

**Exercice 1** Les animaux de la ferme de Gaston sont représentés par un dictionnaire dont les clés sont le nom des animaux et les valeurs le nombre d'animaux. On donne :

```
ferme_gaston = {'lapin': 5, 'vache': 7, 'cochon': 2, 'cheval': 4}
```

1. Quelle instruction permet d'obtenir le nombre de chevaux de Gaston ?
2. Gaston achète 12 poules. Quelle instruction permet de les ajouter à `ferme_gaston` ?
3. Quelle instruction permet d'obtenir la liste des animaux de la ferme de Gaston ?
4. Ecrire une fonction `nombre_animaux(ferme)` qui a pour argument `ferme`, un dictionnaire représentant le nombre d'animaux de la ferme et qui renvoie le nombre total d'animaux.  
Par exemple `nombre_animaux(ferme_gaston)` renvoie 18 (ou 30 après l'ajout des 12 poules).
5. Ecrire une fonction `nombre_animal(ferme, animal)` qui a pour arguments `ferme`, un dictionnaire représentant le nombre d'animaux de la ferme et `animal`, une variable de type `str`. Cette fonction renvoie le nombre d'animal dans la ferme si cet animal est dans la ferme sinon 0.  
Par exemple `nombre_animal(ferme_gaston, 'cochon')` renvoie 2 et `nombre_animal(ferme_gaston, 'oie')` renvoie 0.

**Exercice 2** On donne le tableau suivant indiquant le nom des produits et le prix correspondant :

Nom du produit	Prix TTC (en €)
Sabre laser	229
Etoile de ninja	29,95
Cape	75
Baguette	35
Chapeau	12
Bandeau	12
Balai	130

1. Construire un dictionnaire `BDPrix` correspondant au tableau ci-dessus.
2. Définir une fonction `Dispo(p, D)` renvoyant `True` si le produit `p` est présent dans le dictionnaire `D` de prix et `False` sinon.
3. Définir une fonction `PrixMoyen(D)` qui calcule le prix moyen des produits proposés dans le dictionnaire `D` (qui contient au moins un produit).
4. Définir une fonction `IntervallePrix(m, M, D)` qui renvoie l'ensemble des produits du dictionnaire `D` dont le prix est compris entre `m` et `M`.
5. Le panier est un concept omniprésent dans les sites marchands. En Python, le panier du client sera représenté par un dictionnaire `Panier` de type `dict[str : int]` avec :
  - ★ le nom des produits comme clé,
  - ★ la quantité d'achat comme valeurs associées.
 Donner une expression Python correspondant à l'achat de deux sabres lasers, de trois étoiles de ninja et d'un bandeau.
6. Définir une fonction `TousDispo(D1, D2)` qui à partir du panier d'achat `D2` et d'un dictionnaire des prix `D1`, renvoie `True` si tous les produits sont disponibles ou `False` sinon.
7. Définir une fonction `Total(D1, D2)` renvoyant le total à payer du panier (dictionnaire `D2`) à partir des prix du dictionnaire `D1` (on supposera que tous les articles sont disponibles).

**Exercice 3** Voici un exemple de données permettant de manipuler un livre de recettes de cuisine à partir de la liste des ingrédients des recettes :

Recette	Ingrédients
Gâteau au chocolat	chocolat, oeuf, farine, sucre, beurre
Gâteau au yaourt	yaourt, oeuf, farine, sucre
Crêpes	oeuf, farine, lait, bière
Quatre-quarts	oeuf, farine, beurre, sucre
Kouign-amann	farine, beurre, sucre

On va modéliser en Python un livre de recettes, par un dictionnaire de type `dict(str : list(str))` dans lequel :

- ★ les noms des recettes, de type `str`, comme clés,
  - ★ l'ensemble des ingrédients, de type `list(str)`, comme valeurs associées.
1. Construire un dictionnaire `MesRecettes` correspondant au tableau ci-dessus.
  2. Définir une fonction `NbIngredients(D, nom)` qui à partir d'un dictionnaire de recettes `D`, défini comme décrit ci-dessus, renvoie le nombre d'ingrédients de la recette nommé `nom`.
  3. Définir une fonction `RechercheRecettes(D, I)` renvoyant l'ensemble des recettes du dictionnaire `D` qui utilisent l'ingrédient `I`.
  4. Définir une fonction `RechercheRecettes2(D, I1, I2)` renvoyant l'ensemble des recettes du dictionnaire `D` qui utilisent les ingrédients `I1` et `I2`.
  5. Définir une fonction `RechercheRecettesMulti(D, LI)` renvoyant l'ensemble des recettes du dictionnaire `D` qui utilisent tous les ingrédients de la liste `LI`.
  6. Définir une fonction `RechercheRecettesSansMulti(D, LI)` renvoyant l'ensemble des recettes du dictionnaire `D` qui n'utilisent aucun des ingrédients de la liste `LI`.

**Exercice 4** Un pokémon est modélisé par :

- ★ son nom (chaîne de caractères),
- ★ sa qualité de défense (nombre entier),
- ★ ses points de vie ou HP (nombre entier),
- ★ sa vitesse (nombre entier),
- ★ sa puissance d'attaque (nombre entier),
- ★ son ou ses types (liste de chaîne de caractères).

On représente un pokémon par un dictionnaire. Par exemple,

```
bulbizarre = {'Nom': 'Bulbizarre', 'HP': 45, 'Attaque': 49,
              'Défense': 49, 'Vitesse': 45, 'Type': ['Plante', 'Poison']}
```

correspond au pokémon nommé Bulbizarre qui a 45 points de vie, une attaque et une défense de 49, une vitesse de 45 et qui a deux types : Plante et Poison.

1. On donne le pokémon suivant : `mystere = {'Nom': 'Salamèche', 'HP': 39, 'Défense': 43, 'Vitesse': 455, 'Type': ['Feu']}`
  - (a) Quel est son nom ? Préciser l'instruction permettant de le récupérer.
  - (b) Combien a-t-il de types ? Quelle(s) instruction(s) permet de le(s) récupérer ?
  - (c) La vitesse donnée est inexacte, elle est en fait de 65. Quelle instruction permet de corriger cette erreur ?
  - (d) Il manque la valeur de la puissance d'attaque de ce pokémon. Elle est de 52. Quelle instruction faut-il saisir pour ajouter cette information ?
2. Représenter le pokémon `fragilady` qui a une défense de 75, une attaque de 60, une vitesse de 90, 70 points de vie, de type Plante.
3. Dans les questions qui suivent, on supposera que le pokémon passé en paramètre existe.
  - (a) Ecrire une fonction qui prend un pokémon en paramètre et renvoie sa valeur d'attaque.
  - (b) Ecrire une fonction qui prend un pokémon en paramètres et renvoie son nombre de types.
  - (c) Ecrire une fonction qui prend un pokémon et une valeur entière en paramètres et renvoie `True` si le pokémon a une valeur de défense supérieure à la valeur passée en paramètre, `False` sinon.