

J1IN3001 ASD1 Devoir surveillé n°2 Durée : 30mn Aucun document autorisé 12/11/2013	Nom :  Prénom :
---	-----------------------

**Exercice 1 :**

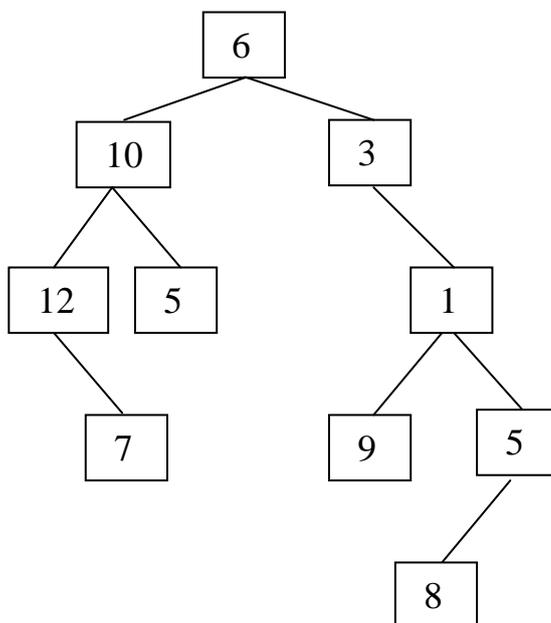


Figure 1

1. On considère l'arbre binaire de la figure 1 ci-dessus. Ecrire les 3 mots suivants :

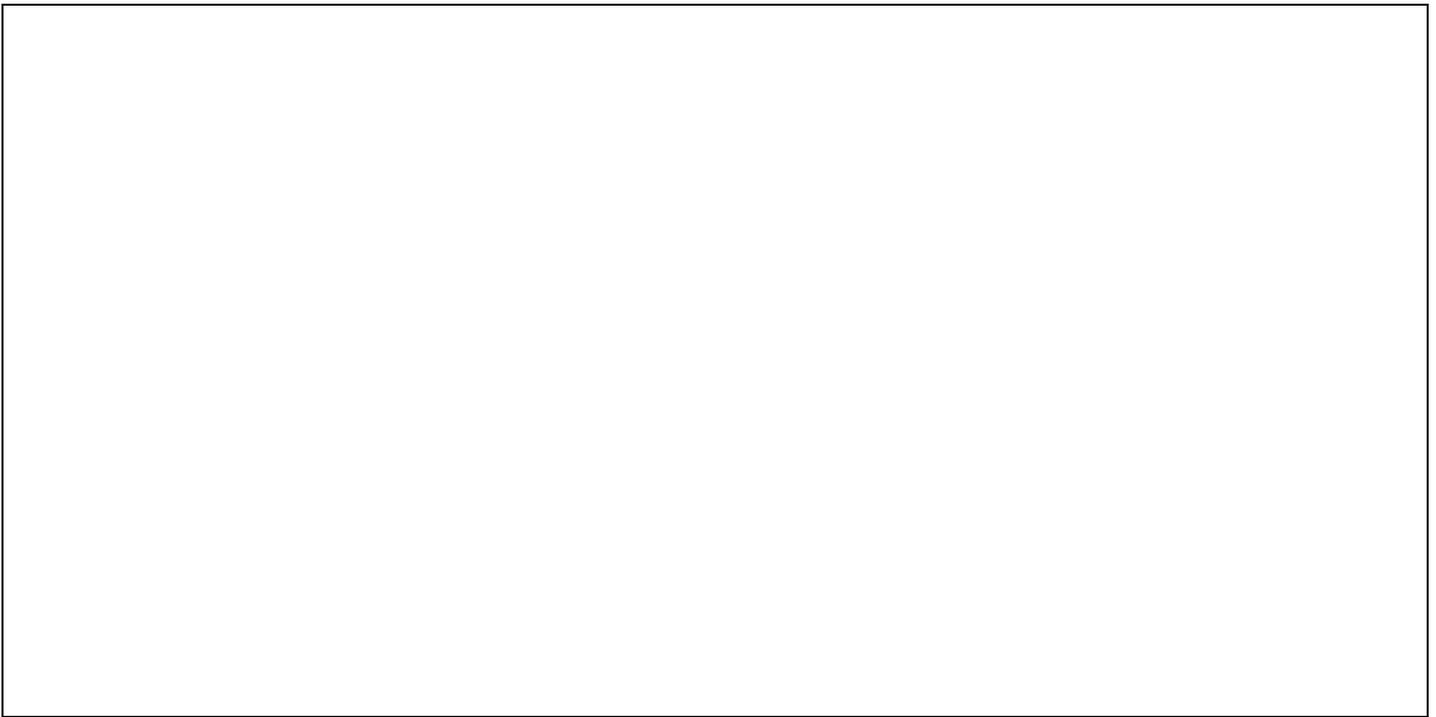
Mot Préfixe :
Mot Infixe :
Mot Suffixe :

2. Donnez le mot produit par le parcours hiérarchique de l'arbre.

--

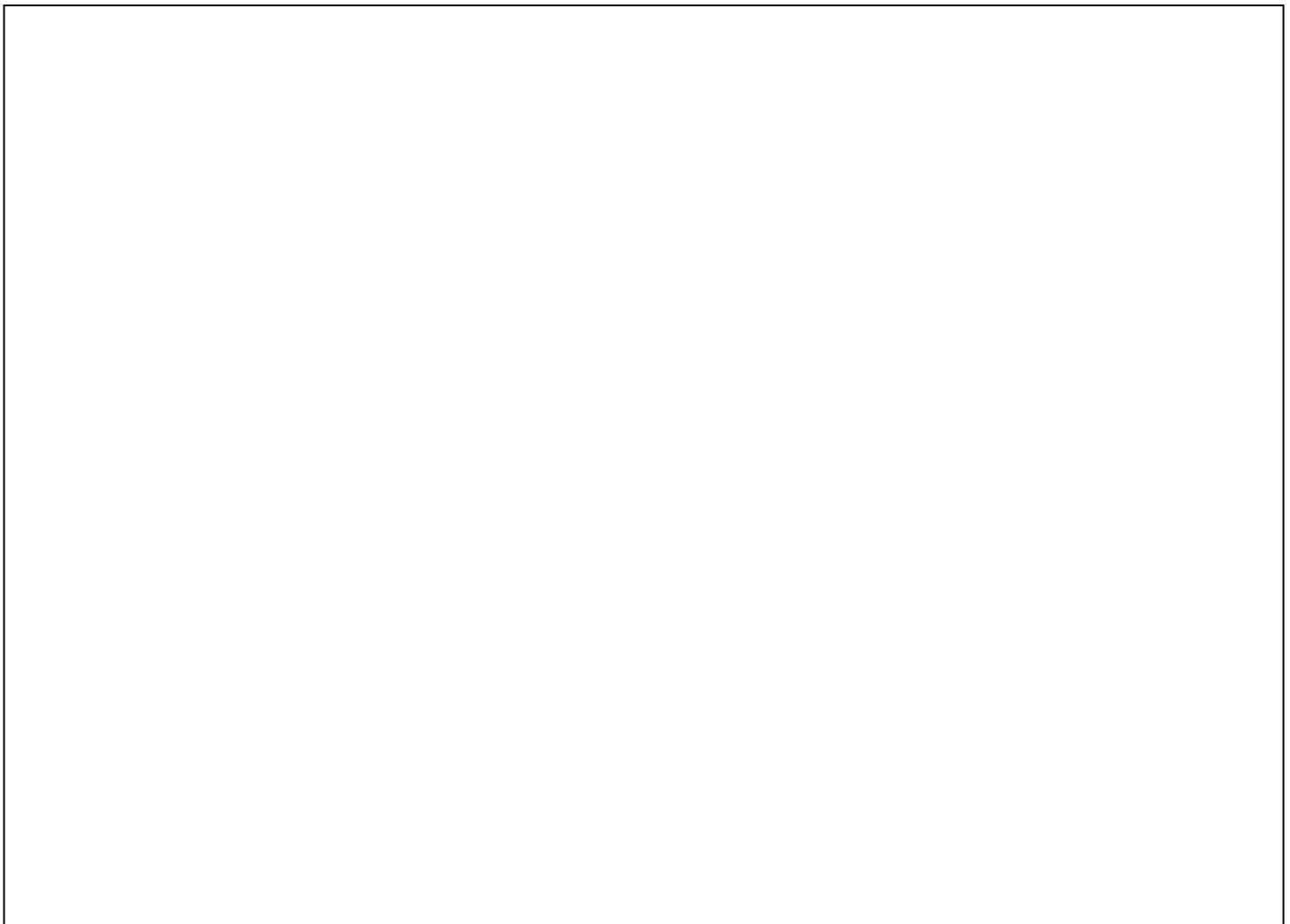
**Exercice 2 :** (Examen Janvier 2013)

Ecrire une fonction qui prend en entrée une file d'entiers et qui fournit en sortie une file telle que tous les nombres multiples de 3 se retrouvent en tête de la file et dans le même ordre que dans la file originale, les autres éléments sont dans la file en ordre inverse. La file de départ reste inchangée. Par exemple, [1,3,5,4,2,6,8] fournit en sortie en [3,6,8,2,4,5,1].



**Exercice 3:**

En utilisant un des algorithmes de parcours pour les arbres binaires (cf. Annexe B) écrire une fonction qui prend en paramètre un arbre binaire et retourne la hauteur minimum pour les feuilles. C'est-à-dire la hauteur de la feuille qui est la plus proche de la racine. Par exemple sur l'arbre de la figure1 la fonction retournera 2 qui est la hauteur de la feuille de valeur 5.



## **Annexe A :**

### **Files**

```
fonction valeur(ref P:pile de objet):objet;  
fonction pileVide(ref P:pile de objet):booléen;  
fonction créerPile(P:pile de objet) :vide  
fonction empiler(ref P:pile de objet;val x:objet):vide;  
fonction dépiler(ref P:pile de objet):vide;  
fonction detruirePile(ref P:pile de objet):vide;
```

### **Files**

```
fonction valeur(ref F:file de objet):objet;  
fonction fileVide(ref F:file de objet):booléen;  
fonction créerFile(F:file de objet);vide;  
fonction enfiler(ref F:file de objet;val x:objet):vide;  
fonction défiler (ref F:file de objet):vide;  
fonction detruireFile(ref F:file de objet):vide;
```

### **Arbres binaires**

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

### **Arbres planaires**

```
fonction valeur(val S:sommetArbrePlanaire):objet;  
fonction premierFils(val S:sommetArbrePlanaire):sommetArbrePlanaire;  
fonction frere(val S:sommetArbrePlanaire):sommetArbrePlanaire;  
fonction pere(val S:sommetArbrePlanaire):sommetArbrePlanaire;  
fonction créerArborescence(val racine:objet):sommetArbrePlanaire;  
fonction ajouterFils(ref S:sommetArbrePlanaire,val x:objet):vide;  
fonction supprimerSommet(ref S:sommetArbrePlanaire):vide;  
fonction detruireArborescence(val racine:objet):sommetArbrePlanaire;
```

## **Annexe B:**

```
fonction parcoursHierarchiqueAB(val A : AB_de_objet) : vide ;
    var s : sommet
        F : file de sommets
        //Declarations locales
    debut
        creerFile(F);
        enfiler(F,A);
        tantque !fileVide(F) faire :
            s=valeurFile(F)
            defiler(F);
            //Traitement s//
            si filsGauche(s) != NIL alors
                enfiler(F,filsGauche(s));
            si filsDroit(s) != NIL alors
                enfiler(F,filsDroit(s));
        finTantQue
    fin

fonction parcoursProfondeurAB (ref A: AB d'objet):vide;
// Déclarations locales
début
// T1;
si estFeuille(A) alors
// T2
sinon
// T3;
si filsGauche(A) !=NIL alors
// T4;
parcoursProfondeurAB (filsGauche(A));
// T5;
finsi
// T6;
si filsDroit(A) !=NIL alors
// T7
parcoursProfondeurAB (filsDroit(A));
// T8;
finsi
// T9;
finsi
// T10;
fin
```