

Frédéric  
LURET



# Python

## Banque de questions

Version du 15 décembre 2024



## 0 Table des matières



### 1 Divers

2

## 1 Divers



### Question 1



Nous avons mis en place un système d'amende pour les chasseurs de notre commune. Chaque chasseur se voit pénaliser d'un certain nombre de point par faute. Le barème des pénalités est le suivant :

- s'il tue une poule : 1 point.
- s'il tue un chien : 3 points.
- s'il tue une vache : 5 points.
- s'il tue un ami : 10 points.

Un point à une valeur de 2€.

Écrire une fonction amende qui reçoit le nombre de victimes du chasseur et qui renvoie la somme due.

Utilisez cette fonction dans un programme principal qui demande le nombre de victimes et qui affiche la somme que le chasseur doit déboursier.

**Code python :**



```
1 points = {"poules": 1, "chiens": 3, "vaches": 5, "amis": 10}
2
3
4 def points_perdus(**kwargs):
5     return sum(kwargs.get(victime, 0) * point for victime, point in
6         ↪ points.items())
7
8 # Définition de fonction
9 ↪ ~~~~~
10 def permisSup(**kargs):
11     return points_perdus(**kargs) * 2
12
13 victimes = {}
14 # Programme principal
15 ↪ =====
16 victimes["poules"] = int(input("Combien de poules ?"))
17 victimes["chiens"] = int(input("Combien de chiens ?"))
18 victimes["vaches"] = int(input("Combien de vaches ?"))
19 victimes["amis"] = int(input("Combien d'amis ?"))
20
21 payer = permisSup(**victimes)
22
23 print(f'{"Rien à payer" if payer == 0 else f"Vous avez à payer {payer}"
24     ↪ " euros"}')
```

q133.py

## Question 2

Soit des comptes bancaires d'individus définis par la liste :



```
1 comptes = {  
2     "compte1": {"nom": "Boismoneau", "prenom": "stephane", "epargne":  
3         ↪ 2500},  
4     "compte2": {"nom": "Jambon", "prenom": "fred", "epargne": 5000},  
5     "compte3": {"nom": "Durois", "prenom": "nicolas", "epargne":  
6         ↪ 10000},  
7     "compte4": {"nom": "Gueux", "prenom": "phillipe", "epargne": 1250},  
8     "compte5": {"nom": "Duchan", "prenom": "alice", "epargne": 4530},  
9     "compte6": {"nom": "Lepenou", "prenom": "amed", "epargne": 2200},  
10    "compte7": {"nom": "Gueux", "prenom": "bernard"},  
11    "compte8": {"nom": "Jambon", "prenom": "steven", "epargne": 1670},  
12    "compte9": {"nom": "Gueux", "prenom": "sylvie", "epargne": 3},  
13    "compte10": {"nom": "Durois", "prenom": "berbard", "epargne":  
14        ↪ 300000},  
15 }
```

q134-comptes.py

On considère que les individus qui portent le même 'nom' sont de la même famille. En cas d'absence de revenu attribué à un individu, nous considérerons que son épargne est nulle (cas de 'Bernard Gueux').

Écrire une fonction qui retourne le nom de la famille la plus pauvre et de la plus riche avec le montant de leur épargne respective. Ici, ('Gueux', 1253) et ('Durois', 310000).

**Code python :**



```
1 def synthese_familles(**kwargs):
2     # On commence par créer un dictionnaire pour ranger les résultats
3     synthese = {}
4     # On traite chaque compte
5     for c in kwargs.values():
6         # Si le nom de famille n'a pas encore été trouvé
7         if c["nom"] not in synthese:
8             # On le crée dans le dictionnaire des résultats
9             synthese[c["nom"]] = 0
10        # Si l'épargne est présente dans le compte...
11        if "epargne" in c:
12            # On peut alors la rajouter au résultat
13            synthese[c["nom"]] += c["epargne"]
14        # Trouver le plus pauvre et le plus riche
15        pauvre = min(synthese, key=synthese.get)
16        riche = max(synthese, key=synthese.get)
17
18        # On retourne les informations récupérées
19        return (pauvre, synthese[pauvre]), (riche, synthese[riche])
20
21 # Test de la fonction
22 comptes = {
23     "cpt1": {"nom": "Boismoneau", "prenom": "stephane", "epargne":
24         ↪ 2500},
25     "cpt2": {"nom": "Jambon", "prenom": "fred", "epargne": 5000},
26     "cpt3": {"nom": "Durois", "prenom": "nicolas", "epargne": 10000},
27
28     "cpt4": {"nom": "Gueux", "prenom": "phillipe", "epargne": 1250},
29     "cpt5": {"nom": "Duchan", "prenom": "alice", "epargne": 4530},
30     "cpt6": {"nom": "Lepenou", "prenom": "amed", "epargne": 2200},
31     "cpt7": {"nom": "Gueux", "prenom": "bernard"},
32     "cpt8": {"nom": "Jambon", "prenom": "steven", "epargne": 1670},
33     "cpt9": {"nom": "Gueux", "prenom": "sylvie", "epargne": 3},
34     "cpt10": {"nom": "Durois", "prenom": "berbard", "epargne": 300000},
35 }
36 print(synthese_familles(**comptes))
```

q134.py