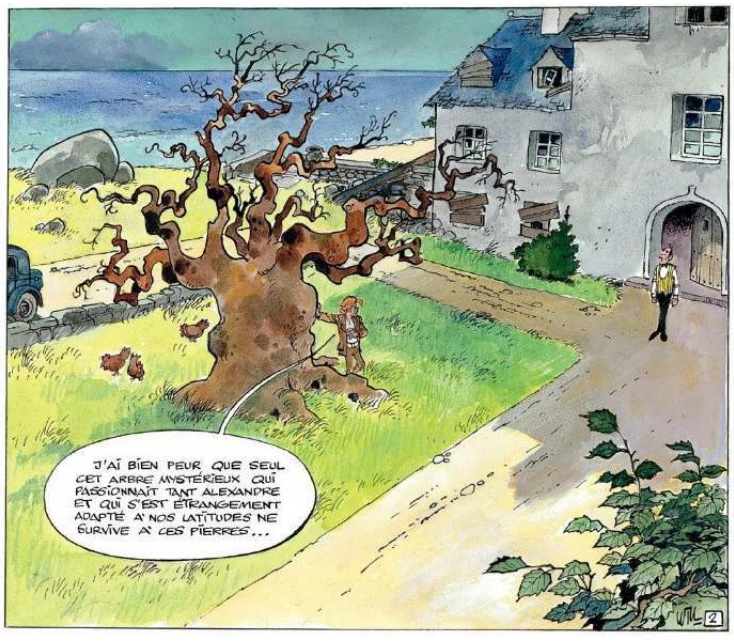
Algorithmique
Contrôle n° 2 (C2)

Info-sup (S2)
Epita

9 Mar. 2016 - 9 :30 (D.S. 307186.87 BW)

Consignes (à lire) :

- ☐ Vous devez répondre sur **les feuilles de réponses prévues à cet effet**.
  - — Aucune autre feuille ne sera ramassée (gardez vos brouillons pour vous).
  - — Répondez dans les espaces prévus, **les réponses en dehors ne seront pas corrigées** : utilisez des brouillons !
  - — Ne séparez pas les feuilles à moins de pouvoir les ré-agrafer pour les rendre.
  - — Aucune réponse au crayon de papier ne sera corrigée.
- ☐ La présentation est notée en moins, c'est à dire que vous êtes noté sur 20 et que les points de présentation (2 au maximum) sont retirés de cette note.
- ☐ **Le code :**
  - — Tout code doit être écrit dans le langage `Python` (pas de C, Caml, Algo ou autre).
  - — **Tout code `Python` non indenté ne sera pas corrigé.**
  - — Tout ce dont vous avez besoin (fonctions, méthodes) est indiqué en **annexe** !
  - — Vos fonctions doivent impérativement respecter les exemples d'applications donnés.
- ☐ Durée : 2h00



1 Des expressions et des arbres

Rappel

On peut représenter une expression par un arbre : les nœuds internes contiennent les opérateurs, les opérandes, elles, sont contenues dans les nœuds externes. Les expressions dont il est question ici sont des expressions arithmétiques utilisant les opérateurs binaires + - * et /.

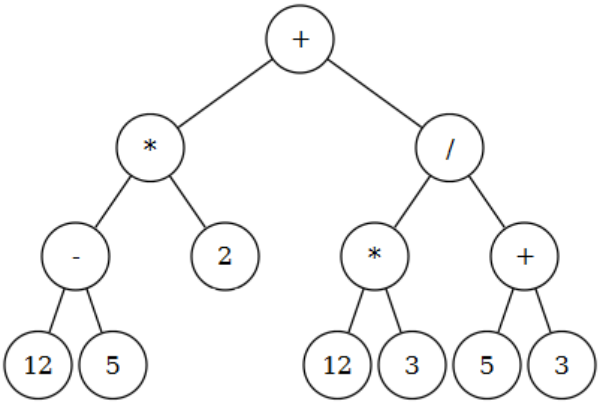


FIGURE 1 – Arbre de l'expression $(12-5)*2+(12*3)/(5+3)$

Exercice 1.1 (Dessine moi – *5 points*)

Les arbres binaires B_1, B_2, B_3 et B_4 représentent des expressions.

Lors du parcours profondeur main gauche

- les valeurs de B_1 sont rencontrées en préfixe dans cet ordre : + - / 8 2 5 + 4 * 2 3
- les valeurs de B_2 sont rencontrées en suffixe dans cet ordre : 8 20 - 2 2 * 2 + /
- les parcours de B_3 et B_4 donnent en infixe le même ordre : 8 + 7 / 5 - 4 * 2
- B_3 et B_4 sont différents mais les deux expressions qu'ils représentent valent toutes les deux -5.

Dessiner les arbres B_1, B_2, B_3 et B_4 et donner les valeurs des expressions représentées par B_1 et B_2.

Exercice 1.2 (Compte moi – *3 points*)

Écrire la fonction `nodes` qui retourne le nombre d'opérateurs, ainsi que le nombre d'opérandes d'une expression représentée par un arbre binaire.

Exemple d'application sur l'arbre de la figure 1 :

```
>>> nodes(B)
(6, 7)
```

Exercice 1.3 (Affiche moi – *3 points*)

Écrire la fonction `exp2str` qui retourne une chaîne contenant l'expression complètement parenthésée représentée par un arbre.

Exemple d'application sur l'arbre de la figure 1 :

```
>>> exp2str(B)
'(((12-5)*2)+((12*3)/(5+3)))'
```

2 Des matrices

1	10	3	0	3	10	1
1	0	1	8	1	0	1
10	9	14	1	14	9	10
10	3	7	11	7	3	10
7	8	5	1	5	8	7

FIGURE 2 – `Mat1`

1	10	3	3	10	1
1	0	1	1	0	1
10	9	4	4	9	10
10	3	7	7	3	10
7	8	15	15	8	7

FIGURE 3 – `Mat2`

1	24	12	18	4
10	15	15	0	18
8	14	0	16	2
22	4	8	14	22
19	7	23	5	5

FIGURE 4 – `Mat3`

Dans les exercices ci-dessous, les matrices sont supposées non vides.

Exercice 2.1 (Minimax – *5 points*)

Écrire une fonction qui cherche la valeur minimale parmi les maximums de chaque ligne d'une matrice d'entiers.

Deux versions possibles pour cette fonction :
— elle retourne la valeur cherchée : `minimax`
— elle retourne la position de la valeur cherchée : `posMinimax`

Exemples d'applications sur la matrice de la figure 4 :

```
>>> minimax(Mat3)
16
>>> posMinimax(Mat3)
(2, 3)
```

Choisissez la version qui vous convient sachant que c'est bien entendu la deuxième version qui rapportera le plus de points. [1]

Exercice 2.2 (Symétrie – *5 points*)

Écrire la fonction `symetric` qui vérifie si une matrice est symétrique selon un axe verticale (symétrie horizontale).

Exemples d'applications sur les matrices des figures 2, 3 et 4 :

```
>>> symetric(Mat1)
True
>>> symetric(Mat2)
True
>>> symetric(Mat3)
False
```

1. Des fois, il vaut mieux peu de points que pas de point.

Annexes

Les arbres binaires

Les arbres binaires manipulés ici sont les mêmes qu'en td.
— `None` est l'arbre vide.
— L'arbre non vide a 3 attributs : `key`, `left`, `right`.

Fonctions et méthodes autorisées

Les fonctions que vous pouvez utiliser :

— `len` sur les listes.

— `range`.

— `str` sur les chaînes : transforme son paramètre en chaîne.

— `min` et `max`, mais uniquement avec deux valeurs entières !

```
>>> max(12, 5)
12

>>> min(12, 5)
5
```

Valeur particulière :

On supposera qu'il existe une valeur `maxint` = le plus grand entier représentable.

Vos fonctions

Vous pouvez également écrire vos propres fonctions, dans ce cas elles doivent être documentées (on doit savoir ce qu'elles font).

Dans tous les cas, la dernière fonction écrite doit être celle qui répond à la question.