```python
"""
MATH20621 - Coursework 1
Student name:jieyi huang
Student id:11108970
Student mail: jieyi.huang@student.manchester.ac.uk
"""

import math

# Feel free to other functions you find useful.
def is_prime(n):
    """
    Returns `True` if `n` is a prime number and `False` otherwise.
    This was problem 3 of the week 3 home exercises.
    """
    if n < 2:
        return False
    for k in range(2, n):
        if n % k == 0:
            return False
    return True

# Problem 1
def limiter(a, b, limitertype):
    """
    add a docstring
    """
    # TODO: add your code
    if limitertype == "WENO":
        wa = 1 / (a**2 + 1e-6)
        wb = 1 / (b**2 + 1e-6)
        result = (wa * a + wb * b) / (wa + wb)

    elif limitertype == "MINMOD":

        if a * b > 0:
            result = min(2 * a, 2 * b, (a + b) / 2)

        else:
            result = 0
    else:
        result = (a + b) / 2
    return result

    # WARNING: DO NOT USE INPUT() HERE!
    # THE VALUES OF a, b and limitertype ARE PROVIDED BY THE FUNCTION CALL.

    return # modify this to return (not print!) the computed value

# Problem 2
def pow_mod(a, b, c):
    """
    add a docstring
    """
    # TODO: add your code def pow_mod(a, b, c):
    if a < 0 or b < 0 or c < 0:

            result = None
    else:
            result=(a**b)%c
    # WARNING: DO NOT USE INPUT() HERE!
    # THE VALUES OF a, b and c ARE PROVIDED BY THE FUNCTION CALL.
    return result# modify this to return (not print!) the computed value


# Problem 3
def two_knodel(m):
    """
    add a docstring
    """
    # TODO: add your code
    if not is_prime(m) and m > 2:
        for i in range(2, m):
            if math.gcd(i, m) == 1:
                if (i **(m - 2)) % m != 1:
                    return False
    # WARNING: DO NOT USE INPUT() HERE!
    # THE VALUE OF m IS PROVIDED BY THE FUNCTION CALL.
    return True# modify this to return (not print!) the computed value

# Problem 3

# main() function for all the testing
def main():
    # TODO (optional): do any testing you wish here
    # This main function will not be assessed
    print("should return 0:   ", limiter(-3, 4, 'MINMOD'))
    print("should return 0.120001...:   ", limiter(0.1, 0.2, 'WENO'))
    print("should return 5...:   ", pow_mod(3, 5, 7))
    print("should return False:  ", two_knodel(9))
    print("should return True:  ", two_knodel(10))

main() # call main() function to run all tests
```