

Nom :

Prénom :

*Aucun document n'est autorisé***Exercice 1 Evaluation d'une expression arithmétique**

On suppose donné un arbre binaire représentant une expression arithmétique contenant des entiers *positifs ou nuls* et des opérateurs binaires + et *. Si l'expression est bien formée (correcte) alors l'arbre binaire est complet. Chaque feuille de cet arbre contient donc un entier et chaque noeud interne un opérateur (l'un des symboles + ou *, codé par -1 et * par -2, ainsi tous les sommets de l'arbre contiennent des entiers.)

Ecrire une fonction `valeurExpression(val A: arbreBinaire)` qui évalue la valeur de l'expression représentée par un arbre binaire *si l'arbre est complet* sinon renvoie NIL, en visitant *une et une seule fois* tous les noeuds de l'arbre (un seul parcours de l'arbre).

Exercice 2 Arbres planaires

Ecrire une fonction de parcours par niveaux (parcours hiérarchique) d'un arbre planaire en utilisant une file.

ANNEXE A Type abstrait *arbreBinaire*

```
arbreBinaire= curseur;  
sommet= curseur;
```

– Création

```
fonction creerArbreBinaire(val Racine:objet):sommet;
```

– Accès

```
fonction getValeur(val S:sommet):objet;  
fonction filsGauche(val S:sommet):sommet;  
fonction filsDroit(val S:sommet):sommet;  
fonction pere(val S:sommet):sommet;
```

– Modification

```
fonction setValeur(ref S:sommet, val x:objet):vide;  
fonction ajouterFilsGauche(ref S:sommet, val x:objet):vide;  
fonction ajouterFilsDroit(ref S:sommet, x:objet):vide;  
fonction supprimerFilsGauche(ref S:sommet):vide;  
fonction supprimerFilsDroit(ref S:sommet):vide;  
fonction detruireSommet(ref S:sommet):vide;
```

ANNEXE B Type abstrait *sommetArbrePlanaire*

```
sommetArbrePlanaire= curseur;
```

– Création

```
fonction creerArbrePlanaire(val Racine:objet):sommetArbrePlanaire;
```

– Accès

```
fonction getValeur(val S:sommetArbrePlanaire):objet;  
fonction premierFils(val S:sommetArbrePlanaire):sommetArbrePlanaire;  
fonction frere(val S:sommetArbrePlanaire):sommetArbrePlanaire;  
fonction pere(val S:sommetArbrePlanaire):sommetArbrePlanaire;
```

– Modification

```
fonction setValeur(ref S:sommetArbrePlanaire, val x:objet):vide;  
fonction ajouterFils(ref S:sommetArbrePlanaire, val x:objet):vide;  
fonction supprimerSommet(ref S:sommetArbrePlanaire):vide;  
fonction detruireArbrePlanaire(ref S:sommetArbrePlanaire):vide;
```