AL 4I - Algorithmique avancée - Contrôle continu

Nom : _____

Prénom : _____

- Durée : 1 heure. Documents interdits, calculatrices interdites.
- Écrivez toutes les réponses directement sur le sujet. Si vous n'avez pas suffisamment de place, écrivez au dos d'une feuille en le **précisant dans la question**.
- En cas de plagiat, l'original et le plagiaire auront **tous les deux** la note $\frac{0}{20}$.

1. 3 points Ecrire une fonction $somme(a)$ retournant la somme des éléments se trouvant dans les noeuds d'un arbre binaire a.

Solution:

```
Fonction somme(a)
    si a = null alors
        retourner 0
    sinon
        a.cle + somme(a.g) + somme(a.d)
    fin
FIN
```

2. 3 points Ecrire une procédure $afficheVersFeuilles(a)$ affichant tous les noeuds de a de sorte que la racine soit affichée en premier et les feuilles en dernier.

Solution:

```
Fonction afficheVersFeuilles(a)
    si a ≠ null alors
        Afficher a.cle
        afficheVersFeuilles(a.g)
        afficheVersFeuilles(a.d)
    fin
FIN
```

3. 3 points Ecrire une fonction $sousArbreGauche(a)$ laissant inchangés les valeurs des feuilles, et affectant à chaque noeud interne la somme des valeurs des noeuds du sous-arbre gauche. $sousArbreGauche(a)$ retourne la somme des éléments se trouvant dans a.

Solution:

```
Fonction sousArbreGauche(a)
    si a = null alors
        retourner 0
    fin
    si a.g = null et a.d = null alors
        retourner a.cle
    sinon
        a.cle ⟵ sousArbreGauche(a.g)
        retourner a.cle + sousArbreGauche(a.d)
    fin
FIN
```

4. [3 points] Ecrire une fonction $copieNoeudsInternes(a)$ copiant tous les noeuds de a mais pas ses feuilles. $copieNoeudsInternes(a)$ retourne la racine du nouvel arbre.

Solution:

```
Fonction copieNoeudsInternes(a)
    si a = null ou (a.g = null et a.d = null) alors
        retourner null
    sinon
        retourner AB(copieNoeudsInternes(a.g), a.cle,
        copieNoeudsInternes(a.d))
    fin
FIN
```

5. [4 points] Ecrire une fonction $supprimeRacine(a)$ supprimant la racine de a et la remplaçant par son fils gauche, le fils gauche étant à son tour remplacé par son fils gauche et ainsi de suite. Cette fonction retourne la racine de l'arbre obtenu après la suppression.

Solution:

```
Fonction supprimeRacine(a)
    si a = null alors
        retourner null
    sinon
        r ⟵ a.g
        a.g ⟵ supprimeRacine(a.g)
        r.g ⟵ a.g
        r.d ⟵ a.d
        retourner r
    fin
FIN
```

6. [4 points] Ecrire une fonction $profondeurMax(a, x)$ retournant la profondeur du noeud de valeur x dans l'arbre a. 0 si x est la racine, -1 s'il ne se trouve pas dans a. S'il y a plusieurs occurrences de x, c'est la profondeur de celle qui est la plus éloignée de la racine qui est retournée.

Solution:

Fonction $profondeurMax(a, x)$
| **si** $a = null$ **alors**
| | **retourner** -1
| **fin**
| $r \longleftarrow$
| $max(profondeurMax(a.g, x), profondeurMax(a.d, x))$
| **si** $r = -1$ *et* $a.cle \neq x$ **alors**
| | **retourner** -1
| **sinon**
| | **retourner** $r + 1$
| **fin**
FIN