

Exercice 1. Questions de cours

On dispose de deux variables :

- La variable `champions` associe à chaque champion ses valeurs d'attaque et de défense ;
- La variable `roles` répertorie les rôles possibles des champions.

Par exemple :

```
champions = {'Ashe': (7, 3), 'Alistar': (6, 9), 'Shaco': (8, 4)}
roles = {'Support', 'Mage'}
```

- 1.1. Quel est le type de la variable `champions` ? Préciser le type de ses éléments.
- 1.2. Donner l'instruction nécessaire pour :
 - a) ajouter dans `champions` le champion "Nunu" qui a pour attaque 4 et pour défense 6.
 - b) obtenir la valeur d'attaque du champion 'Shaco' qui se trouve dans `champions`.
 - c) ajouter 'Tank' à `roles`.
- 1.3. Expliquer quelles sont les différences structurelles entre un ensemble et une liste.

Exercice 2. Représentation de la mémoire

2.1. Donner la représentation de la mémoire (pile et tas) à l'endroit indiqué du code suivant.

Préciser les affichages des deux dernières lignes.

```
persos = ['John', 'Aria', 'Sansa']
persos2 = persos
persos2.append('Ramsay') # ICI
print(persos2)
print(persos)
```

2.2. On exécute le code suivant. Représenter l'état de la mémoire au premier passage par l'endroit indiqué.

```
def double(x):
    y = x + x # ICI
    return y
def triple(x):
    d = double(x)
    return x + d

def suivant(a):
    return a + 1
x = 1
y = suivant(x)
z = triple(y)
```

Exercice 3. Soirée cinéma

Ce soir, c'est soirée cinéma à la maison. Le pop-corn est prêt, les rideaux sont tirés. Au fait, quelles personnes seront présentes ? Et quel film va-t-on regarder ? Les ennuis commencent ... Chaque personne a un film qu'elle voudrait voir.



3.1. Proposez une modélisation possible (autre que celle proposée dans la suite de l'exercice)

On décide finalement de représenter les souhaits des participants sous la forme d'un dictionnaire dont les clés sont les noms des participants, et la valeur associée à chaque clé est le film qu'il veut voir. Voici un exemple d'un tel dictionnaire de souhaits :

```
souhaits_lundi = {
    'M. Becker': 'Le cuirasse Potemkine',
    'Mme. Adobet': 'Bienvenue à Zombieland',
    'M. Robert': 'La menace fantôme'}
```

Les critères d'évaluation pour les questions suivantes (dans l'ordre) :

- il n'y a pas d'erreur de syntaxe
- des tests sont présents ET ils sont lus ET ils sont pertinents
- votre fonction passe les tests avec succès
- S'il y a une boucle, vous avez fait le bon choix de boucle
- votre code est clair et lisible (choix du nom des variables, commentaires si nécessaire ...)

Exceptionnellement, pour cette évaluation, on ne demande pas de documenter les fonctions.

3.2. Celui qui organise la soirée cinéma veut savoir quel sera le nombre des participants satisfaits, c'est-à-dire le nombre de participants dont le souhait sera exaucé . Écrire une fonction `nombre_satisfaits` qui prend en paramètre un dictionnaire de souhaits et le nom d'un film, et qui renvoie le nombre de personnes qui souhaitaient voir ce film.

```
mardi = {
    'M. Becker': 'Le cuirasse Potemkine',
    'Mme. Adobet': 'Le retour du Jedi',
    'M. Robert': 'Le cuirasse Potemkine',
    'M. Melin': 'Le cuirasse Potemkine'}

vendredi = {
    'M. Becker': 'Il etait une fois la Revolution',
    'Mme. Adobet': "l'Empire contre-attaque",
    'M. Robert': "l'Empire contre-attaque",
    'M. Melin': "La vie d'Adele"}

assert nombre_satisfaits(mardi, 'Le cuirasse Potemkine') == 3
assert nombre_satisfaits(vendredi, 'Le cuirasse Potemkine') == 0
```

3.3. Dictionnaire de fréquences

Écrire une fonction `popularite_des_films` qui prend en paramètre un dictionnaire de souhaits, et qui renvoie un dictionnaire dont les clefs sont les films et la valeur associée le nombre de personnes intéressées.

```
assert popularite_des_films(mardi) == {
    'Le cuirasse Potemkine':3,
    'Le retour du Jedi':1}
assert popularite_des_films(vendredi) == {
    'Il etait une fois la Revolution':1,
    'l'Empire contre-attaque':2,
    "La vie d'Adele":1}
```

3.4. On se demande quel est le film le plus populaire de la soirée. Écrire une fonction `film_prefere` qui prend en paramètre un dictionnaire de souhaits, et qui renvoie le nom du film le plus populaire.

```
assert film_prefere(mardi) == 'Le cuirasse Potemkine'
assert film_prefere(vendredi) == "l'Empire contre-attaque"
```

3.5. On se demande s'il est possible de passer une bonne soirée, c'est-à-dire s'il est possible de choisir un film qui satisfasse au moins la moitié des participants. Écrire une fonction `bonne_soiree` qui prend en paramètre un dictionnaire de souhaits, et qui renvoie `True` s'il est possible de passer une bonne soirée, et `False` sinon.

```
souhaits_lundi = {
    'M. Becker': 'Le cuirasse Potemkine',
    'Mme. Adobet': 'Bienvenue à Zombieland',
    'M. Robert': 'La menace fantôme'}
mardi = {
    'M. Becker': 'Le cuirasse Potemkine',
    'Mme. Adobet': 'Le retour du Jedi',
    'M. Robert': 'Le cuirasse Potemkine',
    'M. Melin': 'Le cuirasse Potemkine'}
assert bonne_soiree(mardi)
assert not bonne_soiree(souhaits_lundi)
```

3.6. Il arrive que certains participants ne puissent pas venir à la soirée cinéma. Dans ce cas, On se demande s'il est possible de choisir un film qui satisfasse tout le monde. Écrire une fonction `excellente_soiree` qui prend en paramètre un dictionnaire de souhaits et un ensemble de participants, et qui renvoie `True` si tous les participants sont d'accords sur le film à voir, et `False` sinon.

```
assert excellente_soiree(mardi, {'M. Becker', 'M. Robert'})
assert not excellente_soiree(mardi, {'M. Becker', 'M. Melin', 'Mme. Adobet'})
```

Exercice 4. Exercice - Soirée cinéma, la suite!¹

Comme la soirée cinéma a bien plu, on propose maintenant d'organiser une soirée *marathon* durant laquelle plusieurs films seront visionnés. Il reste à savoir qui organise la soirée, qui est invité et qui apporte quel film.

Pour résoudre ce problème, on dispose d'un dictionnaire des propriétaires de films dont les clefs sont les films éventuellement projetés, et la valeur associée le nom du propriétaire du film.

```
proprietaires = {
    'Il etait une fois la Revolution': 'M. Melin',
    'Le retour du Jedi': 'Mme. Adobet',
    'l'Empire contre-attaque': 'M. Becker',
    'La vie d'Adele': 'M. Robert',
    'Buffy contre les vampires': 'Mme. Adobet'}
```

Pour la questions suivantes, on utilisera la liste de souhaits suivante :

```
vendredi = {
    'M. Becker': 'Il etait une fois la Revolution',
    'Mme. Adobet': "l'Empire contre-attaque",
    'M. Robert': "l'Empire contre-attaque",
    'M. Melin': "La vie d'Adele"}
```

M. Becker décide d'organiser la soirée. Il voudrait voir *Il etait une fois la Revolution*. Il doit donc inviter M. Melin qui possède le film. Comme M. Melin veut voir *La vie d'Adèle*, il faut également inviter M. Robert qui est l'heureux propriétaire de ce film. Comme M. Robert veut regarder *l'Empire contre-attaque* que possède M. Becker, la liste des invités est complète.

Pour cette soirée, seront donc invités : M. Becker, M. Melin et M. Robert Les films vus ce soir là seront : *Il etait une fois la Revolution*, *La vie d'Adèle* et *l'Empire contre-attaque*.

4.1. Et si M. Melin décide d'organiser la soirée, quels seront les invités ? Quels seront les films visionnés ?

4.2. Ecrire une fonction `invitations` qui prend en paramètre un dictionnaire de souhaits, un dictionnaire des propriétaires et le nom de l'organisateur de la soirée. Cette fonction devra renvoyer l'ensemble des invités de la soirée.

```
assert invitations(vendredi, proprietaires, 'M. Becker') == {'M. Becker',
    'M. Melin', 'M. Robert'}
```

1. Pour ceux qui visent une note > 16/20