L'empreinte carbone d'Ada

Sujet n°5 — Corrigé

Question 1 *Total d'une empreinte agrégée*

```
def total_simple(empreinte):
    total = 0
    for categorie in empreinte:
        total = total + empreinte[categorie]
    return total

print(total_simple(chargement_json('empreinte_ada_agr.json')))
```

Empreinte totale d'Ada : 7252 kg de CO_2.

Question 2 *Total récursif*

Une valeur est soit un nombre, qu'on ajoute, soit un dictionnaire, dont on ajoute le total calculé récursivement.

```
def total_rec(empreinte):
    total = 0
    for valeur in empreinte.values():
        if est_dictionnaire(valeur):
            total = total + total_rec(valeur)
        else:
            total = total + valeur
    return total
```

Test plus proche du fichier : `total_rec({'a': {'b': {'c': 4}, 'd': 1}, 'e': 2}) == 7`. Le fichier complet d'Ada donne aussi 7252.

Question 3 *Une alerte qui s'arrête trop tôt*

Cause : `return alerte_valeur_aberrante(valeur, limite)` renvoie le résultat de la *première* sous-catégorie rencontrée, même `False`, et la fonction s'arrête. Avec le fichier d'Ada, la première catégorie est « Alimentation », qui ne dépasse pas 1000 : la fonction renvoie `False` sans jamais examiner « Logement », qui contient « Chauffage individuel » (1500). **Correction** : ne renvoyer `True` que si la sous-catégorie contient une valeur aberrante, et sinon continuer.

```
def alerte_valeur_aberrante(empreinte, limite):
    for categorie, valeur in empreinte.items():
        if est_dictionnaire(valeur):
            if alerte_valeur_aberrante(valeur, limite):
                return True
        else:
            if valeur > limite:
                return True
    return False
```

Question 4 *Un jeu de tests contre les régressions*

Entrée	Attendu	Ce que le test vérifie
`{}`, limite 10	`False`	dictionnaire vide
`{'a': 15}`, limite 10	`True`	valeur au premier niveau
`{'a': 10}`, limite 10	`False`	inégalité stricte
`{'a': {'b': 1}, 'c': {'d': 50}}`, limite 10	`True`	la 1re sous-catégorie n'arrête pas la recherche
`{'a': {'b': {'c': 99}}}`, limite 10	`True`	imbrication profonde
`{'a': {'b': 1, 'c': 2}, 'd': 3}`, limite 10	`False`	aucune valeur aberrante

Le quatrième test aurait détecté l'erreur de la question 3.