

Répondre aux questions de l'énoncé dans un fichier `kholle.py` sur votre ordinateur. Le fichier doit pouvoir s'exécuter correctement lors de la correction.

Vous n'êtes pas obligés de répondre à toutes les questions pour avoir la note maximale. N'effacez pas votre code même s'il ne fonctionne pas, tout essai est pris en compte lors de la correction.

Nom et prénom: _____

Numéro étudiant: _____

La khölle a pour but de vous faire développer un programme qui gère vos pokemons et affronte d'autres dresseurs.

1. Commencer par définir une fonction `genere_nom()` qui génère automatiquement un nom de pokemon. Importez la variable `ascii_lowercase` depuis la librairie `string`. Importez également la librairie `random`. La fonction `genere_nom()` va choisir six lettres aleatoires dans la variable `ascii_lowercase` en utilisant la fonction `randint(a, b)` de la librairie `random` et renvoyer leur concaténation. Les bornes `a`, `b` doivent bien sûr dépendre du nombre de lettres de l'alphabet. Le nom de pokemon risque d'être imprononçable, c'est normal.
2. On ajoute une fonction `pokedex()` qui va renvoyer un dictionnaire contenant 6 pokemons. Pour chacun des pokemons, générer son nom en utilisant la fonction ci-dessus, ça sera sa clé. La valeur associée à chacun des pokemons généré est un dictionnaire `{ 'atk': a, 'def': d }` qui correspond aux valeurs d'attaque et de défense du pokemon. Ces valeurs seront tirées au sort entre 1 et 10 à chaque fois à l'aide de la fonction `randint(a, b)` de la librairie `random`. Utiliser cette fonction pour créer deux dictionnaires, `p1` et `p2`.
3. On ajoute une fonction `combat(x, y)` qui prend en entrée deux tuples de la forme `x = (a1, d1)`, `y = (a2, d2)` qui contiennent les valeurs d'attaque et de défense de deux pokemons et qui simule un combat entre ces derniers. Le combat se déroule de la manière suivante: le premier pokemon attaque l'autre en soustrayant son attaque `a1` à la défense de l'autre `d2`. Si la nouvelle valeur de défense `d2 = d2 - a1` est supérieure à 0, le pokemon 2 attaque alors le pokemon 1 de la même manière (i.e. `d1 = d1 - a2`). Lorsqu'un pokemon atteint une valeur de défense inférieure ou égale à 0, il est KO et le combat s'arrête immédiatement. On renvoie alors les nouvelles valeurs `(a1, d1)`, `(a2, d2)`.
4. Finalement, on simule un combat entre deux dresseurs possédant chacun 6 pokemons (contenus dans les dictionnaires `p1` et `p2`) définis précédemment. Tant que les deux dictionnaires contiennent encore des pokemons, simuler un combat entre les deux premiers pokemons de ces dictionnaires à l'aide de la fonction `combat(x, y)` définie précédemment. Lorsque cette fonction retourne les deux pokemons, retirer le pokemon KO de son dictionnaire associé en utilisant la fonction `del`. Annoncez le vainqueur du combat lorsqu'un des dictionnaires est vide. Bravo, vous êtes le meilleur dresseur.