```python
"""
This module contains all functions which are used to test the program code.
"""

import main_module
import text_module


# *** tests for the text_module ***
# * Caesar *
def test_caesar():
    """
    tests if encrypt_caesar and decrypt_caesar work
    :return: True if the former encrypted text decrypted is equal to the original text
    False if not (-> Something went wrong)
    """
    print("*** TEST_CAESAR STARTED ***")
    key = 4
    original_text = "Hello world"
    # format text
    formatted_text = main_module.format_text(original_text)
    # encrypt text
    encrypted_text = text_module.encrypt_caesar(formatted_text, key)
    # check if decrypted encrypted text is equal to formatted text
    if text_module.decrypt_caesar(encrypted_text, key) == formatted_text:
        return True
    else:
        return False


# * Vigenère *
def test_create_vigenere_table():
    """
    tests to see if create_table_of_vigenere is working, i.e. test some letters in the table at specific fields
    compare result with wikipedia
    :return: table of vigenere
    """
    print("*** TEST_CREATE_VIGENERE_TABLE STARTED ***")
    text_module.create_vigenere_table()


def test_vigenere():
    """
    tests if encrypt_vigenere and decrypt_vigenere work
    :return: True if the former encrypted text decrypted is equal to the original text
    False if not (-> Something went wrong)
    """
    print("*** TEST_VIGENERE STARTED ***")
    key = "magique"
    original_text = "Hello world"
    # format text
    formatted_text = main_module.format_text(original_text)
    # encrypt text
    encrypted_text = text_module.encrypt_vigenere(formatted_text, key)
    # check if decrypted encrypted text is equal to formatted text
    if text_module.decrypt_vigenere(encrypted_text, key) == formatted_text:
        return True
    else:
        return False


# * Enigma *
def test_enigma():
    """
    tests if enigma works
    :return: True if the former encrypted text decrypted is equal to the original text
    False if not (-> Something went wrong)
    """
    print("*** TEST_ENIGMA STARTED ***")
    key1 = "AAA"
    key2 = "ADB"
    original_text = "AABBCCDDEEFFGGHHIIJJKKLLMMNNOOPPQQRRSSTTUUVVWWXXYYZZ"
    other_text = "Ayant le choix ou d’être serf ou d’être libre, quitte la franchise et prend le joug, qui consent à son mal."
    # format text
    formatted_text1 = main_module.format_text(original_text)
    formatted_text2 = main_module.format_text(other_text)
    # check if decrypted encrypted text is equal to formatted text
    # encrypt
    encrypt1 = text_module.enigma(formatted_text1, key1)
    encrypt2 = text_module.enigma(formatted_text2, key2)
    # decrypt
    decrypt1 = text_module.enigma(encrypt1, key1)
    decrypt2 = text_module.enigma(encrypt2, key2)
    if encrypt1 == decrypt1 and encrypt2 == decrypt2:
        return True
    else:
        return False


# *** tests for the main_module ***
def test_format_and_normalise():
    """
    tests if normalise_letter and format_text work
    :return: True if there are only capital letters left in the formatted text
    False if not (-> Something went wrong)
    """
    print("*** TEST_FORMAT_AND_NORMALISE STARTED ***")
    original_text = """àâ test1t!? æ test2test,. ç test3test;+ èéêë test4test-% îï t5t$& ô
    test6test\"/ ùûü t7t{} ÿ t8t[] œ t9t= """
    # format text
    formatted_text = main_module.format_text(original_text)
    print(formatted_text)
    # check if only capital letters
    if formatted_text.isalpha() and formatted_text.isupper():
        return True
    else:
        return False


def test_run():
    print("*** TEST_RUN STARTED ***")
    print("")

```

```python

# run all tests
def run_all_tests():
    """
    general function to run all tests
    :return: None
    """
    test_caesar()
    test_vigenere()
    test_format_and_normalise()
    test_create_vigenere_table()
    test_enigma()


# run all tests
run_all_tests()

```