```python
from doctest import testmod


class Noeud:
    def __init__(self, g, v, d):
        self.gauche = g
        self.valeur = v
        self.droit = d

    def __str__(self):
        return str(self.valeur)

    def est_une_feuille(self):
        '''Renvoie True si et seulement si le noeud est une feuille'''
        return self.gauche is None and self.droit is None


def expression_infixe(e):
    """Parcours infixe d'un arbre pour afficher les noeuds
    séparés par des parenthèses.

    Args:
        e (Noeud): racine de l'arbre

    Returns:
        str: expression bien formée par des parenthèses de l'abre e

    Tests et Exemples:
    >>> e = Noeud(Noeud(Noeud(None, 3, None), '*', Noeud(Noeud(None, 8, None), '+',
Noeud(None, 7, None))), '-', Noeud(Noeud(None, 2, None), '+', Noeud(None, 1, None)))
    >>> expression_infixe(e)
    '((3*(8+7))-(2+1))'
    """
    # initialisation de la chaine à renvoyer
    s = ""
    if e.gauche is not None:
        # appel récursif pour afficher d'abord le sous arbre gauche
        s = s + expression_infixe(e.gauche)
    # ajout de la valeur du noeud
    # en utilisant la méthode dédiée
    s = s + str(e)
    if e.droit is not None:
        # appel récursif pour afficher d'abord le sous arbre gauche
        # s'il est non vide
        s = s + expression_infixe(e.droit)

    # cas de base : le noeud est une feuille
    if e.est_une_feuille():
        # afficher la valeur sans les parenthèses
        return s


    return '('+ s +')'


if __name__ == '__main__':
    testmod()

    e = Noeud(Noeud(Noeud(None, 3, None), '*', Noeud(Noeud(None, 8, None), '+', Noeu
d(None, 7, None))), '-', Noeud(Noeud(None, 2, None), '+', Noeud(None, 1, None)))
    assert expression_infixe(e) == '((3*(8+7))-(2+1))'
```