

INFO-H-100 – Informatique – Programmation – Prof. Th. Massart  
1<sup>ère</sup> année du grade de Bachelier en Sciences de l'Ingénieur  
Examen de juin

---

### Remarques préliminaires

- On vous demande de répondre à **chaque question sur une feuille séparée**.
- N'oubliez pas d'inscrire votre nom, prénom et numéro de matricule sur chaque feuille.
- Vous disposez de 2 heures 45 minutes et vous ne pouvez pas utiliser de notes.
- Si du code vous est demandé,
  - la réponse à la question doit comprendre le code *Python* structuré et conforme aux règles de bonne pratique et conventions,
  - sauf mention contraire, vous ne pouvez utiliser aucune fonction de bibliothèques (pas d'import)
  - veillez à découper votre réponse en fonctions de manière pertinente
  - veillez à utiliser des structures de données appropriées.

### Question 1 - Complexité (3 points)

Les 2 fonctions `equality_list` et `equality_dic` renvoient toutes les deux la valeur vraie si et seulement si la liste `b` est une permutation de la liste `a` (tous les éléments de `a` (resp. `b`) sont dans `b` (resp. `a`) et en même quantité).

```
def equality_dic(a,b):
    res = False
    if len(a)==len(b):
        dic_a = {}
        dic_b = {}
        for i in a:
            dic_a[i]=dic_a.get(i,0)+1
        for i in b:
            dic_b[i]=dic_b.get(i,0)+1
        res = True
        for j in dic_a:
            res = res and (dic_a[j] == dic_b.get(j,0))
        for j in dic_b:
            res = res and (dic_b[j] == dic_a.get(j,0))
    return res

def equality_list(a,b):
    res = False
    if len(a)==len(b):
        res = True
        for i in a:
            if i in b:
                a.remove(i)
                b.remove(i)
            else:
                res = False
    return res
```

En supposant que les éléments soient entiers, donner la complexité maximale en  $O$  du temps d'exécution de chacune des deux fonctions. Veillez à détailler vos réponses en justifiant la complexité de chaque partie du code.

### Question 2 - Compilation / interprétation (2 points)

Expliquez les différentes phases de la compilation d'un programme. Python est un langage interprété. Expliquez la différence avec un langage compilé. La gestion de la mémoire lors de l'exécution d'un programme est-elle différente dans les deux cas (justifiez) ?

### Question 3 - Hachage (4 points)

**ATTENTION** : cette question doit être résolue à l'aide de `map`, `reduce` et `filter`. Vous n'avez pas le droit d'utiliser de boucle (`for`, `while`).

En programmation, il est souvent utile de pouvoir réduire une chaîne de caractères à un nombre (pour des comparaisons grossières et rapides). On vous demande d'écrire une fonction qui reçoit en paramètre une chaîne de caractères et qui renvoie comme résultat un entier calculé de la manière suivante : après avoir retiré tous les caractères qui ne sont pas des lettres, remplacer chaque lettre par sa valeur ASCII et calculer la somme de toutes ces valeurs ASCII. Notez que, par exemple, les mots `ab` et `ba` correspondront au même entier.

Rappel : la fonction `ord(c)` prend en argument un caractère (chaîne de caractères de longueur 1) et donne comme résultat sa valeur ASCII (entier).

## Question 4 - Flatten (5 points)

Ecrire une fonction qui “aplatit” une liste à plusieurs niveaux (une liste dont chaque élément est soit une valeur de n'importe quel type sauf une liste, soit une liste à plusieurs niveaux) en une liste à un seul niveau contenant les mêmes éléments, dans le même ordre. Ici, “dans le même ordre” signifie que si on devait écrire les deux listes littéralement, les valeurs apparaîtraient dans le même ordre.

Exemples :

```
>>> flatten([[[[0]], 5, [3]], 4])
[0, 5, 3, 4]
>>> flatten([5, [2, [3, 7], [[4], 2], [], 8]])
[5, 2, 3, 7, 4, 2, 8]
>>>
```

Rappel : on peut demander à Python quel est le type de la valeur associée à une variable à l'aide de la fonction `type`. Par exemple, le code suivant :

```
x = 10
if type(x) == int:
    print "C'est un entier !"
else:
    print "Ce n'est pas un entier."
```

imprimera à l'écran "C'est un entier!". Les types vus au cours sont `int`, `float`, `bool`, `list`, `str`, `dict` et `tuple`.

## Question 5 - Cryptage de fichier (6 points)

On vous demande d'écrire une fonction `translate` qui reçoit en paramètres deux noms de fichiers. Le premier fichier doit contenir, sur chaque ligne, deux caractères séparés par un espace. Le second fichier est un fichier texte que l'on souhaite crypter à l'aide du premier fichier. Votre programme doit créer un nouveau fichier, dont le nom est similaire au second fichier mais avec une extension différente (la nouvelle extension doit être `crypt`), et dont le contenu est celui du second fichier dans lequel tous les caractères qui apparaissent dans la première colonne du premier fichier ont été remplacés par le caractère correspondant dans la seconde colonne. Les caractères qui n'apparaissent pas dans le premier fichier doivent être laissés intacts.

Par exemple, si le premier fichier, `trans.txt`, contient les deux lignes

```
a b
z d
```

et que le fichier `bonjour.txt` contient le texte

```
Bonjour, comment allez-vous ?
```

l'appel de fonction

```
translate("trans.txt", "bonjour.txt")
```

devrait créer sur le disque le fichier `bonjour.crypt` contenant le texte

```
Bonjour, comment blled-vous ?
```

Rappel : L'extension d'un fichier est le groupe de lettres qui se trouve après le dernier point dans le nom du fichier (`txt` dans `dico.txt` par exemple).

Vous pouvez considérer que les fichiers se trouvent dans le même dossier que le programme.

BON TRAVAIL !