National Sun Yat-sen University
Introduction To Blockchain Technology
Homework 1
Course Number: CSE222, Chapter: 1 & 2

# Notice :
1. No late homework.
2. Please submit your homework to **Cyber University of National Sun Yat-sen University** (https://cu.nsysu.edu.tw/mooc/index.php). It is not allowed to submit assignments to any other location.
3. You only need to submit **Homework 1.docx**, after you paste all screenshot of your code and code execution results of solving following problems.
4. We only accept using **python** to write program code.
5. **Please answer each question according to the requirements below, otherwise no points will be awarded.**

1. Solve these problems in F_{701}:

1-1. $599 \cdot 607 \cdot 613$

1-2. $23 \cdot 223 \cdot 509 \cdot 666$

1-3. $337^{79} \cdot 557^{131}$

# Grading for Problem 1:
- Total: 30%
  - Correct code and code execution result: 30%, 10% for each
    (Only have correct code and correct code execution can get score.)

--- Please paste screenshot of your code and code execution result of solving problem 1-1 below. ---

程式說明:

以 class FieldElement 定義元素的計算方式，包含加法、乘法、減法、除法、次方。

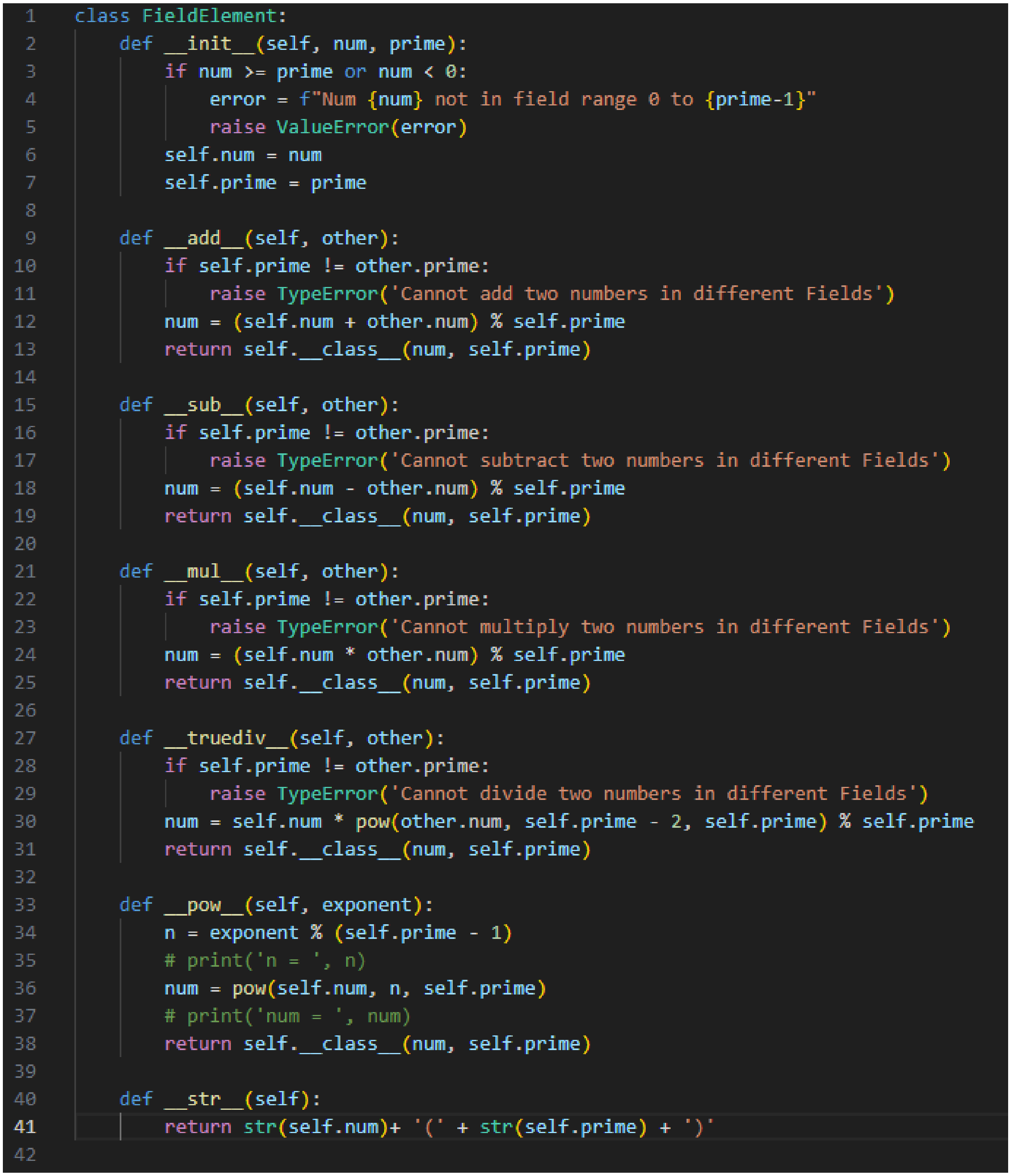
function readEquation 會讀入一個算式，把該算式的數字和運算元分離，將算式中的數字依序存於 list num，算式中的運算元依序存於 list operators。

function FFoperation 會讀入一個運算元，以及兩個 list: num, operators，代表目前尚未運算完的數字和運算元，此 function 會從 operators 中找尋傳入的 operator 的位置，並在

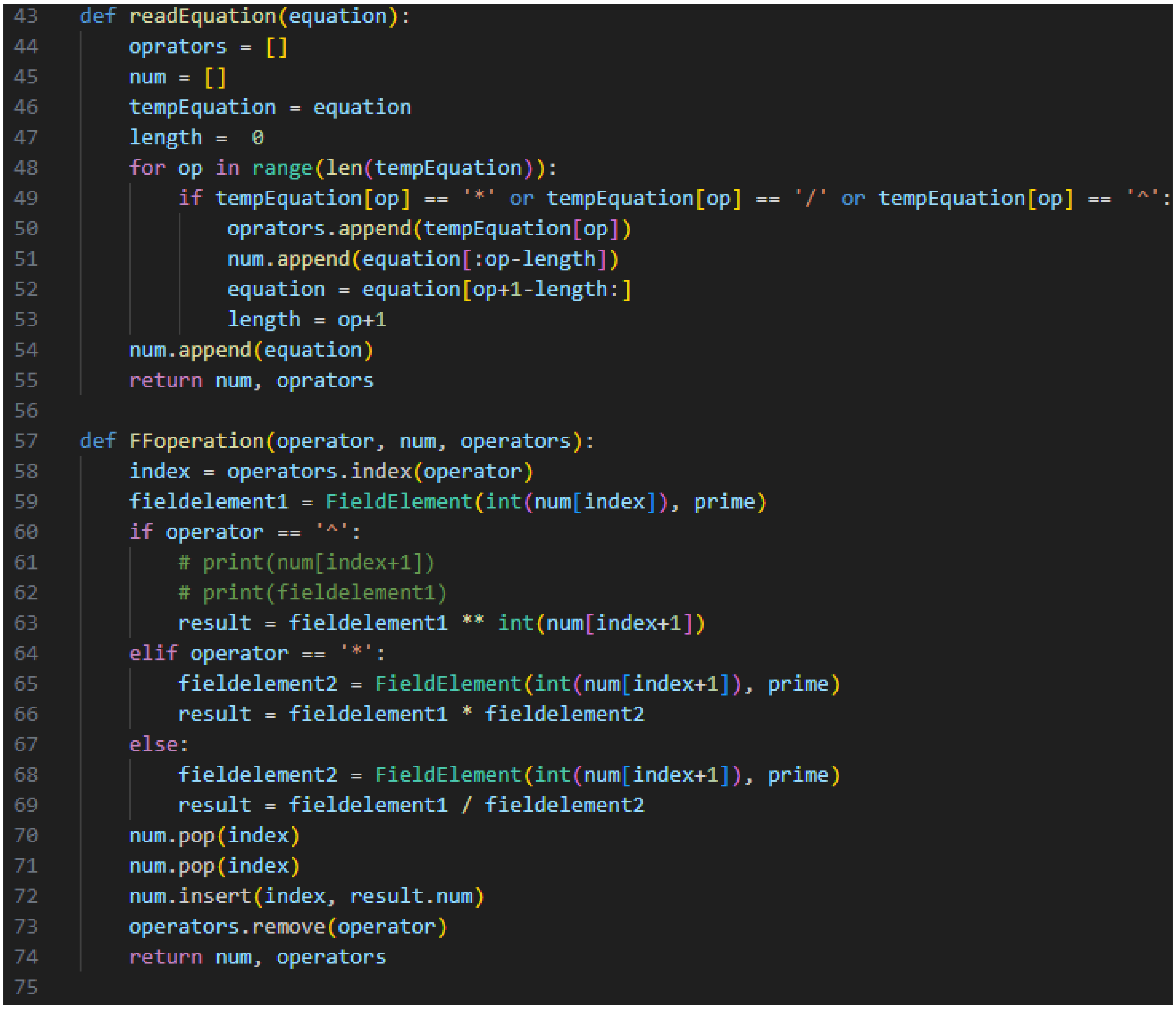
num 中找尋並取出兩個在原運算式中和該運算元相鄰的兩個數字進行該運算元的計算，再把運算完的結果存回 num 中，剛剛取出數字的位置。

執行時，會請使用者輸入一個 prime 作為 finite field 的 order，以及一個算式，其中，使用者若輸入^則代表次方；*代表相乘；/代表相除

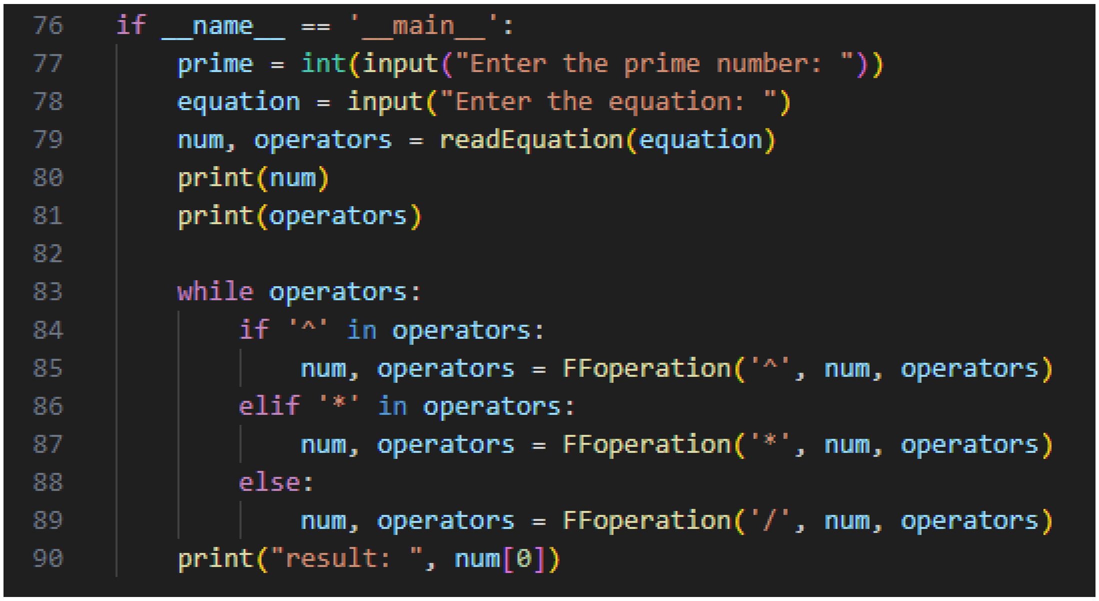
Code:



```python
class FieldElement:
    def __init__(self, num, prime):
        if num >= prime or num < 0:
            error = f"Num {num} not in field range 0 to {prime-1}"
            raise ValueError(error)
        self.num = num
        self.prime = prime

    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 self.__class__(num, self.prime)

    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 self.__class__(num, self.prime)

    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 self.__class__(num, self.prime)

    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 self.__class__(num, self.prime)

    def __pow__(self, exponent):
        n = exponent % (self.prime - 1)
        # print('n = ', n)
        num = pow(self.num, n, self.prime)
        # print('num = ', num)
        return self.__class__(num, self.prime)

    def __str__(self):
        return str(self.num)+ '(' + str(self.prime) + ')'

```

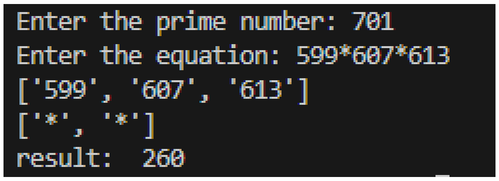


```python
def readEquation(equation):
    oprators = []
    num = []
    tempEquation = equation
    length =  0
    for op in range(len(tempEquation)):
        if tempEquation[op] == '*' or tempEquation[op] == '/' or tempEquation[op] == '^':
            oprators.append(tempEquation[op])
            num.append(equation[:op-length])
            equation = equation[op+1-length:]
            length = op+1
    num.append(equation)
    return num, oprators

def FFoperation(operator, num, operators):
    index = operators.index(operator)
    fieldelement1 = FieldElement(int(num[index]), prime)
    if operator == '^':
        # print(num[index+1])
        # print(fieldelement1)
        result = fieldelement1 ** int(num[index+1])
    elif operator == '*':
        fieldelement2 = FieldElement(int(num[index+1]), prime)
        result = fieldelement1 * fieldelement2
    else:
        fieldelement2 = FieldElement(int(num[index+1]), prime)
        result = fieldelement1 / fieldelement2
    num.pop(index)
    num.pop(index)
    num.insert(index, result.num)
    operators.remove(operator)
    return num, operators

```



```python
if __name__ == '__main__':
    prime = int(input("Enter the prime number: "))
    equation = input("Enter the equation: ")
    num, operators = readEquation(equation)
    print(num)
    print(operators)

    while operators:
        if '^' in operators:
            num, operators = FFoperation('^', num, operators)
        elif '*' in operators:
            num, operators = FFoperation('*', num, operators)
        else:
            num, operators = FFoperation('/', num, operators)
    print("result: ", num[0])
```

Execution result:

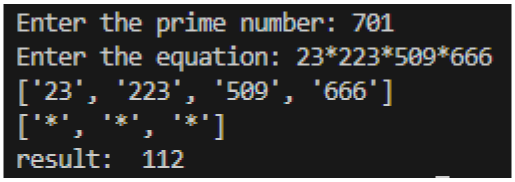


```
Enter the prime number: 701
Enter the equation: 599*607*613
['599', '607', '613']
['*', '*']
result:  260
```

--- Please paste screenshot of <u>your code and code execution result</u> of solving problem 1-2 below. ---

Code: 同 problem 1.1

Execution result:

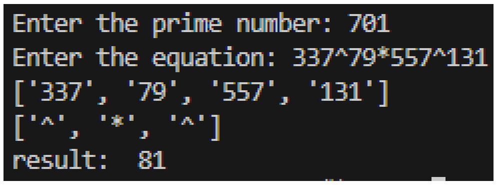


```
Enter the prime number: 701
Enter the equation: 23*223*509*666
['23', '223', '509', '666']
['*', '*', '*']
result:  112
```

--- Please paste screenshot of <u>your code and code execution result</u> of solving problem 1-3 below. ---

Code: 同 problem 1.1

Execution result:



```
Enter the prime number: 701
Enter the equation: 337^79*557^131
['337', '79', '557', '131']
['^', '*', '^']
result:  81
```

2. Solve these problems in F_{881}:

2-1. 800/31

2-2. 201^{-101} · 57

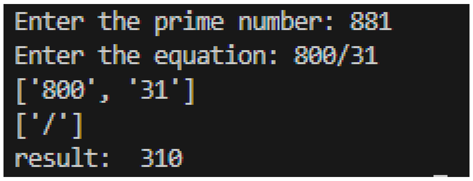
# Grading for Problem 2:

- Total: 20%
  - Correct code and code execution result: 20%, 10% for each
    (Only have correct code and correct code execution can get score.)

--- Please paste screenshot of <u>your code and code execution result</u> of solving problem 2-1 below. ---

Code: 同 problem 1.1

Execution result:

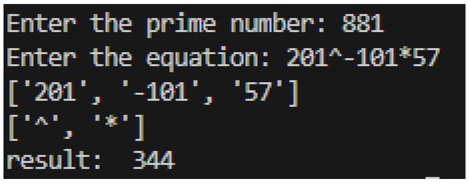


```
Enter the prime number: 881
Enter the equation: 800/31
['800', '31']
['/']
result:  310
```

--- Please paste screenshot of <u>your code and code execution result</u> of solving problem 2-2 below. ---

Code: 同 problem 1.1

Execution result:



```
Enter the prime number: 881
Enter the equation: 201^-101*57
['201', '-101', '57']
['^', '*']
result:  344
```

3. Compute the slope and the sum of the points:

3-1. (−2,−5)+(5,−9) for the curve $y^2=x^3-11x+11$

3-2. (−2,−3)+(3,3) for the curve $y^2=x^3-7x+3$

3-3. (−3,−1)+(5,9) for the curve $y^2=x^3-9x+1$

Python provides the *fractions.Fraction* module, and you can use it to avoid floating-point errors.

# Grading for Problem 3:

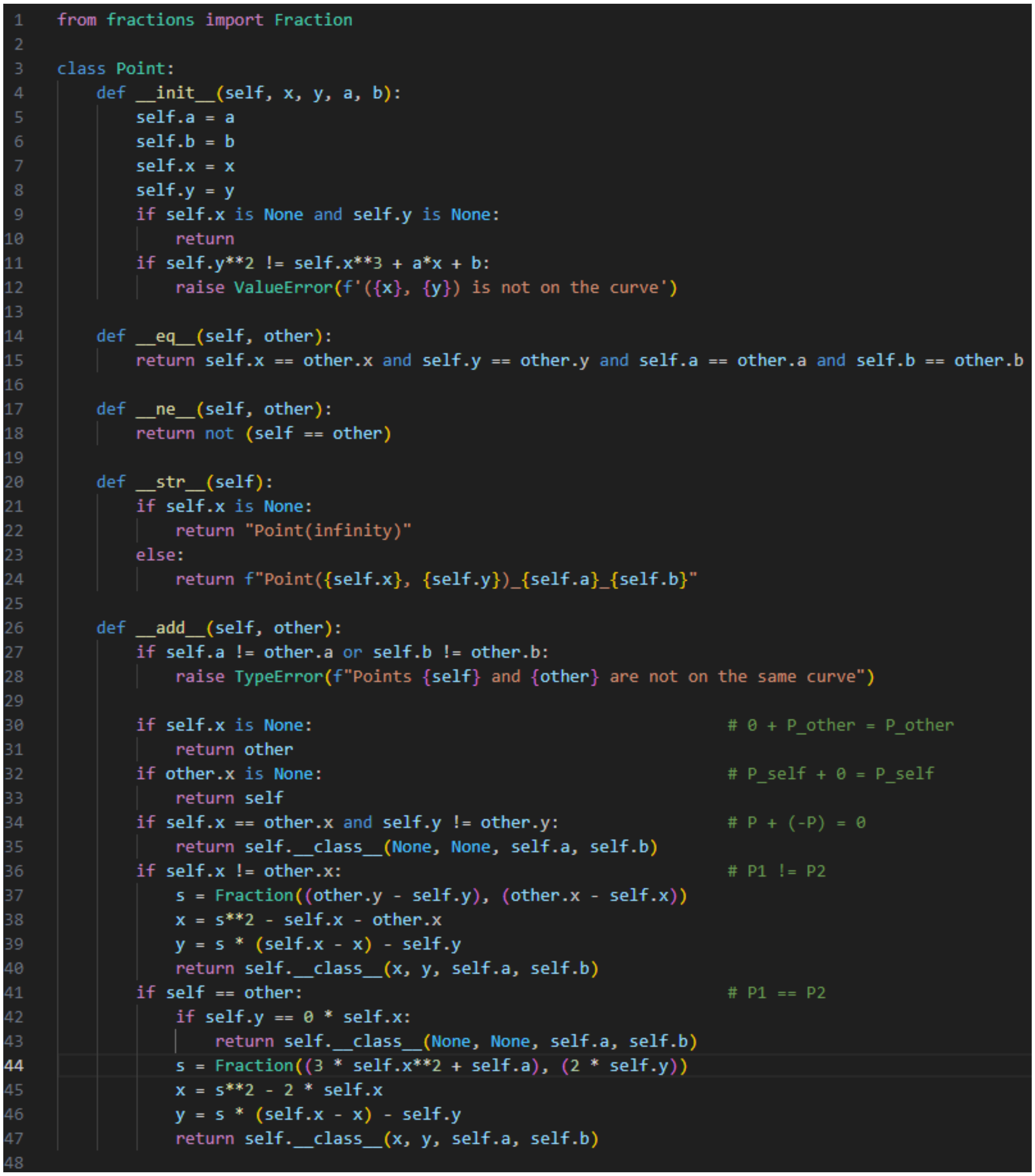
- Total: 30%
  - Correct code and code execution result: 30%, 10% for each (Only have correct code and correct code execution can get score.)

--- Please paste screenshot of <u>your code and code execution result</u> of solving problem 3-1 below. ---

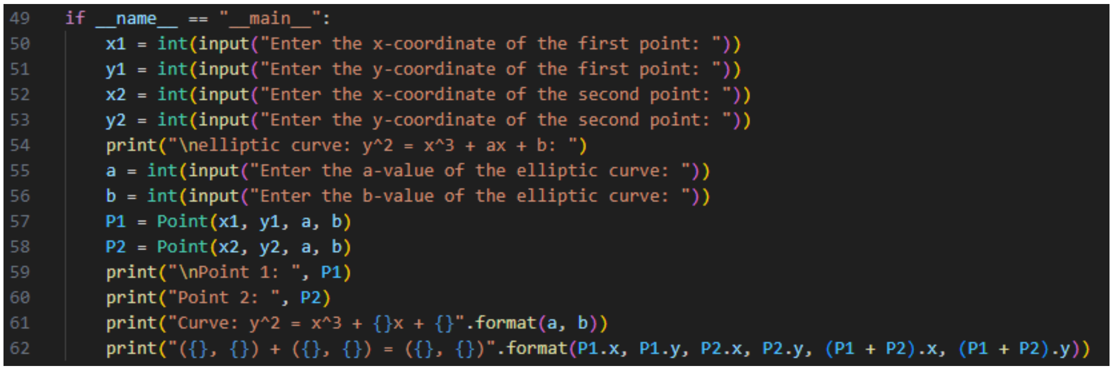
程式說明:

class Point 定義了 elliptic curve 中，兩點的判斷相等、判斷不相等以及相加的方式。程式執行時，會讓使用者輸入兩個欲相加的點的座標，以及 elliptic curve 的 a 與 b 之值，接下來輸出使用者輸入的兩點、elliptic curve 以及兩點相加的結果

Code:

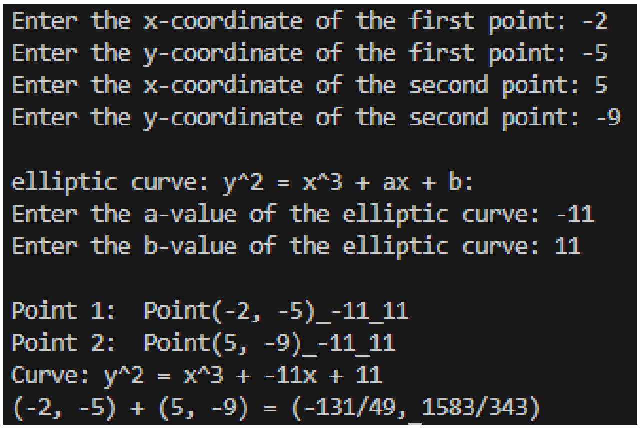


```python
from fractions import Fraction

class Point:
    def __init__(self, x, y, a, b):
        self.a = a
        self.b = b
        self.x = x
        self.y = y
        if self.x is None and self.y is None:
            return
        if self.y**2 != self.x**3 + a*x + b:
            raise ValueError(f'({x}, {y}) is not on the curve')

    def __eq__(self, other):
        return self.x == other.x and self.y == other.y and self.a == other.a and self.b == other.b

    def __ne__(self, other):
        return not (self == other)

    def __str__(self):
        if self.x is None:
            return "Point(infinity)"
        else:
            return f"Point({self.x}, {self.y})_{self.a}_{self.b}"

    def __add__(self, other):
        if self.a != other.a or self.b != other.b:
            raise TypeError(f"Points {self} and {other} are not on the same curve")

        if self.x is None:                                          # 0 + P_other = P_other
            return other
        if other.x is None:                                         # P_self + 0 = P_self
            return self
        if self.x == other.x and self.y != other.y:                 # P + (-P) = 0
            return self.__class__(None, None, self.a, self.b)
        if self.x != other.x:                                       # P1 != P2
            s = Fraction((other.y - self.y), (other.x - self.x))
            x = s**2 - self.x - other.x
            y = s * (self.x - x) - self.y
            return self.__class__(x, y, self.a, self.b)
        if self == other:                                           # P1 == P2
            if self.y == 0 * self.x:
                return self.__class__(None, None, self.a, self.b)
            s = Fraction((3 * self.x**2 + self.a), (2 * self.y))
            x = s**2 - 2 * self.x
            y = s * (self.x - x) - self.y
            return self.__class__(x, y, self.a, self.b)

```



```python
if __name__ == "__main__":
    x1 = int(input("Enter the x-coordinate of the first point: "))
    y1 = int(input("Enter the y-coordinate of the first point: "))
    x2 = int(input("Enter the x-coordinate of the second point: "))
    y2 = int(input("Enter the y-coordinate of the second point: "))
    print("\nelliptic curve: y^2 = x^3 + ax + b: ")
    a = int(input("Enter the a-value of the elliptic curve: "))
    b = int(input("Enter the b-value of the elliptic curve: "))
    P1 = Point(x1, y1, a, b)
    P2 = Point(x2, y2, a, b)
    print("\nPoint 1: ", P1)
    print("Point 2: ", P2)
    print("Curve: y^2 = x^3 + {}x + {}".format(a, b))
    print("({}, {}) + ({}, {}) = ({}, {})".format(P1.x, P1.y, P2.x, P2.y, (P1 + P2).x, (P1 + P2).y))
```

Execution result:

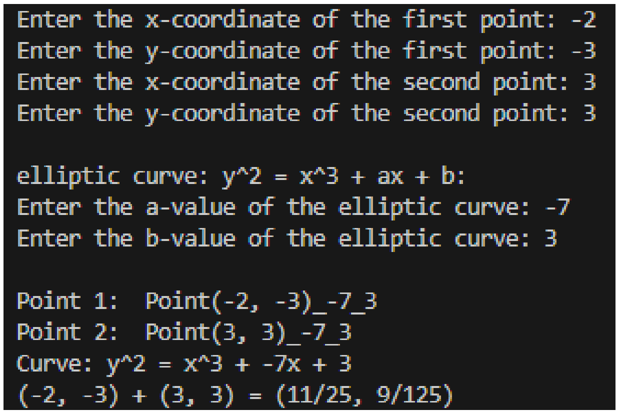


```
Enter the x-coordinate of the first point: -2
Enter the y-coordinate of the first point: -5
Enter the x-coordinate of the second point: 5
Enter the y-coordinate of the second point: -9

elliptic curve: y^2 = x^3 + ax + b:
Enter the a-value of the elliptic curve: -11
Enter the b-value of the elliptic curve: 11

Point 1:  Point(-2, -5)_-11_11
Point 2:  Point(5, -9)_-11_11
Curve: y^2 = x^3 + -11x + 11
(-2, -5) + (5, -9) = (-131/49,_1583/343)
```

--- Please paste screenshot of your code and code execution result of solving problem 3-2 below. ---

Code: 同 problem3-1

Execution result:

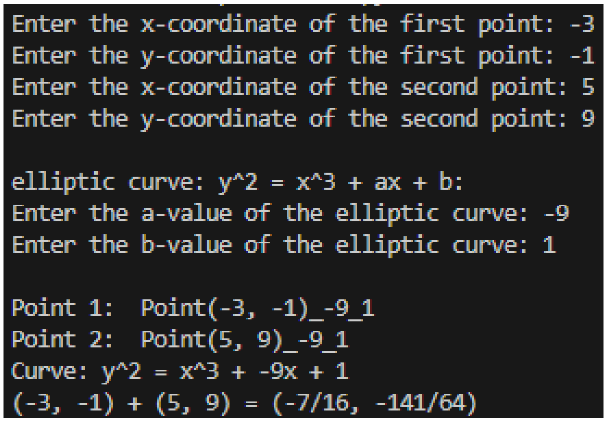


```
Enter the x-coordinate of the first point: -2
Enter the y-coordinate of the first point: -3
Enter the x-coordinate of the second point: 3
Enter the y-coordinate of the second point: 3

elliptic curve: y^2 = x^3 + ax + b:
Enter the a-value of the elliptic curve: -7
Enter the b-value of the elliptic curve: 3

Point 1:  Point(-2, -3)_-7_3
Point 2:  Point(3, 3)_-7_3
Curve: y^2 = x^3 + -7x + 3
(-2, -3) + (3, 3) = (11/25, 9/125)
```

--- Please paste screenshot of your code and code execution result of solving problem 3-3 below. ---

Code: 同 problem3-1

Execution result:



```
Enter the x-coordinate of the first point: -3
Enter the y-coordinate of the first point: -1
Enter the x-coordinate of the second point: 5
Enter the y-coordinate of the second point: 9

elliptic curve: y^2 = x^3 + ax + b:
Enter the a-value of the elliptic curve: -9
Enter the b-value of the elliptic curve: 1

Point 1:  Point(-3, -1)_-9_1
Point 2:  Point(5, 9)_-9_1
Curve: y^2 = x^3 + -9x + 1
(-3, -1) + (5, 9) = (-7/16, -141/64)
```

4. Compute the slope and the sum of the points:

4-1. (5,−9)+(5,−9) for the curve $y^2=x^3-11x+11$

4-2. (5,9)+(5,9) for the curve $y^2=x^3-9x+1$

Python provides the *fractions.Fraction* module, and you can use it to avoid floating-point errors.

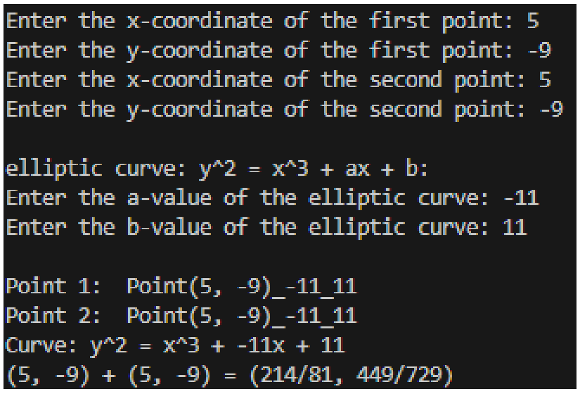
# Grading for Problem 4:

- Total: 20%
  - Correct code and code execution result: 20%, 10% for each
    (Only have correct code and correct code execution can get score.)

--- Please paste screenshot of <u>your code and code execution result</u> of solving problem 4-1 below. ---

Code: 同 problem3-1

Execution result:

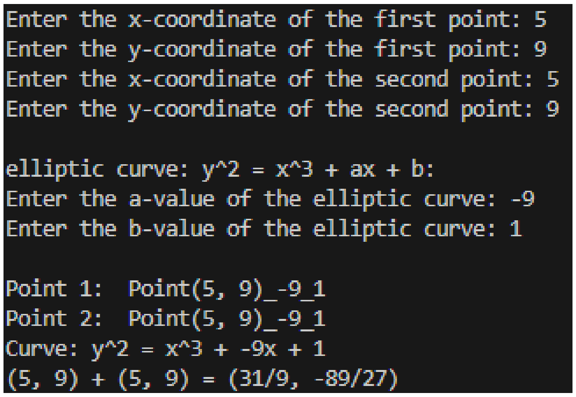


```
Enter the x-coordinate of the first point: 5
Enter the y-coordinate of the first point: -9
Enter the x-coordinate of the second point: 5
Enter the y-coordinate of the second point: -9

elliptic curve: y^2 = x^3 + ax + b:
Enter the a-value of the elliptic curve: -11
Enter the b-value of the elliptic curve: 11

Point 1:  Point(5, -9)_-11_11
Point 2:  Point(5, -9)_-11_11
Curve: y^2 = x^3 + -11x + 11
(5, -9) + (5, -9) = (214/81, 449/729)
```

--- Please paste screenshot of <u>your code and code execution result</u> of solving problem 4-2 below. ---

Code: 同 problem3-1

Execution result:



```
Enter the x-coordinate of the first point: 5
Enter the y-coordinate of the first point: 9
Enter the x-coordinate of the second point: 5
Enter the y-coordinate of the second point: 9

elliptic curve: y^2 = x^3 + ax + b:
Enter the a-value of the elliptic curve: -9
Enter the b-value of the elliptic curve: 1

Point 1:  Point(5, 9)_-9_1
Point 2:  Point(5, 9)_-9_1
Curve: y^2 = x^3 + -9x + 1
(5, 9) + (5, 9) = (31/9, -89/27)
```