

DEVOIR DE CONTROLE DU SEMESTRE 2**Matière : INFORMATIQUE****Classes : 1^{ère} Année MP – PC – T Durée : 1 h**

- Le sujet comporte deux parties.
- Les fonctions demandées devront être écrites en langage Python.
- Lorsque l'énoncé n'exige pas une réponse avec version récursive, c'est la version itérative qui est demandée. Dans le cas contraire, seule la réponse avec une version récursive est acceptée.

Partie 1

L'adresse **IPV4** est un numéro servant à identifier d'une façon unique chaque ordinateur connecté au réseau d'internet. Elle a la forme suivante : **W.X.Y.Z** avec **W**, **X**, **Y** et **Z** sont quatre entiers positifs compris entre 0 et 255 (par exemple : 155.105.50.69 est une **IPV4**).

On se propose dans cette partie de vérifier la validité des adresses **IPV4** stockées en tant que chaînes de caractères dans une liste **L** (par exemple : **L** = ['155.105.50.69', '155.105.50.47', '155.105.50.58', '155.105.50.61']).

Pour vérifier la validité d'une adresse **IPV4**, il suffit de vérifier que la chaîne de caractères, qui la représente, est formée par quatre sous-chaînes **W**, **X**, **Y** et **Z** dont la conversion de chacune en un entier appartient à l'intervalle **[0, 255]**.

Indications :

- La fonction **ch.split(sep)** permet de retourner la liste formée par les sous-chaînes de la chaîne de caractères **ch** séparées par le caractère **sep** donné en argument (par exemple '155.105.50.69'.split('.') donne la liste ['155', '105', '50', '69']).
- La fonction **ch.isdigit()** renvoie **True** si la chaîne de caractères **ch** n'est formée que par des caractères numériques, **False**, sinon.

Travail demandé :

1. Écrire la fonction booléenne **testVal(v)** qui renvoie la valeur **True** si l'entier **v** appartient à l'intervalle **[0, 255]**, **False**, sinon.
2. Écrire la fonction booléenne **valide(ip)** qui permet de vérifier la validité d'une adresse **IPV4** (le paramètre **ip** est une chaîne de caractères représentant l'adresse **IPV4**).
3. Écrire la fonction récursive **validIP(l)** qui renvoie une liste ne contenant que les adresses **IPV4** valides.

Partie 2

Chaque adresse **IPv4** valide appartient à l'une des classes suivantes :

- Classe **A**, si la valeur du premier bit à gauche de la représentation en binaire de **W** est **0**.
- Classe **B**, si la valeur des deux premiers bits à gauche de la représentation en binaire de **W** est **10**.
- Classe **C**, si la valeur des trois premiers bits à gauche de la représentation en binaire de **W** est **110**.
- Classe **D**, si la valeur des quatre premiers bits à gauche de la représentation en binaire de **W** est **1110**.
- Classe **E**, si la valeur des quatre premiers bits à gauche de la représentation en binaire de **W** est **1111**.

Exemple :

L'adresse 155.105.50.69 est valide et elle appartient à la classe **B** car la valeur des deux premiers bits à gauche de la représentation en binaire de 155 qui est 10011011 est 10.

Indication :

La fonction **bin(nb)** permet de convertir en binaire un nombre décimal **nb** (par exemple : **bin(155)** retourne la chaîne de caractères '**0b10011011**').

Travail demandé :

4. Écrire la fonction **binaire(n)** qui retourne la représentation binaire de **n** sans le préfixe '**0b**' qui apparaît dans le résultat de **bin(n)**.
5. Écrire la fonction **classe(ip)** qui retourne la chaîne de caractères représentant la classe d'une adresse **ip** donnée en paramètre.
6. Écrire la fonction **creerList(l)** qui reçoit en argument la liste **l** contenant des adresses **IPv4** et retourne la liste des tuples dont chacun est formé par une adresse **IPv4 valide** ainsi que la classe à laquelle elle appartient (Exemple de tuple dans la liste retournée : ('155.105.50.69', 'B')).
7. Écrire la fonction **réursive nbClass(l, ch)** permettant de compter le nombre d'adresses dans la liste des tuples **l**, passée en premier argument, qui appartiennent à la classe **ch** passée en deuxième argument.
8. Écrire la fonction **creerDict(l)** permettant de retourner le dictionnaire dont les clés sont toutes les classes des adresses ('A', 'B', ..., 'E') et leurs valeurs correspondantes sont leurs nombres d'occurrences dans la liste des tuples **l** donnée en argument.