

Sujet 8

EXERCICE 1 (10 points)

Écrire une fonction recherche qui prend en paramètres un tableau tab de nombres entiers triés par ordre croissant et un nombre entier n, et qui effectue une recherche dichotomique du nombre entier n dans le tableau non vide tab.

Cette fonction doit renvoyer un indice correspondant au nombre cherché s'il est dans le tableau, None sinon.

Exemples :

```
>>> recherche([2, 3, 4, 56], 5)
3
>>> recherche([2, 3, 4, 6,7], 5) # renvoie None
```

EXERCICE 2 (10 points)

On dispose de chaînes de caractères contenant uniquement des parenthèses ouvrantes et fermantes.

Un parenthésage est correct si :

6, le nombre de parenthèses ouvrantes de la chaîne est égal au nombre de parenthèses fermantes;

7, en parcourant la chaîne de gauche à droite, le nombre de parenthèses déjà ouvertes doit être, à tout moment, supérieur ou égal au nombre de parenthèses déjà fermées. Ainsi, ((()) (())) est un

parenthésage correct.

Les parenthésages ()) (() et (()) (() sont, eux, incorrects.

On dispose du code de la classe Pile suivant :

```
class Pile:
    """Classe définissant une structure de pile."""
    def __init__(self):
        self.contenu = []

    def est_vide(self):
        """Renvoie un booléen indiquant si la pile est vide."""
        return self.contenu == []

    def empiler(self, v):
        """Place l'élément v au sommet de la pile"""
        self.contenu.append(v)

    def depiler(self):
        """Retire et renvoie l'élément placé au sommet de la pile, si la pile n'est pas vide. Produit une erreur sinon.
        nflo
        """
        assert not self.est_vide()
        return self.contenu.pop()
```

On souhaite programmer une fonction bon_parenthesage qui prend en paramètre une chaîne de caractères ch formée de parenthèses et renvoie True si la chaîne est bien parenthésée et False sinon.

Cette fonction utilise une pile et suit le principe suivant : en parcourant la chaîne de gauche à droite, si on trouve une parenthèse ouvrante, on l'empile au sommet de la pile et si on trouve une parenthèse fermante, on dépile (si possible) la parenthèse ouvrante stockée au sommet de la pile.

La chaîne est alors bien parenthésée si, à la fin du parcours, la pile est vide. Elle est, par contre, mal parenthésée :

8, si dans le parcours, on trouve une parenthèse fermante, alors que la pile est vide;

9, ou si, à la fin du parcours, la pile n'est pas vide.

Compléter le code de la fonction bon_parenthesage ci-dessous:

```

def bon_parenthesage(ch):
    """Renvoie un booléen indiquant si la chaîne ch est bien parenthésée"""
    p = Pile()
    for c in ch:
        if c == '(':
            p.empiler(c)
        elif c == ')':
            if p.est_vide():
                return False
            else:
                p.depiler()
    return True

```

Exemples:

```

>>> bon_parenthesage("((()())())")
True
>>> bon_parenthesage("()()()")
False
>>> bon_parenthesage("(())()")
False

```