Exercice 1 Modéliser des Pokémons avec des dictionnaires et des tuples

On modélise des informations (nom, taille et poids) sur des Pokémons de la façon suivante :

```
exemple_pokemons = {
        'Bulbizarre' :(70, 7),
        'Herbizarre': (100, 13),
        'Abo': (200, 7),
        'Jungko': (170, 52)}
```

Par exemple, **Bulbizarre** est un Pokémon qui mesure 70 cm et qui pèse 7 kg.

1) Quel est le type de `exemple_pokemons` ?

2) Quelle instruction permet d'ajouter à cette structure de données le Pokémon Goupix qui mesure 60 cm et qui pèse 10 kg ?

3) On donne le code suivant :

```
def le_plus_grand(pokemons):
        grand = None
        taille_max = None
        for (nom, (taille, poids)) in pokemons.items():
              if taille_max is None or taille > taille_max:
                  taillemax = taille
                  grand = nom
        return (grand, taille_max)
```

a. Quelle est la valeur de `le_plus_grand(exemple_pokemons)` ?

b. Écrire le code d'une fonction `le_plus_leger` qui prend des Pokémons en paramètre et qui renvoie un tuple dont la première composante est le nom du Pokémon le plus léger et la deuxième composante est son poids.

```
assert le_plus_leger(exemple_pokemons) == ('Bulbizarre', 7)
```

4) Écrire le code d'une fonction taille qui prend en paramètre un dictionnaire de Pokémons ainsi que le nom d'un Pokémon, et qui renvoie la taille de ce Pokémon

```
assert taille(exemple_pokemons, 'Abo') == 200
assert taille(exemple_pokemons, 'Jungko') == 170
assert taille(exemple_pokemons, 'Dracaufeu') is None
```

Exercice 2 Itérer sur les éléments d'un dictionnaire

Au zoo de Beauval, il y a 5 éléphants d'Asie, 17 écureuils d'Asie, 2 pandas d'Asie...
On représente cet inventaire à l'aide d'un dictionnaire, de la façon suivante :

```
zoo_Beauval={
    'éléphant': ('Asie', 5),
    'écureuil': ('Asie', 17),
    'panda': ('Asie', 2),
    ' hippopotame ' : ('Afrique', 7),
    'girafe': ('Afrique', 4)}
```

On représente de la même façon le Zoo de la Flèche : `zoo_LaFleche={`

```
    'ours': ('Europe', 4),
    'tigre': ('Asie', 7),
    'girafe': ('Afrique', 11),
    ' hippopotame ' : ('Afrique', 3)}
```

1) On souhaite se doter d'une fonction `plus_grand_nombre` qui prend un zoo en paramètre et qui renvoie le nom de l'animal le plus représenté dans ce zoo.

   Par exemple :

```
assert plus_grand_nombre(zoo_LaFleche) == 'girafe'
assert plus_grand_nombre(zoo_Beauval) == 'écureuil'
```

   a) Quel type de boucle peut-on envisager pour le code de cette fonction ?

```
i.    for cle in dico.keys()
ii.   for valeur in dico.values()
iii.  for (cle,valeur) in dico.items()
iiii. Aucune boucle.
```

   b) Ecrire le code de cette fonction.

2) On souhaite se doter d'une fonction `nombre_total` qui prend un zoo en paramètre ainsi que le nom d'un continent, et qui renvoie le nombre d'animaux originaires de ce continent dans le zoo.

   Par exemple :

```
assert nombre_total(zoo_LaFleche, 'Afrique') == 14
assert nombre_total(zoo_Beauval, 'Asie') == 24
```

   a) Quel type de boucle peut-on envisager pour le code de cette fonction ?

```
i.    for cle in dico.keys()
ii.   for valeur in dico.values()
iii.  for (cle,valeur) in dico.items()
iiii. Aucune boucle.
```

   b) Ecrire le code de cette fonction.

3) On souhaite se doter d'une fonction `nombre` qui prend un zoo en paramètre ainsi que le nom d'un animal, et qui renvoie le nombre de représentants de cet animal dans le zoo.

   Par exemple :

```
assert nombre(zoo_LaFleche, 'panda') == 0
assert nombre(zoo_Beauval, 'panda') == 2
```

   a) Quel type de boucle peut-on envisager pour le code de cette fonction ?

```
i.    for cle in dico.keys()
ii.   for valeur in dico.values()
iii.  for (cle,valeur) in dico.items()
iiii. Aucune boucle.
```

   b) Ecrire le code de cette fonction.

Exercice 3 Construire des listes en compréhension

1) Parmi les extraits de programme suivants, lesquels permettent de construire la liste des cinq premiers nombres impairs ?

☐ a.
```
impairs = [1, 3, 5, 7, 9]
```

☐ b.
```
impairs = []
for n in range(5):
    impairs.append(2 * n + 1)
```

☐ c.
```
impairs = [2 * n + 1 for n in range(5)]
```

☐ d.
```
impairs = [n for n in range(1, 11 ,2)]
```

☐ e.
```
impairs = []
n = 0
while len(impairs) != 5:
    if n % 2 == 1:
        impairs.append(n)
    n = n + 1
```

2) Donner plusieurs programmes permettant de construire la liste des 25 premiers nombres pairs.

3) a) Quelle est la valeur de `couples` à la fin de l'exécution du programme suivant ?

```
lettres = ['a', 'b', 'c']
nombres = [1, 5]
couples = [(c, n) for c in lettres for n in nombres]
```

b) Proposer un programme qui permet de construire `couples` en utilisant des boucles bornées.