Modéliser des notes

Partie 1 : Modélisation simpliste

On modélise les notes d'une élève de la façon suivante :

notes_de_lea = [12, 14, 3, 16, 17, 2, 13, 19]

1. Quel est le type de notes_de_lea ?
   - un int
   - une liste
   - un tuple
   - un dictionnaire
   - autre chose
2. Que vaut l'expression notes_de_lea[2] ?
   - 3
   - 14
   - 6
   - 5
   - autre chose
3. Quelle instruction permet d'ajouter une note de 15 à cette structure de données ?
   - notes_de_lea.append(15)
   - notes_de_lea[8] = 15
   - notes_de_lea.append([15])
   - notes_de_lea = notes_de_lea + 15

4. On propose le code suivant :

```
1  def fonction(liste_de_notes):
2      """
3      'liste_de_notes' est une liste de nombres qui
4  modélise
5      les notes d'un élève.
6      Cette fonction renvoie ???
7      """
8      compteur1 = 0
9      compteur2 = 0
10     for note in liste_de_notes:
11         if note >= 10:
12             compteur1 = compteur1 + 1
13         else:
14             compteur2 = compteur2 + 1
15     return (compteur1, compteur2)
16
17 notes_de_lea = [12, 14, 3, 16, 17, 2, 13, 19]
   assert fonction(notes_de_lea) == ???
```

- Quel est le type de retour de cette fonction ?
- Recopier et compléter la ligne 17 de ce code.
- Recopier et compléter la ligne 5 de ce code. On demande ici d'expliquer en quelques mots ce que fait cette fonction.

Partie 2 : Modélisation avec une structure de données imbriquées

La modélisation précédente n'est pas satisfaisante si l'on veut conserver les notes de plusieurs élèves dans une même structure de données.

On propose, dans cette partie, de modéliser les notes des élèves de la façon suivante :

```
notes_de_la_classe = [('Enzo', 3), ('Emma', 16), ('Lucas', 14),
 ('Manon', 13)]
```

1. Quel est le type de notes_de_la_classe ?
   - un int
   - une liste
   - un tuple
   - un dictionnaire
   - autre chose
2. Que vaut l'expression notes_de_la_classe[2] ?
   - 14
   - 'Lucas'
   - ('Lucas', 14)
   - 'Emma'
   - 16
   - autre chose
3. Quelle instruction permet d'ajouter à cette structure de données une note de 15 obtenue par Gabin ?
   ……

4. On veut écrire une fonction nom_du_genie qui prend une telle structure de données en paramètre et qui renvoie le nom de l'élève qui a eu la meilleure note.
   - Proposer un test pour cette fonction.
   - On donne le code mélangé de cette fonction. À vous de le remettre dans l'ordre !

```
1      note_max = note
2    note_max = None
3  def nom_du_genie(les_notes):
4    return genie
5      genie = nom
6    genie = None
7      if note_max == None or note > note_max:
8    for (nom, note) in les_notes:
```

Que vaut l'expression `nom_du_genie([])` ?
- `None`
- `""`
- `0`
- `()`
- rien : cette expression génère une erreur

Partie 3 : Une modélisation plus complète

Dans cette partie, on souhaite modéliser dans une même structure de données les notes des élèves d'une classe en précisant le nom de la matière concernée par la note. On propose la modélisation suivante :

```
notes = {'Enzo' : ('Math', 3), 'Emma' : ('Math', 16), 'Lucas' : ('NSI', 14), 'Manon' : ('Math', 3)}
```

1. Quel est le type de `notes` ?
    - un `int`
    - une liste
    - un tuple
    - un dictionnaire
    - autre chose
2. Que vaut l'expression `notes[2]` ?
    - `14`
    - `'Lucas'`
    - `('NSI', 14)`
    - `3`
    - cette expression génère une erreur
3. Quelle instruction permet d'ajouter la note de 15 obtenue par Gabin en NSI ?
4. Quel est l'affichage généré par l'exécution du code suivant ?

```
for (nom, (matiere, note)) in notes.items():
    if note < 15:
        print(nom)
```

5. On veut écrire une fonction qui prend une telle structure de données en paramètre et qui renvoie le nom de l'élève qui a eu la moins bonne note, toutes matières confondues.
    - Proposer un test pour cette fonction.
    - Écrire le code de cette fonction.
6. On veut écrire une fonction `tri_par-matiere` qui prend une telle structure de données en paramètre et qui renvoie un dictionnaire dont les clés sont les noms des matières, et les valeurs la liste des notes obtenues par les élèves dans chaque matière.

Exemple :

```
>>> notes = {'Enzo' : ('Math', 3), 'Emma' : ('Math', 16), 'Lucas' : ('NSI', 14), 'Manon' : ('Math', 3)}
>>> tri_par_matiere(notes)
{'Math': [3, 3, 16], 'NSI': [14]}
```

Écrire le code de cette fonction.