

## 2022 sujet 24

**Exercice 1**

```
def maxliste(tab):
    maxi = tab[0]
    for i in range(1, len(tab)):
        if tab[i] > maxi:
            maxi = tab[i]
    return maxi
```

**Exercice 2**

```
# Attention !!!
# Ça fonctionne car les assert sont placés dans le bon ordre.
# Il ne faut jamais mettre un objet mutable comme valeur par défaut
# d'un paramètre de fonction car il est évalué une seule fois
# lorsque la fonction est définie.
# Ainsi, dans ce programme le tableau valeur par défaut se conserve
# entre chaque instance !
# À cause de cela un assert parenthesesage("(((())(())")) == True
# placé à la fin ne fonctionnera pas car le tableau valeurs n'est
# pas vide au début.
```

```
class Pile:
    """ Classe définissant une pile """
    def __init__(self, valeurs=[]): # Grosse erreur !!! Il vaut mieux
        # ne pas mettre de valeur par défaut dans ce cas
        self.valeurs = valeurs

    def est_vide(self):
        """Renvoie True si la pile est vide, False sinon"""
        return self.valeurs == []

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

    def depiler(self):
        """
        Supprime l'élément placé au sommet de la pile, à condition
        qu'elle soit non vide
```

```

        """
        if self.est_vide() == False:
            self.valeurs.pop()

def parenthesage (ch):
    """Renvoie True si la chaîne ch est bien parenthésée et False sinon"""
    p = Pile() # Pour éviter tout problème : p = Pile([])
    for c in ch:
        if c == "(":
            p.empiler(c)
        elif c == ")":
            if p.est_vide():
                return False
            else:
                p.depiler()
    return p.est_vide()

assert parenthesage("((()())(()))") == True
assert parenthesage("()()()") == False
assert parenthesage("(())(()") == False

```