Ne rien inscrire dans ce cadre

Prénom : .....

Nom : .....

Promotions : Systèmes, Réseaux et Cloud Computing/ Architectures du logiciel

Année : 3

Module : Algorithmique avancée : listes, tris et arbres

Type de l'épreuve : DE

Session : Principale

Date : 20/07/2023 **Durée :** 2h

Sujet proposé par : **Maha NACEUR**

Calculatrice autorisée : **NON**

Documents autorisés : **NON**

Ordinateur portable autorisé : **NON**

Internet : **NON**

Traducteur électronique, dictionnaire : **NON**

Consigne :

- Merci de restituer uniquement : **« Votre copie »**
- **Répondre directement sur la feuille d'examen**

Rappel :

- Tous les appareils électroniques (téléphones portables, ordinateurs, tablettes, montres connectées …) doivent être éteints et rangés.
- Il est interdit de communiquer.
- Toute fraude ou tentative de fraude fera l'objet d'un rapport de la part du surveillant et sera sanctionnée par la note zéro, assortie d'une convocation devant le conseil de discipline. Aucune contestation ne sera possible. Tous les documents et supports utilisés frauduleusement devront être remis au surveillant.
- Aucune sortie de la salle d'examen ne sera autorisée avant la moitié de la durée de l'épreuve.

Instructions : Les algorithmes peuvent être écrits en pseudo-code ou avec un script python.

Exercice 1 (2 points) :

On ajoute, dans cet ordre, les valeurs A, B, C, D, E et F à une structure linéaire vide.
Pour chacun des ordres de sortie donnés, indiquer si la structure en question peut être : une pile, une file (ce peut être les deux), ou aucune des deux (ni une pile, ni une file).

	pile	file	aucune
$A\ B\ C\ D\ E\ F$			
$F\ E\ D\ C\ B\ A$			
$C\ B\ E\ F\ D\ A$			
$C\ E\ D\ F\ A\ B$			

Exercice 2 (6 points) :

Ecriture polonaise inversée

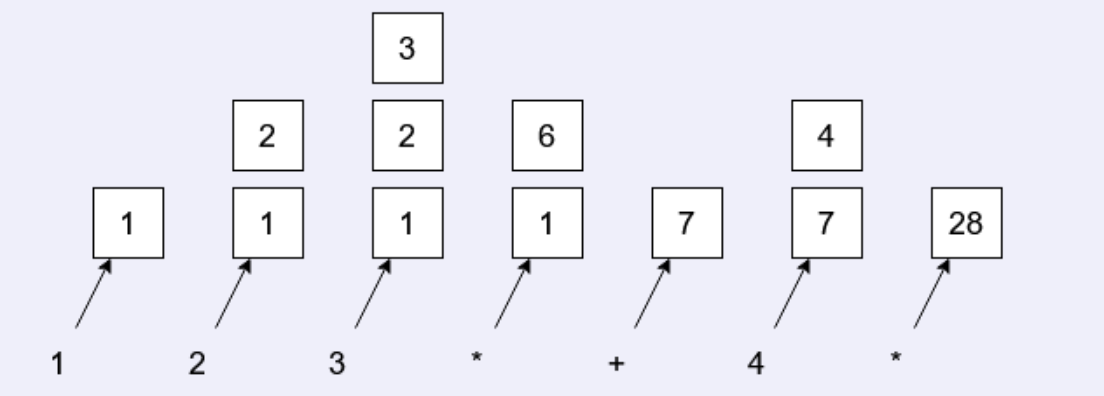
L'écriture polonaise inversée des expressions mathématique place l'opérateur après les opérandes. Cette notation ne nécessite aucune parenthèse ni aucune règle de priorité.

Exemple : **(1 + 2 * 3) * 4** s'écrit 1 2 3 * + 4 *

Pour évaluer une telle expression, on utilise une pile pour stocker tous les résultats intermédiaires. Les éléments de l'expression sont évalués un à un :

- S'il s'agit d'un nombre, on le place dans la pile
- S'il s'agit d'une opération, on réalise l'opération avec les deux premiers nombres de la pile (le deuxième par le premier), puis on replace le résultat sur la pile.

Si l'expression est bien écrite, à la fin du processus, la pile ne comporte plus que le résultat final.



Écrire une fonction **rpn(expr)** prenant en paramètre une chaine de caractères expr et retournant le résultat de l'expression. Cette fonction permet d'évaluer une expression polonaise inverse composée des opérateurs **+**, **-**, ***** et **/** dont les éléments sont séparés par des espaces.

N.B : On supposera que l'expression est bien écrite.
Pour écrire cette fonction, vous disposez de l'implémentation suivante d'une pile :

Interface d'une pile

PileVide : $() \rightarrow Pile$
Empiler : $Element \times Pile \rightarrow Pile$

EstVide : $Pile \rightarrow bool$
Sommet : $Pile \rightarrow Element$
Dépiler : $Pile \rightarrow Pile$

En Python avec des listes :

```
def PileVide(): return []
def Empiler(x,p): p.append(x)
def EstVide(p): return (p == [])
def Sommet(p): return p[-1]
def Depiler(p): p.pop()
```

Vous pouvez utiliser les fonctions ci-dessus sans les implémenter

BONUS (1 point) : Si l'expression est mal écrite, la fonction doit renvoyer une erreur de type SyntaxError.

Exercice 4 (6 points) :

À partir d'un arbre vide, construire l'AVL en insérant successivement les valeurs 25, 60, 35, 10, 20, 5, 70, 65.

Vous dessinerez l'arbre à deux étapes :

— après l'insertion de 20 ;

— l'arbre final.

Indiquez quelles rotations ont été effectuées, dans l'ordre

Exemple : si une rotation gauche a été effectuée sur l'arbre de racine 42, indiquer rg(42).

Arbre créé par insertions de 25, 60, 35, 10, 20 :	*Rotations :*
Arbre après ajout de 5, 70, 65 :	*Rotations :*

Exercice 4 (6 points) :

Concevoir un algorithme récursif qui renvoie vrai si un arbre est un **arbre binaire de recherche** et faux sinon.

Bonne chance