

---

T.D. N°3

---

**Exercice 1 :**

On veut évaluer à laide d'une pile les expressions suivantes :

$$R1 = (((A + B) * C) + D) \text{ avec } A = 1, B = 2, C = 4, D = 3$$

$$R2 = ((A + B) * (C - D)) \text{ avec } A = 4, B = 2, C = 5, D = 3$$

On appelle expression un ensemble de valeurs situées à droite d'une affectation et reliées par des opérateurs. On appelle sous expression une partie d'expression délimitée par un couple de parenthèse. Lors de la lecture de gauche à droite d'une expression une pile est gérée parallèlement avec des règles d'évaluation suivantes :

- Si opérateur, on l'empile,
- Si variable, on empile sa valeur,
- Si parenthèse gauche, on l'empile (début de sous expression),
- Si parenthèse droite, on dépile la sous expression jusqu'à une parenthèse gauche, on évalue la sous expression puis, on empile le résultat.

1. Adresser un tableau qui clarifie les états de la pile d'évaluation des expressions précédentes.
2. Déterminer pour chaque expression le résultat final que contient la pile.

**Exercice 2 :**

On considère une file illimitée contenant des formes. Ces formes, étant identifiés par leurs noms et leurs surfaces, se mettent en file d'attente, de façon que la première forme de la file soit la première à être affichée.

1. Donner les structures de données nécessaires à cette file.
2. Écrire une fonction qui permet d'ajouter une forme dans cette file.
3. Écrire une fonction qui permet de supprimer une forme de cette file.

**Exercice 3 :**

1. Représenter l'arbre binaire de recherche, initialement vide, obtenu en insérant successivement les entiers suivants : **7, 2, 9, 0, 5, 6, 8** et **1**.
2. Afficher le niveau de chaque noeud.
3. Afficher le contenu de cet arbre par le parcours **infixe**.
4. Afficher le contenu de cet arbre par le parcours **préfixe**.
5. Afficher le contenu de cet arbre par le parcours **postfixe**.
6. Donner la représentation graphique de l'arbre obtenu après l'insertion de l'entier **3**.
7. Déterminer la hauteur de l'arbre obtenu.