

TP 3 : Arbres binaires de recherche équilibrés en hauteur

travail à réaliser sur 2 séances (6h)

Des références de produits sont représentées par des valeurs entières positives distinctes. Ces références sont stockées dans un arbre binaire de recherche équilibré déclaré ainsi :

```
typedef struct _produit
{
    int ref; // valeur de la reference produit
    int eq; // valeur du facteur d'equilibre
    struct _produit * fg, * fd; // pointeurs vers les noeuds fils
} Produit, * AVL;
```

Ecrire les fonctions suivantes :

1. une fonction **main** dans laquelle vous déclarerez un AVL *a* initialisé à NULL et un tableau *t* de *N* entiers, initialisé avec les références {4,2,1,8,6,7,3,9,5}. Ce tableau sera utilisé pour tester les fonctions suivantes sur *a*.
2. une fonction **afficher** pour afficher un AVL de racine *r* sous une forme d'arbre, comme ci-dessous. Les facteurs d'équilibre sont entre parenthèses après la référence.

```
----9 (0)
--8 (0)
----7 (0)
6 (1)
-----5 (0)
----4 (0)
-----3 (0)
--2 (-1)
----1 (0)
```

On vous donne la fonction **inserer_reeq** ci-dessous, qui insère une nouvelle référence *n* à sa place dans un AVL de racine *a* et rééquilibre l'arbre si nécessaire. Compléter en écrivant les fonctions :

3. **creer** : crée et renvoie un AVL sur un nouveau Produit de référence *n*,
4. **maj_eq** : met à jour les facteurs d'équilibre d'un AVL, récursivement à partir de la racine,
5. **reequilibre** : rééquilibrage global (voir TD3). Cette fonction fait appel à des fonctions de rotation (à écrire aussi).

```
AVL inserer_reeq(AVL a, int n)
{
    AVL p=NULL;

    if (a)
    {
        if (n <= a->ref) a->fg=inserer_reeq(a->fg,n);
        else a->fd=inserer_reeq(a->fd,n);
        maj_eq(a);
        if ((a->eq < -1)|| (a->eq > 1)) a= reequilibre(a);
        return a;
    }
    else
    {
        p=creer(n);
        return p;
    }
}
```

6. Tester avec les valeurs du tableau *t*. L'affichage de l'arbre obtenu avec la suite de valeurs de *t* insérées dans l'ordre avec rééquilibrage à chaque insertion de valeur donne l'arbre ci-dessus.