Lab - Python Program of Finite Fields

I apologize in advance, the lab instructions weren't clear on what is expected. I think we were meant to implement a finite fields class in python with its simple properties (and test them) and that's what I did for this lab.

```
# Constructing a Finite Field Class

class FieldElement:

  # Check that num is between 0 and prime-1 inclusive, i.e., F_5 = {0, 1, 2, 3, 4}.
  def __init__(self, num, prime):
    if num >= prime or num < 0:
      error = 'Num {} not in field range 0 to {}'.format(num, prime - 1)
      raise ValueError(error)
    self.num = num
    self.prime = prime

  def __repr__(self):
    return 'FieldElement_{}({})'.format(self.prime, self.num)

  # Equal method, checks if two objs are equal.
  def __eq__(self, other):
    if other == None:
      return False
    return self.num == other.num and self.prime == other.prime

  # Not equal method, check if two objs are not equal.
  def __ne__(self, other):
    return not self.__eq__(other)

 # Addition method with modulo arithmetic for adding two objs.
  def __add__(self, other):
    if self.prime != other.prime:
      raise TypeError('Cannot add two numbers in different Fields')
    num = (self.num + other.num) % self.prime
    return FieldElement(num, self.prime)

  # Subtraction method with modulo arithmetic for subtracting two objs.
  def __sub__(self, other):
    if self.prime != other.prime:
      raise TypeError('Cannot subtract two numbers in different Fields')
    num = (self.num - other.num) % self.prime
    return FieldElement(num, self.prime)

  # Exponentiation method, overriding the ** operator.
  def __pow__(self, exponent):
    fermat = exponent % (self.prime - 1)
    num = pow(self.num, fermat, self.prime)
    return FieldElement(num, self.prime)

  # Multiplication (*) method for the multiplication of two finite field elements.
  def __mul__(self, other):
    if self.prime != other.prime:
      raise TypeError('Cannot multiply two numbers in different Fields')
    num = (self.num * other.num) % self.prime
    return FieldElement(num, self.prime)

  # Division (/) method utuilizing Fermat's little theorem for the division of two field elements.
  def __truediv__(self, other):
    if self.prime != other.prime:
      raise TypeError('Cannot divide two numbers in different Fields')
    num = self.num * pow(other.num, self.prime - 2, self.prime) % self.prime
    return FieldElement(num, self.prime)
```

First, we test the equals (== or __eq __) method and see if two objects are equal.

Then, we also test the not equal (!= or __ne __) method and see if two objects are not equal.

```
# Equal method testing
a = FieldElement(7, 13)
b = FieldElement(6, 13)
print(a == b)
print()
print(a == a)
```

```
False

True
```

```
# Not equal method testing
a = FieldElement(2, 31)
b = FieldElement(2, 31)
c = FieldElement(15, 31)
print(a != c)
print()
print(a != b)
```

```
True

False
```

Second, we test the addition (addition closed) method and see if two objects add correctly and the result is still in the set.

```
a = FieldElement(7, 13)
b = FieldElement(12, 13)
c = FieldElement(6, 13)
print(a+b) # Results in 6, which is in the set F_13.
print()
print(a+b==c)
```

```
FieldElement_13(6)

True
```

Third, we test the subtraction (or additive inverse) method and see if two objects subtract correctly and the result is still in the set.

```
a = FieldElement(29, 31)
b = FieldElement(4, 31)
c = FieldElement(25, 31)
print(a-b) # Results in 25, which is in the set F_31.
print()
print(a-b==c)
```

```
FieldElement_31(25)

True
```

Fourth, we test the multiplication (or multiplication closed) method and see if two objects multiply correctly and the result is still in the set.

```
a = FieldElement(3, 13)
b = FieldElement(12, 13)
c = FieldElement(10, 13)
print(a*b) # Results in 10, which is in the set F_13.
print()
print(a*b==c)
```

```
FieldElement_13(10)

True
```

Fifth, we test the division (or multiplication inverse) method and see if two objects divide correctly and the result is still in the set.

```
a = FieldElement(3, 31)
b = FieldElement(24, 31)
c = FieldElement(4, 31)
print(a / b) # Results in 4, which is in the set F_31.
print()
print(a / b == c)
```

```
FieldElement_31(4)

True
```

References:

https://www.oreilly.com/library/view/programming-bitcoin/9781492031482/ch01.html

https://colab.research.google.com/drive/1TKSaknUcCgkpi_0CIH5Eas1wGPFrfjgo?usp=sharing&pli=1#scrollTo=QUhJYbi3ZVNH
