```
"""
Author: Pascal Padilla
Source: correction de l'exercice 1
    * du sujet 19 des épreuves pratiques NSI 2022
    * du sujet 21 des épreuves pratiques NSI 2022
"""


from doctest import testmod


# version itérative
def multiplication(n1: int, n2: int) -> int:
    """Renvoie le résultat de la multiplication de n1 par n2
    (avec n1 et n2 entiers relatifs).
    Petit défi : fonction programmée uniquement avec
                 les opérateurs + et -.

    Args:
        n1 (int): premier facteur
        n2 (int): deuxième facteur

    Returns:
        int: résultat de n1 × n2

    Tests et exemples:
    >>> multiplication(3,5)
    15
    >>> multiplication(-4,-8)
    32
    >>> multiplication(-2,6)
    -12
    >>> multiplication(-2,0)
    0
    """
    # résultat du produit initialisé à 0
    # car si aucune boucle ne s'exécute (n1 ¿ 0)
    # alors le résultat reste correct
    produit = 0

    # cas de n1 positif:
    #   faire n1 additions successives pour obtenir "n1 fois n2"
    if n1 >= 0:
        for i in range(n1):
            produit = produit + n2

    # cas de n1 négatif:
    #   faire n1 soustractions successives pour obtenir "n1 fois n2"
    else:
        for i in range(-n1):
            produit = produit - n2

    return produit


# version récursive (pour se faire plaisir ;) )
def multiplication_r(n1: int, n2: int) -> int:
    """Renvoie le résultat de la multiplication de n1 par n2
    (avec n1 et n2 entiers relatifs).
    Petit défi : fonction programmée uniquement avec
                 les opérateurs + et -.

    Args:
        n1 (int): premier facteur
        n2 (int): deuxième facteur

    Returns:
        int: résultat de n1 × n2

    Tests et exemples:
    >>> multiplication_r(3,5)
    15
    >>> multiplication_r(-4,-8)
```

```
    32
    >>> multiplication_r(-2,6)
    -12
    >>> multiplication_r(-2,0)
    0
    """
    # l'idée est de remarquer pour n1 POSITIF:
    # ¿ n1 × n2 = n2 + (n1 − 1) × n2
    # ce qui s'écrit de façon fonctionnelle:
    # ¿ multiplication(n1, n2) = n2 + multiplication(n1-1, n2)
    #
    # pour le cas n1 NÉGATIF:
    # ¿ n1 × n2 = (n1 + 1) × n2 − n2
    #
    # cette remarque permet une définition récursive de la multiplication
    #   ¿ avec n1 qui diminue de 1 à chaque appel récursif
    #   ¿ jusqu'à atteindre le cas de base 1

    # cas de bases (1) et (2):
    if n1 == 0 or n2 == 0:
        return 0

    # cas de base (3)
    if n1 == 1:
        return n2

    # cas positif:
    if n1 > 1:
        return n2 + multiplication(n1 - 1, n2)
    # cas négatif
    else:
        return multiplication(n1 + 1, n2) - n2


# tests avec des affichages
print(multiplication(3,5))
print(multiplication(-4,-8))
print(multiplication(-2,6))
print(multiplication(-2,0))


# tests de la fonction avec doctest
testmod()
```