

# ET SDD Pile Files Arbres

## Exercice 1 : parenthésage (14 points)

Cet exercice composé de deux parties A et B, porte sur les structures de données.

### Partie A : Expression correctement parenthésée

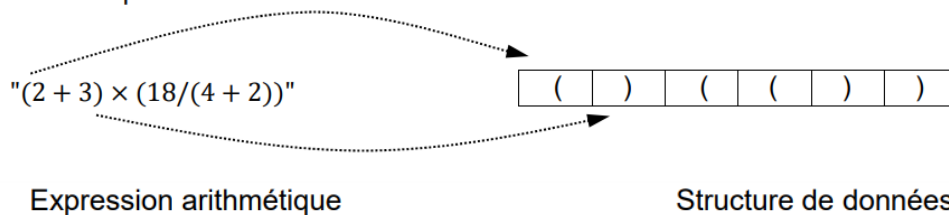
On veut déterminer si une expression arithmétique est correctement parenthésée.

Pour chaque parenthèse fermante ")" correspond une parenthèse précédemment ouverte "(".

Exemples :

- L'expression arithmétique  $"(2 + 3) \times (18 / (4 + 2))"$  est correctement parenthésée.
- L'expression arithmétique  $"(2 + 3) \times (18 / (4 + 2"$  est non correctement parenthésée.

Pour simplifier les expressions arithmétiques, on enregistre, dans une structure de données, uniquement les parenthèses dans leur ordre d'apparition. On appelle expression simplifiée cette structure.



1. Indiquer si la phrase « les éléments sont maintenant retirés (pour être lus) de cette structure de données dans le même ordre qu'ils y ont été ajoutés lors de l'enregistrement » décrit le comportement d'une file ou d'une pile. Justifier.

Pour vérifier le parenthésage, on peut utiliser une variable `controleur` qui :

- est un nombre entier égal à 0 en début d'analyse de l'expression simplifiée ;
- augmente de 1 si l'on rencontre une parenthèse ouvrante "(" ;
- diminue de 1 si l'on rencontre une parenthèse fermante ")".

Exemple : On considère l' expression simplifiée A :  $"()()"$

Lors de l'analyse de l'expression A, `controleur` (initialement égal à 0) prend successivement pour valeur 1, 0, 1, 2, 1, 0. Le parenthésage est correct.

2. Écrire, pour chacune des 2 expressions simplifiées B et C suivantes, les valeurs successives prises par la variable `controleur` lors de leur analyse.

Expression simplifiée B :  $"((( )))"$

Expression simplifiée C :  $"(( )))(""$

3. L'expression simplifiée B précédente est mal parenthésée (parenthèses fermantes manquantes) car le `controleur` est différent de zéro en fin d'analyse. L'expression simplifiée C précédente est également mal parenthésée (parenthèse fermante sans parenthèse ouvrante) car le `controleur` prend une valeur négative pendant l'analyse.  
Recopier et compléter uniquement les lignes 13 et 16 du code ci-dessous pour que la fonction `parenthesage_correct` réponde à sa description.

```
1 def parenthesage_correct(expression):
2     ''' fonction retournant True si l'expression arithmétique
3     simplifiée (str) est correctement parenthésée, False
4     sinon.
5     Condition: expression ne contient que des parenthèses
6     ouvrantes et fermantes '''

7     controleur = 0
8     for parenthese in expression: #pour chaque parenthèse
9         if parenthese == '(':
10             controleur = controleur + 1
11         else: # parenthese == ')'
12             controleur = controleur - 1
13             if controleur ... : # test 1 (à recopier et compléter)
14                 #parenthèse fermante sans parenthèse ouvrante
15                 return False
16     if controleur ... : # test 2 (à recopier et compléter)
17         return True #le parenthésage est correct
18     else:
19         return False #parenthèse(s) fermante(s) manquante(s)
```

### Partie B : Texte correctement balisé

On peut faire l'analogie entre le texte simplifié des fichiers HTML (uniquement constitué de balises ouvrantes `<nom>` et fermantes `</nom>`) et les expressions parenthésées :

Par exemple, l'expression HTML simplifiée :

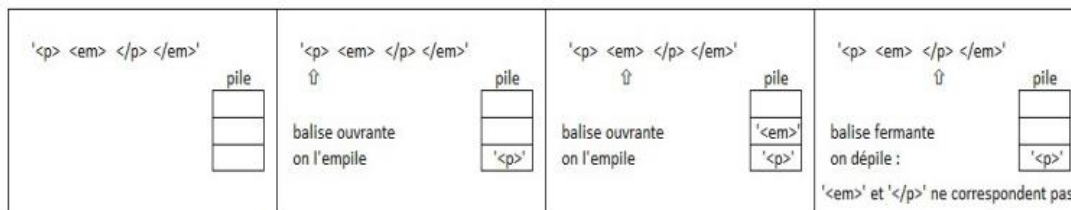
"`<p><strong><em></em></strong></p>`" est correctement balisée.

On ne tiendra pas compte dans cette partie des balises ne comportant pas de fermeture comme `<br>` ou `<img ...>`.

Afin de vérifier qu'une expression HTML simplifiée est correctement balisée, on peut utiliser une pile (initialement vide) selon l'algorithme suivant :

On parcourt successivement chaque balise de l'expression :

- lorsque l'on rencontre une balise ouvrante, on l'empile ;
- lorsque l'on rencontre une balise fermante :
  - si la pile est vide, alors l'analyse s'arrête : le balisage est incorrect ,
  - sinon, on dépile et on vérifie que les deux balises (la balise fermante rencontrée et la balise ouvrante dépilée) correspondent (c'est-à-dire ont le même nom) si ce n'est pas le cas, l'analyse s'arrête (balisage incorrect).



État de la pile lors du déroulement de l'algorithme

4. Cette question traite de l'état de la pile lors du déroulement de l'algorithme.
  - a. Représenter la pile à chaque étape du déroulement de cet algorithme pour l'expression "<p><em></em></p>" (balisage correct).
  - b. Indiquer quelle condition simple (sur le contenu de la pile) permet alors de dire que le balisage est correct lorsque toute l'expression HTML simplifiée a été entièrement parcourue, sans que l'analyse ne s'arrête.
5. Une expression HTML correctement balisée contient 12 balises.  
Indiquer le nombre d'éléments que pourrait contenir au maximum la pile lors de son analyse.

## Exercice 2 : Arbres binaires (6 points)

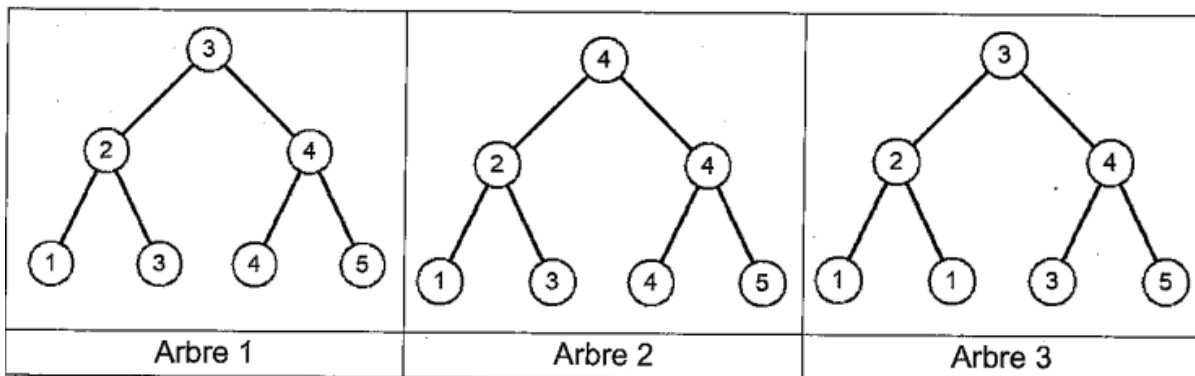
Cet exercice porte sur le thème "Algorithmique", les arbres binaires de recherche et leurs parcours.

**Rappel :** Un arbre binaire de recherche (ABR) est un arbre binaire étiqueté avec des clés tel que :

- Les clés du sous arbre gauche sont inférieures ou égales à celle de la racine ;
- Les clés du sous arbre droit sont strictement supérieures à celle de la racine ;
- Les deux sous arbres sont eux-mêmes des arbres binaires de recherche.

### Partie A : Préambule

- 1) Recopier sur votre copie le ou les numéro(s) correspondant aux arbres binaires de recherche parmi les arbres suivants :



- 2.a) Dans un ABR, où se trouve le plus petit élément ? Justifier.

Pour rechercher une clé dans un ABR, il faut comparer la clé donnée avec la clé située à la racine. Si cette clé est à la racine, la fonction renvoie vrai sinon il faut procéder récursivement sur les sous arbres à gauche ou à droite.

3. On considère l'ABR ci-contre :

- 3.a) Dire à quel type de parcours correspond le résultat suivant où les clés sont triées dans l'ordre croissant :

1 - 2 - 3 - 5 - 6 - 7 - 8 - 9 - 10

- 3.b) Donner le parcours préfixe de cet arbre.  
3.c) Donner le parcours suffixe de cet arbre.  
3.d) Donner le parcours en largeur de cet arbre.

