



TP4: arbre

1. Récupérez sur le site de ASD2 les fichiers `arbre.h` et `arbre.c`. Créez un nouveau fichier `main.c` et écrivez-y un « `main()` » vide pour l'instant. Complétez le fichier `arbre.c` en y ajoutant les définitions des fonctions et procédures suivantes :

- `initialiserArbre`
- `insererElement`
- `rechercheElement`
- `hauteur`
- `testamentArbre`

Remarque: Pour certains de ces sous-programmes, vous devrez utiliser une fonction ou procédure auxiliaire travaillant sur un sous-arbre, et prenant donc comme paramètre supplémentaire l'adresse du nœud dans lequel le sous-arbre est enraciné. Ces sous-programmes auxiliaires restent internes au module `Arbre` : ils n'apparaissent donc pas dans le `.h`, et sont précédés du mot-clé **static** dans le `.c`.

Dans le fichier `main.c`, complétez le `main()` pour qu'il insère dans un arbre binaire de recherche 255 entiers aléatoires compris entre 1 et 100000, puis qu'il calcule la hauteur maximale de l'arbre et sa profondeur moyenne. Le programme devra bien entendu se terminer proprement, c'est-à-dire en vidant l'arbre. Testez que tout fonctionne bien, en utilisant notamment la procédure d'affichage d'arbre qui vous est fournie. Vérifiez que vous n'obtenez pas le même arbre si vous exécutez deux fois le programme.

Aide : pour le tirage de nombres aléatoires, vous pouvez vous inspirer du petit programme suivant :

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#define VALEUR_MAX 100000
int main()
{
    int alea1, alea2;
    /* Initialisation du générateur, à ne faire qu'une fois
    dans le programme. En l'initialisant avec l'heure
    à laquelle l'exécution est lancée, on aura des valeurs
    différentes à chaque exécution, ce qui est souhaitable ! */
    srand((unsigned) time(NULL));
    /* Tirage de deux nombres aléatoires compris entre 0 et
    VALEUR_MAX incluse */
    alea1 = rand()%(VALEUR_MAX + 1);
    alea2 = rand()%(VALEUR_MAX + 1);
    printf("%d %d\n", alea1, alea2);
    return 0;
}
```

3. Complétez le `main` pour qu'il recherche 100 nombres aléatoires entre 0 et 100000 dans l'arbre (bien évidemment, parmi ces 100 nombres, certains seront effectivement présents dans l'arbre et d'autres non). En plus de la hauteur maximale et de la profondeur moyenne de l'arbre, le `main` devra afficher le nombre moyen de nœuds visités par opération de recherche.