...     | ... | ...     | ... | ...     | ... |

On sent venir la suite : le pixel  $(i, j)$  traité aura donc comme valeur de niveau de gris :

$$130a + 104b + 95c + 34d + 234e + 56f + 146g + 213h + 21i$$

Cette valeur peut être normalisée. Le même filtre étant appliqué à tous les pixels de l'image à traiter.

**334** Avant toute programmation quels problèmes peut-on déjà planifier ?

## Premier filtre

**335** .....

Déterminer ce que fait le filtre suivant :

|   |   |   |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

**337** .....

Déterminer ce que fait le filtre suivant :

|    |    |   |
|----|----|---|
| -2 | -1 | 0 |
| -1 | 0  | 1 |
| 0  | 1  | 2 |

**336** *Flou*

Proposer un filtre pour flouter une image :

|     |     |     |
|-----|-----|-----|
| ... | ... | ... |
| ... | ... | ... |
| ... | ... | ... |

**338** *Réhausseur de contraste*

Proposer un filtre permettant de réhausser le contraste d'une image :

|     |     |     |
|-----|-----|-----|
| ... | ... | ... |
| ... | ... | ... |
| ... | ... | ... |

**339** Reprendre l'exercice précédent en modifiant la valeur centrale (4,7,8,9,17...)

# Stéganographie

La stéganographie est l'art de cacher une image dans une autre.

Le principe est très simple. Nous supposerons que nous disposons de deux images en noir et blanc qui ont les mêmes dimensions.

Le niveau de gris d'un pixel de l'image 1 s'écrit en binaire **1011 1100**. Le niveau de gris de ce pixel dans l'image 2 s'écrit en binaire 0110 **0101**.

Pour cacher l'image 2 dans l'image 1, une troisième image de même taille va être créée. Le pixel correspondant va alors prendre la valeur **1011 0101**. Ainsi ce pixel aura le même ton que celui de l'image 1 à  $2^4 = 16$  valeurs près sur 255.

## 340 Nom

- Comment appelle-t-on dans un octet les 4 premiers bits (en partant de la droite)?
- Comment appelle-t-on dans un octet les 4 derniers bits (en partant de la droite)?
- Justifier ces appellations.

## 341 Planification

- Déterminer tous les sous problèmes à résoudre pour mener à bien ce projet.
- Déterminer les spécifications et les prototypes des différentes fonctions nécessaires à la résolution du problème de la stéganographie.
- Télécharger deux photos noir et blanc libre de droit et de même taille.
- C'est parti.

## 342 En couleurs ?

Résoudre le même problème mais pour des images couleurs.

# Traitements en parallèle

## 343 Négatifs d'images en programmation séquentielle

1. Télécharger une image couleur en jpg de votre choix sur internet
2. Écrire une fonction python permettant de faire le négatif de l'image.
3. Combien de temps met votre processeur pour en faire le négatif ?
4. Télécharger une deuxième image et faites le négatif de vos deux images. Combien de temps cela prend-il ?
5. Télécharger plusieurs images et créer une fonction qui va faire le négatif de toutes vos images. Combien de temps cela prend-t-il ?

## 344 Premier processus

Écrire un programme qui fera le négatif de toutes les images dans un processus. Combien de temps le programme met-il ?

## 345 Négatif d'images en programmation parallèle : deux processus

1. Réfléchir à l'implémentation d'un programme permettant le traitement de toutes les images en utilisant deux processus.
2. Programmer votre idée.
3. Déterminer et comparer les temps mis par ce nouveau programme par rapport à votre programme de l'exercice précédent.

## 346 Plusieurs processus

1. Modifier votre programme de manière à pouvoir choisir votre nombre de processus.
2. Après avoir déterminé le temps mis par votre programme dans différentes configurations que peut-on remarquer ?

## 347 A votre avis

1. Un programme comprenant 8 processus est-il 8 fois plus rapide qu'un programme n'en contenant qu'un seul pour notre traitement d'images ?
2. Cela ne sert à rien de mettre 9 processus si notre machine ne contient que deux coeurs. Vrai ou faux ?
3. Avoir deux process sur une machine contenant plusieurs coeurs permet systématiquement d'accélérer le traitement ?

# Images et POO.

## 348 Récupération d'images

1. Télécharger divers images libres de droits (petite taille et grande taille) aux formats PBM, PGM et PPM.
2. Pourquoi on se restreint à ces formations et on ne prend pas des PNG et JPEG ?

## 349 Création de la classe et premiers attributs

1. Créer une classe Image. L'initialisation prendra comme seul paramètre une chaîne de caractère qui sera le nom de l'image. La détermination du type de l'image sera encapsulée dans la classe avec un attribut qui aura pour nom *format*.
2. Rajouter un attribut *size* qui sera un 2-uplet.
3. Rajouter un attribut *mode* qui indiquera si l'image est noir ou blanc, en niveau de gris ou en couleur.
4. Rajouter un attribut *pixels* qui sera un tableau à deux dimensions de taille *size*.

## 350 Méthode copy

1. Créer une méthode *copy()* qui permettra de construire une copie d'une instance d'Image.
2. Créer une méthode *deep\_copy()* qui permettra de faire une copie profonde d'une instance d'Image.

## 351 getter/setter

1. Créer une méthode *getpixel(i,j)* qui permettra de récupérer la valeur du pixel (ou le 3-uplet si l'image est en couleur) en ligne *i* et colonne *j*
2. Créer une méthode *putpixel(i,j)* qui permettra de modifier la valeur du pixel (ou le 3-uplet si l'image est en couleur) en ligne *i* et colonne *j*

## 352 Méthode save

Créer une méthode *save(nom)* qui créera l'image sur le disque.

## 353 Encapsulation

Encapsuler dans des méthodes toutes les fonctions déjà vues permettant la retouche d'images.

## 354 Méthode show

Créer une méthode *show* qui permet d'ouvrir l'image avec un logiciel de lecture d'images

$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\binom{n}{k} p^k (1-p)^{n-k}$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\frac{\bar{X} - \mu}{\sigma} \xrightarrow{\text{distrib. normale}} \mathcal{N}(0,1)$$

$$\mathbb{P}\left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1,\alpha}^2\right) \approx 1 - \alpha$$

**1** Un octet

Byte ne signifie pas octet. L'amalgame est souvent fait et bien aidé par les constructeurs de disques durs qui jouent sur l'ambiguïté. Les anglais sont comme nous ils utilisent octet pour les mots binaires de 8 bits.

Byte est "la plus petite unité « logiquement » adressable par un programme sur un ordinateur".

La longueur d'un byte dépend donc du programme ou de l'ordinateur. Même si bien souvent cette plus petite unité est normé sur un octet (codage ascii, RGB...), il existe certains ordinateurs ou programme qui travaille de base avec une autre unité de base.

Pour aller plus loin : <https://fr.wikipedia.org/wiki/Byte>

**2** A Faire**3** A FAIRE**4** A FAIRE**5** A FAIRE**6** A FAIRE**7** A FAIRE

**8** CF tablette de babylone ! Cf existence de zéro (arrivé tardivement)

**9** A FAIRE**10** A FAIRE**11** A FAIRE

**12** 47 objets se regroupent en 9 paquets et 2 unités, puis les 9 paquets se regroupent en 1 paquet de paquets et 4 paquets.  $47 = 9 \times 5 + 2 = (1 \times 5 + 4) \times 5 + 2 = (1 \times 5^2) + (4 \times 5^1) + (2 \times 5^0)$ .

Donc  $47 = 142$  (en base cinq)

Donc  $944 = 12234$  (en base cinq)

Donc  $289 = ????$  (en base cinq)

**13**  $401302_5 = 12702$ .

**14** A FAIRE**15** A FAIRE**16** A FAIRE**17** A FAIRE**18** A FAIRE**19** A FAIRE**20** A FAIRE**21** A FAIRE

**22** Il faut rajouter des chiffres. Le codage hexadecimal propose a,b,c,d,e et f?

**23** A FAIRE

$$\mathbb{E}(A) = \sum_{i \in I} \mathbb{P}^i(A) \mathbb{E}^i(B)$$

$$\mathbb{E}(\Phi(X)) = \int \Phi(x) q \mathbb{P}^X(x)$$

$$\left( \sum_{i \in I} \mathbb{P}^i(J - B_i) \right)_{i \in I}$$

$$\mathbb{E} \left( \sum_{i \in I} \mathbb{P}^i (J - B_i) \right)_{i \in I} = J - \alpha$$

**24** A FAIRE**25** A FAIRE**26** 2 lettres pour représenter un octet**27** Base 4**28** A faire**29** A faire**30**

1. Dépassement de capacité
2. fds
3. ds
4. Représentation facile ou algorithme facile ?

**31** A faire**32** A faire**33** A faire**34** A faire**35**

Dépassement de capacité.

**36** A faire**37** 0101 0011 et 1010 1110**38**

- 1.
- 2.
- 3.
4. Le principe est respecté.

**39** A faire**40** 0101 0100 et 1010 1111**41**

- 1.
- 2.
- 3.
4. Le principe est respecté.

**42**

- 1.
- 2.
3. Cette représentation permet d'effectuer les opérations arithmétiques usuelles naturellement. Représentation complexe, algorithme simple à implémenter

**43** A faire**44** Python possède les entiers long, mais à un autre type les long dont les tailles est dynamique et qui ne dépendent que de la puissance de la machine. -> pep 237.

$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i} (A) \mathbb{P}(B_i)$$

$$\left( \frac{n}{k} \right) p^k (1-p)^{n-k}$$

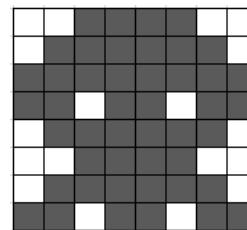
$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\frac{\bar{X} - \mu}{\sigma} \xrightarrow{\mathcal{N}(0,1)}$$

$$\mathbb{P}\left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1, \alpha}^2\right) \approx 1 - \alpha$$

**45** Chiffrement

- chaque pixel noir est remplacé par 0 et chaque pixel blanc est remplacé par 1 ;
- chaque ligne peut donc être vue comme une suite de 8 bits ;
- chaque suite de 8 bits est considéré comme un entier relatif, codé en binaire par la méthode du complément à 2 sur 8 bits ;
- une image correspond donc à une suite de 8 entiers relatifs. C'est cette suite de 8 entiers relatifs qu'ils s'envoient l'un à l'autre.

**46** A faire**47** A FAIRE**48**

- $42 = 0010\ 1010$  et  $-43 = 1101\ 0101$  (on représente  $-43 + 2^8 = 212 = 2^7 + 2^6 + 2^4 + 2^2 + 2^0$ ).  
On remarque que la représentation binaire de  $42 + (-43)$  est  $1111\ 1111$  soit  $-1$  (normal !).
- l'addition binaire de  $m$  et  $m'$  donne  $1111\ 1111$  soit  $-1$ .  
On a donc :  $n + n' = -1$
- Ainsi  $-n = n' + 1$ . Donc la représentation binaire de  $-n$  est égale à la représentation binaire de  $n'$  à laquelle on ajoute 1.

**49** A faire**50** A faire**51** A faire**52** A faire**53** A faire**54** A faire**55**

- Chiffre : 1 :A, 2 :B, 3 :C  
On obtient Bonjour
- 1 :A, 2 :B, 3 :C
- Même chiffre mais avec 27 : espace et 28 :!  
Exemple pour illustrer les caractères non visibles et qu'il y a plus de 26 lettres.
- Il y en a moins de 128 : d'où une table ASCII sur 7 bits

**56**

- Question de place mémoire : Un bit c'était un bit.  
Le byte n'est pas toujours égal à un octet.
- ponctuation/majuscule/minuscule/caractère de contrôle.
- lié à l'implémentation : en mettant un écart de 32 seul le 6ème bit change entre les majuscules et les minuscules.

**57** a faire**58**

- a faire

2. (a) Détection d'erreur dans les transmissions  
 (b) On ne sait pas quel bit est faux. Si deux bits sont faux ça passe.

**59**

1. A faire
2. A faire
3. Il est différent !
4. Il est différent !
5. Non trop de caractères.
6. Place perdu en machine si on reste en europe par exemple.

Vers la norme UTF8 : encodage flottant 1 2 4 octets selon la rareté du caractère

**60** A faire**61** A faire**62**

1. Il correspond à l'euro. ord donne le point de code au regard de l'unicode
2. A faire soit 2 octets.
3. Encodage UTF-8 : 0xE2 0x82 0xAC  
 Encodage UTF-16 : 0x20AC  
 Encodage UTF-32 : 0x000020AC
- 4.

**63**

1. A faire
2. point de code : 233, représentation en mémoire : 0xC3 0xA9

3. Sinon chaque caractère serait codé sur 4 octets!!!  
 4. Lorsqu'une machine lit un texte codé en UTF8, elle regarde le premier bit. si c'est 0, elle "décode" le caractère selon la valeur obtenue avec les 7 bits restants de l'octet selon l'encodage ascii. Si c'est 1, elle va chercher le les octets qui suivent pour décoder selon un protocole beaucoup plus compliqué : voir exo suivant ! Unicode n'est pas un encodage comme ASCII. C'est juste un catalogue numéroté de caractères et un "paquet" complet de diverses spécifications  
 Les encodages correspondants s'appellent UTF-8 (très répandu et conseillé par défaut), UTF-16, UTF-32 et diffèrent par le degré de compression. Il est plus exact de les citer eux avec ASCII et ISO-8859-1, sinon on compare des pommes et des oranges.

**64** a faire**65** A faire

**66** Windows et surtout les applis qui tournent sur windows qu'il faut toutes reprogrammer... ça prend du temps !

**67** A faire**68** A faire**69** A faire**70** A faire**71** A faire**72** A faire

$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\frac{\bar{X} - \mu}{\sigma} \xrightarrow{\mathcal{N}(0,1)}$$

$$\mathbb{P}\left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1,\alpha}^2\right) \approx 1 - \alpha$$

**73** A faire**74** A faire**75** A faire**76** A faire**77** A faire**78** A faire**79** A faire

**80** Exemple à ne pas faire... même si le et est prioritaire sur le ou (convention), on écrit : a ou (b et c)

**81** A faire**82** A faire**83** A faire**84**

1. A ou B et  $\overline{\overline{A}}$  et  $\overline{\overline{B}}$  ont la même table de vérité. Ils définissent la même expression booléenne.
2. Toutes fonction booléennes ou fonction booléenne peut s'écrire en n'utilisant que deux opérateurs booléens : *non* et *et*.

**85**

| a | b | a xor b |
|---|---|---------|
| 0 | 0 | 0       |
| 1 | 0 | 1       |
| 0 | 1 | 1       |
| 1 | 1 | 0       |

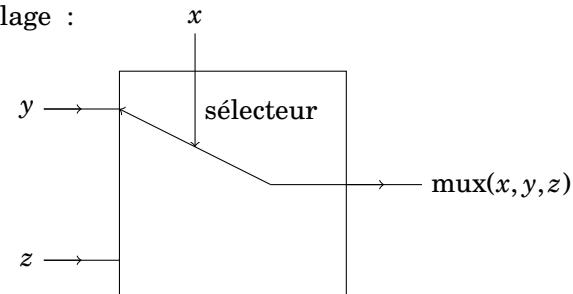
1.

**2.** A faire

**86** La fonction booléenne multiplexeur noté mux est la fonction de trois variables booléennes définie par :

$$\begin{cases} \text{mux}(x, y, z) = y \text{ si } x = 0 \\ \text{mux}(x, y, z) = z \text{ si } x = 1 \end{cases}$$

L'état de la variable *x* permet de faire le choix de *y* ou de *z* à la manière d'un aiguillage :

**87** si et seulement si**88** implique**89** A faire**90** A faire

**91** Python évalue séquentielle une expression booléenne. Ainsi : 'A AND B' Python regarde d'abord A. Si A est True, Python évalue ensuite B ?

$$\mathbb{E}(X) = \sum_{x \in \Omega} \mathbb{P}(B_x) x = \int_{\Omega} x \mathbb{P}(dx)$$



92 A faire

93 A faire

94 A faire

95 A faire

96 A faire

97 A faire

98 A faire

99 A faire

100 T["age"] et T[s]

101 A faire

102 A faire

103 A faire

104 A faire

105 A faire

106 A faire

107 A faire

108 A faire

109 A faire

110

```

1. def plusGrandEcart(tab) :
2     return max(tab) - min(tab)

```

Cette fonction a un coût linéaire.

```

1. def plusPetitEcart(tab) :
2     S = abs(tab[1] - tab[0])
3     n = len(tab)
4     for i in range(n) :
5         for j in range(n) :
6             if i != j and abs(tab[i] - tab[j]) < S :
7                 S = abs(tab[i] - tab[j])
8
return S

```

Cette fonction a un coût quadratique.

111

```

1. def ConvertVersBase10(tab, base) :
2     n = len(tab)
3     S = 0
4     for i, x in enumerate(tab) :
5         S = S + x * base**(n-1-i)
6
return S

```

112 A faire

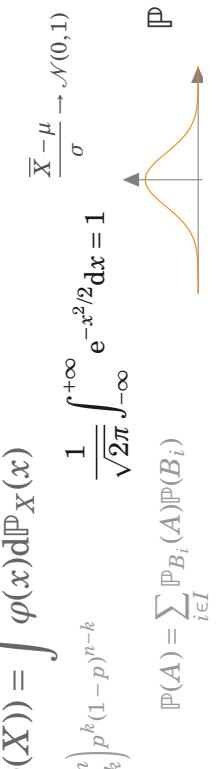
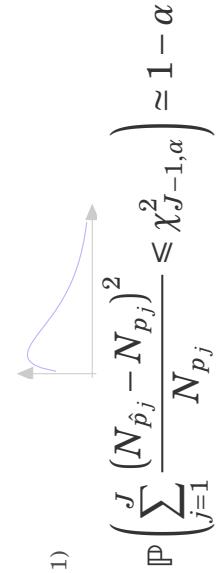
**113** A faire**114** A faire**115** A faire**116** A faire**117**

```
1. def RechercheValeur(dico, cle) :
2     if cle in dico :
3         return dico[cle]
```

```
2. def RechercheCle(dico, val) :
3     for cle in dico :
4         if dico[cle] == valeur :
5             return cle
```

**118** Même principe que l'exercice précédent.**119** Même principe que les deux exercices précédents.**120**

```
1 def CreerDictionnaire(listeCles, listeValeurs) :
2     dico = {}
3     n = len(listeCles)
4     for i in range(n) :
5         dico[listeCles[i]] = listeValeurs[i]
6     return dico
```

**121**

```
1 with open("communes.csv", "r", encoding="utf-8") as fichier :
2     ligne = fichier.readline()
3     descripteurs = ligne.strip().split(" ; ")
4     ligne = fichier.readline()
5     table = []
6     while ligne != "" :
7         objet = ligne.strip().split(" ; ")
8         table.append(CreerDictionnaire(descripteurs, objet))
9         ligne.readline()
```

**Remarque :** une boucle Pour au lieu d'une boucle Tant que fonctionne aussi :

```
1 with open("communes.csv", "r", encoding="utf-8") as fichier :
2     ligne = fichier.readline()
3     descripteurs = ligne.strip().split(" ; ")
4     table = []
5     for ligne in fichier :
6         objet = ligne.strip().split(" ; ")
7         table.append(CreerDictionnaire(descripteurs, objet))
```

**122** Attention aux données manquantes dans le fichier CSV : certaines communes n'ont pas de coordonnées géographiques. il faut donc penser à traiter ce cas pour éviter une erreur.

Pour le centre de la France, il faut exclure les communes de Corse.

**123** A faire

**124** Cet exercice à faire seul ou en binôme, permet d'introduire le mot de vocabulaire objet.

**125** Cet exercice, permet de présenter le terme de collection d'objet et pose les préalables à la recherche d'un objet dans une collection.

**126** Commentaires : Ici les élèves où les groupes d'élèves sont invités à consulter le plus possible de tweets écrit par la classe. On peut introduire le vocabulaire de donnée et faire émerger le terme générique de descripteurs d'une donnée (titre, auteur, date de parution, nombre de pages...). Une fois cette formalisation faite, on peut alors extraire tous les descripteurs auxquels, globalement, les élèves ont pensé même si ce n'est qu'une seule fois. On peut alors les inviter, pour prolonger, à en trouver d'autres. Une fois cela fait, on s'interroge aux types d'une donnée et sortir les trois classifications : texte, nombre, booléan. En s'intéressant maintenant aux données de type nombre, on peut s'intéresser aux différents formats permettant d'écrire ces nombres (date, taille, entier, décimal ...) On termine en se posant la question des données permettant à coup sûr d'identifier le livre : on peut parler du numéro ISBN (sous réserve que nous n'ayons pas deux fois le même livre dans la collection). On peut évoquer à ce moment là la notion de données personnelles en précisant que l'on ne réserve cette notion qu'aux personnes physiques.

### 127 Exercice d'application

**128** Exercice d'application qui permet d'aborder ce qu'est une donnée personnelle : (numéro sécurité sociale...)

**129** Le pad est un bloc note collaboratif, tous les élèves peuvent s'y connecter en même temps. Cela peut se faire avec les téléphones des élèves si l'on a pas d'ordinateur. La majorité des ENT intègrent un tel pad. Si un pad n'est pas disponible ou si l'on n'est pas à l'aise avec cet outil, on peut récupérer en fin de séance tous les tweets écrits des élèves et les intégrer en dehors de la classe dans un unique fichier .txt ou .odt pour la séance suivante. Le faire faire individuellement en classe par les élèves prend un certain temps.

**130** Naturellement, une majorité d'élèves (tous ?) vont s'orienter vers le tableau. La première ligne contiendra les descripteurs et chaque

autre ligne correspondra à un objet. Quelques uns auront un fichier symétrique : les descripteurs seront dans la première colonne et chaque colonne correspondra à un objet. Ils ont naturellement créé un fichier .csv !

**131** Une recherche rapide sur un moteur de recherche permet de trouver rapidement où télécharger une archive des données que possède GOOGLE et FACEBOOK. La construction de l'archive peut cependant prendre plusieurs jours. L'unité est l'octet, ou le kilo octet, ou le méga-octet... mais bien souvent c'est du giga-octet.

**132** Commentaires : Même si la visualisation est naturelle avec un tableau, un fichier .csv n'est rien d'autre qu'un fichier texte. On y trouve un objet par ligne, la première ligne correspondant aux descripteurs. Les données sont séparées par un caractère séparateur (, ; : tabulation espace).

**133** A faire

**134** A faire

**135** A faire

**136** A faire

**137** A faire

**138** A faire

**139** A faire

**140** chargement du fichier csv dans un tableau

1. Ok

2. Ok

```

1.3. descripteurs = ["nom_région",
2   "chef-lieu_région", "numero_département",
3   "nom_département", "préfecture", "nom_commune",
4   "codes_postaux", "code_insee", "latitude",
5   "longitude"]

```

C'est une variable de type liste dont les éléments sont des chaînes de caractères.

```

1.4. {"chef-lieu_région" : "Lyon",
2   "code_insee" : "1024",
3   "codes_postaux" : "1340",
4   "latitude" : "46.283333",
5   "longitude" : "5.166667",
6   "nom_commune" : "Attignat",
7   "nom_département" : "Ain",
8   "nom_région" : "Rhône-Alpes",
9   "numéro_département" : "1",
10  "préfecture" : "Bourg-en-Bresse"}

```

Il s'agit d'une variable de type dictionnaire. Les clés de ce dictionnaire sont les descripteurs.

5. 1) On ouvre le fichier communes.csv
- 2) On le charge dans une variable appelée tableCSV
- 3) On initialise un tableau vide
- 4) On ajoute dans le tableau les objets de la table communes.csv en parcourant la variable tableCSV
- 5) On crée la liste des descripteurs.

141

1. Ok
2. On parcourt les objets du tableau tab\_communes. Si la valeur correspondant au descripteur "numéro\_département" de l'objet est égale à '25', alors on augmente la valeur de nb de 1.

La variable nb contient donc à la fin le nombre de communes du Doubs.

3. nb = 595

```

1 nb = 0
2 for objet in tab_communes :
3   if objet["numéro_département"] == "25" :
4     nb = nb + 1

```

```

1.4. nb35 = 0
2 nb29 = 0
3 nb56 = 0
4 nb22 = 0
5 for objet in tab_communes :
6   if objet["numéro_département"] == "35" :
7     nb35 = nb35 + 1
8   elif objet["numéro_département"] == "29" :
9     nb29 = nb29 + 1
10  elif objet["numéro_département"] == "56" :
11    nb56 = nb56 + 1
12  elif objet["numéro_département"] == "22" :
13    nb22 = nb22 + 1

```

nb22 = 373, nb29 = 284, nb35 = 355 et nb56 = 261)

```

15. lat_max = 0
1 com_max = ""
2 for objet in tab_communes :
3   if objet["latitude"] != "" and float(objet["latitude"]) >
4     lat_max = float(objet["latitude"])
5   com_max = objet["nom_commune"]
6   long_max = float(objet["longitude"])
7

```

com\_max = "Bray-Dunes"

Pour visualiser

```

1 import webbrowser
2 zoom="10"
3 webbrowser.open("https://www.openstreetmap.org/#map=" + zoom
4 lat_max )+ "/" + str ( long_max))

```

## 6. Centre de la France

```

1 S_lat = 0
2 S_long = 0
3 nb = 0
4 for objet in tab_communes :
5     if objet["latitude"] != "" :
6         S_lat = S_lat + float(objet["latitude"])
7         S_long = S_long + float(objet["longitude"])
8         nb = nb + 1

```

$$S_{\text{lat}}/nb = 47.03833497426952$$

$$S_{\text{long}}/nb = 2.666824438227952$$

On peut également finir avec la bibliothèque folium

142

```

1. import csv
2
3 def charger(chemin_fichier) :
4     with open(chemin_fichier,"r",encoding = "utf-8") as fichCSV :
5         tableCSV = csv.DictReader(fichCSV,delimiter=";")
6         tab = []
7         for objet in tableCSV :
8             tab.append(objet)
9         descripteurs = tableCSV.fieldnames
10        return (descripteurs,tab)
11
12 desc_cinema, tab_cinema = charger("fichiers CSV/cinema.csv")
13 desc_villes, tab_villes = charger("fichiers CSV/villes.csv")

```

```

12. # nom cinema + coordonnées villes
13 for objet_cinema in tab_cinema :
14     for objet_villes in tab_villes :
15         if objet_cinema["ville"] == objet_villes["ville"] :
16             print(objet_cinema["cinema"], objet_villes["latitude"],
17                   objet_villes["longitude"])

```

```

13. # ville + nb salles
14 # on récupère le nom des villes dans une liste
15 liste_villes = []
16 for objet_villes in tab_villes :
17     if objet_villes["ville"] not in liste_villes :
18         liste_villes.append(objet_villes["ville"])
19
20 # parcours de la table cinema pour récupérer le nombre de salles
21 for ville in liste_villes :
22     S = 0 # on initialise le nombre de salles à 0
23     for objet_cinema in tab_cinema :
24         if objet_cinema["ville"] == ville :
25             S = S + int(objet_cinema["nombre de salles"])
26     print(ville, S)

```

143 A faire

144 A faire

145 A faire

146 A faire

147 A faire

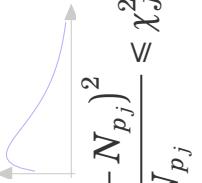
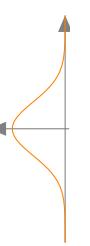
$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$${n \choose k} p^k (1-p)^{n-k}$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\frac{\bar{X} - \mu}{\sigma} \xrightarrow{\text{ } \mathcal{N}(0,1)}$$

$$\mathbb{P}\left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1, \alpha}^2\right) \approx 1 - \alpha$$



**148** A faire

```
left(45)
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90+45)
forward(sqrt(100**2+100**2))
left(90+45)
forward(100)
left(90+45)
forward(sqrt(100**2+100**2))
```

**149** A faire**150** A faire**151**

```
forward (100)
left (90)
```

**152** A faire**153** A faire

```
left(45)
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
```

**154** A faire**155** A faire**156** A faire**157**

```
forward(100)
left(60)
forward(100)
left(60)
forward(100)
left(60)
forward(100)
left(60)
forward(100)
left(60)
forward(100)
left(60)
```

**158** A faire

**159**

```
angle = 360/20
longueur = 50
for k in range(20) :
    forward(longueur)
    left(angle)
```

ou encore :

```
def polygone(n,cote) :
    """n represente le nombre de coté
    cote est la longueur du côté"""
    angle = 360/n
    for k in range(n) :
        forward(cote)
        left(angle)

def carre(cote) :
    polygone(4,cote)
def decagone(cote) :
    polygone(10,cote)
```

**160**

```
for k in range(360) :
    forward(1)
    left(1)
```

**161**

1. Solution 1

```
color("red")
for k in range(10) :
    forward(100)
    left(90)
```

```
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
up()
forward(15+100)
down()
```

2. Solution 2

```
def carre() :
    forward(100)
    left(90)
    forward(100)
    left(90)
    forward(100)
    left(90)
    forward(100)
    left(90)
```

```
color("red")
for k in range(10) :
    carre()
    up()
    forward(15+100)
    down()
```

3. Solution 3 :

```
def carre() :
    for k in range(4) :
        forward(100)
        left(90)
```

```
color("red")
for k in range(10) :
```

$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$${n \choose k} p^k (1-p)^{n-k}$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\frac{\bar{X} - \mu}{\sigma} \xrightarrow{\text{distrib. normale}} \mathcal{N}(0,1)$$

$$\mathbb{P}\left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1, \alpha}^2\right) \approx 1 - \alpha$$

162

```
carre()
up()
forward(15+100)
down()

def carre(cote) :
    for k in range(4) :
        forward(cote)
        left(90)
```

```
for k in range(5) :
    carre(100)
    up()
    forward(100+15)
    down()
    carre(50)
    up()
    forward(50+15)
    down()
```

163

```
for k in range(1,10) :
    carre(10*k)
    up()
    forward(10*k+15)
    down()
```

164

```
carre()
up()
forward(15+100)
down()
```

```
for i in range(10) :
    for j in range(10) :
        carre(20)
        up()
        forward(20+5)
        down()

        up()
        backward(250)
        right(90)
        forward(20+5)
        left(90)
        down()
```

165

```
for i in range (10) :
    for j in range (10) :
        if i == 3 and j == 7 :
            color ( " red " )
        else :
            color ( " black " )
        carre (30)
        up ()
        forward (30+5)
        down ()

        up ()
        backward (350)
        right (90)
        forward (30+5)
        left (90)
        down ()
```

166 A faire

167 A faire

168 A faire

169 A faire

170 A faire

171 A faire

172 A faire

173 A faire

174 A faire

175 A faire

176 A faire

177 A faire

178 à faire

150 A-fine

180

181

102

183 A faire

184 A faire

185 A faire

186 A faire

187

1. Deux variable début et fin permettant de savoir les indices entre lesquels travailler.
  2.  $\text{fin} < \text{début}$

```
13. def RechercheDichotomique(tab, element) :
 14.     début = 0
 15.     fin = len(tab) - 1
 16.     while début <= fin :
 17.         milieu = (début+fin)//2
 18.         if tab[milieu] == element :
 19.             return True
 20.         elif tab[milieu] < element :
 21.             début = milieu + 1
 22.         else :
 23.             fin = milieu - 1
 24.     return False
```

188 A faire

189 A faire

190 A faire

191

1. Il manque le cas où debut > fin. Dans ce cas, on renvoie False.

$$\Phi(X) = \begin{cases} \Phi(x) & x \in \mathbb{R} \\ 0 & \text{otherwise} \end{cases}$$

```

1.2. def rechercheDicho(tab, element) :
    """ rechercheDicho (list, int) -> bool """
    ...
    if debut > fin :
        return False
    else :
        milieu = (debut + fin)//2
        if tab[milieu] == element :
            return True
        elif tab[milieu] < element :
            return rechercheDichoRec(tab, element, milieu + 1, fin)
        else :
            return rechercheDichoRec(tab, element, 0, n - 1)

    n = len (tab )
    return rechercheDichoRec(tab, element, 0, n - 1)

1.3. assert rechercheDicho(tab, 0) == True
      assert rechercheDicho(tab, 999**2-1) == False
      assert rechercheDicho(tab, 500**2) == True
      assert rechercheDicho(tab, 300**2+1) == False
      assert rechercheDicho(tab, 998**2+1) == False
      assert rechercheDicho(tab, 250**2) == True

```

**192** A faire**193** A faire**194** A faire**195** Sujet du CAPES epreuve 2 2020.

Objectif : dichotomie et trichotomie

**196**

La question 2c est difficile pour les élèves. En particulier, le terme adapter n'est pas suffisamment précis. Proposer, en les justifiant, une ou plusieurs aides que vous pourriez proposer aux élèves afin de leur permettre de répondre à cette question. (Répondre en 6 lignes maximum.)

En transposant les notions de variantes et d'invariants, vues pour l'algorithme de recherche dichotomique dans un tableau trié en classe de 1ère NSI, donner à vos élèves un schéma de la preuve de correction de l'algorithme par trichotomie. (Répondre en 10 lignes maximum.)

**197**

Lorsque l'on tri un jeu de tarot, on fait de l'insertion. Spontanément la majorité des élèves évoque l'idée du tri par sélection quelquefois d'autre tri sorte : fusion, à bulles.

**198** A faire**199** A faire**200** -sort(-tab)**201** A faire**202** Enoncé léger pour favoriser la créativité et voir ce qui sort.**203** A faire**204** A faire**205** A faire

**206** A faire**207** A faire**208** A faire**209 Algorithme du tri par sélection**

```

1 Pour i allant de 0 à n - 2 :
2   # Recherche de l indice de l élément minimal
3   # dans tab[i :n]
4   i_mini = i
5   Pour j allant de i + 1 à n - 1 :
6     Si tab[i_mini] > tab[j] :
7       i_mini = j
8   fin de Pour
9   # échanger tab[i_mini] et tab[i]
10  temp = tab[i]
11  tab[i] = tab[i_mini]
12  tab[i_mini] = temp
13 Fin de Pour

```

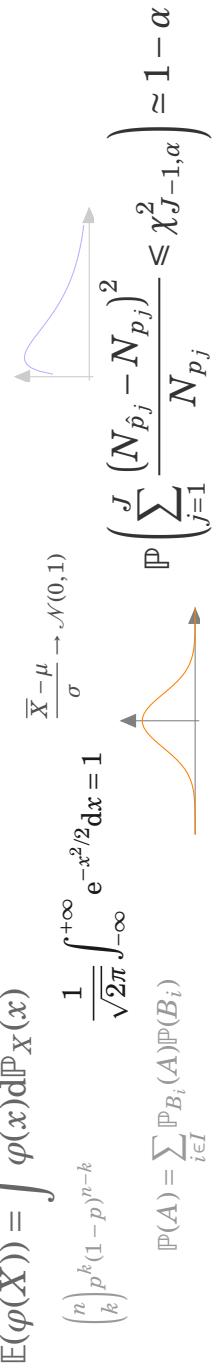
**215** A faire**216** A faire**217** A faire**218 Algorithme du tri par insertion**

```

1 Pour i allant de 1 à n - 1 :
2   mise en mémoire de tab[i] dans temp
3   temp = tab[i]
4   # décalage vers la droite les éléments plus grands
5   # que temp
6   j = i
7   Tant que j > 0 et tab[j - 1] > temp :
8     tab[j] = tab[j - 1]
9     j = j - 1
10  fin de Tant que
11  # positionner temp dans la place laissée par le
12  # décalage
13  tab[j] = temp
14 Fin de Pour

```

**210** A faire**211** Enoncé léger pour favoriser la créativité et voir ce qui sort.**212** A faire**213****214** A faire**219** A faire**220** Pas de teste d'égalité entre flottant. En pratique on teste la différence plus petite qu'un seuil. Mais il y a toujours possibilité de trouver un jeu de donnée pour faire renvoyer une réponse fausse au programme.**221** A faire



**222** Le but de l'exercice est d'aller chercher le arange de la bibliothèque numpy

**223** A faire

**224** A faire

**225** En lecture, une fonction recherche des variables dans cet ordre :

1) dans la fonction elle-même 2) dans la fonction englobante 3) dans l'environnement global

**226** L'instruction `x = x + ...` dans la fonction crée une nouvelle variable `x` à portée locale. En revanche, la méthode `append` crée un effet de bord.

**227** A faire

**228**

```
1. def puissanceRec(x, n) :
    if n==0 :
        return 1
    else :
        return x * puissanceRec(x, n - 1)
```

```
1.2. def puissanceIter(x, n) :
    S = 1
    for i in range(n) :
        S = S * x
    return S
```

**229**

```
1. def sommeDigitale(n) :
    if n<10 :
        return n
    else :
        return (n%10) + sommeDigitale(n//10)
```

```
12. def racineDigitale(n) :
    if n < 10 :
        return n
    else :
        return racineDigitale(sommeDigitale(n))
```

**230**

```
1. def effectif(tab, element) :
    if tab == [] :
        return 0
    elif tab[0] == element :
        return 1 + effectif(tab[1 :], element)
    else :
        return effectif(tab[1 :], element)
```

```
1. def maximum(tab) :
    if len(tab) == 1 :
        return tab[0]
    else :
        return max(tab[0], maximum(tab[1 :]))
```

**Inconvénient** : mauvaise complexité spatiale à cause de `tab[1 :]`.

```
1. def maximum(tab) :
2.
3.     def maximumRec(tab, i) :
```

```

4     if len(tab) == i+1 :
5         return tab[i]
6     else :
7         return max(tab[i], maximumRec(tab, i+1))
8
9     return maximumRec(tab, 0)

```

**231** Pour la question 4., on peut programmer une fonction (réursive) qui compte le nombre d'appels récursifs de la fonction FiboRec. Pour 100, on trouve environ  $1,15 \times 10^{21}$  appels récursifs.

**232** A faire

**233** A faire

**234** A faire

**235** A faire

**236** A faire

**237** A faire

**238** A faire

**239** A faire

**240** A faire

**241** A faire

**242** A faire

**243** A faire

**244** A faire

**245** A faire

**246** A Faire

**247** A Faire

**248** A Faire

**249** A faire

**250** Félicitations, vous avez créez votre première classe !

**251** Félicitations, vous avez créez votre premier objet. A est une instance de la classe Point.

**252**

1. Félicitations, vous avez créeé votre premier attribut.
2. A faire
3. A faire
4. A faire
5. A faire
6. Pouvoir rentrer :

`1 B = Point(2,5)`

... Encapsulation

7. Le type de self.x et de self.y : entier ou flottant... encapsulation

**253** A faire

**254** A faire

**255**

1. Une méthode ne dépend que de l'objet et peut donc être encapsulée dans la class
2. La distance à l'origine.

**256**

1. `A.distance_origine()`
2. `sqrt` is not defined. Il faut encapsuler l'import.

```
1 class Point(object) :
2     "point geometrique"
3     import math as mt
4     def __init__(self,x = 0,y = 0) :
5         self.x = x
6         self.y = y
7     def distance_origine(self) :
8         return mt.sqrt(self.x**2+self.y**2)
9     def distance(self,A) :
10        return mt.sqrt((self.x-A.x)**2+(self.y-A.y)**2)
```

**257** A faire

**258** il peut être très pratique de sécuriser certaines données de notre objet, par exemple faire en sorte qu'un attribut de notre objet ne soit pas modifiable, ou alors mettre à jour un attribut dès qu'un autre attribut est modifié. Les cas sont multiples et c'est très utile de pouvoir contrôler l'accès en lecture ou en écriture sur certains attributs de notre objet.

**259** A faire

**260** A faire

**261** A faire

**262** A faire

**263** A faire

**264** A faire

**265** A faire

**266** A faire

**267** A faire

**268** A faire

**269** A faire

**270** A faire

$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\binom{n}{k} p^k (1-p)^{n-k}$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\frac{\bar{X} - \mu}{\sigma} \rightarrow \mathcal{N}(0,1)$$

$$\mathbb{P}\left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1,\alpha}^2\right) \approx 1 - \alpha$$

**271** A faire

**272**

1.

Charge la valeur stockée en mémoire à l'adresse 78 dans le registre R0

2.

Place dans le registre R1 la valeur immédiate 25

3.

Range à l'adresse mémoire 131 le contenu du registre R2

4.

Additionne le contenu du registre R1 et la valeur immédiate 37 et place le résultat dans le registre R0

5.

Soustrait le contenu du registre R2 au contenu du registre R1 et place le résultat dans le registre R0

6.

Compare le contenu du registre R0 et la valeur immédiate 18. Si ces valeurs sont égales, la prochaine instruction exécutée se trouve à l'adresse représentée par l'étiquette laBas

7.

Interruption du programme.

**273**

1. ADD R0, R0, R1

2. LDR R3, 251

3. STR R0, 124

4. B monLabel

5. CMP R0, #4

**274**

```

1 if a > c :
2     b = c
3 else :
4     b = a

```

**275**

```

1 MOV R0, 1
2 MOV R1, 0
3 testTantque :
4     CMP R0, #100
5     BLT tantQue
6     B finTantQue
7 tantQue :
8     ADD R0, R0, R0
9     ADD R1, R1, #1
10    B testTantque
11 finTantQue :
12    HALT

```

**276** A faire

**277** A faire

**278**

1. A faire

2. On suppose que x est dans R0 et y est dans R1.

$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\left( \frac{n}{k} \right) p^k (1-p)^{n-k}$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\frac{\bar{X} - \mu}{\sigma} \rightarrow \mathcal{N}(0,1)$$

$$\mathbb{P}\left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1, \alpha}^2\right) \approx 1 - \alpha$$

**279**

```

1. MOV R0, #0
2. MOV R1, #0
3. pour :
4.     CMP R1, #11
5.     BGT finPour
6.     ADD R0, R0, R1
7.     ADD R1, R1, #1
8.     B pour
9. finPour :
10.    HALT

```

**280** A faire

**281** Pour représenter les résultats, on peut se poser la question de toutes les routes possibles : Arbre des chemins possibles, chaque branche ne peut contenir deux fois le même sommet pour des raisons évidentes de boucles inutiles.

```

1. MOV R0, #7
2. MOV R1, #12
3. CMP R0, R1
4. BLT cas1
5. BEQ cas2
6. SUB R0, R0, R1
7. HALT
8. cas1 :
9.     ADD R0, R0, R1
10.    HALT
11. cas2 :
12.     ADD R1, R0, R1
13.     HALT

```

**282** Le but de cet exercice est de présenter l'algorithme de Djikstra.

**283** L'unité peut être des kilomètres ou des minutes, le principe pour mesurer le plus court chemin est indépendant de l'unité.

**284** A faire

**285** La réponse en terme de saut correspond au routage historique RIP pour déterminer le chemin le plus court. Il est progressivement remplacé/complété par le routage OSPF qui tient compte lui de la bande passante.

**286**

1. C
2. G
3. A → C → E → G
4. F → G → E → H → D → C

**287** A faire**288**

1. Non, il n'y a pas d'unicité. Par exemple, pour le routeur F, si on veut envoyer un message à C, on peut passer par B.
2. Chaque envoi de message nécessite un coût : temporel, énergétique (consommation d'électricité), financier. Ainsi, une table sera plus efficace qu'une autre si les messages vont plus vite, c'est-à-dire si le chemin suivi est le plus court. Par exemple, dans le réseau précédent, aller de F à C en passant par H n'est pas le plus court chemin, il vaut mieux passer par B. De plus, si un routeur voit passer tous les messages, il risque fort d'être surchargé (comme dans la circulation automobile) et il vaut mieux éviter les embouteillages. On peut donc utiliser des itinéraires BIS.

289

- Un câble peut être coupé, un routeur peut être en panne, de nouveaux routeurs ou de nouvelles connexions peuvent être ajoutés, etc.
- Les tables de routage initiales ne sont évidemment plus valides et il faut les mettre à jour.

290

- OK
- (a) Au final, il faut envoyer sa table de routage.
- (b) L'éliminer de notre table ainsi que tous les chemins qui passaient par lui.
- (c) Je note que le chemin vers ce routeur est direct : je lui envoie.
- (d) Eliminer le chemin vers ce routeur qui passait par lui.
- (e) Je note que j'ai un chemin pour accéder à ce routeur et qu'il passe par mon voisin.
- (f) Chaque routeur, quand il modifie sa table, sait qu'il va devoir la renvoyer à ses voisins. Quand plus personne ne veut envoyer de message c'est fini (en vrai, dans un réseau, on ne sait pas que plus personne ne veut envoyer de message ; d'ailleurs les routeurs relancent régulièrement leur mise à jour de table).

291 A faire

292

- Les routeurs  $C$  et  $D$  retirent les destinations impossibles, le routeur  $A$  note qu'il a le routeur  $C$  comme voisin et envoie sa table :

| Routeur A |        | Routeur B |        | Routeur C |         | Routeur D |         |
|-----------|--------|-----------|--------|-----------|---------|-----------|---------|
| Dest.     | Lien   | Dest.     | Lien   | Dest.     | Lien    | Dest.     | Lien    |
| $B$       | $B(1)$ | $A$       | $A(1)$ | $A$       | $B(2)$  | $A$       | $\dots$ |
| $C$       | $C(1)$ | $C$       | $C(1)$ | $B$       | $B(1)$  | $B$       | $\dots$ |
| $D$       | $B(3)$ | $D$       | $C(2)$ | $D$       | $\dots$ | $C$       | $\dots$ |

- Le routeur  $B$  pouvait déjà aller en  $C$  directement, donc le message de  $A$  ne change rien pour lui ; il n'envoie donc rien. En revanche, le routeur  $C$  met sa table à jour :

| Routeur A |        | Routeur B |        | Routeur C |        | Routeur D |         |
|-----------|--------|-----------|--------|-----------|--------|-----------|---------|
| Dest.     | Lien   | Dest.     | Lien   | Dest.     | Lien   | Dest.     | Lien    |
| $B$       | $B(1)$ | $A$       | $A(1)$ | $A$       | $B(2)$ | $A$       | $\dots$ |
| $C$       | $C(1)$ | $C$       | $C(1)$ | $B$       | $B(1)$ | $B$       | $\dots$ |
| $D$       | $B(3)$ | $D$       | $C(2)$ | $D$       | $A(4)$ | $C$       | $\dots$ |

- Comme le routeur  $C$  se rend compte que la distance obtenue est plus grande que le diamètre du réseau (avec quatre routeurs, la distance maximale doit être de 3), il enlève le chemin vers  $D$  via  $A$  dans sa table et renvoie sa table qui ne contient pas de chemin vers  $D$ . Le routeur  $A$  ne modifie rien, mais le routeur  $B$  peut modifier sa table :

| Routeur A |        | Routeur B |         | Routeur C |         | Routeur D |         |
|-----------|--------|-----------|---------|-----------|---------|-----------|---------|
| Dest.     | Lien   | Dest.     | Lien    | Dest.     | Lien    | Dest.     | Lien    |
| $B$       | $B(1)$ | $A$       | $A(1)$  | $A$       | $B(2)$  | $A$       | $\dots$ |
| $C$       | $C(1)$ | $C$       | $C(1)$  | $B$       | $B(1)$  | $B$       | $\dots$ |
| $D$       | $B(3)$ | $D$       | $\dots$ | $D$       | $\dots$ | $C$       | $\dots$ |

- Le routeur  $A$  n'a pas changé sa table, donc n'envoie rien. En revanche, lorsqu'il reçoit la table du routeur  $B$ , il peut mettre à jour la sienne :

| Routeur A |         | Routeur B |         | Routeur C |         | Routeur D |         |
|-----------|---------|-----------|---------|-----------|---------|-----------|---------|
| Dest.     | Lien    | Dest.     | Lien    | Dest.     | Lien    | Dest.     | Lien    |
| $B$       | $B(1)$  | $A$       | $A(1)$  | $A$       | $B(2)$  | $A$       | $\dots$ |
| $C$       | $C(1)$  | $C$       | $C(1)$  | $B$       | $B(1)$  | $B$       | $\dots$ |
| $D$       | $\dots$ | $D$       | $\dots$ | $D$       | $\dots$ | $C$       | $\dots$ |

5. Cette dernière table va-t-elle bouger après l'envoi de  $A$  de sa nouvelle table ? **Non.**

**298** A faire

**293**

#### Pour le réseau en étoile :

- La distance entre deux routeurs est de maximum deux câbles (c'est bien).
- Le routeur central  $A$  va avoir bien plus de travail que les autres. De plus, s'il tombe en panne, plus personne ne peut communiquer. Il faut donc soit changer la topologie (fortement conseillé), soit surveiller  $A$  de très près.

**299** A faire

**300** A faire

**301** A faire

**302** A faire

**303** A faire

**304** A faire

**305** A faire

**306** A faire

**307** A faire

**308** A faire

**309** A faire

**310** A faire

**311** BUG

**312** BUG

#### Pour le réseau linéaire :

- La distance entre les routeurs  $A$  et  $F$  est très grande.
- Si un routeur tombe en panne, cela coupe en général le réseau en deux, mais il peut y avoir tout de même des communications restantes entre certains routeurs.
- Plus un routeur est vers le centre, puis il risque d'avoir du travail.

**294** a faire

- 295** Pour que deux machines puissent communiquer, il faut qu'elles soient situées sur le même sous-réseaux.

Ici, le masque de sous-réseau est 255.255.255.0. Donc dans la dernière question, la première machine est sur le sous-réseau 192.168.1.0 et la deuxième sur le sous-réseau 192.168.2.0. Elles ne peuvent donc pas communiquer entre elles.

- 296** La table du switch se met à jour à chaque fois que le switch voit passer des informations.

**297** A faire

$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\frac{\bar{X} - \mu}{\sigma} \rightarrow \mathcal{N}(0,1)$$

$$\mathbb{P}\left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1,\alpha}^2\right) \approx 1 - \alpha$$

**313** BUG**314** BUG**315** BUG**316** BUG**317** BUG**318** A faire**319** A faire**320**

1. négatif papillon :

```
def negatif_papillon() :
    """renvoie le negatif du papillon"""
    pap=Image.open("papillon.png")
    for l in range(10) :
        for c in range(10) :
            valeur = pap.getpixel((c,l))
            if valeur == 0 :
                pap.putpixel((c,l),255)
            else :
                pap.putpixel((c,l),0)
    pap.save("image.png","png")
```

2. Négatif Joconde\_10

```
def negatif_joconde10() :
    """renvoie le negatif de Joconde_10.png"""
    lena=Image.open("Joconde_10.png")
    for l in range(10) :
        for c in range(10) :
            valeur = lena.getpixel((c,l))
            nouvelle_valeur = 255-valeur
            lena.putpixel((c,l),nouvelle_valeur)
    lena.save("image.png","png")
```

**321** A faire**322** A faire**323** A faire**324** A faire**325** A faire**326** A faire**327** A faire**328** A faire**329** A faire**330** A faire**331** A faire

**332** Il faut balayer l'image pixel par pixel en imbriquant deux boucles pour y parvenir. On obtient des bouts de code du genre :

$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\binom{n}{k} p^k (1-p)^{n-k}$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\frac{\bar{X} - \mu}{\sigma} \xrightarrow{\text{distrib.}} \mathcal{N}(0,1)$$

```

1 pour i = 0 jusqu'à hauteur-1           # i est la ligne du pixel 343 A faire
2   pour j = 0 jusqu'à largeur-1          # j est la colonne du pixel
3     traitement du pixel (i,j)
4   fin pour
5 fin pour

```

**333** A faire**334** Problème d'arrondi ou encore nombre résultant non compris entre 0 et 255.**335** Identité**336**

|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

**337** A faire**338** A faire**339** A faire**340** A faire**341** A faire**342** A faire

# i est la ligne du pixel 343 A faire

# j est la colonne du pixel

**344** A faire**345** A faire**346** A faire**347** A faire**348** Plus difficile de construire la matrice de l'image**349** A faire**350** A faire**351** A faire**352** A faire**353** A faire**354** A faire