

# I21 - Algorithmique élémentaire

## TP 7

Semestre 2 2018/20719

Pour réaliser l'ensemble des exercices de cette planche vous devrez faire débiter vos scripts par l'instruction `from pilefile import *` après avoir sauvegardé le fichier `pilefile.py` dans votre répertoire de travail.

### EXERCICE 1. Manipulation élémentaire des piles/files

1. Se Familiariser avec les différentes fonctions de manipulation des piles/files (commande `help("pilefile")` dans l'interprète par exemple).
2. Écrire des fonctions `pile_affiche(P)` et `file_affiche(F)` qui affiche le contenu de la pile P sous forme de pile et de la file F sous forme de file. Par exemple:

```
>>> P = pile_init()          >>> F = file_init()
>>> empiler(P,5)             >>> enfiler(F,5)
>>> empiler(P,12)           >>> enfiler(F,12)
>>> empiler(P,4)            >>> enfiler(F,4)
>>> pile_affiche(P)         >>> file_affiche(F)

-----
|  4 |
-----
| 12 |
-----
|  5 |
-----

-----
| 4 | 12 | 5 |
-----
```

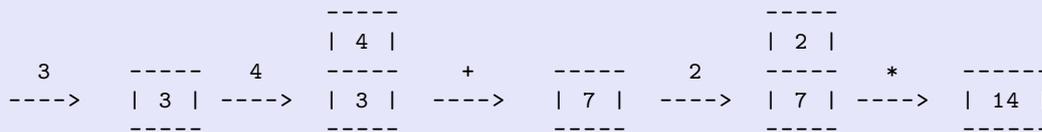
3. Écrire une fonction `parenthese(s)` qui vérifie qu'une expression sous forme de chaîne de caractères est correctement parenthésée (une chaîne pourra comporter des parenthèses, des crochets et des accolades). Par exemple `parenthese( "3*[2-(6)]-(a+b)" )` retournera `True` et `parenthese( "(3+[2])" )` retournera `False`.
4. Écrire une fonction `affiche_ternaire(n)` qui affiche toutes les chaînes ternaires de longueur n. Par exemple:

```
>>> affiche_ternaire(2)
00
01
02
10
11
12
20
21
22
```

## EXERCICE 2. Notation polonaise inversée

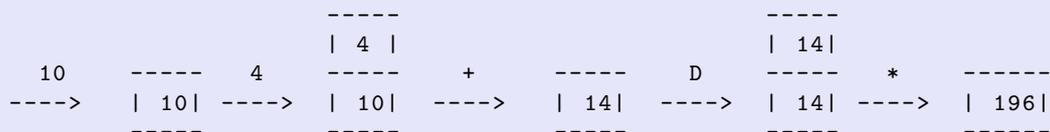
La notation polonaise inversée (NPI) est une écriture non ambiguë des expressions arithmétiques permettant de se passer des parenthèses dans laquelle les opérandes sont placées avant les opérateurs. Par exemple l'expression  $2*(3+4)$  pourra s'écrire `3 4 + 2 *`. L'évaluation d'une telle expression peut se faire à l'aide d'une pile: on empile lorsqu'on lit un nombre et lorsqu'on lit un opérateur on dépile les deux données précédentes et effectue l'opération et on empile le résultat. Par exemple:

```
exp = "3 4 + 2 *"
```



1. Écrire une fonction `eval_npi(exp)` qui évalue et retourne le résultat d'une expression arithmétique écrite en NPI en ne considérant que des nombres écrit sur un seul chiffre.
2. Améliorer la fonction précédente pour gérer les nombres sur plusieurs chiffres (les nombres et les opérateurs seront séparés par des espaces).
3. Ajouter la gestion d'un opérateur supplémentaire D qui duplique le résultat un résultat intermédiaire. Par exemple `"10 4 + 10 4 + *"` pourra s'écrire `"10 4 + D *"`.

```
exp = "10 4 + D *"
```



## EXERCICE 3. Challenge: Jeu du Morpion

Le Tic-Tac-Toe, ou jeu du Morpion, est un jeu se déroulant sur une grille de 3x3 cases dans lesquelles deux joueurs vont à tour de rôle dessiner un symbole qui leur a été attribué. Le gagnant est celui qui réussit à aligner trois symboles identiques, que ce soit en longueur, largeur ou diagonale.

Le but de cette exerce est d'écrire un programme capable de compter le nombre de parties possibles du jeu du Morpion.

### Indications:

1. Après avoir choisit soigneusement la manière de représenter la grille de jeu ou pourra dans un premier temps écrire les fonctions suivantes:

```
def coups(grille):
    """Retourne la liste des coups possibles a jouer dans <grille>
    ou chaque coup est de la forme (lig,col)
    """

def jouer(grille, symb, coup):
    """Place le symbole <symb> dans <grille> aux coordonnees donnees par
    le coup <coup>

    """

def fini(grille):
    """Retourne True si <grille> correspond a une partie terminee (un des
    joueurs a gagne ou la grille est pleine) et False sinon

    """
```

- 
2. Pour faire une copie d'un tableau à deux dimensions il faut utiliser la fonction `deepcopy` du module `copy`
  3. À partir des positions 1, 2 et 3 ci-dessous ils restent respectivement 5, 14 et 592 continuations possibles (on considère que les X jouent en premier).

$\begin{array}{c} \text{X} \mid \text{O} \mid \text{X} \\ \hline 1) \quad \text{O} \mid \text{X} \mid \\ \hline \text{O} \mid \quad \mid \end{array}$	$\begin{array}{c} \text{O} \mid \text{O} \mid \\ \hline 2) \quad \quad \mid \text{X} \mid \\ \hline \text{X} \mid \text{X} \mid \end{array}$	$\begin{array}{c} \text{X} \mid \quad \mid \\ \hline 3) \quad \text{O} \mid \quad \mid \\ \hline \text{X} \mid \quad \mid \end{array}$
---	--	--