TD3. Conditionnelles

Bases de la Programmation Impérative – Chapitre 1 : bases
version du 15 décembre 2024

L'objectif premier est de continuer à travailler avec la structure de contrôle conditionnelle `if` mais sur des programmes plus complexes que dans le TD précédent. Le second objectif consiste à s'interroger sur la qualité du code que nous allons écrire.

Exercice 1 : heure à la seconde suivante

Question 1

Écrire une fonction `tictac_1(heures, minutes, secondes)` renvoyant l'heure obtenue en incrémentant d'une seconde le temps donné en paramètre.

Question 2

Écrire une fonction `tictac_2(heures, minutes, secondes)` renvoyant l'heure obtenue en incrémentant d'une seconde le temps donné en paramètre sans utiliser de structure de contrôle conditionnelle.

Exercice 2 : carrés

Un carré peut être caractérisé de nombreuses façons, par exemple :

- avec uniquement des tests d'orthogonalité (il en faut 4) ;
- avec uniquement des tests d'égalité de longueur (4 également) ;
- avec des tests d'égalité de longueur (il en faut 4) et un test d'orthogonalité.

Question 1

Écrire une fonction `est_carre(quadrilatere)` prenant en argument un tuple contenant 4 tuples de taille 2 (chacun représentant un point) formant un quadrilatère (les points sont donc ordonnés) et renvoyant s'ils forment un carré ou non. Penser à découper son code en sous-fonctions quand cela est pertinent.

Exercice 3 : échecs

Dans cet exercice, nous nous intéressons aux mouvements de certaines pièces sur un échiquier entre une case source et une case destination. Une case est représentée par deux entiers (ses coordonnées x et y). Nous considérons uniquement des cases valides, c'est-à-dire faisant partie de l'échiquier dont la taille est `8 x 8`. Nous considérons également que toutes les cases entre le départ et la destination sont inoccupées.

Question 1

Donner l'implémentation d'une fonction vérifiant si le déplacement d'une tour est valide.

Question 2

Donner l'implémentation d'une fonction vérifiant si le déplacement d'un fou est valide.

Question 3

Donner l'implémentation d'une fonction vérifiant si le déplacement d'un cavalier est valide.

Question 4

Sans se fatiguer, c'est-à-dire en réutilisant les fonctions déjà écrites, donner l'implémentation d'une fonction vérifiant si le déplacement d'une dame est valide.

Exercice 4 : expressions conditionnelles (pour aller plus loin)

Question 1

Que penser du programme ci-dessous ?

```python
#!/usr/bin/env python3

"""Construire une chaîne de caractère en fonction d'un booléen."""


def construit_chaine_1(est_une_femme):
    """Version avec structure de contrôle conditionnelle."""
    if est_une_femme:
        return "Bonjour madame,"
    return "Bonjour monsieur,"


def construit_chaine_2(est_une_femme):
    """Version avec expression conditionnelle."""
    return "Bonjour " + ("madame," if est_une_femme else "monsieur,")


def teste():
    """Teste les deux fonctions ci-dessus."""
    print(construit_chaine_1(True) + " (première fonction)")
    print(construit_chaine_1(False) + " (première fonction)")
    print(construit_chaine_2(True) + " (deuxième fonction)")
    print(construit_chaine_2(False) + " (deuxième fonction)")


teste()
```

Question 2

Dans la deuxième fonction, quelle est la différence fondamentale avec la structure de contrôle conditionnelle `if` que nous avons vue jusqu'à présent ?

Question 3

Est-il possible de faire la même chose que dans la deuxième fonction dans d'autres langages impératifs ?