Método

armstrong.py

```python
import logging

logging.basicConfig(level=logging.INFO, format='%(asctime)s\t%(levelname)s\t%(message)s')

def ArmstrongNumber(n) -> bool:
    """
    Calculates whether a given number is an Armstrong number or not.

    An Armstrong number is a number that is equal to the sum of its own digits raised to the power of the number of digits in the number.

    Parameters:
    n (int): The number to be checked.

    Returns:
    bool: Whether n meets the conditions for an Armstrong number or not.
    """

    logging.debug(f'The input {n} has type {type(n)}')
    logging.info('The method ArmstrongNumber has been called.')

    if not str(n).strip('-').isnumeric():
        logging.error('Input is not an integer.')
        raise TypeError()
    else: n = int(n)

    if n < 0:
        logging.warning('Input is a negative number.')
        return False

    return n == sum([int(i) ** len(str(n)) for i in [*str(n)]])
```

test.py

```python
import unittest as ut
from armstrong import ArmstrongNumber

class TestArmstrongNumber(ut.TestCase):
    def test_return(self):
        pass
```

```
        self.assertTrue(ArmstrongNumber(3))
        self.assertTrue(ArmstrongNumber(153))
        self.assertTrue(ArmstrongNumber(371))

        self.assertFalse(ArmstrongNumber(10))
        self.assertFalse(ArmstrongNumber(-1))
        self.assertFalse(ArmstrongNumber(100))

    def test_invalid(self):
        """
            ArmstrongNumber("Hello!")
        with self.assertRaises(TypeError):
        """
        with self.assertRaises(TypeError):
            ArmstrongNumber(True)
        with self.assertRaises(TypeError):
            ArmstrongNumber("Hello!")

if __name__ == '__main__':
    ut.main()
```

Output & logs