

# LIFAP1 – Séquence 1

## Sujet B

Contrôle Continu – TP10 - durée 1h30 mn

Mardi 12 décembre 2017

### Consignes

Aucun accès au WEB, aux pages de l'UE, ni à vos anciens TP n'est autorisé. Téléphones portables interdits.

Vous devrez compiler et tester votre programme. Vous donnerez votre nom au fichier source.

Dans votre fichier, vous mettrez en commentaire vos nom et prénom ainsi que votre numéro d'étudiant et le sujet qui vous a été distribué (A ou B).

La note tiendra compte du respect des consignes, de la qualité de la présentation et de la lisibilité du code, des algorithmes, et du bon fonctionnement du programme.

Une fois le programme terminé et testé (ou à la fin du temps imparti), vous devrez déposer le fichier source (.cpp) via **TOMUSS** (en cliquant sur "déposer" dans la case Depot\_TP10 de l'UE LIFAP1).

### Travail à réaliser

On souhaite créer une petite application de gestion d'une pépinière. Un arbre est identifié par les informations suivantes : un `nom`, un `type` (feuillus, résineux), une `hauteur` et un `âge`. Une pépinière contient quant à elle un tableau d'au maximum `MAXARBRE` arbres et le nombre d'arbres de la plantation.

Les sous-programmes demandés dans les questions suivantes doivent être écrits en C/C++ et devront être testés au fur et à mesure.

- 1- Définir deux constantes `MAXCHAR` (qui sera utilisée pour toutes les chaînes de caractères) et `MAXARBRE` (nombre maximum d'arbres dans la pépinière) ayant pour valeurs respectives 64 et 100.
- 2- Définir les structures `arbre` et `pepiniere`.
- 3- Ecrire une **fonction** `Creer_Arbre` permettant de créer un nouvel arbre en demandant à l'utilisateur son `nom`, son `type`, sa `hauteur` et son `âge`. Attention, les saisies de l'âge et de la `hauteur` devront être recommencées tant que les valeurs proposées ne sont pas strictement positives.
- 4- En utilisant la fonction précédente, écrire une **procédure** `Ajouter_Arbre` permettant d'ajouter un arbre à la pépinière. L'ajout ne pourra se faire que si le tableau contient encore au moins une case vide !
- 5- Ecrire un sous-programme `Affiche_Type` permettant, à partir d'un `type` passé en paramètre (feuillus ou résineux), d'afficher la liste des arbres (`nom`) de ce `type`.
- 6- Ecrire un sous-programme `Plus_Bas_Plus_Haut` permettant de "renvoyer" au programme principal la `hauteur` de l'arbre le plus petit **et** celle de l'arbre le plus haut. Aucun affichage n'est demandé ici.
- 7- Ecrire le programme principal permettant, en utilisant les sous-programmes écrits précédemment, de remplir le tableau d'arbres de la `pepiniere` avec autant d'arbres que l'utilisateur le voudra, puis d'afficher la liste des arbres du `type` choisi, et enfin d'afficher la `hauteur` de l'arbre le plus haut et de l'arbre le plus bas.