

22 NSI 24

```
In [ ]: class Pile:
    """ Classe définissant une pile """
    def __init__(self, valeurs=[]):
        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()
```

```
In [ ]: def parenthesage (ch):
    """Renvoie True si la chaîne ch est bien parenthésée et False sinon"""
    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()
```

```
In [ ]: assert parenthesage("((()())())") == True
assert parenthesage("())(())" == False
assert parenthesage("(())(())" == False
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js