

**LIFAP1 : Session 2**  
**Contrôle Continu Terminal** (Durée : 1h30)  
Jeudi 21 juin 2018

***Recommandations :** Les documents, calculatrice, téléphone portable sont interdits.*

*La qualité de l'écriture et de la présentation sera prise en compte dans la note finale.*

*Vous veillerez à **respecter** les notations et les règles d'écriture des algorithmes vues en cours et en TD. Le barème est donné à titre indicatif.*

NOM :

.....

PRENOM :

.....

Numéro Etudiant :

.....

**Partie A : Algorithmique (10 pts)**

La suite de Lucas est définie de la manière suivante :

$$\mathcal{L}_0 = 2$$

$$\mathcal{L}_1 = 1$$

$$\mathcal{L}_{n+2} = \mathcal{L}_{n+1} + \mathcal{L}_n$$

- 1- Écrire une **fonction** `saisie_valeur` qui effectue la saisie d'un nombre entier **appartenant** à l'intervalle  $[a, b]$  avec  $a$  et  $b$  des paramètres de la fonction. Attention, on recommencera la saisie tant que la valeur choisie n'est pas dans l'intervalle.

- 2- Écrire un sous-programme `remplit_lucas` qui remplit et "retourne" un tableau avec les  $n$  premiers termes de la suite de Lucas,  $n$  étant passé en paramètre.

- 3- Écrire le programme principal qui remplit un tableau de taille maximale 32 avec les  $n$  premiers termes de la suite de Lucas,  $n$  étant compris entre 3 et 32. On réutilisera les fonctions précédentes.

## Des chaines de caractères ....

Ecrire l'algorithme d'une procédure `enleve_chiffres` qui transforme une chaine de caractères `ch_init` contenant **uniquement** des caractères alphanumériques en une chaine `ch_res` ne contenant que les caractères alphabétiques de la chaine initiale.

Exemple : `ch_init = "10trucs33MACHINSet21choses"` → `ch_res = "trucsMACHINSetchoses"`

## Partie B : Langage C / C++ (10 pts)

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.

- 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 pépinière.

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 `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.

- 6- Ecrire le programme principal permettant, en utilisant les sous-programmes écrits précédemment, de remplir le tableau d'arbres de la pépinière avec 5 arbres, puis d'afficher la hauteur de l'arbre le plus haut et de l'arbre le plus bas.