

"""

Author: Pascal Padilla

Source: correction de l'exercice 2 du sujet 24 des épreuves pratiques NSI 2022

Remarques:

* ne JAMAIS mettre un tableau VIDE comme paramètre par défaut d'une fonction.

Ici le bug est dans la définition de `__init__` de la Pile:

```
def __init__(self, valeurs = []):...
```

Ça fait des erreurs assez complexes à trouver...

Exemple d'erreur de fou :

```
>>> parenthesage("((()())())")
```

```
True
```

```
>>> parenthesage("((()())())")
```

```
False
```

```
);
```

```
... et oui, c'est dingue !
```

"""

```
class Pile:
```

```
    """ Classe définissant une pile """
```

```
    # modification d'énoncé sinon c'est FAUX...
```

```
    # def __init__(self, valeurs=[]):
```

```
    #     self.valeurs = valeurs
```

```
def __init__(self):    # suppression de la valeur par défaut
    self.valeurs = []  # initialisation avec une pile vide
```

```
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"""
    # création d'une pile vide
    p = Pile()
```

```
    # parcours de la chaîne, caractère par caractère
    for c in ch:
        # empiler une parenthèse ouvrante
        if c == '(':
            p.empiler(c)
```

```
    # cas des parenthèses fermantes
    elif c == ')':
        # mauvais parenthésage avec une fermante alors
        # que la pile est vide
        if p.est_vide():
            return False
```

```
    # dépiler la parenthèse ouvrante au sommet de la pile
    else:
        p.depiler()
```

```
    # parenthésage correct <=> pile vide à la fin
    return p.est_vide()
```

```
assert parenthesage("((()())())") == True
```

```
assert parenthesage("()()") == False
```

```
assert parenthesage("(" (()) ()) == False
```