

Corrigé

A) Pile ou file...

On suppose qu'il faut lire les ordres de sortie de gauche à droite. Il fallait bien comprendre que l'on peut retirer des éléments de la structure entre les opérations d'insertion. Sinon, le premier ordre correspond à une file et le dernier à une pile, et c'est tout.

Remarquons que dans une file, le premier élément sorti est le premier entré. Seule la séquence de sortie commençant par A peut correspondre à une structure de file.

- **A B C D E F** : ce peut être une file (on peut retirer n'importe quand) ou une pile (si l'on retire les éléments immédiatement après les avoir ajoutés);
- **D E C B F A** : pile uniquement :

Opération	État de la pile
ajouter A, B, C, D	[A; B; C; D[
retirer D	[A; B; C[
ajouter E	[A; B; C; E[
retirer E, C, B	[A[
ajouter F	[A; F[
retirer F, A	[

- **B D E F A C** : ni pile ni file : pour retirer D il faut avoir ajouté A et C dans la pile et alors on ne peut pas retirer A avant C;
- **F E D C B A** : pile possible (on ajoute tout puis on retire tout).

B) Tri fusion...

1. Il suffit de couper la liste au milieu (au vu des exemples proposés, cette démarche convient bien). On utilise la simplicité donnée par le *slicing* en PYTHON.

PYTHON

```
def partition(liste):
    milieu = len(liste) // 2
    return liste[:milieu], liste[milieu:]
```

On peut aussi donner une version récursive. Il faut évidemment une fonction de complexité linéaire en la taille de la liste.

2. Les listes sont supposées triées et il faut donner une version de complexité linéaire en la somme des tailles des deux listes. On égrène simultanément les deux listes et il suffit de comparer leur tête pour savoir quel est l'élément suivant à insérer dans la liste résultat. C'est le rôle des variables *i1* et *i2*.

PYTHON

```
def merge(liste1, liste2):
    i1, i2 = 0, 0
    resultat = []
    while i1 < len(liste1) and i2 < len(liste2):
        if liste1[i1] < liste2[i2]:
            resultat.append(liste1[i1])
            i1 += 1
        else:
            resultat.append(liste2[i2])
            i2 += 1
    # L'une des deux listes est vide, reste éventuellement
    # à vider l'autre
    while i1 < len(liste1):
        resultat.append(liste1[i1])
        i1 += 1
    while i2 < len(liste2):
        resultat.append(liste2[i2])
        i2 += 1
    return resultat
```

On peut aussi donner une version récursive.

3. Il reste simplement à recoller les morceaux, en faisant bien attention aux cas de base (il faut considérer la liste vide *et* la liste à un élément).

PYTHON

```
def merge_sort(liste):
    if len(liste) <= 1:
        return liste
    liste1, liste2 = partition(liste)
    return merge(merge_sort(liste1), merge_sort(liste2))
```

C) Maximum Gap...

On commence par écrire une fonction qui calcule le « gap » d'une liste.

PYTHON

```
def gap(liste):
    """Renvoie l'écart maximum entre deux valeurs
    de la liste, supposée non vide"""
    mini, maxi = liste[0], liste[0]
    for i in range(1, len(liste)):
        if liste[i] < mini:
            mini = liste[i]
        elif liste[i] > maxi:
            maxi = liste[i]
    return maxi - mini
```

On en déduit la fonction demandée qui se contente de chercher le maximum sur les lignes.

PYTHON

```
def max_gap_matrix(M):
    """Renvoie le gap maximum des lignes de M,
    supposée non vide."""
    max_gap = gap(M[0])
    for i in range(1, len(M)):
        # Ne pas calculer deux fois ceci car O(n)
        new_gap = gap(M[i])
        if new_gap > max_gap:
            max_gap = new_gap
    return max_gap
```

Remarque 1

Je préfère en général utiliser les fonctions `min` et `max` de PYTHON qui permettent de simplifier un peu le code, mais j'ai choisi ici de faire sans.

On peut écrire, de manière très pythonique, la fonction demandée en une ligne, mais ce n'est probablement pas ce qui était attendu

PYTHON

```
def max_gap_matrix2(M):
    return max(max(ligne) - min(ligne) for ligne in M)
```

D) Matrices : symétrique...

On vérifie que la matrice est bien carrée (on vérifie également que A encode bien une matrice, c'est-à-dire que toutes ses lignes ont même dimension) et que pour tout coefficient de la partie triangulaire supérieure on a $a_{i,j} = a_{j,i}$.

PYTHON

```
def is_symetric(A):
    n = len(A)
    for i in range(n):
        if len(A[i]) != n:
            return False
        for j in range(i + 1, n):
            if A[i][j] != A[j][i]:
                return False
    return True
```

QCM

Question 1

Parmi les termes suivants, lequel a un lien direct avec les réseaux informatiques ?

- ✗ UPS (*Uninterruptible power supply* : onduleur électrique, pas directement lié aux réseaux)
- ✓ TCP (*Tranfert Control Protocol* : protocole de la couche transport des réseaux, RFC 793)
- ✗ ICBM (*Intel Chip Based Macintosh* : Mac utilisant une puce Intel, pas lié aux réseaux)

- ✓ DHCP (*Dynamic Host Configuration Protocol* : configuration automatique des IP d'un réseau)

Question 2

Le noyau de Linux a été initialement écrit par

- ✗ Richard Matthew Stallman (projet GNU, GPL, Emacs, gcc, etc.)
- ✓ Linus Torvalds (en 1991, à 22 ans ; à l'origine aussi de git)
- ✗ Andrew Tannenbaum (MINIX, clone de UNIX : « précurseur de Linux »).

Question 3

Informaticien célèbre

- ✓ Alan Turing (1912-1954), inventeur de l'ordinateur, héros de la seconde guerre mondiale, précurseur de l'intelligence artificielle, et bien d'autres choses...
- ✗ Edgar J. Hoover (1895—1972), premier directeur du FBI.
- ✗ Alan Parker (1944-), réalisateur, compositeur, producteur.
- ✓ Edsger W. Dijkstra (1930-2002), nombreux travaux sur les langages de programmation (Algol, *Go To Statement Considered Harmful*, etc.), et puis bien sûr l'algorithme de Dijkstra pour les plus court chemins.
- ✗ Donald Trump (1946-), président actuel des États-Unis.
- ✓ Donald E. Knuth (1938-), pionnier de l'algorithmique, *The Art of Computer Programming*, \TeX , etc.

Question 4

Laquelle des affirmations suivantes est vraie

- ✗ L'électronique numérique fonctionne essentiellement selon trois états : 0 (pas de courant), 1 (courant) ou haute impédance : les sorties trois états (Hi-Z) sont bien utilisées et essentielles (par exemple pour les Bus) mais on ne peut pas dire que c'est *la* composante essentielle de l'électronique numérique aujourd'hui.
- ✓ Certains robots mobiles utilisent pour se déplacer des moteurs pas-à-pas : les moteurs pas-à-pas transforment une impulsion électrique en un déplacement angulaire sans nécessairement d'asservissement pour obtenir un mouvement précis, ce qui les rend attractifs pour la robotique.

- ✗ La monnaie virtuelle bitcoin est générée par des « miners » virtuels hébergés par le jeu Minecraft : les « miners » bitcoin n'ont rien de virtuel et ne sont pas hébergés par un jeu. L'émeraude est une monnaie virtuelle sur Minecraft.
- ✓ Android est un système d'exploitation pour téléphones, tablettes, et ordinateurs : c'est un système d'exploitation mobile fondé sur le noyau Linux, principalement conçu pour les téléphones et les tablettes, mais qui équipe aussi des ordinateurs (e.g. Android-x86).

Question 5

Que fait le code suivant :

PYTHON

```
def fubarp(n, k):
    if k * k > n:
        return n
    if n % k:
        return fubarp(n, k + 2)
    else:
        return k

def fubar(n):
    if n % 2:
        return fubarp(n, 3)
    else:
        return 2
```

La fonction `fubarp` renvoie le plus petit diviseur de n qui soit plus grand que k . La fonction `fubar` renvoie le plus petit diviseur non trivial de $n \geq 1$. Ce programme utilise le pythonisme `if m` avec $m \in \mathbb{N}$, qui s'évalue à `True` si et seulement si $m \neq 0$.

- ✗ Il affiche une décomposition en facteurs premiers
- ✗ Il calcule une factorielle
- ✓ Il détermine le plus petit facteur non trivial d'un nombre
- ✗ Il calcule le PGCD de deux nombres

Question 6

La technologie Java appartient à

- ✗ Sun Microsystems (n'existe plus, a été racheté en 2009 par Oracle). Est-ce un piège ? Faut-il répondre vrai ou faux ?
- ✗ Android
- ✗ Microsoft
- ✓ Oracle

Question 7

Noms de microprocesseurs

- ✓ sparc (*Scalable Processor Architecture*)
- ✗ tinkerbelle (Fée clochette)
- ✗ coldwater (film)
- ✓ arm (*Advanced RISC Machines* ou *Acorn RISC Machine*)
- ✗ head&shoulders (shampooing)
- ✗ mip-mips (Bip bip et coyote ?)
- ✓ mips (*microprocessor without interlocked pipeline stages*)

Question 8

L'acronyme GAFa englobe les plus grosses compagnies informatiques liées à internet, il comprend

- ✓ Google
- ✗ Android (n'est plus une compagnie, ancienne startup)
- ✗ Facetime (application, pas compagnie)
- ✗ Twitter (géant du Web aussi)
- ✓ Apple

Remarque 2

GAFa : Google, Apple, Facebook, Amazon

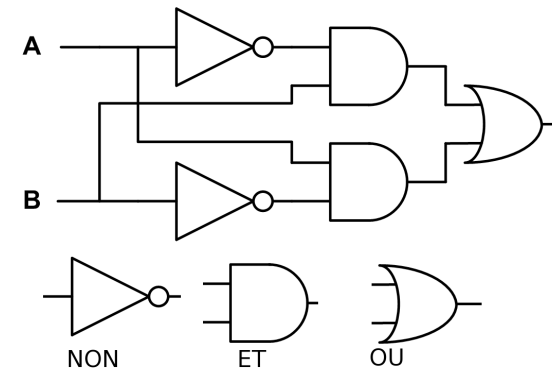
Question 9

Certaines calculatrices ont un mode BCD (binaire codé décimal). Le principe : on découpe un octet en deux groupes de quatre bits, et chaque groupe code un chiffre en base 10, ce qui évite d'avoir à convertir pour l'affichage, et permet d'avoir un nombre de chiffres "exact" en représentation. Avec ce schéma, la valeur 85 donnera en binaire

- ✓ 10000101 ($8 = \overline{1000}^2, 5 = \overline{0101}^2$)
- ✗ 10001001
- ✗ 01010101
- ✗ 10000100

Remarque 3

Je n'ai pas compris grand chose aux explications.



Question 10

Le schéma précédent combine les entrées A et B et forme

- ✗ La négation du "et" de A et B
- ✓ Le ou exclusif de A et B
- ✗ La retenue de la somme de A et B
- ✓ La somme de A et B, en oubliant la retenue