

Nom : ..... Prénom : .....

**Exercice 1.** On suppose écrite une classe `Chien` dont chaque instance de classe `c` possède les attributs : `nom` (le nom du chien `c`) et `poids` (le poids du chien `c`). On suppose également écrite la méthode `aboie` qui simule l'aboiement d'un chien. Appliquée à un chien dont l'attribut `nom` vaut 'Milou' affiche :

⚙️ ➤ Résultat

Milou fait Ouaf !

1. a. Donner une instruction permettant d'instancier un objet `c1` de type `Chien`, représentant un chien nommé Médor qui pèse 12 kilos.

.....

b. Représenter par un schéma l'état de la mémoire après l'instanciation de la variable `c1`.

c. Écrire une méthode `mange` de la classe `Chien` simule le repas d'un chine. Elle modifie l'attribut `poids` du chien `self` en lui ajoutant la valeur de l'argument `ration` (le poids de la ration, en kilogrammes). On ajoute les contraintes suivantes concernant la méthode `mange`.

- On vérifiera que la valeur de `ration` est comprise entre 0 (exclu) et un dixième du poids du chien (inclus). Si le poids du chien ne satisfait pas cette condition, alors on ne modifie pas l'attribut `poids` et on considère que le chien n'a pas mangé.
- Si le chien a mangé, alors il doit aboyer.
- La méthode renverra `True` si le chien a été nourri, `False` sinon.

Code python

```
1 def mange(self, ration):
2     """ Chien, float -> bool """
3     .....
4     .....
5     .....
6     .....
7     .....
8     .....
```

d. Donner l'instruction python permettant d'appliquer la méthode `mange` au chien `c1`, avec une ration de 0.8 kilogrammes.

.....

2. On s'intéresse dans cette question à la classe `Chenil`, dont chaque instance de classe possède un unique attribut `contenu` qui est la liste des variables de type `Chien` appartenant au chenil en question.

Code python

```
1 class Chenil:
2     def __init__(self, c):
3         """ Chenil, [Chien] -> None """
4         self.contenu = c
5
6
```

On suppose instanciées les variable  $c_1, \dots, c_4$  qui représentent les chiens suivants :

Nom de la variable	$c_1$	$c_2$	$c_3$	$c_4$
Nom	Médor	Milou	Rex	Beethoven
Poids	12	10	22	18

**a.** Donner l'instruction python permettant d'instancier un objet maison de type `Chenil`, qui représente un chenil qui abrite les quatre chiens Médor, Milou, Rex, et Beethoven.

.....

**b.** Écrire une méthode `nourrir` de la classe `Chenil` qui nourrit chaque chien du chenil `self` avec la ration `ration`. La méthode renverra la liste des noms des chiens qui ont été effectivement nourris.

Code python

```

1  def nourrir(self, ration):
2      """ Chenil, float -> [str] """
3      .....
4      .....
5      .....
6      .....
7      .....
8      .....
9      .....
10     .....
```

Code python

```

1  chiens_nourris = maison.nourrir(1.5)
2  print(chiens_nourris)
```

 Résultat

```

Rex fait Ouaf !
Beethoven fait Ouaf !
['Rex', 'Beethoven']
```

**Exercice 2.** On considère la suite  $(u_n)_{n \in \mathbb{N}}$  définie par la relation :

$$u_n = \begin{cases} 3 & \text{si } n = 0 \\ 3u_{n-1} + 5 & \text{sinon.} \end{cases}$$

**1. a.** Écrire une fonction récursive  $u$  qui étant donné un entier  $n$  renvoie le nombre  $u_n$ .

Code python

```

1  def u(n):
2      """ int -> int """
3      .....
4      .....
5      .....
6      .....
7      .....
8      .....
```

**b.** Justifier que la fonction  $u$  que vous avez écrite est récursive.

.....  
.....  
.....

**2.** Dresser l'arbre d'appel de l'instruction `u(3)`.