

Algorithmique 1

Interrogation 3 groupe CSB3A2

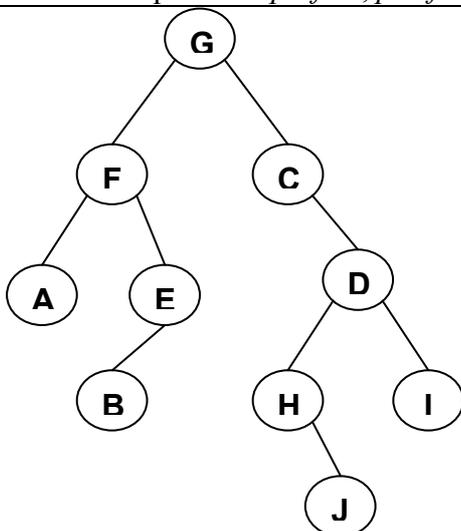
NOM :	Prénom
-------	--------

Exercice 1

Ecrire une fonction qui affiche les valeurs des sommets d'un arbre planaire dans le parcours par niveau. On utilisera une file de sommets.

Exercice 2

Donner les parcours *préfixe*, *postfixe* et *infixe* de l'arbre binaire suivant :



<i>Préfixe :</i>
<i>Postfixe :</i>
<i>Infixe :</i>

Cet arbre binaire peut-il représenter un arbre planaire ? Si OUI dessinez-le, si NON expliquez pourquoi.

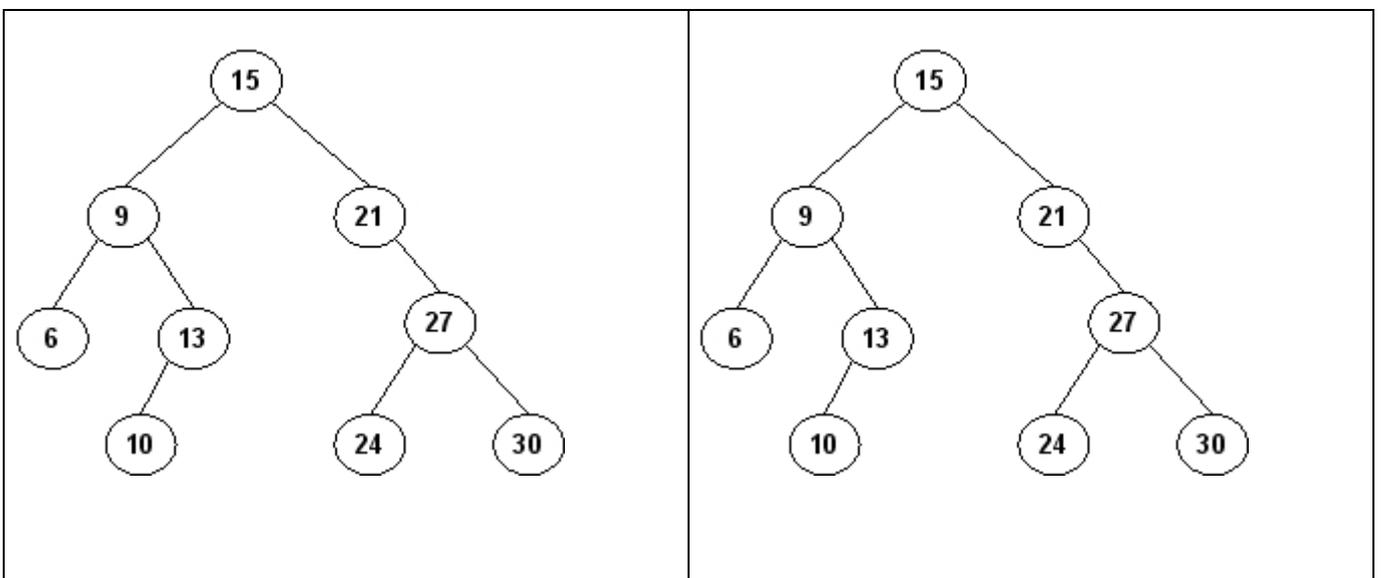
Exercice 3

1- Ecrire une fonction `nbOccurrences` qui, étant donné `a` un arbre binaire contenant des entiers et un entier `x`, renvoie le nombre de sommets de `a` dont la valeur est égale à `x`.

Exercice 4

Ajouter la valeur **23** à l'ABR ci-dessous

Ajouter la valeur **13** à ABR ci-dessous



Dessiner l'arbre **initial** après suppression de **21**

Dessiner l'arbre **initial** après suppression de **9**

--	--

Exercice 5

Ecrire une fonction `nbElemSupérieurs` qui, étant donné `a` un arbre binaire de recherche contenant des entiers et un entier `x`, renvoie le nombre de sommets de `a` dont la valeur est strictement supérieure à `x`.

Annexe A Type abstrait *sommetArbrePlanaire*

– Création

fonction creerArborescence(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;

Annexe B Implémentation du type abstrait *sommetArbrePlanaire*

```
cellule= structure
    info: objet;
    premierFils: sommet
    frere: sommet;
    pere: sommet
finstructure
```

```
sommet= ^cellule;
sommetArbrePlanaire= sommet;
```

Annexe C Type abstrait *arbreBinaire*

– 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 D Implémentation du type abstrait *arbreBinaire*

```
cellule=structure
    info:objet;
    gauche: sommet;
    droit: sommet;
    pere: sommet;
finstructure
```

```
sommet=^cellule;
arbreBinaire= sommet;
```