

# Algorithmes et Structures de Données

**Mardi 16 Janvier 2018**

Durée 3H – Cours et TD **NON** autorisés

## Exercice 1 : Algorithme d'Euclide (6 pts)

Pour calculer le PGCD de deux entiers naturels non nuls  $a$  et  $b$  ( $a > b$ ) avec l'algorithme d'Euclide, on utilise la séquence d'instructions suivante :

- (1) Diviser  $a$  par  $b$  ; obtenir le reste  $r$ .
- (2) Si  $r = 0$ , alors l'algorithme se termine :  $\text{PGCD}(a, b) = b$ .
- (3) Si  $r \neq 0$ , remplacer  $a$  par  $b$ ,  $b$  par  $r$  et recommencer à partir de (1).

1.2. Ecrire en pseudo-langage la **fonction itérative** qui réalise ce calcul.

1.2. Ecrire en pseudo-langage la **fonction récursive** qui réalise ce calcul.

## Exercice 2 : Position d'une liste chaînée dans une autre (4 pts)

On souhaite écrire une fonction qui retourne la position d'une liste chaînée  $L2$  dans une autre  $L1$  ( $L1$  et  $L2$  sont de type `liste`) ; 0 si  $L2$  n'apparaît pas dans  $L1$ .

```
Type liste : ^cellule
      cellule : Enregistrement
                val : entier
                suiv : ^cellule
                FinEnregistrement
```

Ecrire en pseudo-langage la **fonction itérative** qui réalise cette opération.

## Exercice 3 : Pointeurs - Les vins en France (10 pts)

On souhaite construire une base de données des régions de France avec leurs vins.

```
Type tab-région : tableau[1..18] de région      {18 régions en France}
      région = Enregistrement
                nomr : chaîne                    {nom de la région}
                appel : ^vin                     {liste des appellations}
                FinEnregistrement
      vin = Enregistrement
            nomv : chaîne                        {nom du vin}
            tcep : tableau [1..5] de ^cépage    {pas + de 5 cépages par vin}
            vin-suiv : ^vin                     {pointeur sur le vin suivant}
            FinEnregistrement
      cépage : Enregistrement
                nomc : chaîne                    {nom du cépage}
                FinEnregistrement
```

3.1. Faire un dessin de `tr` de type `tab-région` avec les données suivantes : 3 régions avec les appellations de vins (origine d'un vin) et leurs cépages (entre parenthèses), qui sont des types de vigne.

- Grand Est : Gewurztraminer (Gewurztraminer), Riesling (Riesling)
- Bourgogne-Franche-Comté : Bourgogne Aligoté (Aligoté, Chardonnay), Chablis (Chardonnay), Côte de nuit rouge (Pinot Noir)

- Nouvelle Aquitaine : Graves rouges (Cabernet Sauvignon, Merlot), Médoc (Cabernet Sauvignon, Merlot), Saint Estèphe (Cabernet Sauvignon, Merlot), Saint Julien (Cabernet Sauvignon, Merlot)

3.2. Ecrire en pseudo-code une **procédure** qui initialise le tableau `tr` de type `tab-région` à partir d'un fichier texte contenant les régions **triées dans l'ordre alphabétique** (une région par ligne dans le fichier). Ces régions n'ont pas encore d'appellation attribuée (mettre les pointeurs à nil).

3.3. Ecrire en pseudo-code une **procédure** qui initialise la liste des appellations d'une région donnée à partir d'un fichier texte contenant les appellations des vins **triés dans l'ordre alphabétique** (une appellation par ligne dans le fichier). Ces appellations n'ont pas encore de cépages attribués (mettre les pointeurs à nil). **Faire des dessins.**

3.4. Ecrire en pseudo-code une **procédure** qui ajoute un cépage (les tableaux de cépages sont **triés dans l'ordre alphabétique**) à une appellation de vin donnée et une région donnée. **Faire des dessins.**

3.5. Ecrire en pseudo-code une **procédure** qui écrit à l'écran les appellations qui contiennent un cépage donné.