```
# -*- coding: utf-8 -*-
"""
Created on February 16, 2024
@authors:
    Elif KARTAL, Istanbul University, Faculty of Economics, Department of
Management Information Systems
    Beyaz Basak ESKISEHIRLI, Istanbul University, Faculty of Science,
Department of Mathematics
    Fatma CALISKAN, Istanbul University, Faculty of Science, Department of
Mathematics
    Zeki OZEN, Istanbul University, Faculty of Economics, Department of
Management Information Systems

@title: p-adic Module
@ver: 1.1
"""

import numpy as np
from fractions import Fraction

# p-parameter of p-adic distance must be prime.
# This function controls if the p is prime or not.
def is_prime(n):
    for i in range(2, n):
        if (n % i) == 0:
            return False
    return True

# This function returns how many of one number (div) are in another number
(num).
def is_div(num, div):
    counter = 0
    while num % div == 0:
        num //= div
        counter += 1
    return counter

# This function calculates the p-adic absolute value of the given number.
def p_adic_val(m_num, p_num):
    if is_prime(p_num):
        if m_num != 0:
            m_fraction = Fraction(m_num).limit_denominator()
            return np.round(p_num ** (is_div(m_fraction.denominator, p_num) -
is_div(m_fraction.numerator, p_num)),6)
        else:
            return 0
    else:
        print("p should be a prime number!")
```

```
# This function calculates the p-adic order (my_alpha) and the p-adic
absolute value of the given number.
def p_adic_ord_val(my_num, p_numb):
    if is_prime(p_numb):
        if my_num != 0:
            my_fraction = Fraction(my_num).limit_denominator()
            my_alpha = (is_div(my_fraction.numerator, p_numb)-
is_div(my_fraction.denominator, p_numb))
            my_padic_val = np.round(p_numb ** (-my_alpha), 6)
            return my_alpha, my_padic_val
        else:
            return 0
    else:
        print("p should be a prime number!")
```