L2
UFR de Mathématiques et Informatique
Université de Paris
F. Cloppet

TD-TP n° 12 Programmation Impérative
TDA - Arbres

Exercice 1:
Un arbre binaire est un ensemble fini de nœuds qui est soit vide soit constitué d'une racine et de 2 arbres binaires disjoints, appelés le sous-arbre gauche et le sous-arbre droit. Nous désirons coder un arbre binaire avec une représentation chaînée où chaque nœud de l'arbre stockera un caractère (fourni par l'utilisateur), son numéro de création (calculé automatiquement), l'adresse du nœud représentant son fils gauche et l'adresse du nœud représentant son fils droit.

a) Ecrire la structure de données **nœud** la plus adaptée pour représenter un nœud de l'arbre.

b) Ecrire une fonction **nouvNoeud** qui prend en paramètre un caractère **carac** permet d'allouer de l'espace mémoire pour un nœud et d'initialiser les champs du nœud (val avec **carac**, num avec le n° de création du nœud automatiquement incrémenté à chaque appel de **nouvNoeud**, et le fils gauche et le fils droit à NULL). Cette fonction devra renvoyer le nœud créé et initialisé.

c) On suppose que **vous disposez** d'une fonction **RechercheNoeud** qui prend en paramètre un arbre(ou sous-arbre) et le numéro du nœud recherché et qui renvoie le nœud (dont le champ contenant le n° de création est égal au numéro de nœud passé en paramètre).
Le code de la fonction **rechercheNoeud** est le suivant :

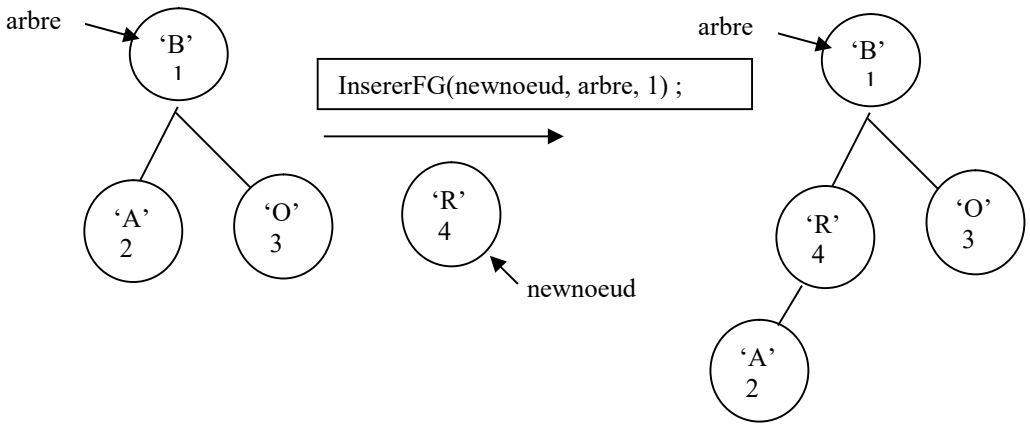
```
nœud * rechercheNoeud( nœud *n, int numNœud){
    noeud  *tmpNoeud;

    if(n == NULL)
            return(NULL);
    if(n->num == numNoeud)
            return(n);

    tmpNoeud = rechercheNoeud(n->filsG, numNoeud);
    if(tmpNoeud!=NULL)
            return(tmpNoeud);

    return(rechercheNoeud(n->filsD, numNoeud));
}
```

Ecrire une fonction **insererFG**, qui prend en paramètres le nœud à insérer, l'arbre (nœud qui correspond à la racine) dans lequel il doit être inséré et le numéro du nœud sous lequel il doit être inséré en fils gauche. Cette fonction ne renvoie rien.
Rq : cette fonction fera appel à la fonction **rechercheNoeud** pour trouver le nœud où le nœud doit être inséré en fils gauche. Si celui-ci possède déjà un fils gauche alors il deviendra le fils gauche du nœud inséré.



d) Le parcours préfixe d'un arbre revient à partir de la racine de l'arbre, à "visiter" le nœud rencontré et à parcourir la branche gauche du nœud en visitant tous les nœuds rencontrés. Ce parcours continue jusqu'à un nœud NULL. A ce stade, on retourne à l'ancêtre le plus proche qui a un fils droit et on continue ….
Ecrire une fonction récursive **parcoursPrefixe** qui prend en paramètre l'adresse d'un nœud (racine d'un arbre ou sous-arbre), et ne renvoie rien. Cette fonction affichera le caractère contenu dans chacun des nœuds visités.

En supposant que l'on dispose d'une fonction **insererFD** (vous pouvez la coder s'il vous reste du temps), et que l'on insère comme fils droit du nœud de numéro 4 (sur la figure précédente), un nœud contenant la lettre 'V', dites ce qui serait affiché à l'écran après l'appel de la fonction **parcoursPrefixe**.