

# LIFAP1 – Séquence 5

## Sujet B

Contrôle Continu – TP10 - durée 1h30 mn

Vendredi 14 décembre 2018

### 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 de Pokemon dans une Pokeball. Un `pokemon` est identifiée par les informations suivantes : un `nom`, une `espece`, une `force` et un `point_de_vie`. Une `pokeball` contient quant à elle un tableau d'au maximum `MAXPOKEMON` `pokemon` et le `nombre_de_pokemon` présents dans la balle.

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 `MAXPOKEMON` (nombre maximum de `pokemon` dans une `pokeball`) ayant pour valeurs respectives 64 et 10.
- 2- Définir les structures `pokemon` et `pokeball`.
- 3- Ecrire une **fonction** `Creer_Pokemon` permettant de créer un nouveau `pokemon` en demandant à l'utilisateur son `nom`, son `espece`, sa `force` et son `nombre point_de_vie`. Attention, les saisies de la `force` et du `nombre de point_de_vie` devront être recommencées tant que les valeurs proposées ne sont pas strictement positives.
- 4- En utilisant la fonction précédente, écrire une **procédure** `Ajouter_Pokemon` permettant d'ajouter un `pokemon` à une `pokeball`. L'ajout ne pourra se faire que si le tableau contient encore au moins une case vide !
- 5- Ecrire un sous-programme `Affiche_Espece` permettant, à partir d'une `espece` passée en paramètre, d'afficher la liste des `pokemon` de cette `espece`.
- 6- Ecrire un sous-programme `Plus_Fort_Plus_Faible` permettant de "renvoyer" au programme principal la `force` du `pokemon` la plus fort **et** celui du `pokemon` le plus faible (le moins fort). Aucun affichage n'est demandé ici.
- 7- Ecrire le programme principal permettant, en utilisant les sous-programmes écrits précédemment, de remplir la `pokeball` avec autant de `pokemon` que l'utilisateur le voudra (on demandera à chaque saisie à l'utilisateur s'il souhaite poursuivre ou non), puis d'afficher les `pokemon` de la `pokeball` ayant une `espece` choisie, et enfin d'afficher la `force` du `pokemon` le plus fort et celui le moins fort.